PROJECT FINAL REPORT

Research Project No. (EO06/12)

ENGLISH TITLE:	Toward Enhanced Arabic Speech Recognition Systems				
ARABIC TITLE:	نحو نظام مطور للتعرف على الكلام العربي المتصل				
TOTAL BUDGET *:	42250 KD				
START DATE:	01/01/2015	**EXTENSION TAKEN:**	0 months	**END DATE **:**	31/12/2017
PAPER(S) SUBMITTED:		**PAPER(S) ACCEPTED:**			
PAPER(S) PUBLISHED:		**CONFERENCE PAPER(S):**			

RESEARCH TEAM

First: Principal Investigator			
Name:	FAWAZ S. AL-ANZI	**Rank:**	PROFESSOR
Faculty:	FACULTY OF ENGINEERING & PETROLEUM	**Department:**	COMPUTER ENGINEERING
Second: Co-Investigator(s)			
1. Name:	DIA EDDIN ABUZEINA	**Rank:**	ASST_PROF
Faculty:		**Department:**	
2. Name:		**Rank:**	
Faculty:		**Department:**	
3. Name:		**Rank:**	
Faculty:		**Department:**	
4. Name:		**Rank:**	
Faculty:		**Department:**	
Third: Contributor(s)			
1. Name:		**Rank:**	
Faculty:		**Department:**	
2. Name:		**Rank:**	
Faculty:		**Department:**	
3. Name:		**Rank:**	
Faculty:		**Department:**	
4. Name:		**Rank:**	
Faculty:		**Department:**	

* Including Supplementary Budget taken (if any)

** Including Extension(s) taken (if any)

Prof. Fawaz S. Al-Anzi
Computer Engineering Department
College of Engineering and Petroleum
Kuwait University
E-mail: Fawaz.Alanzi@ku.edu.kw

FINAL REPORT
EO06/12: 2017-2018
TOWARD ENHANCED ARABIC SPEECH RECOGNITION SYSTEMS

By
Prof. Fawaz S. Al-Anzi (PI)
Dr. Dia AbuZeina (CoI)

31 December 2017

Table of Contents

CHAPTER 1:
EXECUTIVE SUMMARY

Abstract- This is final report that presents the findings and achievements of project number EO 06/12 entitles "TOWARD ENHANCED ARABIC SPEECH RECOGNITION SYSTEMS" that was funded by Kuwait University, Research Administration. The project has concluded its activities by 31st of December 2017 after completing all the promised tasks as in it initial submitted and approved proposal. The duration of the project was 3 years where it successfully produced one of the largest Arabic language speech corpus in the world. The corpus has been carefully created and revised for the use of this project and future research projects in the area of Arabic Speech Recognition (ASR) and Natural Language Processing (NLP). We think that the completion of this corpus can constitute a benchmark for the researchers in this field in the future. A number of high quality journal and conference papers were produced as contribution of the project.

The investigators of this research project would like to thank the Research Administration at Kuwait University for the funding and their continuous support of this research project. We would also like to thank the referees for monitoring the progress of this project and sending us their valuable comments in reviewing the initial proposal and the annual reports of this project.

1. General Introduction

A. Research Subject:

Arabic is the most widely spoken Semitic language today that recently has received significant attention for automatic speech recognition (ASR). ASR is a component of the natural language processing (NLP), which is used to automate the communication process between human and machine, i.e., the man-machine interaction. In this regard, much research has been devoted to introducing innovative technologies in dialogue systems for automation purposes (e.g., banking services, cars, control machines, etc.). However, employing ASR technology in Arabic NLP applications is still limited due to various challenges about within the Arabic language itself. For instance, it is difficult to obtain corpora for dialects that are spoken rather than written, i.e., there is no common writing standard, difficulty in obtaining a sizable diacritized text as Arabic allows writing without diacritics, and an enormous number of word forms due to the morphology richness

of Arabic. In fact, one of the most difficult tasks in Arabic ASR is preparing a large diacritized text for ASR systems, which is a time-consuming preprocessing stage. In order to promote research on the Arabic ASR, we considered the corpora availability problem by producing a manually diacritized large-vocabulary speaker-independent continuous speech corpus for Modern Standard Arabic (MSA). The contents of the prepared corpus belong to general broadcast news.

In this project, we used the prepared corpus for an experimental evaluation of two off-the-shelf open source speech recognition toolkits, namely the Carnegie Mellon University (CMU) Sphinx [1] and the Hidden Markov Model Toolkit (HTK) [2]. In fact, it is important to find the performance of the popular speech engines using an identical speech collection because it reveals the unique characteristics of each engine for further understanding of their behavior in NLP and ASR applications. With the growing interest in ASR technology, it becomes more important to evaluate recent ASR systems in order to find the best-suited system for the tasks in question. For instance, the study in [3] demonstrated a large-scale evaluation of open-source speech recognition toolkits that include Sphinx, HTK, and Kaldi [4]. The study in [3] indicated that Kaldi is better than Sphinx and HTK in terms of results and training recipes (for the German and English languages, however, this project considers the Arabic language). That is, performing an ASR task using different recognizers will increase researcher knowledge regarding which engine is the best fit for particular target applications, as well as enhancing research in this field.

B. Main Objectives: This project has three main objectives as the following:

1. Facilitating the research in Arabic speech recognition by producing a survey of the literature to be a starting point for the researchers in this field. The review includes the basic algorithms that includes n-gram language models, acoustic models, pronunciation dictionaries, Viterbi, Gaussian mixture models, etc.

2. Preparing a large Arabic speech corpus (about 64,000 words) to be a test bed for speech recognition research as well as for other research domains. In fact, preparing a large speech corpus facilitates many NLP research problems such as speech synthesis, Arabic phonemes research (discovering the basic units of Arabic sound), speech features analysis, morphology and its role to enhance Arabic speech recognition, searching in speech files (i.e. identifying word spot), etc. Hence, preparing such corpus is a hope for many linguistic researches as well as for graduate students who are in need for large Arabic speech corpora. In addition, the

textual part of the corpus can be used in many NLP tasks such as machine translation, part of speech (PoS) tagging, sentiment analysis, question answering system, etc.

3. Having such corpus will accurately measure the performance of the two well-known speech recognition systems, the CMU Sphinx [1] and the HTK [2] systems. However, despite we used the prepared corpus for the CMU Sphinx and the HTK; it is possible as a future work to use it for the Kaldi [4] toolkit, which is now very widely used.

In addition to the above main objectives, the corpus might be used for other research activities. For instance, the corpus is beneficial for language models research such as perplexity [5], entropy [6], WordNet (semantic and syntactic relationships) [7]. Other research areas such as text classification and text clustering might use the textual part of the prepared corpus.

C. Methodology: preparing a speech corpus is the essential part to fulfill the listed objectives. Therefore, we follow the below steps to compile a large corpus for modern standard Arabic (MSA) as well as the other steps to evaluate the performance:

- **Step 1**: Preprocessing stage by extracting the useful short speech files (i.e. 30 – 60 seconds length) from the newscasts. Useful here means speech files that are suitable for speech recognition in term of noise, background music, and live translation.
- **Step 2**: A very short silence (i.e. 0.1 second length) is added at the beginning and at the end of each speech file. This silences help in the training process.
- **Step 3**: Having the collected speech files transcribed, (i.e. .wav files ➔.txt files).
- **Step 4**: Having the prepared textual files diacritized, (i.e. .txt files ➔ diacritized .txt files).
- **Step 5**: Using the diacritized textual files and the phonemes list, the speech dictionary created. The created dictionary contains the phonetic transcriptions of each word.
- **Step 6**: Using the diacritized textual files, the n-gram language model is created. The language model is used to constraint the range of possible utterances during recognition phase.
- **Step 7**: Installing and configuring SPHINX speech recognition system.
- **Step 8**: Installing and configuring HTK speech recognition system.
- **Step 9**: Measuring the performance using both the CMU Sphinx and the HTK systems.

D. Plan to Execute: What have done in this project includes the following:

1. [**Successfully Completed**] ➔ **Step 1:** Preprocessing and preparing the speech files that contains 4,071 short speech files of exactly 28:59:42 hours: minutes: seconds. Nevertheless, we initially prepared the speech files of (30 seconds to 60 seconds) length; however, we found it is necessary to reduce the speech files to about 30 seconds maximum particularly for the HTK systems. This process added more overhead as we lately discovered that HTK fails with long speech files.

2. [**Successfully Completed**] ➔ **Step 2:** Short silence were added at the beginning and at the end of each speech file. We developed a Python based program that automatically adds the silence for a set of speech files.

3. [**Successfully Completed**] ➔ **Step 3:** Having the speech files transcribed. The prepared textual files include 207,258 words and 47,727 unique words. We emphasize that we initially aimed to collect about 64,000 unique words.

4. [**Successfully Completed**] ➔ **Step 4:** Diacritization. We emphasize that the diacritization process includes two phases: the automated process using Harakat program and then a manual investigation of each word as a second phase.

5. [**Successfully Completed**] ➔ **Step 5:** Pronunciation dictionary. Based on the prepared part of the corpus, we generated pronunciation dictionaries for different speech recognition tasks using a Python based program. We used our "proposed" phonemes set (46 phonemes) for creating the pronunciation dictionaries. The CMU sphinx can handle the Arabic script while the HTK expects Roman characters. Hence, we prepared the program to generate dictionaries for both cases. In our experiments, we used both the Arabic script for the CMU Sphinx system and the Roman script for the HTK system. This is the first distinction (it could be also the first difficulty) that raised at the beginning of the evaluation process.

6. [**Successfully Completed**] ➔ **Step 6:** The language model. Based on the textual part of the corpus, we generated the n-gram language models for different speech recognition tasks using the software available in the CMU sphinx and the HTK tools. More information about statistical language model can be found in [8].

7. [**Successfully Completed**] ➔ **Step 7:** We installed CMU Sphinx system and performed the training and the recognition process based on the prepared corpus. The CMU Sphinx toolkit

includes the latest available releases as follows: ‘Sphinxbase – 5Prealpha’, ‘PocketSphinx - 5prealpha’, and ‘SphinxTrain - 5prealpha’.

8. [**Successfully Completed**] ➔ **Step 8:** We installed HTK system and performed the training and decoding process based on the prepared corpus. More information regarding the HTK version 3.4.1 (HVite decoder) can be found in [9].
9. [**Successfully Completed**] ➔ **Step 9:** We performed performance evaluation for the needed experiments. We used word error rate (WER) for performance evaluation.

Despite that we prepared 4,071 speech files as a final output, however, we only used 2,014 speech files for training and decoding due to time constraint to timely deliver our finding based on the prepared corpus. Another constrain is that the HTK fails when using long speech files while we did not observe this problem with CMU Sphinx. This is the reason why we discard some long speech files during training and decoding. In our main experiment, we used 12.74 hours (1,611 speech files) for training and 3.19 hours (403 speech files) for testing. The following figures show the accuracy achieved using CMU Sphinx and the HTK toolkit. Figure 1 shows the CMU Sphinx related results while Figure 2 shows the HTK related results. For more information, the project papers have the details. In general, CMU Sphinx engine outperforms the HTK engine.

Experiment	*Densities*	*Senones*	*WER (%)*	*Accuracy (%)*
1	*8*	*500*	*22.6*	*77.4*
2	*8*	*1000*	*22.2*	*77.8*
3	*8*	*2000*	*21.5*	*75.5*
4	*16*	*500*	*21.8*	*78.2*
5	*16*	*1000*	*21.1*	*78.9*
6	***16***	***2000***	***20.7***	***79.3***
7	*32*	*500*	*21.8*	*78.2*
8	*32*	*1000*	*21.3*	*78.7*
9	*32*	*2000*	*21.3*	*78.7*

Figure 1. The performance of the Sphinx recognizer

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 18 19:01:41 2017
  Ref : words.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=403, N=403]
WORD: %Corr=67.50, Acc=65.31 [H=15582, D=1044, S=6459, I=506, N=23085]
===================================================================
```

Figure 2. The performance of the HTK recognizer

E. Budget

The budget of the project was utilized in the following activities:

- Professional manpower employment including: Research Postdoc, and Research Assistants
- Temporary manpower for data entry: Recoding, Filtering, Segmentation, Text entry, Typesetting, Labeling, Diacritization,
- Conference attendance: according to the university rules.
- High quality audiovisual equipment
- A high-speed machine for speech recognition systems. Both the Sphinx and the HTK employ time-consuming algorithms for training and decoding that require a high-speed machine.
- Petty cash expenditures for miscellaneous purchases (Storage media, filing and backups)

2. Results Attained and Obstacles

A. **Results:** The results so far include preparing the speech corpus (29 hours) along with the corresponding diacritized transcription. Based on the prepared corpus, we have contributed in a number of conference and journal papers as the following:

Conference Papers:

1. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina ,"Arabic Speech Recognition: A Survey of the Literature", The 10th International Conference on Informatics and Systems Cairo, Egypt, May 9-11, 2016.
2. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Utilizing Long Distance Word Dependencies for Automatic Speech Recognition", The International Conference on Innovations in Information Technology (IIT'16) , United Arab Emirates University,28 - 30 November 2016.
3. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "An Empirical Study of Arabic Continuous Speech Recognition Performance", The International Conference on Computer Applications & Technology, ICCAT' 2017, from 28-29 January, in Cairo, Egypt.
4. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Effect of diacritization on Arabic Speech Recognition" The 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). Oct 11 - 13, 2017.

5. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Phonetic Tied-Mixture PTM Acoustic Model for Arabic Continuous Speech Recognition", The 18th International Arab Conference on Information Technology (ACIT'2017), Yassmine Hammamet, Tunisia.
6. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", ICNMLKD 2017 : 19th International Conference on Network, Machine Learning and Knowledge Discovery, Bangkok, Thailand, October 26 - 27, 2017.
7. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Exploring the Language Modeling Toolkits for Arabic Text", The International Conference on Electrical and Computing Technologies and Applications, 2017 (ICECTA'2017), November, 21-23, 2017, AURAK, UAE.
8. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "A Survey of Markov Chain Models In Linguistics Applications", Fifth International Conference on Data Mining & Knowledge Management Process (CDKP 2016) , November 12-13, 2016, Dubai, UAE
9. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina ," A Literature Survey of Arabic Speech Recognition", Second International Conference on Computing Sciences and Engineering (ICCSE 2018), March 11th to 13th 2018 - Kuwait University, Kuwait.

Journal Papers:

1. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Statistical Markovian Data Modeling for Natural Language Processing", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.7, No.1, January 2017. (Extended Version of Conference Paper 8).
   http://aircconline.com/ijdkp/V7N1/7117ijdkp03.pdf
2. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Impact of Phonological Rules on Arabic Speech Recognition", International Journal of Speech Technology.
   https://link.springer.com/article/10.1007/s10772-017-9440-2
3. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:11, No:10, 2017. (Extended Version of Conference Paper 6).
   https://waset.org/journal/Computer

4. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina, “Theoretical and practical models for Arabic speech recognition”, Submitted to International Journal of Applied Mathematics and Computer Science.
5. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina, " Performance Evaluation of Sphinx and HTK Speech Recognizers for Spoken Arabic Language”, Submitted to Engineering Applications of Artificial Intelligence.

B. **Obstacles:** we faced some obstacles such as:

- Repetitions in the speech files. During transcription process, we found that many audio files contains almost same contents for local news. The source of repetitions is the repeated news in As-Sabah TV archive, the source of our speech files. The repetition is not good in our work since the corpus has to have a wide range of different words. Intuitively, the quality of any corpus is related to the words diversity.
- The speech files contain some slang words such as :{ هوشة وهوشات قوية – الشرهه مو عليك الشره على الي مقعدك في بيته – غبقة – التكهك – اقعد مكانك – لاتحاتي – ريح واستريح – ياظالم لك يوم – لاتتغشمر - احنا عيال اليوم – چال بدل قال}.
- Some speakers pronounce sentences quickly, which make things harder for the transcriber to realize what is being spoken. In addition, such words are not suitable to be added to the developed corpus, as they are not clear.
- Some cases that the speaker wrongly pronounces a word in the first time and then repeat pronouncing it correctly in the second time. This case is not suitable to be added to the developed corpus as it will leads to problems in the language model.
- Some speakers are not linguistically qualified as they wrongly pronounce some words. This problem leads to ambiguity in words’ diacritization.
- Sometimes there is music along with speech. The problem is that if we cut the music part, this leads to ambiguity in the context which is not good in the training the acoustic model and the language model.
- The volume of the speech files is very low from the source that make things harder for the transcriber.

- As we have to return back As-Sabah TV archive, we spent long time to copy the archive in our hard disk. Then we prepared 5,579 speech files. Finally, we end up with 4,071 speech files with 47,727 unique words.
- Other reasons for selecting only 4,071 files is the difficulty of the transcription process. We realized that the allocated time is not enough to transcribe all prepared speech files. Another reason is the fixed number of speakers (30 speakers) in which increasing the size of the corpus is not beneficial with such limited number of speakers.
- We had a device crash during transcription, and it took about one month to continue the work after obtaining another machine.
- Our transcriber had eye exhaustion (i.e overwork) which require for some pauses (off) during the transcription process.
- Our transcriber has no special skills for quick typing which might be one reason of the delay. On the same time, it is not easy to replace the transcriber and retraining another person.
- The diacritization is very sensitive process that demands high level of concentration. The transcriber has to repeat the speech files many times in order to catch all information in the file. Hence, the diacritization is also a difficult task in this work. Hence, this process also includes some pauses for relaxing.

3. Closeness to the Original Research Plan

Based on the initial plan, we fulfilled the promised work. However, some slightly issues and differences from the proposed base plan and time line due to the manpower availability and time and space constrained of the available hardware. However, these issues did not affect the overall project output and promised research performance. We successfully prepared a corpus that contains 47,727 unique diacritized text and audio words. To the best of our knowledge, this is considered to be the largest known corpus of Arabic spoken language (multi speakers & mixed gender diacritized text and audio) that is fully revised and authenticated to the high standards. Regarding employing the CMU Sphinx and the HTK, the literature review shows that this is the first attempt to experimentally compare the CMU Sphinx and HTK recognizers for continuous Arabic speech. We performed the comparison that explores the differences between both speech recognition engines. The main conclusion shows that the CMU Sphinx outperforms the HTK recognizer. Sphinx is also better in some issues such as handling long speech files, since some of

the long speech files were discarded due to failure execution using the HTK (i.e., the training fails using long speech files). Sphinx is also better in terms of execution time as it takes less training and decoding time compared to the HTK. Finally, we have found that it is easier to perform an ASR task with Sphinx than HTK. The only issue with Sphinx is that it fails when the phonemes set has capital and small letters. For instance, if we use the character <B> to indicate a specific phoneme and, at the same time, use the character <b> to indicate another phoneme, then we get an error during training. On the other hand, this error did not appear in the HTK system. We also have found that HTK is better documented than Sphinx. In conclusion, more research is required to understand the reasons for the performance difference between both systems.

4. Chapters of the Report

This final report consists of 10 chapter and an appendix. The first chapter of this report is this executive summary. The next chapters of this final report will present the details of the main achievement and findings of the project including the following:

- Chapter 2: Literature Survey of Arabic Speech Recognition
- Chapter 3: Utilizing Long Distance Word Dependencies
- Chapter 4: Empirical Study of Arabic Continuous Speech
- Chapter 5: Effect of Diacritization on Arabic Speech Recognition
- Chapter 6: Phonetic Tied-Mixture PTM Acoustic Model
- Chapter 7: Modeling Capacity of Mel Frequency Cepstral Coefficients
- Chapter 8: Language Modeling Toolkits for Arabic Text
- Chapter 9: Markov Chain Models In Linguistics
- Chapter 10: Performance Evaluation of Sphinx and HTK Speech Recognizers for Spoken Arabic
- Appendix Journal & Conference Papers

5. Closing Remarks

Upon completion this project, we have some recommendations that can help to produce more research based on the prepared corpus. For example, we used the HTK version 3.4.1 (HVite decoder). However, the HTK toolkit has another famous decoder, which is 'HDecode', but in this work, we used HVite. Similarly, we used CMU PocketSphinx decoder; however, CMU Sphinx has other decoders such as Sphinx 3 and Sphinx 4. In addition, a great research could be initiated regarding the Arabic phonemes to find the optimal phonemes set of the Arabic language. Language models are also good candidate for further research to investigate the possible of implementing "any-word" language model (i.e not n-gram). As a final statement, since we had to use two types of the scripts (Arabic and Roman) this mean that we need to find a way to unified the script for any comparison between two or more speech recognizers. It is also worthy to compare the findings of this project with other previously published works such as what available in [10][11][12][13].

References

[1] Available: https://cmusphinx.github.io/

[2] Available: http://htk.eng.cam.ac.uk/

[3] Gaida, Christian, et al. "Comparing open-source speech recognition toolkits." Tech. Rep., DHBW Stuttgart (2014).

[4] Available: http://kaldi-asr.org/doc/index.html

[5] Jelinek, Frederick, et al. "A Dynamic Language Model for Speech Recognition." HLT. Vol. 91. 1991.

[6] Brown, Peter F., et al. "An estimate of an upper bound for the entropy of English." Computational Linguistics 18.1 (1992): 31-40.

[7] Ruiz-Casado, M., Alfonseca, E. and Castells, P. 2007. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. Data & Knowledge Engineering. 61, 3 (2007), 484-499.

[8] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. Pearson, 2014.

[9] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

[10] About LDC | Linguistic Data Consortium: 2015. https://www.ldc.upenn.edu/about. Accessed: 2015- 12- 22.

[11] Linguistic Data Consortium - Linguistic Data Consortium: 2015. https://catalog.ldc.upenn.edu/. Accessed: 2015- 12- 22.

[12] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. Int J Speech Technol. 10, 4 (2007), 183-195.

[13] Hyassat, H. and Abu Zitar, R. 2006. Arabic speech recognition using SPHINX engine. Int J Speech Technol. 9, 3-4 (2006), 133-150.

CHAPTER 2:
LITERATURE SURVEY OF ARABIC SPEECH RECOGNITION

1. INTRODUCTION

Having human speech interpreted by a computer is called Automatic Speech Recognition (ASR). It is defined as the process of converting spoken language (sound waves) into a machine-readable text. With the fast growth of powerful communication devices, it is making man-machine interfaces even more valuable and pervasive. Developing commercial speech recognizers have been shown to be successful business interactive solutions in various industry sectors such as healthcare, telecommunication, banking and finance, retail and mall management, education, hospitality, governmental institutions, and travel,[1]. In the last decade, there has been great enthusiasm by developers to have this attractive property that can be of great advantage in the new technologies such as search engines, voice maps, communications, etc. Recently, speech was used in security and protection fields for authentication purposes (also called voiceprints). Findbiometrics list five unique applications of voiceprints; targeting the developers, hands free interface, call center authentication, proof of life, and multi-factor logical access control [2]. Nuance lists some of benefits of using voiceprints that include simpler authentication, wipe out fraud, and almost-instant return on investment [3]. Even though the utilizing of voiceprint in banking industry is not new, this technology has been transferred to Arab countries. For examples, Kuwait Finance House (KFH) uses a speech recognition platform for users' authentications. The service that was initially only available to VIP customers is now in the process of being rolled out to KFH's entire customer base, [4]. Abu Dhabi Commercial Bank (ADCB) has also turned to voiceprints, [5]. Fortunately, the communications infrastructure that already exist help to spread the voiceprint biometric rather than installing a new machine such as image scanner to read the fingerprint before sending the figure print for authentication.

In fact, globally utilizing speech recognition in natural language processing (NLP) and linguistic applications has pushed to utilize this technology for the Arabic language that has more than 380 million speakers [6]. Most importantly, the holy Quran that was revealed in Arabic has to be read in Arabic by the entire Muslim world. This constraint might reinforce Arabic speech research, and therefore the technology to server the holy Quran readers and learners. Reference [7] presented a speech recognition technique for verification of Quranic recitation of sound files and media. Reference [8] provided a structural overview of

speech recognition system for developing Quranic verse recitation recognition with Tajweed checking rules function.

However, speech recognition is not an easy task and there is a long way for efficiently utilizing speech recognition to fulfil people requirements. Despite the successive research attempts, the high accurate transcription of human natural spoken words (speech-to-text) is still a difficult task problem. In fact, speech processing is much complicated than other pattern recognition problems such as text or images classification. For illustration, while locating a particular text or an image has achieved great success, locating a particular speech segment of a particular word in a speech collection audio file is still an active research problem.

Speech recognition is classified as a multidiscipline field that includes machine learning, phonetics, linguistics, and signal processing. Accordingly, significant integration is required for satisfied performance. Unfortunately, while the remarkable evolution in the research toward enhanced ASR; Arabic research is still behind when compared to other languages such as English. In general, varying acoustic conditions, pronunciation variations, dialects, accent, age and others factors are all degrade the performance. Reference [9] presented various factors that affect the way in which words are pronounced such as assimilation, co-articulation, reduction, deletion, and insertion. The Arabic language has even more challenges such as morphological complexity and diacritization wherein short vowels are usually missed in formal writings.

In addition to the Arabic intrinsic challenges, there are some other logistic researching challenges such as resources availability. The absence of unified large continuous speech corpora is an obstacle that might restrain the research in this flourishing domain. It has been noticed that almost all-Arabic speech recognition studies have been investigated using in-house small corpora. This is unlike English language that has many common large corpora such as North America business (NAB) and Broadcast News switchboard, [10]. Working on common corpora saves time as well as gradually enhance the research since the outputs can be compared and improved. It is known in speech recognition community that creating a large speech corpus is time demand and extremely expensive task. Consequently, it is hard and might be inconvenient for individuals to perform such task. Reference [11] indicated that preparing large training corpora for dialectal Arabic acoustic modeling is too difficult compared to Modern Standard Arabic (MSA).

Even though-isolated words speech recognition is an important task for some digits and commands applications, the continuous or conversational speech recognition has had even more interest. For several years, there have been two well-known ASR engines that are used for speech-to-text task; Carnegie Mellon University (CMU) Sphinx and the Cambridge University Hidden Markov Model Toolkit (HTK). Both are

statistical based engines that are based on Hidden Markov models (HMMs). Developing a speech recognition engine is a complex task and requires highly expert staff; therefore, most researchers used the free (black box) Sphinx and/or HTK engines. However, some other techniques have been used in the literature such as artificial neural networks (ANNs) and support vector machines (SVM). Sphinx engine does support Arabic, but HTK does not support Arabic and conducting speech recognition research requires a transliteration process on the training text that can be performed, as an example, using the transliteration system available at [12]. Figure 1 shows the architecture of an ASR that include three knowledge databases: the acoustic models contains the trained HMMs, the language mode represents the statistical words co-occurrences, and the dictionary (also called pronunciation dictionary or vocabulary) has the pronunciation of each words in terms of phonemes, the basic unit of sounds.

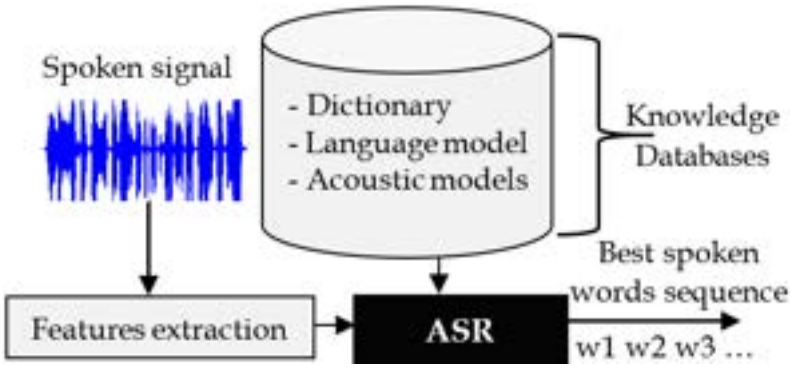


Figure 1. Automatic Speech Recognition (ASR) Architecture

The goal of this study is to present the recent advances in Arabic ASR with a particular highlight of the essential components of a typical ASR. The topics include the corpora used for both isolated-words and continuous speech. The corpora information include vocabulary size, nature of data, topics, speakers, number of male and female, age, etc. features extraction methods, classification approaches, phonemes sets, pronunciation dictionary, language models (LMs). The study also includes some new directions that could be investigated for Arabic.

This study is organized as follows. The next section presents the Arabic speech corpora of both isolated-words and continuous speech. Section 3 presents the phonemes set and pronunciation dictionary, followed by language models in section 4. The performance evaluation is demonstrated in section 5, and the conclusion and future works are presented in section 6.

2. ARABIC SPEECH CORPORA

The preliminary work in speech recognition requires in the first place to specify the type of speech; either isolated-words or continuous speech. Therefore, the speech corpora will be the first topic to be discussed in this survey. In isolated-words speech (also called discrete words), a pause are existed between digits or words while such constraint is not existed in continuous or conversational speech. Isolated-words speech recognition is characterized by easy to implement when compared to the continuous speech recognition that it suffers from co-articulation phenomenon, the critical factor for recognition results and performance.

2.1 Isolated-words speech recognition

In this subsection, we present the isolated-words corpora, the speech features, and the classifiers used. In general, isolated-words size is represented using the number of recorded speech files while continuous speech file is represented using the number of hours (all recorded speech files length). In the following, MFCC is the shorthand for Mel-Frequency Cepstral Coefficients, the widely used features for speech recognition; LPCC is the shorthand of Linear Predictive Cepstral Coefficients. The research contributions presented in Table 1 are date ordered, the earliest first. Otherwise indicated, all the researches listed in Table 1 are categorized as isolated-words speech recognition, MFCC features, and HMM-based classifiers.

Table 1. Isolated-words corpus information

Ref.	Year	Corpus Information, Features, Classifier/s
[13]	2001	The training set is composed by 50 speakers each of them uttered three times the ten digits. The test set comprises two groups, 30 speakers and 10 speakers. LPCC features were used.
[14]	2002	They used fuzzy NNs for recognition of isolated words. Cannot have access to the paper to present corpus information.
[15]	2003	Speech corpus consists of the 10 isolated digits, with 20 repetitions for each digit, using single male speaker. LPCC and MFCC features were used.
[16]	2004	The corpus contains of a total 1800 digits pronounced by 60 speakers (30 males and 30 females). For testing, they used 1000 digits pronounced by 50 others speakers (25 males and 25 females) .NNs classifier was used.
[17]	2006	The corpus consists of 92 speakers. (46 male and 46 female) pronounce each word two times where 20/92 of the corpus used for learning. (HMM,SVM) classifiers were used.

[18]	2007	Training set: 300 token (10 digits * 5 repetitions * 6 Moroccan speakers). Testing set: 30 token of different individuals.
[19]	2007	128 words for training and other 7 words for testing. Dynamic time warping (DTW) was used as similarity measure for classification.
[20]	2008	The corpus consists of 600 utterances (10 speaker, 10 words, 6 repetitions) split into 300 utterances for training and 300 utterances for testing. NNs was used for recognition.
[21]	2008	The corpus contains about 1.5 hours of commands and less than 1 hour of digits.
[22]	2008	The corpus contains Egyptian 59 men, (33 speakers for and 26 for testing). Speakers asked to utter 16 sentences of proverbs.
[23]	2008	The training set contains 340 tokens (17 speakers × 2 repetitions × 10 digits). For testing 1,700 tokens (17 speakers × 10 repetitions × 10 digits) were used. NNs was used for classification.
[24]	2009	The corpus was created from all 10 Arabic digits. A number of 60 Moroccan speakers (35 males and 25 females) were asked to utter all digits 5 times.
[25]	2010	The corpus contains 3650 speech files recorded by 13 speakers. Training set contains 3000 speech files and 650 for testing.

2.2 Continuous speech recognition

The research contribution toward continuous Arabic speech recognition is less than what we have seen in isolated-words that make sense as previously indicated regarding the difficulty of preparing a continuous speech corpus. However, there is considerable work initiated by the Linguistic Data Consortium (LDC). The LDC is an open consortium of universities, libraries, corporations and government research laboratories, [26]. More information about LDC Arabic speech corpora can be found at LDC catalog [27] that contains hundreds of (not free) holdings. One of important Arabic speech contribution is the work fielded by IBM in the Gale project that used LDC corpora. Gale project has many phases that gradually improve the performance. The Gale acoustic training set composed of approximately 1800 hours of transcribed Arabic broadcasts provided by LDC. The published work the described the phases include: The IBM 2006 Gale Arabic ASR System [28], The IBM 2009 GALE Arabic speech transcription system [29], and the IBM 2011 GALE Arabic speech transcription system [30]. LDC produced CallHome (CH) corpus of Egyptian Colloquial Arabic (ECA) [31]. This corpus is a collection of informal phone conversations between close friends or family members. This corpus has many sets as shown in Table 2, [32].

Table 2. LDC ECA CallHome datasets

collection	Number of conversations	Number of words	Number of hours
train	80	146,298	14
dev	20	32,148	3.5
eval96	20	15,584	1.5
eval97	20	17,670	1.8
h5_new	20	16,752	1.8
eval03	10	11,015	1.9

In speech recognition systems, it is highly recommended and even more accurate to use large speech collections. Reference [33] explains the meanings of large vocabulary speaker-independent continuous speech recognition as follows. Large vocabulary means that the corpus has vocabulary (unique words) of about 20,000 to 60,000 words. Speaker-independent is the classifier's ability to recognize the speech of people whose speech has never been exposed before. The continuous mean that the speech is recorded according to the human natural language.

One of the earliest attempts to develop a speech corpus was made by reference [34] who describes the OrienTel speech dataset. They indicated that the OrienTel is the first time makes an effort to create speech data on a large scale. The participants of OrienTel collected standard and colloquial varieties of Arabic in Saudi Arabia, the UAE, Egypt, Palestine, Tunisia and Morocco. GlobalPhone project produced a read speech corpus that was designed for the development and evaluation of large continuous speech recognition systems, [35]. Reference [36] presented a Saudi accented Arabic telephone speech database. It contains 96 hours that was collected on a telephone network during 2002 and 2003 using 1033 native speakers (51% males, 49% females). Reference [32] used CallHome corpus for morphology-based LMs at different stages of Arabic ASR. New versions were used by authors as Reference [37] used a new LDC test set called Dev07 that was distributed by LDC in March 2007 and consisting of 2.5 hours of speech (18186 words). The Reference [37] research work consists of using neural network LMs for Arabic ASR. Reference [38] developed an MSA broadcast news speech recognition system. The system was trained on 7.0 hours of a 7.5 hours and tested on the remaining half an hour. The corpus contains a total of 235 news items, 41 news items cover sport news and the rest of the items cover mainly economic news. Among the

speakers, 88 of the news items were by female speakers. Reference [21] presented a Holly Qura'an corpus that contains about 18.5 hours. However, the actual challenge is developing a broadcast news corpus since the holy Quran recordings are already available. However, Reference [21] indicated that it takes about 732 working hours to build their holy Quran corpus. Reference [11] used MSA acoustic models as multilingual models to decode Egyptian dialect. They chose the Nemlar broadcast news speech corpus to build the acoustic models. The corpus consists of 40 hours of MSA news broadcast. The total number of speakers is 259 with a lexicon of 62,000 words. Reference [39] presented a MSA continuous speech corpus composed of 200 sentences pronounced by 300 Algerian native speakers selected from eleven regions of Algeria. Reference [40] developed an Arabic ASR system based on phonetically rich and balanced speech corpus. That work was based on 8,043 utterances gathered from 8 (5 male and 3 female) speakers resulting about 8 hours of speech. The round robin testing approach was applied.

The speech dataset is used to train the acoustic models that are the statistical representations of the MFCC speech features vectors. Hence, the acoustic models represent the statistical co-occurrences between phonemes. HMM is one of the most common type to represent acoustic models as the example represented Figure 2. In the figure, each number represents a phoneme and the all three phonemes represent what is called triphone that represents a phoneme surrounded by specific left and right phonemes. There are even more details as each phoneme is internally represented using three states (beginning, middle, end) using mixture density Gaussian distributions. Hence, Figure 2 shows how HMMs used to represent acoustic models.

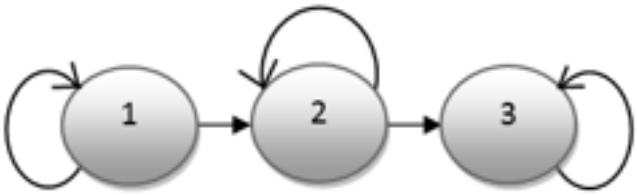


Figure 2. HMM-based Triphone

The procedures that can be applied to produce a continuous speech corpus are summarized as follows. A number of audio files are recorded (under equal conditions) from radio and/or TV broadcast news. It is highly recommended to cooperate with local or international stations of radio and TVs to get some prerecorded speech collections (e.g. a part of the archive). This cooperation avoids the labor and cost intensive manual generation to produce speech corpora. If the lengths of audio files are too long, these audio files should be split into small audio files of 10-30 seconds. In fact, there is no problem to have

more than 30 seconds, but the speech recognizer might fail during the training process to align the recordings with their phonetic transcription. Hence, it will be productive and efficient if the recordings are short. However, if the recordings are long say 10 minutes, initial alignment might be fail and therefore causing a problem during training process. During recording to create a corpus, the following parameters are set as indicated by Sphinx [41]: Sampling Rate = 16 kHz (or 8 kHz, depending on the training data), No. of Bits = 16 bits, No. of Channels = Mono (= single channel), and File Format = ".wav". Once have the recordings completed, the spoken words in the audio files are transcribed and then diacritized. Based on the diacritized text, the pronunciation dictionary is produced. After this step two models can be created, the acoustic model and the language model. The three knowledge bases (acoustic models, pronunciation dictionary, and language model) will be ready for setting up and testing a continuous speech recognition system.

3. ARABIC PHONEMES SET

Speech recognition task is generally performed using one of three approaches based on the basic units of classification. The units include words, syllables, and phonemes. Word-based recognition has a drawback that it needs large number of data for training. A syllable that is a single unit of written or spoken word has relatively smaller number of used units and runs faster than word-based recognition. However, the recognizer which depends on the distinctive unit of sound (i.e., phoneme) is a wide spread approach since it is easy to train, [22]. Either making a research using isolated words or continuous words speech, phonemes set has to be defined before training stage, of course if the research is based on phoneme-based method. Language linguistics define the phoneme set of a particular language after studying and careful classification of speech sounds. In this section, we demonstrate the phonemes-based research that found in the literature. Many studies indicated that Arabic language has 34 phonemes (28 consonants and 6 vowels) such as References [20], [25], and [38]. However, the number of Arabic phonemes is a debated by researchers. For example, Reference [42] used a phoneme set that contains 46 phonemes for developing a tool that is used to create a pronunciation dictionary. The tool generated by Reference [42] had later used in some researches such as Reference [38] and Reference [43]. Reference [19] indicated that Arabic has at least one hundred twelve phonemes as they considered that every letter has four diacritics, therefore, four phonemes. Reference [23] used 37 MSA phoneme as given by Language Data Consortium (LDC). Reference [11] indicated that MSA consists of 38 phonemes, 28 are original consonants, 4 are foreign and rare consonants and 6 are vowels. An example of pronunciation dictionary's entries as produced by

Reference [42] is shown in Figure 3. It shows some words and the phonetic transcription (the phonemes) of each word.

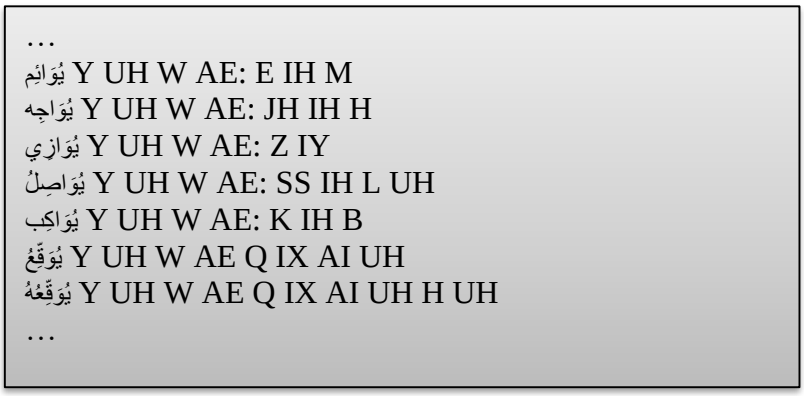


Figure 3. Some entries of a pronunciation dictionary

4. LANGUAGE MODELS

Language models is a statistical component of automatic speech recognition systems that is used to estimates the most likely co-occurring sequence of words (possible words) in the language. In combination with acoustic models, language models are used to recognize the spoken words given the sequence of speech signal features vectors. Reference [15] indicated that the language model can be incorporated to constrain the recognizer to recognize only valid word sequences. Reference [25] demonstrated that one benefit of the language model is to reduce the search recognition probability by forcing the recognition to follow certain rules to ensure a better accuracy in the recognition output. Reference [23] indicated that the absence of diacritics in Arabic text decreases predictability in the language model. In general, much larger amounts of text leads to develop more powerful and enhance the goodness of the language models. In speech recognition, N-grams language model is used to indicate for a contiguous sequence of n items that are also called unigram, bigrams, and trigrams of the language text. In case of isolated words speech recognition, context free grammar is used. While language models are mainly used in large continuous speech recognition systems, Context-Free Grammar is also used in speech recognition to to predict subsequent words in small corpora or isolated words speech recognition. As language models Context-Free Grammar is used to reduce the possible words to be considered as a next word. While statistical language models generally describe complex language, Reference [44] indicated

that grammars describe very simple type of the language for command and control, and they are usually written by hand. Therefore, grammars usually do not have probabilities for word sequences, but some elements might be weighed. The perplexity is the common way to evaluate N-gram language model, it is an indication of the average number of words that can follow a given word. Reference [39] used a bigram language model for continuous speech recognition of the Arabic language. Reference [38] used both bi-grams and trigrams for the language model. Reference [40] used different language models (bigram and context free grammar) for continuous Arabic speech recognition system. Reference [21] used the tool available in SPHINX-IV to generate the N-grams language model. Reference [32] used n-gram models up to an order of n = 6 for improving the perplexity of the language models. Reference [37] indicated that using of neural network language models for Arabic broadcast news and broadcast conversations outperforms the 4-Grams based language model. Reference [25] used an Arabic grammar file that contains some words and commands to be represented in their isolated words speech system. Reference [45] investigated the use of morphology-based language model at different stages in a speech recognition system for conversational Arabic. Reference [18] used the tool available in CMUSphinx to specify the grammars of spoken Arabic digits recognition system. Figure 4 shows an example of the entries in a language model generated using CMU statistical language tool [44] of the corpus produced by Reference [38]. The number besides the n-grams in Figure 4 is related to the probabilities of occurring for each case.

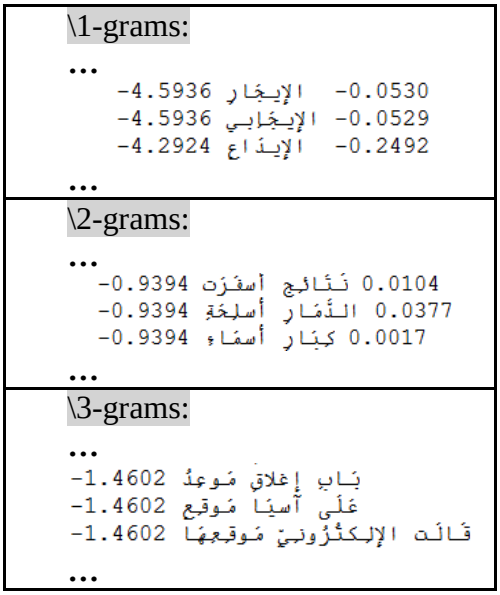


Figure 4. Some entries of a language model

5. PERFORMANCE EVALUATION

The isolated-words speech recognition is generally measured using recognition accuracy rate that is the percentage of correctly recognized patterns such as words or digits. However, in continuous speech, word error rate (WER) is the common metric to measure performance of ASRs. WER is computed using the following formula: WER=(S+D+I)/N, Where: S is the number of substitutions words errors, D is the number of the deletions words errors, I is the number of the insertions words errors, N is the number of words in the testing set. The word accuracy can be measured using WER formula: Word Accuracy = 1 – WER.

Reference [21] said that the WER of MSA is in the range: 15–20%. However, WER depends on some other parameters such as the size of training corpus, the number of words in the dictionary, and the perplexity of the language model. We emphasize that measuring performance is required to show if the achieved WER is statistically significant. It has been noticed that the majority of Arabic speech recognition authors do not use the appropriate statistical significant test to prove their performance. Therefore, a baseline system should be developed and tested, and then the output of the proposed method is compared with the baseline results to find the statistically significance enhancement. If the result does not have statistically significant enhancement, then the enhancement is considered accidental and the proposed

method is not as strong method to enhance the classification performance. Reference [46] has a description how to perform a statistical significant test.

In this section, we present some of performance evaluations of continuous speech since it has more interest and challenge than isolated-words speech recognition. Reference [32] performed performance evaluation for morphology based LMs. The WER improved by 1.8% and 1.5% for two different test sets. Reference [37] achieved WER improvements by 0.8% and 3.8% for 2 different configurations of neural probabilistic models. Reference [38] achieved a WER of 13.66% of broadcast news corpus. Reference [21] achieved a WER of 46.182% using the holy Quran corpus. Reference [11] reported a recognition accuracy of 99.34% for Egyptian Colloquial Arabic. Reference [39] presented an accuracy rate of 91.65 % for MSA continuous speech corpus. Reference [40] demonstrated a WER of 11.27% and 10.07% with and without diacritical marks respectively for MSA continuous speech corpus. Table 3 shows some of WER for some systems on different English speech corpora, [33].

Table 3. WERs for a number of ASRs (English corpora)

Pronunciation Corpus	Vocabulary	WER %
TI Digits	11 zero-nine, oh	0.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech (CST)	64000+	20

6. CONCLUSION

The review of Arabic speech recognition shows that the research is still in the raw stage specially the continuous speech type. Most of the research attempts belong to isolated-words speech recognition. However, there are some research activities towards continuous speech recognition. The major obstacle is the corpora availability. Hence, reinforcing Arabic speech recognition needs affording the professionality producing of large Arabic continuous speech collections (corpora).

The following are some research topics that can be investigated for Arabic. Even there are some studies regarding pronunciation variations such as Reference [43], more studies are needed to tackle this phenomenon. The researches could also investigate using long distance words relationships in the

language models. The traditional N-grams language models assume that a word is only influenced by a few proceeding words, typically one or two. However, it is much better to account for longer-distance constraints. Data mining association rules algorithms such as Apriori could help to find some words association rules. Then, N-best ASR hypotheses can be used in combination with words association rules for rescoring the ASR outputs for better accuracy. The semantic relationships can also be used to enhance ASR performance. The semantic and syntactic relationships can be obtained using knowledge bases such as WordNet [47]. The syntactic relationships can be obtained in combination of part of speech tagging and data mining algorithms. The deep neural network hidden Markov model (DNN-HMM) hybrid architecture is another direction that can be employed for Arabic [48]. Further research is needed to clarify the realistic Arabic phoneme set. Reference [49] had some preliminary work in this direction. However, an approach is needed for automatic extracting the Arabic phonemes using data-driven approaches and clustering methods. There has been current research to train a recognizer using synthesized data, because if it is possible then we could get as much data as we want in a preferred domain or with a new and large vocabulary,[38].

References

[1] Emerging Technologies: 2017. *http://www.em-t.com/*. Accessed: 2017- 12- 4.

[2] Voice Month: 5 Unique Applications of Voice Biometrics - FindBiometrics: 2017. *http://findbiometrics.com/voice-month-5-unique-applications-of-voice-biometrics-22186/*. Accessed: 2017- 12- 4.

[3] Nuance | Nuance - PDF, Customer Service, HIM, and Speech Recognition Solutions: 2017. *http://www.nuance.com/*. Accessed: 2017- 12- 4.

[4] Speech Recognition Case Study Kuwait Finance House (KFH): 2017. *http://*www.*em-t.com/content/speech-recognition-case-study-kuwait-finance-house-kfh*. Accessed: 2017- 12- 4.

[5] ADCB: 2017. *http://www.adcb.com/*. Accessed: 2017- 12- 4.

[6] Mubarak, Hamdy, and Kareem Darwish. 2014.Using Twitter to collect a multi-dialectal corpus of Arabic. *ANLP* (2014): 1.

[7] Mohammed, A., Sunar, M. and Hj Salam, M. 2015. Quranic Verses Verification using Speech Recognition Techniques. *Jurnal Teknologi*. 73, 2 (2015).

[8] Jamaliah Ibrahim, N., Yamani Idna Idris, M., Razak, Z. and Naemah Abdul Rahman, N. 2013. Automated tajweed checking rules engine for Quranic learning. *Multicultural Education & Technology Journal*. 7, 4 (2013), 275-287.

[9] Strik, H. and Cucchiarini, C. 1999. Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*. 29, 2-4 (1999), 225-246.

[10] Rabiner, L. and Schafer, R. 2007. Introduction to Digital Speech Processing. *FNT in Signal Processing*. 1, 1â€“2 (2007), 1-194.

[11] Elmahdy, Mohamed, et al. 2009. Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition. *Natural Language Processing*. *SNLP'09. Eighth International Symposium* on. IEEE, 2009.

[12] Arabic Transliteration/Encoding Chart: 2017. *http://languagelog.ldc.upenn.edu/myl/ldc/morph/buckwalter.html*. Accessed: 2017- 12- 4.

[13] Bahi H, Sellami M. 2001. Combination of vector quantization and hidden Markov models for Arabic speech recognition. *ACS/IEEE international conference on computer systems and applications*, 2001

[14] Alimi AM, Ben Jemaa M. 2002. Beta fuzzy neural network application in recognition of spoken isolated Arabic words. *Int J Contr Intell Syst* 30(2), Special issue on speech processing techniques and applications

[15] Elmisery FA, Khalil AH et al.2003. A FPGA-based HMM for a discrete Arabic speech recognition system. In: Proceedings of the *15th international conference on microelectronics, 2003. ICM 2003*

[16] Amrouche, Abderrahmane, and Jean Michel Rouvaen. 2003. Arabic isolated word recognition using general regression neural network. *Circuits and Systems, 2003 IEEE 46th Midwest Symposium* on. Vol. 2. IEEE, 2003

[17] Bourouba H, Djemili R et al. 2006. New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition. *2nd Information and Communication Technologies*, 2006. ICTTA'06

[18] Satori H, Harti M, Chenfour N.2007. Introduction to Arabic speech recognition using CMU Sphinx system. *Information and communication technologies international symposium proceeding ICTIS07*, 2007

[19] Haraty, R. and El Ariss, O. 2007. CASRA+: A Colloquial Arabic Speech Recognition Application. *American Journal of Applied Sciences*. 4, 1 (2007), 23-32.

[20] Essa EM, Tolba AS et al. 2008. A comparison of combined classifier architectures for Arabic speech recognition. *International conference on computer engineering and systems*, 2008. ICCES 2008

[21] Hyassat, H. and Abu Zitar, R. 2006. Arabic speech recognition using SPHINX engine. *Int J Speech Technol*. 9, 3-4 (2006), 133-150.

[22] Azmi M, Tolba H,Mahdy S, FashalM.2008. Syllable-based automatic Arabic speech recognition in noisy-telephone channel. In: *WSEAS transactions on signal processing proceedings, World Scientific and Engineering Academy and Society (WSEAS)*, vol 4, issue 4, pp 211–220

[23] Alotaibi, Y. 2008. Comparative Study of ANN and HMM to Arabic Digits Recognition Systems. *eng*. 19, 1 (2008), 43-60.

[24] Satori, Hassan, et al. 2009. Investigation Arabic speech recognition using CMU sphinx system. *Int. Arab J. Inf. Technol*. 6.2 (2009): 186-190.

[25] Al-Qatab, Bassam AQ, and Raja N. Ainon. 2010. Arabic speech recognition using hidden Markov model toolkit (HTK). *Information Technology (ITSim), 2010 International Symposium* in. Vol. 2. IEEE, 2010.

[26] About LDC | Linguistic Data Consortium: 2017. *https://www.ldc.upenn.edu/about*. Accessed: 2017- 12- 4.

[27] Linguistic Data Consortium - Linguistic Data Consortium: 2017. *https://catalog.ldc.upenn.edu/*. Accessed: 2017- 12- 4.

[28] Soltau, Hagen, et al. 2007. The IBM 2006 Gale Arabic ASR system. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference* on. Vol. 4. IEEE, 2007.

[29] Kingsbury, Brian, et al.2011. The IBM 2009 GALE Arabic speech transcription system. *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on. IEEE, 2011.

[30] Mangu, Lidia, et al. 2011. The IBM 2011 GALE Arabic speech transcription system. *Automatic Speech Recognition and Understanding (ASRU)*, 2011 IEEE Workshop on. IEEE, 2011.

[31] CALLHOME Egyptian Arabic Speech - Linguistic Data Consortium: 2017. *https://catalog.ldc.upenn.edu/LDC97S45*. Accessed: 2017- 12- 4.

[32] Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K. and Stolcke, A. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*. 20, 4 (2006), 589-608.

[33] Jurafsky, D. and Martin, J. 2000. Speech and language processing. Prentice Hall.

[34] Siemund, Rainer, et al. 2002. OrienTel—Arabic speech resources for the IT market. *LREC 2002 Arabic Workshop*. 2002.

[35] Schultz, Tanja. 2002. Globalphone: a multilingual speech and text database developed at karlsruhe university. *INTERSPEECH*. 2002.

[36] Alghamdi, M., Alhargan, F., Alkanhal, M., Alkhairy, A., Eldesouki, M. and Alenazi, A. 2008. Saudi Accented Arabic Voice Bank. *Journal of King Saud University - Computer and Information Sciences*. 20, (2008), 45-64.

[37] Emami, Ahmad, and Lidia Mangu. 2007. Empirical study of neural network language models for Arabic speech recognition. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop* on. IEEE, 2007.

[38] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. *Int J Speech Technol*. 10, 4 (2007), 183-195.

[39] Selouani, Sid Ahmed, and Malika Boudraa. 2010. Algerian Arabic speech database (ALGASD): corpus design and automatic speech recognition application. *Arabian Journal for Science and Engineering* 35.2C (2010): 158.

[40] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. 2012. Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. *International Arab Journal of Information Technology (IAJIT)* 9.1 (2012): 84-93.

[41] Frequenty Asked Questions (FAQ) [CMUSphinx Wiki]: 2017. *http://cmusphinx.sourceforge.net/wiki/faq*. Accessed: 2017- 12- 4.

[42] Ali, M., Elshafei, M., Al-Ghamdi, M. and Al-Muhtaseb, H. 2009. Arabic Phonetic Dictionaries for Speech Recognition. *Journal of Information Technology Research*. 2, 4 (2009), 67-80.

[43] AbuZeina, D., Al-Khatib, W., Elshafei, M. and Al-Muhtaseb, H. 2011. Cross-word Arabic pronunciation variation modeling for speech recognition. *Int J Speech Technol*. 14, 3 (2011), 227-236.

[44] Building Language Model [CMUSphinx Wiki]: 2017. *http://cmusphinx.sourceforge.net/wiki/tutoriallm*. Accessed: 2017- 12- 4.

[45] Vergyri D., Kirchhoff K., Duh K., and Stolcke A. 2004. Morphology Based Language Modeling for Arabic Speech Recognition. *in Proceedings of Interspeech, Germany*, pp. 2245-2248, 2004.

[46] Plötz T. 2005. Advanced stochastic protein sequence analysis, Ph.D. thesis, *Bielefeld University*

[47] Ruiz-Casado, M., Alfonseca, E. and Castells, P. 2007. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data & Knowledge Engineering*. 61, 3 (2007), 484-499.

[48] Dahl, G., Dong Yu, Li Deng, and Acero, A. 2012. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*. 20, 1 (2012), 30-42.

[49] Nahar, Khalid MO, et al. "Data-driven Arabic phoneme recognition using varying number of HMM states." Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on. IEEE, 2013.

[50] Prof Allan Ramsay (BSc, MSc, PhD) research profile - research | The University of Manchester: 2017. *http://www.manchester.ac.uk/research/Allan.ramsay/research*. Accessed: 2017- 12- 4.

CHAPTER 3:
UTILIZING LONG DISTANCE WORD DEPENDENCIES

1. Introduction

Statistical language models, also called n-gram models, are very popular and successfully used in different computational linguistic fields. For example, statistical language models are integral part of the state-of-the-art automatic speech recognizers (ASR) systems such as Sphinx [1] and HTK [2]. Trigram is the basis of the classical language models that assumes the next words is predicted based on the two immediately preceding words. Hence, n-gram model assumes that a word is only influenced by the (n-1) proceeding words, typically one or two words. It is indicated in [3] that due to memory and computation requirements, the value of (n) is restricted to two or three for bigram or trigram language models, respectively. In speech recognition community, language models also called grammars. Despite language models are known as probabilistic models, however, it could be non-probabilistic (i.e. "any word" grammar).

Language modeling research has long history of improvements as indicated by [4]. He indicated that there are many improvements over trigram simple model including caching, clustering, higher-order n-grams, skipping models, and sentence-mixture models. However, not all of the previous methods proved to be useful. For example, reference [4] showed that even using a very large corpus for n-gram model, very small improvements occurred, where n is larger than 5.

Even the statistical language models are quite success and have proven to be reasonable, however, longer contexts deserved for better language modeling representation. Such longer contexts (i.e. long words' relationships) go by the name of long-distance dependencies (LDDs) that obtained by looking at a wider window beyond n-gram. Discussing LDDs is not new, for example, it is indicated in [5] that the n-gram is weak in terms of capturing LDDs. Therefore, it could be better and more appropriate if the textual corpus used to extract long-distance words relationships. This research discusses a new knowledge-based method to mine the words co-occurrences among a data collection of textual sentences.

In fact, employing LDDs is extremely important since it is the key to the performance of many natural language processing (NLP) applications such as ASR systems, parsing, tagging, translation, etc. In this regard, the important aspect is how to find such words-relationships and how to exploit them in NLP systems. In this study, we propose using the predictive Apriori data-mining algorithm to extract the LDDs. We also propose an algorithm for scoring the N-best list of an ASR system. Intuitively, the proposed method is not a

replacement of the celebrated conventional language models, but it is a complementary knowledge base for farther enhancement.

The predictive Apriori algorithm used in this work is available in Weka machine learning tool [6]. In fact, finding LDDs is computational expensive and is feasible for small textual corpora that have relatively short sentences. This is the reason why we chose to demonstrate this work for ASR corpora that usually have short recordings that corresponds to short sentences. The reason of such short recordings is that the speech recognizer should be able during training to align the recordings with their phonetic transcription; if the recordings are long then initial alignment might be fail and therefore causing a problem during training process.

In the next section, we present the language models limitations. Section 3 presents the literature review, followed by the experiments setup and results in section 4 and the ASR N-best rescoring using LDDs in section 5. We conclude in section 6.

2. Language Models Limitations

The standard language model has some limitations and constrains as demonstrated by a number of researchers. It is indicated in [7] that words relevant to predicting the next word might lay in any position beyond the scope of a word trigram. The n-gram models are constrained in their inability to take advantage of dependencies longer than *n* in a sentence [3]. The work in [9] demonstrated two drawbacks when considering neighbor words with fixed window size: 1) the actual relation between words within the sentence is ignored, thus the long-distance context cannot be captured for training. 2) Word embeddings learned from flat context are not isomorphic among languages due to the order difference between diverse languages. The classic text representation methods seldom consider the role of the words order in the texts for the semantic representation, and it is supposed that the words are independent of each other [10]. In the same meaning, reference [5] demonstrated that the trigram model is unable to characterize word dependence beyond the span of three successive words.

The study in [8] used clustering as an alternative way of dealing with the data sparseness problem to explore the effectiveness of cluster-based higher-order n-gram models. Similarly, the presented study in [11] shows that the language models do not describe the constraint relationships between words and words or sentences and sentences. They also indicated that LDDs could be used in other domains such as text classification, text clustering, text summarization, and so on. For farther exploring the the deficiency of standard language models, the work in [12] demonstrated that the n-gram model works in terms of discrete units that have no inherent relationship to one another. The work in [13] presented that the local structure (n-gram) constraint is

a key limitation in many tasks, since natural language contains a great deal of nonlocal structure (LDDs). The vector space model (VSM) that is widely used for text representation in information retrieval (IR) models assumes that words occur independently (i.e. bags-of-words), which is not exactly appropriate to natural language. It is demonstrated in [15] that many of the context dependencies in natural language occur beyond a three-word window.

Even the Arabic language is widely spoken by hundreds of millions (approximately 400 millions); still, there is little research to improve the the linguistic applications through LDDs or lexical databases. Most research studies found in the literature show that Arabic research focusses on the conventional standard language models that restrict n-gram to the neighboring words with fixed window size. That is, the research in the Arabic language mainly discusses local dependencies (i.e. short-distance model). For the above-mentioned deficiencies of standard language model, we have initiated this research.

3. Literature Review

Capturing long distance dependences was discussed in many studies as follows. An empirical study was presented in [6] on two techniques that used to generate LDDs; linguistically motivated word skipping and predictive clustering. They presented that the two techniques significantly outperform word trigram. The work in [3] used dynamic cache language models and context-free grammars to captures topic-related dependencies of words within and across sentences. Reference [10] proposed head-driven phrase structure grammar (HPSG) for LDDs. HPSG includes three aspects: surface oriented, constraint-oriented and strict lexicalism. A method for capturing the long distance dependency presented in [5] as word activation forces-based language model. A graph-based long-distance dependency method for LDDs language models presented in [11]. Reference [16] discussed the notion of using probabilistic context free grammars for modeling LDDs. The work in [13] showed how to account for the long distance structure with Gibbs sampling, a simple Monte Carlo method used to perform approximate inference in factored probabilistic models. Reference [14] presents topical n-grams, a topic model that discovers topics as well as topical phrases. Reference [15] proposed a language modeling approach to capture the preferred relationships between words over a short or long distance through the concept of mutual-information (MI)-Trigger pairs. A method based on extended the fixed surrounding words presented in [9], the approach based on learning distributed representations from dependency structure of a sentence that can capture long distance relations. A method for capturing the long distance common syntactical rules for the Holy Quran was presented in [24].

4. Experiment Setup and Results

1) Data Set

In this section, we describe the corpus that used to generate the LDDs. The corpus contains 6145 short sentences that belong to sport and economy categories. The total number of unique words in the vocabulary is 11,576. This textual collection is a part of a speech corpus that contains 7.57 hours speech recordings for training and testing. Naturally, the audio files were not used in this study, as generating LDDs only requires the transcription (i.e. the textual forms) of the 6,145 audio files. The maximum sentence's length of the corpus is 30 words. The corpus was originally designed for Arabic continuous speech recognition systems as described in [17].

It is worthy to indicate that we initially started working using Reuters-21578 data set, but it was noticed that extracting LDDs for long sentences with huge vocabulary is infeasible. That is, we run the predictive Apriori algorithm to find LDDs, but we had no results even with so long time waiting, therefore, we stopped execution and moved towards a smaller corpus. Therefore, for efficient extracting LDDs, a high performance computing (HPC) environment is required especially for long sentences. Section VI explains how to exploit LDDs to enhance ASR systems

2) Predictive Apriori

To extract the LDDs, we implemented the predictive Apriori data-mining algorithm. In general, data mining algorithms are used in many areas such as health, marketing, communications, etc. For example, three algorithms (Apriori, Predictive Apriori and Tertius) presented in [19] for analyzing the information available on sick and healthy individuals and taking confidence as an indicator, females were seen to have less chance of coronary heart disease then males. For more information on association rules, reference [20] has a survey on such rules. The predictive Apriori algorithm is a Weka class implementing the predictive Apriori algorithm to mine association rules. It searches with an increasing support threshold for the best 'n' rules concerning a support-based corrected confidence value. The implementation of this algorithm in Weka follows the reference [21], as the rule is added if the expected predictive accuracy of this rule is among the 'n' best and it is not subsumed by a rule with at least the same expected predictive accuracy.

3) **Implementation**

The corpus described in the previous section was used to obtain the best association rules. The Weka machine-learning tool used to implement predictive Apriori algorithm. The experiments were implemented using a relatively high-speed machine with the following specifications: Intel(R) i7, CPU 3.4GHz, and 16.0 GB of RAM. However, for large data, HPC is required. In our execution environment, it took about 130 processing hours to produce the best 300 rules as shown in Fig. 1. Nevertheless, it took few hours to generate the rules using Apriori algorithm, as the n best rules are not sorted. The results included two types of LDDs relations. The consecutive and the nonconsecutive words relations. The following subsections demonstrated examples of such relationships. Fig. 1 shows the algorithm output starting from the highest accuracies. We chose to extract the best 300 rules. In the figure, x is the shorthand of word, so x5 means the word at the position 5.

```
=== Associator model (full training set) ===
PredictiveApriori
==========================
Best rules found:
1. x5= المئة 36==> x4= في  36   acc :(0.97338)
2. x1=  يذكر 17 ==> x2= أن   17 acc:(0.94723)
3. x10= المئة 17 ==> x9= في    17 acc:(0.94723)
...
299. x1= وتجدر 2 ==> x2=الإشارة x3= إلى  2  acc:(0.74998)
300. x1= وتجدر 2 ==> x2=الإشارة x4= أن 2    acc:(0.74998)
```

Fig. 1. 300 Best rules Using Predictive Apriori

Fig. 1 shows a part of the best 300 rules, even the rules shown in Fig. 1 are related to consecutive rules; however, there are some other rules that are related to nonconsecutive words as shown in the following subsections. Despite we focus on nonconsecutive words relationships; nevertheless, the consecutive words relationships are also important for NLP applications.

4) **Consecutive Words Relations**

The standard language model represents words relationships of consecutive words. The predictive Apriori algorithm provides similar relationships sorted based on common rules found in the corpus. The only difference is that the standard language models generate plain words sequences with the corresponding probabilities, while the predictive Apriori algorithm generate the association rules based on words co-occurrences with the corresponding accuracies. Table I presented some examples of such rules. The table also shows the accuracies associated with each rule.

TABLE I. EXAMPLE OF CONSECUTIVE WORDS RELATIONS

Rule type	Example
2 consecutive words	x5=المئة 36==> x4=في 36 acc:(0.97338) The rule indicates that the occurrence of word (“المئة” =”percent”) at the fifth position in the sentence occurred with the word (“في” = preposition means “in, at, on”) 36 times in the entire corpus. The rule also shows that the accuracy of this rule is 0.97. This rule is the highest rule among the obtained rules.
3 consecutive words	x2=ارتفاع x4=النفط 6 ==> x3=أسعار 6 acc:(0.87495) The rules indicates that the word (“ارتفاع” =”rise”) and the word (“النفط” = “oil”) associate with the word (“أسعار”= “prices”). This rule appears 6 times in the entire corpus with accuracy equal to 0.87.
4 consecutive words	x1=المزيد x4=في 6 ==> x2=من x3=التفاصيل 6 acc:(0.87495) This rule shows the relation between four words.

5) **Nonconsecutive Words Relations**

The nonconsecutive words includes associative rules between two, three, and four words. Table II presented some examples of such relations. The table also shows the accuracies associated with each rule.

TABLE II. EXAMPLE OF NONCONSECUTIVE WORDS RELATIONS

Rule type	Example
2 nonconsecutive words	x1=البالغ 4==> x4=مليون 4 acc:(0.8333) This rule shows the occurrence of a word at position # 1 and a word at the position # 4 among the corpus sentences. These nonconsecutive words relation appears 4 times with accuracy of 0.833.
3 nonconsecutive words	x1= وأربعة 3 ==> x3=سنتا x4= للبرميل 3 acc:(0.79997) This rule shows a rule between a word at the position #1 and a word at the position #3 and a word at the position #4.
4 nonconsecutive words	x2=النصف x3=الأول x6=العام 5==>x5=هذا5 acc:(0.8571) This rule shows a rule between a word at the position #2, a word at the position #3, a word at the position #5, and a word at the position #6.

Even the examples provided show the relations up to four words, however, more than 4 words relations can be extracted if such co-occurrences found in the corpus.

5. N-best List Rescoring

Speech recognition is the process of converting speech into machine-readable text. Three components are mainly used to perform recognition task; the acoustic models, the pronunciation dictionary, and the language model. An ASR generally has the option to produce the N-best list that contains the best recognition hypotheses. It is indicated in [21] that the Viterbi speech-decoding algorithm is an approximation algorithm. It actually computes an approximation of the most probable word sequence, instead of computing the most probable word sequence. The reason is that the pronunciation variants probabilities' mass is split up among different pronunciations. Therefore, the Viterbi algorithm ignores the correct word that has many-pronunciations and favor an incorrect word with only one pronunciation path. Hence, performance could be enhanced using N-best hypotheses rescoring. Hence, we propose to exploit the extracted LDDs for N-best list rescoring as shown in Fig. 2. The figure shows that the ASR decoder (supposing) recognize a speech file as Sentence 1. However, it might be better if the N-best list is rescored using LDDs as a double check step to obtain better results (it is supposed Sentence 4). The rescoring could be based on counting the number of association rules in each hypothesis. We have not empirically investigated the performance using rescoring step as the number of the obtained rules relatively low. However, it is a future research direction to implement this method for large corpus with very rich rules.

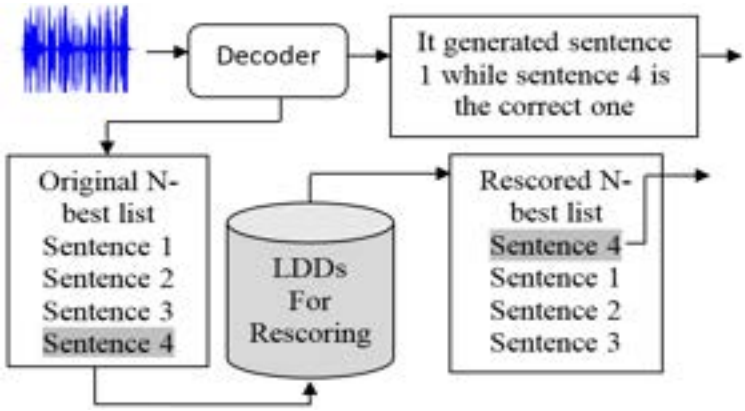


Fig. 2. Rescoring N-best list

Rescoring N-best list is not new. For example, Reference [22] demonstrated the usefulness N-best rescoring using syntactic trigrams. Reference [23] compared the efficacy of a variety of language models for rescoring word graphs and N-best lists generated by a large vocabulary continuous speech recognizer. Hence, the main

contribution of this work is to develop an intelligent method that can raise the hypothesis number four (as an example in the top part of Fig. 3) to be the first choice as shown in the lower part of Fig. 3.

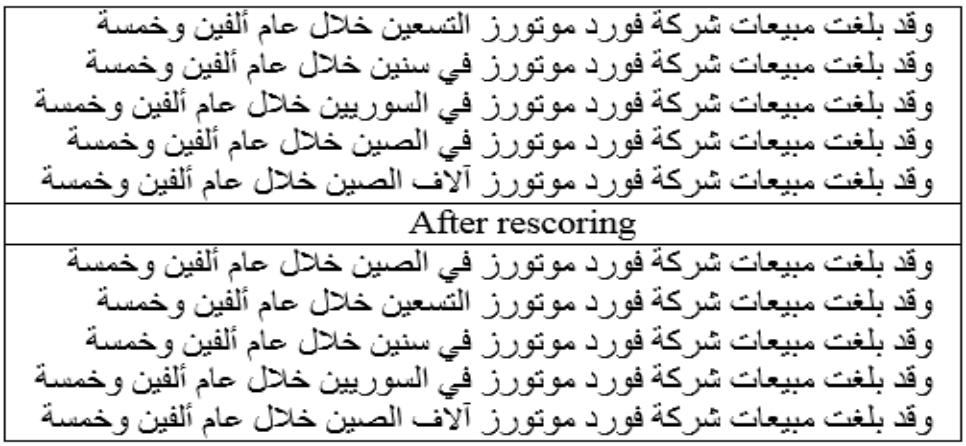


Fig. 3. N-best list befor and after rescoring

6. Conclusion

This study presents a new method to obtain the LDDs for ASR N-best rescoring. The method based on predictive Apriori algorithm that generate best association rules. The study shows that generating LDDs is computational expensive and require high-speed machines. It is a future research to utilize LDDs in the implementation of speech recognition systems and other NLP applications.

References

[1] Available :http://www.speech.cs.cmu.edu/sphinx/doc/sphinx-FAQ.html

[2] Available :http://htk.eng.cam.ac.uk/

[3] Iyer, Rukmini M., and Mari Ostendorf. "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models." Speech and Audio Processing, IEEE Transactions on 7.1 (1999): 30-39.

[4] Goodman, Joshua T. "A bit of progress in language modeling." Computer Speech & Language 15.4 (2001): 403-434.

[5] Qin, Min, et al. "Word Activation Forces-Based Language Modeling and Smoothing." Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on. Vol. 1. IEEE, 2013.

[6] Available :http://www.cs.waikato.ac.nz/ml/weka/

[7] Gao, Jianfeng, and Hisami Suzuki. "Long distance dependency in language modeling: an empirical study." Natural Language Processing–IJCNLP 2004. Springer Berlin Heidelberg, 2004. 396-405.

[8] Gao, Jianfeng, Joshua Goodman, and Jiangbo Miao. "The use of clustering techniques for language modeling–application to Asian languages."Computational Linguistics and Chinese Language Processing 6.1 (2001): 27-60.

[9] Zhao, Yinggong, et al. "Learning word embeddings from dependency relations." Asian Language Processing (IALP), 2014 International Conference on. IEEE, 2014.

[10] Xu, ZhiHai, et al. "Research on language model of long-distance dependency." 2010 International Conference on Advances in Energy Engineering. 2010.

[11] Zhou, Faguo, and Xingang Yu. "Graph-Based Language Model of Long-Distance Dependency." Asian Language Processing (IALP), 2011 International Conference on. IEEE, 2011.

[12] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013.

[13] Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.

[14] Wang, Xuerui, Andrew McCallum, and Xing Wei. "Topical n-grams: Phrase and topic discovery, with an application to information retrieval." Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. IEEE, 2007.

[15] GuoDong, Zhou, and Lua KimTeng. "Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition." Computer Speech & Language 13.2 (1999): 125-141.

[16] Manning, Christopher D., and Hinrich Schütze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.

[17] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[18] Nahar, Jesmin, et al. "Association rule mining to detect factors which contribute to heart disease in males and females." Expert Systems with Applications 40.4 (2013): 1086-1093.

[19] Malik, Meenakshi, and R. P. Agarwal. "A Survey On Association Rule Mining." International Journal of Research in Engineering and Applied Sciences 5.6 (2015): 48-56.

[20] Tobias Scheffer, T. (2005). "Finding association rules that trade support optimally against confidence." Intell. Data Anal. 9(4): 381-3

[21] Jurafsky D, Martin J (2009) Speech and language processing, 2nd edn. Pearson, NJ

[22] Salgado-Garza, Luis R., and Richard M. Stern. "N-Best list rescoring using syntactic trigrams." Mexican International Conference on Artificial Intelligence. Springer Berlin Heidelberg, 2004.

[23] Wang, Wen, Yang Liu, and Mary P. Harper. "Rescoring effectiveness of language models using different levels of knowledge and their integration."Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on. Vol. 1. IEEE, 2002.

[24] AbuZeina, Dia, and Mahmoud Hasan Alsaheb. "Capturing the Common Syntactical Rules for the Holy Quran: A Data Mining Approach." Advances in Information Technology for the Holy Quran and Its Sciences (32519), 2013 Taibah University International Conference on. IEEE, 2013.

CHAPTER 4:
EMPIRICAL STUDY OF ARABIC CONTINUOUS SPEECH

1. Introduction

Automatic Speech recognition (ASR) has recently received significant attention as one of successful trend in information retrieval (IR) and intelligent systems. Converting speech into text is an important since it felicitates deploying online audio contents and make it more accessible. However, developing high-quality speech recognition systems is a challenging task and still is a promising research area. Recently, there has been growing interest in speech recognition for the Arabic language as one of the most common languages worldwide. In fact, there is a real need for software tools to transcribe speech into text. Moreover, Arabic is the language of the holy writings of Islam (e.g. the holy Quran) that raises the demand for software to dictate such huge speech resources. Reference [1] indicated that ASR research is currently moving from mere speech-to-text systems towards "rich transcription" systems, which annotate recognized text with non-verbal information such as speaker identity, emotional state for customer care purposes.

Nevertheless, speech recognition is not a straightforward task, as it requires dynamic programming algorithms along with different stages for training and decoding. Reference [2] demonstrated why speech recognition is difficult; among the reasons include body language, noise, differences between spoken language and written language, etc. Therefore, obtaining an accurate freely available software is difficult to achieve. However, free and commercial software tools are available for Arabic speech recognition. In this study, we consider Soundflower [3] Mac utility that is a free, open-source speech application. The goal is to employ this utility to evaluate the performance of the Arabic speech recognition in terms of word error rate (WER) and the recognition accuracy. The study also aims at comparing the recognition performance of male and female Arabic native speakers. Reference [4] indicated that the performance of speech recognizers for female speakers is usually worse than that obtained for male speakers. In fact, the research in speech recognition contains different sources of pronunciation variations such as continuous or isolated speech, age, gender, emotion, dialects, noise, different accents, etc. Reference [5] presented the main phonetic differences between the speech of male and female speakers. The previous studies on Arabic speech recognition has not considered speaker's gender on speech recognition. The little research in this domain motivates the authors to to take over this research to find the effect of gender in speech recognition.

We have organized the rest of this study as follows. In the next section, we present the literate review. In section 3, we present speech recognition overview followed by the male and female speech recognition in

section 4. The speech corpus information presented in section 5. The experimental results presented in section 6. Finally, conclusion and future work presented in section 7.

2. Literature Review

In this section, we survey the reported contributions of Arabic speech recognition. Because speech recognition is a wide multidiscipline topic that contains vast and diverse subtopics, this literature focuses on the software tools developed for dictating (i.e. audio-to-text) Arabic speech. Soundflower [3] is a free audio system extension that allows applications to pass audio to other applications as one usage option. However, it has another option that is a speech to text converter with the following characteristics based on [6]: Soundflower is a Mac system extension, easy to use, simply presents itself as an audio device, allowing any audio application to send and receive audio with no other support needed.

Sakhr software company developed a commercial ASR [7] engine that has some features such as noisy environments, speaker independent, high accuracy, supports different Arabic accents. The DARPA-funded Babylon project [8] contains Arabic speech recognition as a part of the developed speech-to-speech translation systems. Hidden Markov Model Toolkit (HTK) [9] is a portable toolkit for speech recognition research. However, the HTK assumes that the textual files written using ASCII rather than Unicode, so if the training input text is stored using the standard Arabic character set then it has to transcribe to something that the HTK can handle. The obvious thing to use is the Buckwalter transcription [10]. CMUSphinx toolkit [11] is another option in the research community that used to build speech recognition systems. CMUSphinx is an open source speech software from Carnegie Mellon University (CMU) [12]. For example, Reference [23] employed CMUSphinx for cross-word Arabic continuous speech recognition. Unlike HTK, CMUSphinx does support Arabic language that used directly within the CMUSphinx components such as phonetic dictionaries and the language models. Choosing either HTK or CMUSphinx depends on some aspects such as implementation structure, supporting mobile platform, programming language, etc. nevertheless, both well-known ASR engines share the theoretical background for training and decoding that should give relatively similar outputs. As existing literature shows, little work devoted to serve the Arabic language compared to the English language. Dragon [13] is an example of software that used to convert audio text for English. The developer [13] claimed that Dragon is the fastest and most accurate way to interact with your computer. Gotranscript [14] provides speech recognition service for English. They listed some features of the product such as uncompromising quality, rates within the budget, highly accurate transcripts, timely and convenient delivery. Google [15] cloud speech application program interface (API) enables developers to convert audio to text by applying powerful neural network models in an easy to use API. Reference [16] lists the best 2016 voice

recognition software for English. Reference [17] compared the performance of three commercially available continuous speech recognition software packages for the English language. The packages include the IBM software that was found to have the lowest mean error rate (7.0 to 9.1 percent) followed by the L&H software (13.4 to 15.1 percent) and then Dragon software (14.1 to 15.2 percent).

3. Speech Recognition Overview

Speech recognition mainly contains two stages, training and decoding. The training stage requires two datasets: a set of speech files and a set of files containing the phonetic transcriptions of the speech files. There are various ways of getting phonetic transcriptions. The easiest is to use phonetic dictionary in combination with the training textual transcription. Some ASR engines such as HTK have a tool for doing this, or it can be prepared manually. Writing a phonetic dictionary is hard, and if the vocabulary has many words then it will be quite time-consuming. For Arabic, it is reasonable to approximate each Arabic character to a single phoneme. So, for instance, assuming that the phonetic transcription of "kataba" is "k a t a b a", Buckwalter transcription [6]. This method of transcription has two advantages, namely that everyone uses it, so that data can easily be made available to other people and it let the researchers to use other people's data; and that it uses one Roman character for each Arabic character, which is helpful, and which most of the other options don't do. There is, however, a problem, which is that it uses a number of non-alphabetic characters that have a reserved meaning in some ASR engines. Another option to represent words in the phonetic dictionary is by using Arabic characters such as "كَتَبَ" with the the phonemes "K AE T AE B AE", as an example. Reference [18] has more information of how generate phonemes (could say the phonetic dictionary) for Arabic words. Of course, there are other ways to generate phonetic dictionary for better performance. Linguistic scholars and phonetic specialists might help to in this regards.

In addition to the phonetic dictionary, the training stage also contains declaring language models that also called grammars. There are all sorts of kinds of grammars to use. The choice of the grammar is, indeed, the key to the performance of the recognizer. The more of constrains in the range of possible utterances, the more accurate the recognizer will be. In general, one can extract two types of grammars from a set of training textual transcription. One says that the target utterance may be an arbitrary sequence of words drawn from the training textual transcription (in short "any word" grammar); the other says that it must be one of the training textual transcription. The first is almost entirely not constraining, and leads to very poor accuracy (but lets researchers experiment with the effects of different transcriptions, because it relies entirely on the acoustic model); the other is very tightly constraining, and often leads to 100% accuracy. Naturally, there are other options to write grammars such as probabilistic N-Grams, the well-known approach for language modeling.

Using the phonetic transcriptions of the textual versions of the training speech, the audio files, and the list of phonemes, we can start training phase using the desired machine-learning tool such as hidden Markov models (HMMs). The output of the training stages is the acoustic models that used for testing, also called decoding process. The grammars are required throughout testing process. The testing stage employs a dynamic programming algorithm such as Viterbi algorithm to find the most likely phonemes sequence to find the textual words sequence of the spoken words. In fact, speech recognition is a complicated process that needs to handle different aspects such as Gaussian mixtures model, speech features such as Mel-frequency cepstral coefficients (MFCCs), Baum–Welch algorithm, triphone, pruning, etc.

MacOS recently introduced dictation (speech-to-text) as a feature usable in any application that takes text as input [19]. Reference [19] presented some technical issues that help to run Soundflower application. Fig. 1 shows the Soundflower starting page.

Figure 1. A snapshot of the Soundflower speech application

4. Male and Female Speakers

One goal of this work is to investigate the speech recognition performance of male and female Arabic speakers. The research on Arabic speech recognition has tended to focus on mixed male-female speech recognition rather than on gender based speech recognition. That is, the training corpus usually has mixed male and female speech that ignore the acoustic differences between female and male voices. Vogt in reference [20] indicated that the differences in speech features for male and female speakers are a well-known problem and the gender-dependent emotion recognizers perform better than gender-independent ones. Reference [21] separated the training dataset based on the gender. This separation yielded gender dependent HMMs that found significantly improve the word recognition accuracy over the gender independent method.

Reference [4] indicated that separating training corpora into male and female acoustic-phonetic models is a common solution to enhance the speech recognition performance.

5. The Speech Corpus

The speech corpus used in this work is an in-house corpus that contains of 275 audio files recorded by 20 Arabic native speakers (10 male and 10 female). Each male speaker utter 15-speech items, while some of female speakers utter less than 15- speech items (see Table III). The speech files mainly contains local and international news recorded from Al-Sabah TV channel in Kuwait. The modern standard Arabic (MSA) is the language used by all speakers. The speech file were prepared to have a fixed length between 30-60 seconds. The speech items sampled at 16 kHz and sum up to 2.63 hours of speech. The training textual transcription of the speech files were prepared by transcribing the audio files according to speakers' utterance. Table I composed of the corpus information. We emphasize that the corpus prepared in this work used for testing. That is, we have no training part in this work, as the goal is to investigate the trained models in Soundflower.

TABLE III. THE CORPUS INFORMATION

#	Gender	Number of Speakers	Number of speech files	Length (hour)	Number of Unique words
1	Male	10	150	1.53	5,149
2	Female	10	104	1.10	3,738
	Total	**20**	**254**	**2.63**	**8,887***

*the number of unique words in the entire corpus is 7,386 because of common words

6. Experimental Results

Using the speech corpus described in the previous section, we evaluated the performance for three cases; male only, female only and mixed case (male and female) speech files. The accuracy used to measure the accuracy that based on WER. The WER measured using the following formula [22]: WER= (D+S+I)/N, where D is the deletion errors, S is the substitution errors, I is the insertion errors, and N is the total number of labels (i.e words) in the reference (actual) transcriptions. The accuracy expressed as:

$$\text{Accuracy} = (1\text{-WER}) \times 100\%$$

Fig. 2 shows an example of the Soundflower output of a particular speech file after recognition process. This textual output aligned with the actual transcription to find D, S, I, N, to calculate the WER according to what we have recognized, either for a single speech file or for the entire speech files collection. Of course, there

are some recognition errors in Fig. 2 outputs (e.g. the word "لمنع"). This is natural like any other classification or pattern recognition system to have some misclassification rate.

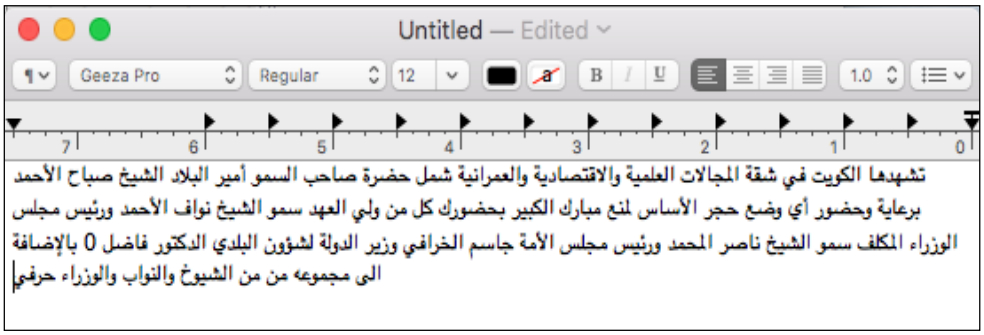


Figure 2. An example of Soundflower output

In the first case, the performance measured using the male speech files. That is, Soundflower employed to measure the accuracy of 150 speech files that belong to 10 male speakers. Table II shows the achieved results of each speaker. The table also shows the range of accuracy [42.26%, 70.39%]. The difference in the scored accuracy related to several factors such as speaker's anatomy of vocal tract, the speed of the speech, and the accent. Table II also shows that the WER is (100% - 55.33% = 44.67%). For the used wave files that is between 30-60 seconds length, the results presented in Table II and Table III show the range of accuracies starts at (almost) 40% up to 70%. Hence, it might be a future work to investigate the performance using less or more wave files lengths.

TABLE IV. ACCURACY FOR MALE ONLY SPEECH

#	Male Speakers	Number of speech files	Length (min:sec)	Accuracy (%)
1	Speaker 1	15	9:29	70.39
2	Speaker 2	15	10:12	48.80
3	Speaker 3	15	9:47	64.93
4	Speaker 4	15	8:46	59.07
5	Speaker 5	15	8:59	42.26
6	Speaker 6	15	9:55	54.67
7	Speaker 7	15	8:12	57.87
8	Speaker 8	15	8:30	55.16
9	Speaker 9	15	9:36	44.58
10	Speaker 10	15	8:54	55.66
	Total	**150**	**92:20**	**Avgerage =55.33%**

For female speakers, 104 speech files used to evaluate the accuracy. Table III shows the accuracy of each person of 10 female speakers. The accuracy range was [46.52%, 68.73%]. This range is close to what we achieved for male speakers. This reveals that the male and female speech recognition is very close in the case of using Soundflower tool. This result calls for more research to find the effect of acoustic differences between male and female speakers on Arabic speech recognition. Table III also shows that the WER is (100% - 56.97% = 43.03%).

TABLE V. ACCURACY FOR FEMAL ONLY SPEECH

#	**Female Speakers**	**Number of speech files**	**Length (min:sec)**	**Accuracy (%)**
1	Speaker 1	3	2:00	60.51
2	Speaker 2	15	9:42	68.73
3	Speaker 3	15	10:34	57.19
4	Speaker 4	7	5:12	52.07
5	Speaker 5	15	8:00	56.53
6	Speaker 6	15	9:15	50.85
7	Speaker 7	15	8:45	46.89
8	Speaker 8	2	1:27	62.83
9	Speaker 9	2	1:29	67.63
10	Speaker 10	15	9:56	46.52
	Total	**104**	**66:20**	**Average=56.97%**

The average of accuracies for the previous two cases indicates that the female speech recognition outperforms the male speech recognition. The third case separates the corpus for male and female speech to find the accuracy separately. Finally, we evaluated for the mixed male and female case for all speech files combined. Table IV shows the results of the mixed case.

TABLE VI. ACCURACY FOR MIXED MALE AND FEMALE SPEECH

Gender	**Total number of speakers**	**Number of speech files**	**Length (min:sec)**	**Accuracy (%)**
Male	10	150	92:20	54.66
Female	10	104	66:20	55.17

Male & Female	20	254	158:40	54.02

Fig. 3 shows the information provided in Table IV as a bar chart graph. The figure shows that the accuracy for Arabic speech is relatively low as the maximum-scored accuracy was 54.02%. The WER of the mixed speech corpus found to be (100% - 54.02% = 45.98%). However, we conducted our experiments on a small corpus that is unacceptable to generalize for the overall Arabic speech recognition. This result motivates the research to investigate and to enhance the performance of Arabic speech recognition.

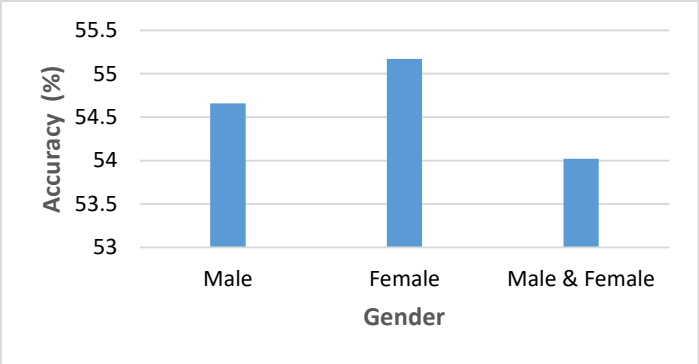


Figure 3. The accuracies of different testing cases

Even though gender is an important factor that has to consider in speech recognition. However, the experimental evaluation did not show clear performance differences using both the prepared corpus and the Soundflower tool. Despite we expect to have less accuracy in the case of female speech, as reported in some literature such as [4], it found that the female speech recognition performance outperforms the male speech recognition performance.

7. Conclusion

The study demonstrated the performance of speaker independent Arabic continuous speech recognition. A free MAC software tool used to find the recognition accuracy. It found that the maximum-scored accuracy is 54.02% for mixed speech of male and female. The experimental results did not show obvious difference between the accuracies based on the gender. As a future work, we propose more investigation of the effect of gender on Arabic speech recognition.

References

[1] Metze, Florian, et al. "Comparison of four approaches to age and gender recognition for telephone applications." Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. Vol. 4. IEEE, 2007.

[2] Forsberg, Markus. "Why is speech recognition difficult." Chalmers University of Technology (2003).

[3] http://soundflower.en.softonic.com/mac

[4] R. Vergin, A. Farhat and D. O'Shaughnessy, "Robust gender-dependent acoustic-phonetic modelling in continuous speech recognition based on a new automatic male/female classification," Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on, Philadelphia, PA, 1996, pp. 1081-1084 vol.2.

[5] Simpson, Adrian P. "Phonetic differences between male and female speech." Language and Linguistics Compass 3.2 (2009): 621-640.

[6] https://code.google.com/archive/p/soundflower/

[7] http://www.sakhr.com/index.php/en/solutions/speech-technologies

[8] Waibel, Alex, et al. "Speechalator: two-way speech-to-speech translation in your hand." Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations-Volume 4. Association for Computational Linguistics, 2003.

[9] Available: http://htk.eng.cam.ac.uk/

[10] Available: http://www.qamus.org/transliteration.htm

[11] Available: http://cmusphinx.sourceforge.net/wiki/tutorialoverview

[12] Available: http://www.speech.cs.cmu.edu/

[13] Available: http://shop.nuance.co.uk/store/nuanceeu/en_GB/DisplayHomePage

[14] Available: https://gotranscript.com/

[15] Available: https://cloud.google.com/speech/

[16] Available: http://voice-recognition-software-review.toptenreviews.com/

[17] Devine, Eric G., Stephan A. Gaehde, and Arthur C. Curtis. "Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports." Journal of the American Medical Informatics Association 7.5 (2000): 462-468.

[18] Ali, M., Elshafei, M., Al-Ghamdi, M. and Al-Muhtaseb, H. 2009. Arabic Phonetic Dictionaries for Speech Recognition. Journal of Information Technology Research. 2, 4 (2009), 67-80.

[19] Available: http://telestreamblog.telestream.net/2013/12/using-dictation-to-turn-recorded-audio-to-text-2/

[20] Vogt, Thurid, and Elisabeth André. "Improving automatic emotion recognition from speech via gender differentiation." Proc. Language Resources and Evaluation Conference (LREC 2006), Genoa. 2006.

[21] Abdulla, W. H., N. K. Kasabov, and Dunedin–New Zealand. "Improving speech recognition performance through gender separation." changes 9 (2001): 10.

[22] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. *Int J Speech Technol*. 10, 4 (2007), 183-195.

[23] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

CHAPTER 5:
EFFECT OF DIACRITIZATION ON ARABIC SPEECH RECOGNITION

1. Introduction

Automatic speech recognition (ASR) is of particular interest to human computer interface (HCI) and natural language processing (NLP). Recently, Arabic large-vocabulary speaker-independent continuous speech recognition system has recently received significant attention in the NLP research community. However, Arabic ASR poses some challenges such as the difficulty to obtain corpora for dialects that are spoken rather than written (i.e. no common standard for writing), difficulty in obtaining a large diacritized text as the Arabic allows writing without diacritics, and enormous number of word forms due to the morphology richness of the Arabic. Recently, there has been much interest in diacritization for better performance in ASR systems. Diacritization is the process of marking the letters using optional orthographic symbols that are called diacritics or short vowels. For Arabic ASR, the problem of short vowels is that they are generally pronounced, but almost never written. The study in [1] indicates that the non-diacritized text leads to problems for both acoustic and language modeling and therefore may lead to a loss in recognition accuracy. Similarly, it is reported in [2] that missing of short vowels leads to a significant increase in both the language model perplexity and the word error rate.

The importance of diacritization is that it enhances the supposed closely match between the training textual files and the corresponding speech files. In fact, it is extremely important that the phonemes of the pronunciation dictionary to adequately represent the actual training speech files. It is indicated in [3] that the performance of ASR is improved by shrinking the mismatch between the speech and the text used in training the acoustic model. In the case of training using non-diacritized text, many of phonetic segments will be lost because the short vowels are not there. Despite short vowels help the reader to realize the meaning of a particular word, however, not fully diacritized text might lead to ambiguity as the same word might have several meanings. For instance, the word "جنة: jnp" has three different meanings based on the short vowels (u:ُ,a:َ,i:ِ) on the first letter: (جُنة, جَنة, جِنة) (junp, janp, jinp) that means (protection, paradise, jinn), respectively. In the previous ward, the Buckwalter scheme was used for Arabic transliteration [4]. More on Arabic diacritization and some other related challenges are found in [5]. Reference [5] mainly discussed the differences in the pronunciation and the meaning of a particular word according to its diacritization. Nevertheless, obtaining a sizable diacritized text for ASR and NLP applications

is extremely difficult as well as time-consuming task. This is the motivation of this work as we produced a manually diacritized continuous speech corpus for the modern standard Arabic (MSA). In this study, we employed the latest Carnegie Mellon University (CMU) PocketSphinx ASR engine [6] for exploring the Arabic ASR performance based on diacritized and non- diacritized text. PocketSphinx includes the latest available releases as follows: sphinxbase - 5prealpha, PocketSphinx - 5prealpha, SphinxTrain - 5prealpha. In the experiments, we used a new "in house" diacritized text corpus that contains 13.5 hours for training and 4.1 hours for testing. This study also presents the intermediate steps for training and decoding such as the proposed and used phonemes set, the pronunciation dictionary, the acoustic model, and the language model. We emphasize that this work is a preliminary step toward further research using the newly created corpus. This corpus has been fully supported by Kuwait University. The size of the corpus in this work is 17.6 hours; however, we aim at increasing the size to about 30 hours.

In next section, we present the literate review. In section 3, we present the phonemes set followed by a background of acoustic models in section 4. The proposed method is described in section 5 and the experiment results in section 6. Finally, we conclude in section 7.

2. Literature Review

Despite that Arabic is one of the popular languages. However, little research has been devoted to tackle the different ASR aspects such as the dialects, the diacritization, and the morphological complexity. In this literature, we focus on the studies that consider diacritization. The study in [7] demonstrates a news transcription system for MSA. It compares the performance using diacritized and non-diacritized text for broadcast news. The word recognition accuracy of the non-diacritized case outperforms the diacritized case. This is due to the errors that are introduced by missing of the short vowels in the diacritized case. However, this might be not a problem since the Arabic native speakers can infer the missing short vowels based on the prior knowledge and the words context. The studies in [8][9] presents methods for capturing the acoustic differences (pronunciation variations) at cross-words and within words for Arabic ASR systems.

The study in [2] demonstrates a comparison between script transcriptions (i.e. non-diacritized) and romanized transcription that is phonologically rich by vowels information. The romanized transcription case outperforms the standard Arabic script that has no diacritics. The work in [10]

produces three different speech diacritized corpora that include a holy Qur'an corpus, a command corpus, and a digits corpus. The results were demonstrated based on the diacritized text. The experimental results in [11] show that the non-diacritized case slightly outperforms the diacritized text case for a phonetically rich and balanced Arabic speech corpus. The Sphinx tools along with SAMPA Romanization method were used in [12] for dialectal Arabic speech recognition. The research in [13] found that the diacritized text improved the acoustic model more than undiacritized orthography. Most of the previous works were performed using relatively small corpora; however, we used a larger corpus to explore the effect of diacritization on Arabic ASR.

3. Phonemes Set

The phoneme is the basic unit of speech that represents a distinctive sound of the language's phonology. Hence, a change of a particular phoneme in a word makes a change in the meaning of the word. Phonemes play a vital role in the performance of ASR and text to speech systems. In this work, we propose a phoneme set that is used to evaluate the recognition performance of the prepared corpus. The pronunciation dictionary is prepared using the proposed phonemes set by a mapping process between the Arabic letters (the language's vowels and consonants) and their corresponding phonemes. However, in some cases, morphologically driven rules are used for phonetic rich dictionary. In addition, some pronunciation exceptions might be manually processed for better acoustic representation. The studies [14] and [15] elaborate on Arabic phonemes and the pronunciation rules.

In general, creation a dictionary of a particular language requires linguistic experts and deep knowledge of the language sounds. However, the choice of the phoneme in the phonemes set is not straightforward as it has some constraints. For instances, it should represent the relevant information of the language sounds, it should consider the surrounding context between the letters, and it should carefully estimate the starting and the ending of the letters. No doubt, the phonemes that are used to represent the training words characterize the quality of the acoustic models and, therefore, the overall performance. Table I shows the phonemes set used in this work. It contains 46 phonemes. In addition to the Arabic letters, the table includes the short vowels that are Fatha (◌َ), Damma (◌ُ), and Kasra (◌ِ). In this work, we discarded the Shadda (◌ّ) as our experimental evaluation showed that it has no difference in the performance. We also used three phonemes to

represent the Fatha that precedes Alif (ا) ➔ ـَا as a single phoneme that is (AUA), the Damma that precedes Waw (و) ➔ ـُو as a single phoneme that is (AWW), and the Kasra that precedes Ya (ي) ➔ ـِي as (AIY). The reason of handling these cases as a single phoneme is that the pronunciation of the short vowels is different when it precedes the long vowels. Hence, it would be correctly transcribed as single phonemes.

TABLE I. THE ARABIC LETTERS AND THE PHONEMES SET

#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	24	ظ	ZZ
2	آ	AA	25	ع	AE
3	أ	O	26	غ	GH
4	ؤ	EW	27	ف	F
5	إ	I	28	ق	Q
6	ئ	EY	29	ك	K
7	ا	A	30	ل	L
8	ب	B	31	م	M
9	ة	P	32	ن	N
10	ت	T	33	ه	H
11	ث	TH	34	و	W
12	ج	J	35	ى	AY
13	ح	HH	36	ي	Y
14	خ	KH	37	ـُ	N
15	د	D	38	ـٌ	N
16	ذ	DH	39	ـٍ	N
17	ر	R	40	ـَ	AU
18	ز	Z	41	ـُ	AW
19	س	S	42	ـِ	AI
20	ش	SH	43	ـً	ignored
21	ص	SS	44	ـَا	AUA
22	ض	DD	45	ـُو	AWW
23	ط	TT	46	ـِي	AIY

The selection of a phoneme symbol is an optional and it does not matter what phoneme symbol is used for an individual letter. In the training stage, each phoneme is modelled using a sequence of Hidden Markov Model (HMM) states for computing the acoustic model. In the decoding stage, the phoneme is initially recognized and then used to find the most likely spoken words based on

the best-matched phonemes between the speech file in question (the observations) and the trained HMMs of the acoustic model.

4. Acoustic Models

The training stage of an ASR system consists of building an acoustic model that is a major component of ASR engines. Acoustic models statistically represent the relationships between the speech signals and the language phonemes. These representations come into the form of probabilistic matrices that are known as three matrices: initial probability, transition probability, and the observation likelihoods or emission probability. There are different methods to train acoustic models; the most common method is HMM. It has been long observed that the HMM based acoustic models successfully implemented in the state of the art speech recognizers. However, there are other approaches such as artificial neural networks (ANN) [16] and support vector machine (SVM) [17].

CMU Sphinx speech engines support three types for acoustic modeling. For instances, the CMU Sphinx configuration file "Sphinx_train.cfg" has the commands to enable or disable the desired acoustic model. The types of acoustic models include the traditional fully continuous, the semi-continuous, and the phonetic tied-mixture (PTM) models [18]. Despite the common implementation of fully continuous and semi-continuous in Arabic ASR, however, PTM has less experimental studies for Arabic speech recognition. PTM is a recent method that compromises between important factors such as speed and performance. It is characterized by fast decoding as well as its ability to handle large amount of speech collections.

Hence, PTM might be good option if the decoding time is more important than the accuracy. There are some advantages of PTM based acoustic models. For instance, the PTM model consider pronunciation variations modeling such the work in [19]. Reference [19] proposes a state-dependent PTM model with variable codebook size to improve the coverage of phonetic variations while maintaining model discriminative ability. One reason of speed is that the PTM model used relatively low fixed Gaussians that speed up the recognition time.

In the decoding stage, the HMM states of each phoneme is compared with the query acoustic feature vectors to find the best-matched phonemes and, then, likely sequence of words. The HMMs parameters are estimated using special algorithms such as Baum-Welch re-estimation and

expectation maximization (EM). The corpus vocabulary and the size of the speech corpus determines some training parameters such as the number of Senones (tied-state) and the number of Gaussians. Table II shows the approximation number of Senones and the densities according to the vocabulary and the size of some English speech corpora [20].

TABLE II. APPROXIMATION NUMBER OF SENONES AND DENSITIES

Vocabulary	Hours	Senones	Densities	Example
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

5. Proposed Method

We evaluated the Arabic ASR performance using the prepared corpus. We got the speech files that belong to MSA broadcast news form As-Sabah TV [21] in Kuwait. The first step was the preprocessing that includes segmenting the long speech files into short segments of 30-60 seconds. The produced speech files cover different news stories and it sums up to 17.6 hours of 29 speakers (19 male speakers and the rest are for female speakers). The speech files were sampled at 16 KHz mono. A silence of 0.1 seconds at the beginning and at the end of each speech file. We collected 1660 speech files that were transcribed and manually diacritized. We divided the speech files into two parts; the training set that contains 1,269 (13.5 hours) speech files and the testing that contains 391 speech files (4.1 hours). Hence, the testing part is 23% of the overall corpus, which follows the training-testing splitting percentage that is usually used in world of statistical classification (i.e. 20% ~ 25%). The vocabulary size of the training set is 29,843 words. The proposed method is summarized in the algorithm in Fig.1.

Part One

- *Prepare the speech collection.*
- *Prepare the transcription of the speech files.*
- *Diacritize the textual transcription*
- *Define the phonemes set.*
- *Create the pronunciation dictionary.*
- *Define the number of Gaussians.*
- *Define the senones (n_tied_states).*
- *Train the acoustic models.*
- *Prepare the language model.*
- *Run PocketSphinx for training and decoding.*

Part Two

- *Repeat the pervious steps without diacritization.*

Fig. 1. The proposed method

In addition to the previous steps, the CMU SphinxTrain performs some internal tasks such as computing features from audio files, training context independent models, training context dependent models, build trees, prune trees, lattice generation, lattice pruning, and finally decoding using the trained models. Once having the trained acoustic model, the PocketSphinx is used for decoding by use other components such as the pronunciation dictionary and the language model.

6. Experimental Results

This section presents the experimental results based on the introduced MSA speech corpus. We conducted the experiments for two cases, diacritized and non-diacritized text. In this work, we used three emitting states of HMMs that corresponds to the subphones at the beginning, middle, and end of the phone. The acoustic models were calculated using context-dependent HMM triphones. Our acoustic models are all trained using SphinxTrain for PTM PocketSphinx. For language model, we used the CMU language toolkit [22] to calculate the statistical N-grams (i.e. 1-grams, 2-grams, and 3-grams) based on the corpus transcription. The pronunciation dictionary was generated using a Python based program. The total number unique words in the diacritized based system is 29,843 while it is 19,581 words in the non-diacritized case.

The word error rate (WER) was measured for different parameters such as the number of the Gaussian densities and the number of the Senones. The PocketSphinx configuration file indicates that the PTM based models have to use the same initial and final Gaussian densities, 256 Gaussians as indicated in [20]. However, we investigated different values as indicated in Table III. In Table III, we demonstrate the WER and the accuracy achieved for different parameter settings. The lowest WER is 36.2% and it achieved using 64 Gaussians densities and 1000 Senones. We investigated a wide range of parameters to clarify its effect on the performance as shown in the table.

TABLE III. THE DIACRITIZED BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	64	200	37.0	63.0
2	128	200	36.8	63.2
3	256	200	36.6	63.4
4	64	500	36.4	63.6
5	128	500	63.5	63.5
6	256	500	36.5	63.5
7	**64**	**1000**	**36.2**	**63.8**
8	256	1000	36.8	63.2
9	256	2000	36.8	63.2
10	256	3000	36.9	63.1
11	256	4000	37.1	62.9
12	256	5000	37.4	62.6
13	256	6000	37.5	62.5
14	256	7000	38.0	62.0
15	256	8000	38.2	61.8
16	256	9000	38.4	61.6
17	256	10000	38.7	61.3

This low accuracy is reasonable since we used a relatively small size corpus. Ideally, ASR requires 200-300 hours speech corpus. It is indicated in [23] that at least 1 gigabyte of texts for language models and 50 hours for acoustic models are required for reasonable performance. It is also reported in [20] that the WER for 10-hours task should be around 10%. For a large task, it could be around 30%. Table IV shows the WER for some ASR systems on different English speech corpora [24].

TABLE IV. ROUGH WERS FOR A NUMBER OF ENGLISH CORPORA

Speech collection	Vocabulary	WER %
TI Digits	11 (zero-nine, oh)	0.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech	64000+	20

One more reason for the obtained relatively low accuracy is that the used corpus has no filler dictionary. Filler dictionary generally contains noise and inhalation speech that are appropriately handled during the training phase. The fillers require indicating the noises and inhalations in the transcription of the speech files, which is an extremely difficult task for our corpus. It is worthy to

point out to a recent study [25] of Arabic ASR that considers the impact of phonological rules on Arabic ASR performance.

For non-diacritized case, Table V shows the performance using different densities and Senones. The lowest WER is 23.6% that achieved using 128 Gaussians densities and 500 Senones as highlighted in Table V. Therefore, the information in Table V indicates that the best accuracy is achieved using 128 densities and 500 Senone, which might be the performance of the baseline for future work. Of course, this result belongs to the used corpus that contains 17.6 hours. Other Arabic speech corpora might have different results. Regarding the execution time of the training and the decoding stages for both diacritized and non-diacritized system, we found that the execution time of non-diacritized case was clearly less than the diacritized case.

TABLE V. THE NON-DIACRITIZED BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	8	200	27.8	72.2
2	16	200	25.7	74.3
3	32	200	24.8	75.2
4	64	200	24.3	75.7
5	128	200	24.1	75.9
6	256	200	23.9	76.1
1	8	500	26.9	73.1
2	16	500	24.8	75.2
3	32	500	24.2	75.8
4	64	500	23.9	76.1
5	**128**	**500**	**23.6**	**76.4**
6	256	500	23.7	76.3
1	8	1000	26.4	73.6
2	16	1000	24.7	75.7
3	32	1000	23.9	76.1
4	64	1000	23.9	76.1
5	128	1000	23.7	76.3
6	256	1000	23.8	76.2

1	8	5000	25.9	74.1
2	16	5000	24.9	75.1
3	32	5000	24.0	76.0
4	64	5000	24.2	75.8
5	128	5000	24.5	75.5
6	256	5000	25.6	74.4
1	128	10000	26.3	73.7
2	256	10000	28.0	72.0

In ASR, the training phase is time-consuming. Hence, we consider speeding up the execution time using an option in CMU PocketSphinx. The configuration file is called "sphinx_train.cfg". This file has an option for multiprocessing mode. The two options that can be used for reducing the training and the decoding time are as follows. $CFG_NPART = 10 ➔ the number of parts to run Forward-Backward estimation; $DEC_CFG_NPART = 10 ➔ how many pieces to split decode in. The number 10 is specified by the user according to the desired factor to reduce the execution time. The default values of these two parameters is 1. This option is helpful since it clearly reduces the execution time by use a number of processors in multicore machines. We also conducted some experiments to compare the execution time of the continuous and PTM based acoustic models. We found that the PTM based acoustic model has less execution time. The PTM based acoustic model is three times faster than the continuous acoustic model.

7. Conclusion

This study presents an experimental evaluation of Arabic ASR performance using a new continuous speech manually diacritized corpus. In the experiments, we consider two cases of the Arabic text; diacritized and non-diacritized. The experimental results show that the non-diacritized based system outperforms the diacritized based system even with a smaller vocabulary. However, the diacritized based system gives vowelized text output, which is not produced by a non-diacritized based system. As a future work, it is worthy to reinvestigate and analysis the main conclusion of this work; which indicates that the accuracy of using diacritized text is lower than the accuracy of using non-diacritized text.

References

[1] Vergyri, Dimitra, and Katrin Kirchhoff. "Automatic diacritization of Arabic for acoustic modeling in speech recognition." Proceedings of the workshop on computational approaches to Arabic script-based languages. Association for Computational Linguistics, 2004.

[2] Kirchhoff, Katrin, et al. "Novel approaches to Arabic speech recognition-final report from the JHU summer workshop 2002." John-Hopkins University, Tech. Rep (2002).

[3] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[4] http://www.qamus.org/transliteration.htm

[5] Al-Anzi, Fawaz S., and Dia AbuZeina. "Stemming impact on Arabic text categorization performance: A survey." Information & Communication Technology and Accessibility (ICTA), 2015 5th International Conference on. IEEE, 2015.

[6] http://cmusphinx.sourceforge.net/wiki/download

[7] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[8] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[9] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[10] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[11] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[12] Elmahdy, Mohamed, et al. "Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition." Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on. IEEE, 2009.

[13] Vergyri, Dimitra, et al. "Development of the SRI/nightingale Arabic ASR system." Interspeech. 2008.

[14] Ali, Mohamed, et al. "Generation of Arabic phonetic dictionaries for speech recognition." Innovations in Information Technology, 2008. IIT 2008. International Conference on. IEEE, 2008.

[15] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[16] Wilinski, Piotr, et al. "Toward the border between neural and Markovian paradigms." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28.2 (1998): 146-159.

[17] Guo, Guodong, and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines." IEEE transactions on Neural Networks 14.1 (2003): 209-215.

[18] Lee, Akinobu, et al. "A new phonetic tied-mixture model for efficient decoding." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 3. IEEE, 2000.

[19] Liu, Yi, and Pascale Fung. "State-dependent phonetic tied mixtures with pronunciation modeling for spontaneous speech recognition." IEEE transactions on speech and audio processing 12.4 (2004): 351-364.

[20] Training Acoustic Model, Available:
http://cmusphinx.sourceforge.net/wiki/tutorialam

[21] AL-SABAH TV, Available:
http://www.alsabahpress.com/

[22] Building language model, Available:
http://cmusphinx.sourceforge.net/wiki/tutoriallm
CMU Sphinx Speech Recognition Toolkit, Available:
https://sourceforge.net/p/cmusphinx/discussion/help/thread/1f102f95/?limit=25#9d3d

[23] Jurafsky D, Martin J (2009) Speech and language processing, 2nd edn. Pearson, NJ

[24] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology(2017): 1-9.

CHAPTER 6:
PHONETIC TIED-MIXTURE PTM ACOUSTIC MODEL

1. Introduction

Automatic speech recognition (ASR) has recently attracted much attention for more convenient in human-computer interaction (HCI). Speech recognition aims at converting the spoken language into machine-readable format. For this task, a speech recognizer generally employs several components such as pronunciation dictionary, language model, and acoustic model. The effective implementation of acoustic modeling is a critical process that aims at capturing the realistic acoustic features of the speech signal. The ASR training phase includes hidden Markov model (HMM) [1] parameters estimation that produces the learned acoustic models. That is, HMM is the popular method to implement acoustic models. The types of acoustic model include the traditional fully continuous, the semi-continuous, and the phonetic tied-mixture (PTM) models. Despite the

common implementation of fully continuous and semi-continuous in Arabic ASR, however, PTM has less experimental studies for Arabic speech recognition. PTM is a recent method that compromises between important factors such as speed and performance. Regardless of the method used to implement the acoustic model, the training of ASR includes the optimal estimation of the acoustic parameters based on the input acoustic feature vectors, especially for large vocabulary continuous speech collections.

Fully continuous acoustic models use a large number of Gaussians to compute the score of each frame. In contrast, the semi-continuous models use extremely less Gaussians, of course, on the account of the accuracy. However, for insufficient training speech collections, semi continuous might give better performance than the fully continuous models. It is indicated in [2] that the use of limited number of mixture densities can not only improve the performance but also significantly reduce the amount of computation. Semi continuous is also good for memory constraint cases such as in mobile devices. For handling such constraints, PTM was released to enhance the performance in the semi-continuous models as well as reducing the heavy use of the computational resources in the fully continuous models. That is, PTM models are somewhere between semi-continuous and fully continuous models, offering the speed of the continuous models with the ability to effectively use large amounts of training data. That is, PTM model aims at reducing the number of parameters of continuous model while significantly reducing the execution time (i.e. the training and decoding time) [3].

This study aims at demonstrating an experimental study of modern standard Arabic (MSA) speech recognition performance based on different acoustic models. We prepared a corpus of Arabic continuous speech that contains 13.5 hours for training and 4.1 hours for testing. We used a pronunciation dictionary based on a proposed phonemes set that contains 44 phonemes. In this work, we used the latest CMU speech recognizer, the PocketSphinx. This tool includes the latest available releases as follows: Sphinxbase - 5prealpha, PocketSphinx - 5prealpha, Sphinxtrain - 5prealpha [4]. We demonstrate the results for different training settings such as different number Gaussians, tied states (Senones), and parts to run Forward-Backward estimation run training in parallel to fully load the machine processing units.

In next section, we present the literate review. In section 3, we present the phonemes set followed by a background of acoustic models in section 4. The proposed method is described in section 5 and the experiment results in section 6. Finally, we conclude in section 7.

2. Literature Review

There are quite studies in Arabic ASR employing semi-continuous and fully continuous acoustic models. For instances, Reference [5] demonstrated an ASR study for the Arabic speech. It used five-state HMM for triphone continuous acoustic models, with 8 and 16 Gaussian mixture distributions. The acoustic model of Reference [5] was then used in other publications such as [6] and [7]. Reference [8] employed CMU Sphinx tools for three different acoustic models that belong to three different speech collections. Reference [9] used CMU Sphinx tool for Arabic speech recognition. They developed three acoustic models for three different speech collections. They demonstrated the performance using using different number of Gaussians. Reference [10] used continuous based acoustic models for a phonetically rich and balanced Arabic speech corpus.

The following are some studies that discuss the three above-mentioned acoustic models. Reference [11] indicated that the impression that continuous HMMs are the best choice of acoustic model, however, semi-continuous might have an advantage in small amount of training data due to the need for estimating a large number of parameters. Reference [12] used semi-continuous HMM to reduce the number of parameters and the computational complexity. Reference [13] highlights the problem when using fully continuous with limited training data. In indicated that when large number of basic HMMs has only a few observations in the training data, then the sparse mixture-weight distributions cannot be estimated robustly and are expensive to store. In PTM based model, the number of Gaussians is reduced according to the shared triphone states of the same phoneme. There are some studies employed this method. For instances, Reference [14] showed that PTM systems, if properly trained, can significantly outperform the currently dominant state clustered HMM-based approach. Reference [15] presented that the PTM based acoustic model is easily trained, reliably estimated, and enable the decoder to perform efficient Gaussian pruning.

3. Phonemes Set

The phoneme is the basic unit of speech that describes the pronunciation of a word. It is also the basic unit that is used for speech recognition in ASR. It is used to represent the language's vowels and consonants thorough straightforward one-to-one rules. The pronunciation dictionary is prepared using the phonemes set by mapping the words pronunciations (e.g. Arabic letters) to their corresponding phonemes based on the given phonemes set. However, in some cases, morphologically driven rules are used for phonetic rich dictionary. In addition, the pronunciation exceptions might manually processed for better acoustic representation. In general, the creation a dictionary requires linguistic experts and deep knowledge of the language sounds.

The selection of a phoneme symbol is an optional and it does not matter what phoneme symbol is used for an individual letter. Each phoneme is modelled using a sequence of HMM states in the acoustic model that is later used to find the most likely spoken words based on the best-matched phonemes. However, the choice of the phoneme in the phonemes set is not straightforward as it has some constraints. For instances, it should represent the relevant information of the language sounds, it should consider the the surrounding context between the letters, and carefully estimating the start and the end of the letters. Table I shows the phonemes set used in this work. It contains 46 phonemes.

TABLE I. THE ARABIC LETTERS AND THE PHONEMES SET

#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	24	ظ	ZZ
2	آ	AA	25	ع	AE
3	أ	O	26	غ	GH
4	ؤ	EW	27	ف	F
5	إ	I	28	ق	Q
6	ئ	EY	29	ك	K
7	ا	A	30	ل	L
8	ب	B	31	م	M
9	ة	P	32	ن	N
10	ت	T	33	ه	H
11	ث	TH	34	و	W
12	ج	J	35	ى	AY
13	ح	HH	36	ي	Y

14	خ	KH	37	ً	N
15	د	D	38	ٌ	N
16	ذ	DH	39	ٍ	N
17	ر	R	40	ًَ	AU
18	ز	Z	41	ُ	AW
19	س	S	42	ِ	AI
20	ش	SH	43	ّ	ignored
21	ص	SS	44	ىا	AUA
22	ض	DD	45	ئو	AWW
23	ط	TT	46	ئِي	AIY

4. Acoustic Models

The essence of this work is that the growing amount of training speech collections adds more complexities for efficient parameter estimation methods in speech recognition. Intuitively, the goal is, hopefully, to decrease the error rate as well as to minimize the processing time in large vocabulary ASR systems. Acoustic model is a major component of ASR engines that statistically represents the relationship between the speech signals and the phonemes. HMM is one most common type of acoustic models to compile the statistical representation of each phoneme. It has been long observed that the HMM based acoustic models successfully implemented in the state of the art speech recognizers. However, there are other approaches such as artificial neural networks (ANN) [16] and support vector machine (SVM) [17].

CMU Sphinx speech engines support all acoustic modeling types. For instances, the configuration file "Sphinx_train.cfg" of the CMU PocketSphinx has the commands to enable of disable a particular acoustic model type as shown in Fig. 1. The figure shows that the corresponding system implements PTM based acoustic model since the PTM command is enabled. It also shows that the PTM model is supported in PocketSphinx. However, the reader can check the CMU Sphinx website to investigate the latest versions and updates.

```
#$CFG_HMM_TYPE = '.cont.'; # Sphinx 4, PocketSphinx
#$CFG_HMM_TYPE  = '.semi.'; # PocketSphinx
$CFG_HMM_TYPE  = '.ptm.'; # PocketSphinx
```

Fig. 2. CMU configuration settings for acoustic models

In the decoding process, the HMM states of each phoneme is compared with the query acoustic feature vectors to find the probabilities of the best matched phonemes using specials algorithm. The HMMs parameters are estimated using the training speech files and the corresponding text

transcription. The size of corpus vocabulary as well as the the size of the speech corpus determines some training parameters such as the number of Senones and the number of Gaussians. Table II shows the approximation number of Senones and the densities according to the vocabulary and the size of some English speech corpora [18].

TABLE II. APPROXIMATION NUMBER OF SENONES AND DENSITIES

Vocabulary	Hours	Senones	Densities	Example
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

The major difference between the three acoustic modeling is related to the accuracy and the processing time. The fully continuous model is assigned a separate Gaussian mixture model for each Senone which hugely increase the the Gaussians in the model. On the other hand, the semi-continuous model allows sharing the Gaussians that increases the computations speed. Still, the semi-continuous is more flexible that performs well in the limited amount of training speech hours. PTM model is in between that used relatively low fixed Gaussians that speed up the recognition time. PTM is characterized by fast decoding as well as its ability to handle large amount of speech training collections. Hence, the model selection is based on the speech and accuracy constrains. Hence, PTM might be good option if the decoding time is more important than the accuracy. The PTM model has been farther enhance for better pronunciation variations modeling such the work in [19]. Reference [19] proposes a state-dependent PTM model with variable codebook size to improve the coverage of phonetic variations while maintaining model discriminative ability.

5. Proposed Method

Preparing a continuous speech corpus is extremely difficult task that demand so much time. It require different phases such as collecting the audio files, segmentation, transcription, and diacritization. Hence, most of the Arabic ASR research studies employed small corpus of isolated words. For this reason, we compile a continuous speech corpus that contains 105,531 words

(22,545 unique words) of 1660 speech file. We got the speech files form As-Sabah TV [20] in Kuwait. We performed a preprocessing to divide the file into short files of 30-60 seconds. The speech files cover different news stories and it sums up to 17.6 hours of 29 speakers (19 male speakers and the rest are for female speakers). We split the the corpus into two parts, 13.5 hours for training (1,269 speech files) and 4.1 hours for testing (391 speech files). The speech files were sampled at 16 KHz mono. A silence of 0.1 seconds at the beginning and at the end of each speech file. The proposed method is summarized in the algorithm shown in Fig.2.

- *Prepare the desired speech collection.*
- *Prepare the textual transcription of the collected speech files.*
- *Define the phonemes set.*
- *Based on the training textual transcription and the phonemes set, create the pronunciation dictionary.*
- *Define the senones (n_tied_states).*
- *Train the desired acoustic models.*
- *Prepare the language model using CMUCLMTK [21]*
- *[Optional for speed] How many parts to run Forward-Backward estimation in.*
- *[Optional for speed] Define how many pieces to split decode in.*

Fig. 3. CMU configuration settings for acoustic models

The CMU Sphinxtrain is used for training by performing the following steps: computing feature from audio files, training context independent models, training context dependent models, build trees, prune trees, lattice generation, lattice pruning, and finally decoding using the trained models. The PocketSphinx is used for decoding using the learned acoustic model and the other components such as the pronunciation dictionary and the language model.

6. Experimental Results

This section presents the experimental results based on the introduced MSA speech corpus. The performance is measured based on different parameters such as the number of Senones and the number of Gaussians. The PocketSphinx configuration file indicates the following constrains regarding the number of Gaussians:

- For fully continuous models, the initial has to be less than the final number of densities.
- For semi-continuous models, the initial and final models have the same density.
- For PTM models, the initial and final models have the same density.

- If you are training semi-continuous or PTM model, use 256 Gaussians [18].

Word Error Rate (WER) was used to evaluate the ASR performance for different acoustic models. Table III shows the WER and the accuracy for the fully continuous model based on different values of Senones and densities. The performance is low for all investigated parameters. The lowest scored WER is 55.9%. This is reasonable since we used a relatively small size corpus. Ideally, ASR requires 200-300 hours speech corpus. One more reason for this low accuracy result is that the used corpus has no filler dictionary. Filler dictionary generally contains noise and inhalation speech that are not considered as phonemes.

TABLE III. THE CONTINUOUS BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	1➔8	200	44.1	55.9
2	1➔16	1000	42.3	57.7
3	1➔32	1000	42.0	58.0
4	1➔256	1000	43.6	56.4
5	1➔256	5000	89.8	
6	1➔256	200	42.2	57.8

Table IV shows the results of the semi-continuous based models. The results are slightly better than the PTM based model that are shown in Table V. It is clear that the size of the used corpus is suitable for semi-continuous model. The best WER is 64.0% at 256 Gaussians and 1000 Senones.

TABLE IV. THE SEMI-CONTINUOUS BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	256	200	36.8	63.2
2	256	1000	35.8	64.2
3	256	2000	35.9	64.1
4	256	3000	36.1	63.9
5	64	500	36.2	63.8
6	32	1000		

Table V shows the results using PTM based acoustic models. The best WER was found to be 63.5% using 256 Gaussians and 500 and 750 Senones. The experimental results show that the PTM based system is noticeably faster than the semi-continuous based model.

TABLE V. THE PTM BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	256	200	36.6	63.4
2	256	500	36.5	63.5
3	256	750	36.5	63.5
4	256	1000	36.8	63.2
5	256	2000	36.8	63.2
6	256	3000	36.9	63.1
7	256	4000	37.1	62.9
8	256	5000	37.4	62.6
9	256	6000	37.5	62.5
10	256	7000	38.0	62.0
11	256	8000	38.2	61.8
12	256	9000	38.4	61.6
13	256	10000	38.7	61.3

We also considered speeding up the execution time using CMU PocketSphinx configuration (i.e. number of parts to run Forward-Backward estimation➔ $CFG_NPART = 10; and how many pieces to split decode in➔ $DEC_CFG_NPART = 10;). This option is helpful since it clearly reduces the execution time by utilizing a number of processors in multicore machines.

7. Conclusion

We evaluated the performance of three different implementations of acoustic models that include semi-continuous, fully continuous, and PTM acoustic models. We employed CMU PocketSphinx with a speech corpus that contains 17.6 hours of Arabic speech. The results show that the semi-continuous acoustic model slightly outperforms the PTM acoustic models. However, the PTM has less execution time.

References

[1] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[2] Huang, Xuedong D., and Mervyn A. Jack. "Semi-continuous hidden Markov models for speech signals." Computer Speech & Language 3.3 (1989): 239-251.

[3] Available: http://cmuSphinx.sourceforge.net/wiki/acousticmodeltypes

[4] Available: http://cmuSphinx.sourceforge.net/wiki/download

[5] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[6] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[7] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[8] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[9] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[10] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[11] Riedhammer, Korbinian, et al. "Revisiting semi-continuous hidden Markov models." Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012.

[12] Huang, Xuedong, et al. "The SPHINX-II speech recognition system: an overview." Computer Speech & Language 7.2 (1993): 137-148.

[13] Digalakis, Vassilios, and Hy Murveit. "High-accuracy large-vocabulary speech recognition using mixture tying and consistency modeling." Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, 1994.

[14] Sankar, Ananth. "A new look at HMM parameter tying for large vocabulary speech recognition." ICSLP. 1998.

[15] Lee, Akinobu, et al. "A new phonetic tied-mixture model for efficient decoding." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 3. IEEE, 2000.

[16] Wilinski, Piotr, et al. "Toward the border between neural and Markovian paradigms." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28.2 (1998): 146-159.

[17] Guo, Guodong, and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines." IEEE transactions on Neural Networks 14.1 (2003): 209-215.

[18] Avialable: http://cmusphinx.sourceforge.net/wiki/tutorialam

[19] Liu, Yi, and Pascale Fung. "State-dependent phonetic tied mixtures with pronunciation modeling for spontaneous speech recognition." IEEE transactions on speech and audio processing 12.4 (2004): 351-364.

[20] Available: http://www.alsabahpress.com/

[21] Available: http://cmusphinx.sourceforge.net/wiki/cmuclmtkdevelopment

Chapter 7:

Modeling Capacity of Mel Frequency Cepstral Coefficient

1. Introduction

Automatic speech recognition (ASR) is an attractive user friendly technology to felicitate human computer interface (HCI) in different domains. In the last years, there has been a growing interest to reinforce natural man-machine communication through speech technology. In this regard, much research has been devoted to introduce innovative ideas in the industry for automation purpose (e.g. banking services, cars, control machines, etc). In general, sound is made out of vibrations of an object to generate a type of energy. The energy causes a movement in the air particles that propagate as audible waves. The air particles movement keeps going until they run out of energy. Humans can hear sound waves with frequencies between about 20 Hz (cycles per second) and 20 kHz. However, the most sensitive limit of human hearing is in the 2000 - 5000 Hz frequency range. In general, machine-learning systems perform feature extraction process at the first place in order to produce the feature values based on the input patterns, these speech features are then pass to an ASR system.

Mel frequency cepstral coefficients (MFCC) is the classical front-end analysis in speech recognition to produce the sequence of real-valued numbers that represent feature vectors based on the input signal. Since 1980, it has dominated the ASR feature extraction methods due to its good performance. The success of MFCC makes it the standard choice in the state-of-the-art speech recognizers such as the Carnegie Mellon University (CMU) Sphinx [1], the Markov Model Toolkit (HTK) [2], and the Kaldi speech recognizer [3]. The literature shows that there is a variety of feature extraction methods; however, it is clearly observed that MFCC is extensively used in the most speech classification tasks. An example of another feature extraction method is Perceptual Linear Prediction (PLP) [4]. In fact, previous studies shows that MFCC is

an appropriate choice to maximize the recognition performance as reported by [5]. It indicates that the MFCC is characterized by better performance and ability of the frequency domain to model adequately the sound. Reference [6] indicated the MFCC and the relative spectral analysis PLP are the most commonly used due to their ability to provide more robust features in adverse conditions. Similarly, Reference [7] demonstrated that the most of today's ASR systems are based on some types of MFCC, which have proven to be effective and robust under various conditions.

The rest of this study is organized as follows. In the next section, we present some of the challenges of speech features. In section 3, we present the background of MFCC technique followed by the literate review in section 4. Finally, we conclude in section 5.

2. Speech Features Challenges

Due to the difficulties of handling speech features, it has been long observed that the ASR researches employ off-the-shelf toolboxes for features extraction. It is clear that employing MFFC, or even other speech features, for speech applications is not a straightforward task since some of the intermediate functions are difficult for non-specialist researchers. For instance, writing a program for fast Fourier transform (FFT), which is the heart of computing MFCC, requires highly qualified scientists or engineers who have a solid background in complex mathematics, and then, can understand and write FFT program from the scratch. No doubt, conducting valuable research that include speech processing (e.g. speech recognition or speech synthesis) requires deep understanding of signal processing. Speech features pose some challenges in terms of the nature of the data. For instance, textual data or even images features are constant, which remain fixed wherever appear. To clarify, the features of an article (i.e. the words or the roots are always same for a particular text), however, speech features are not constant as they are continuously changed according to different aspect such as gender, accent, age, etc. In simple, it is hard to directly compare the speech features due to (small) differences in vibrations that leads to completely different sound. The speech-recording environment might have noise such as background music, a second speaker, unwanted inhalation, the quality of the microphone, or the health and psychological state of person. Reference [8] has a thorough study of the pronunciation variations sources that degrade the performance of ASR systems. In fact, humans can easily interpret signals by extracting relevant information, however, this task is more complex when performed using

signal processing and machine learning algorithms. More problems can be observed regarding the speech context. Sounds are quite substantially changed by the surrounding context. The vocal tract goes through different stages getting from ‘t’ to ‘a’ and getting from ‘r’ to ‘a’, and the parameters during the transition will be different as indicated in [9] . Moreover, sounds can last different amounts of time. Deciding where one ends and the next one starts is hard.

Moreover, the speech extraction process is a tricky task that requires care and skill. The input waveform is sliced up into frames (usually of 20~30 milliseconds) to generate speech spectrum, which is the distribution of energy as a function of frequency for a particular sound source [10]. Therefore, the waveform is transformed into spectral features (i.e. acoustic feature vectors) as shown in Fig. 1. The figure is obtained from reference [11], which has more details of speech and language processing. For general overview of the difficulty to handle speech recognition, reference [12] elaborates on some of the difficulties with ASR.

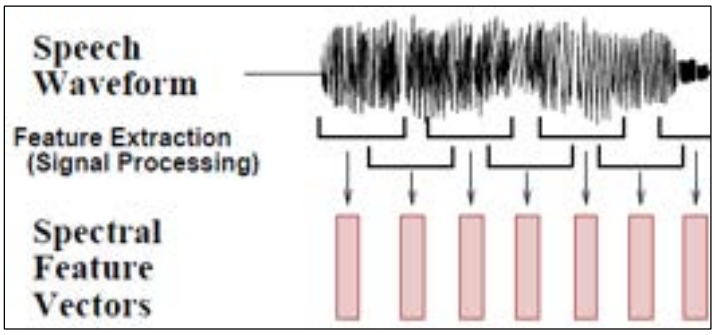


Fig. 1. Extracting features by dividing the signal into frames.

3. MFCC Background

To compute the MFCC, the time domain representation of the input speech signal is used to produce the spectral properties, as the patterns are more evident in the frequency domain. The MFCC consists of a set (39 coefficients) that represents the speech signal by dividing it to a set of overlapping short segments called frames. In particular, MFCC coefficients represent the spectral envelope of the speech signal on the mel-frequency scale. Fig.2 shows the steps to extract the MFCC of a speech signal. For better performance, the temporal properties might be considered to obtain the first and the second derivative (named respectively ΔMFCC and ΔΔMFCC) of the first order 13 coefficients. We emphasize that the first step, which is sampling and quantization, is

performed by the sound card (i.e. a hardware related issue) and is not a part of the MFCC process. However, it is shown in the figure as an indication of nature of the input data for the Preemphasis stage. The goal of sampling and quantization (also called digitization) step is to convert the analog signal to digital forms for further processing. The sampling rate is the number of samples taken per second, while quantization is the process of representing real-valued numbers as integers. It is worthy to indicate that the MFCC process is not invertible; it is impossible to get the signal back from the set of MFCCs.

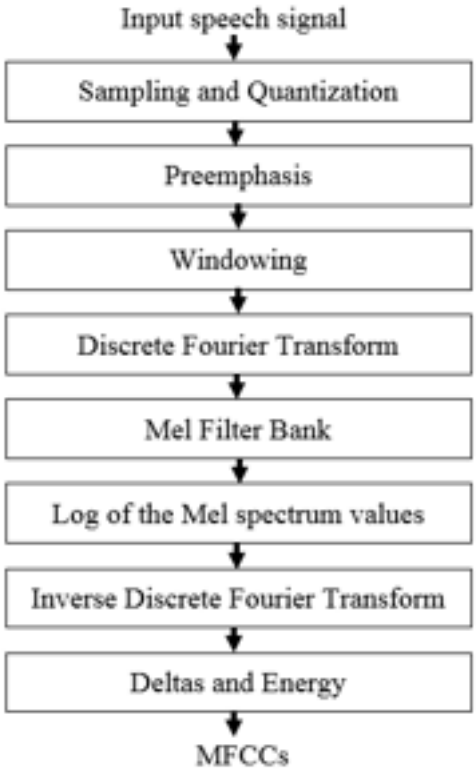


Fig. 2. Extracting features using MFCC algorithm.

Reference [13] highlighted some reasons of MFCC popularity in parametric representation of the spectrum as the follows. First, the calculation of these parameter leads to a source-filter separation. Second, the parameters have an analytically tractable model. Third, experience proves that these parameters work well in recognition applications. The following is a brief description of the tasks to extract the speech features:

Preemphasis: Preemphasis is performed after the digitization step. It aims at increasing the amplitude of high frequency bands and decrease the amplitudes of lower bands. That is, this stage

is to attain the high frequency formants that carry the relevant information. Without Preemphasis, it might be difficult for the receiver to interpret the signal due to the suppression during the sound production mechanism. Hence, the purpose of Preemphasis is to apply to the signal with the proper weight sometime called alpha. The Preemphasis is also considered as noise reduction module as it leaves the desired signal untouched, but reduces the noise power considerably.

Windowing: the preemphasised speech signal is subjected to the short-time Fourier transform analysis with frame durations of 20~30ms, frame shifts overlap of around 10ms. In this stage, the speech signal is analyzed to extract the stationary portion of speech using a window function, which can be characterized by minimizing the discontinuities of the signal.

Discrete Fourier Transform: This stage is the basis of spectral analysis to extract the speech features based on magnitude spectrum computation. It is performed by decomposing an N point time domain signal to obtain the magnitude frequency response of each frame. That is, it calculate the N frequency spectra corresponding to the N time domain signals

Mel Filter Bank: Computing the mel-frequency spectrum is performed after the discrete Fourier transform by passing the spectrum through Mel-Filters to obtain Mel-Spectrum. To produce the filter-bank energies, a number of triangular filters are used that are uniformly spaced on the mel scale between lower and upper frequency. It is used to approximate the frequency resolution of the human ear. That is, the Mel scale approximates the sensitivity of the human ear. The Mel frequency scale is linear up to 1000 Hz and logarithmic thereafter.

Log of the Mel spectrum values: The range of the values generated by the Mel filter bank is reduced by replacing each value by its natural logarithm. This is done to make the statistical distribution of the spectrum approximately Gaussian.

Inverse Discrete Fourier Transform: This transform is used to compress the spectral information into a set of low order coefficients. This representation is called the Mel-cepstrum.

Deltas and Energy: the previous step provides the 12 cepstral coefficient for each frame. This step is to add the 13th feature: the energy from the frame. It is useful to identify phone identity.

To explain the output of the MFCC algorithm, we used a small speech file that contains a single word "الاسهم" that means "stocks". The speech waveform of this word is shown in Fig. 3. In addition, the spectrogram of this word is shown in Fig. 4. The spectrogram is a visualization tool that is used to understand the information in the signal using time and frequency. Acoustic phones and their

properties are better observed in spectrogram. The spectrogram representation of the speech signal is based on short-time Fourier analysis. In the spectrogram, if gray scale is used, the higher the amplitude (the energy), the darker the corresponding region. However, if color scale is used, the blue represents the low energy while the red parts represents high energy [14].

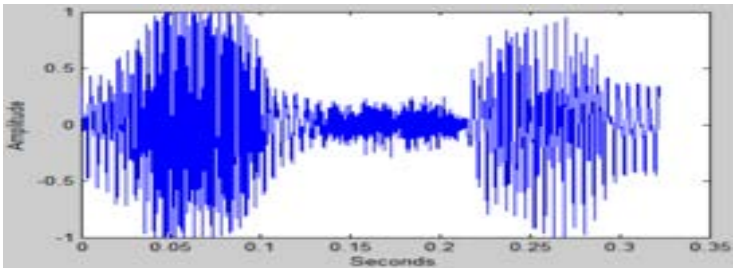


Fig. 3. A speech signal waveform of the Arabic single word "as'hum".

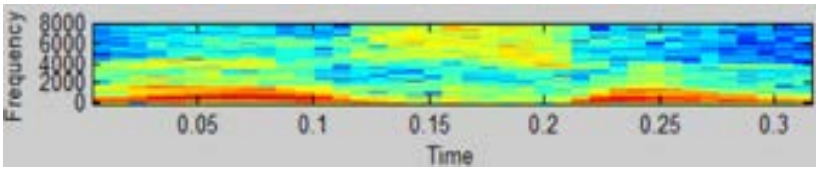


Fig. 4. The spectrogram of a single Arbic word speech file "as'hum".

The HTK system was used to extract the MFCC speech features of the single word speech file that is represented in Fig. 3. The speech file is of length 0.323 seconds and uses a sampling rate of 16 kHz with 16-bit quantization for each sample. Table I shows the first 12-order of the MFCC coefficients after completing the feature extraction process. Each column represents the 13 features (the 13th feature is the energy from the frame) of a 25 milliseconds frame.

TABLE I. MFCC OF A SINGLE WORD SPEECH FILE

Frame					
Feature	1	2	3	4➔29	30
1	-1.81965	-3.15548	-3.76447	…	1.40033
2	-3.10861	-7.92128	-9.68467	…	1.19619
3	1.95010	-2.75036	-4.08556	…	2.44150
4	-12.21996	-14.16043	-15.84370	…	-8.50239
5	-8.21085	-10.14035	-13.00282	…	-11.45462
6	-14.98533	-13.45490	-17.62610	…	-0.93472
7	-22.24395	-23.61402	-13.59459	…	-10.85606
8	2.53291	-1.30409	10.01444	…	-4.10245
9	-8.75291	-14.95345	-3.51132	…	-15.78435
10	-7.62615	-4.33472	-2.60953	…	-13.07826
11	-4.17761	-6.58369	-8.04277	…	3.56489
12	-8.05171	-7.96873	-10.30831	…	-7.61198

13	80.21140	83.59767	86.78981	…	76.75751

The extracted MFCC speech features shown in Table I were extracted using the HTK-HCopy command and the default parameters [2] shown in Table II. A configuration file (generally called config) is needed which specifies all of the conversion parameters. The HCopy command is used as the following, supposing that the input speech file is "sample.wav":

```
HCopy -C config.txt sample.wav sample.mfc
```

However, the HCopy command creates a binary file (special format non-text file) that contains the MFCC data. Therefore, another option to obtain the MFCC data in textual form is by using HTK-HList command as the following:

```
HList -C config.txt -r sample.wav
```

TABLE II. TYPICAL HTK SETTINGS – CONFIGURATION FILE

Coding parameters	Comments
SOURCEFORMAT = WAV	The format of the source file
TARGETKIND = MFCC_0	Cepstral C_0 coefficient appended
TARGETRATE = 100000.0	10ms frame rate
SAVECOMPRESSED = T	Save the output file in compressed form
SAVEWITHCRC = T	Attach a checksum to output parameter file
WINDOWSIZE = 250000.0	25ms window
USEHAMMING = T	Use a Hamming window
PREEMCOEF = 0.97	Set pre-emphasis coefficient
NUMCHANS = 26	Number of filterbank channels
CEPLIFTER = 22	Cepstral liftering coefficient
NUMCEPS = 12	Number of cepstral parameters

4. Speech Features Challenges

Based on a thorough review of the Arabic speech recognition literature, it is observed that MFCC is extensively used in the most of Arabic ASR studies. Table III shows some of the previous studies. However, some of the studies employ other feature extraction method such as the first

work in Table III. in the table, the LPCC is the shorthand of linear prediction spectrum coefficients, which is one of the famous speech features extraction method. As illustrated, the information in the table belongs to two main categories of speech recognition; the isolated and the continuous speech recognition. Table III also reveals that Arabic speech recognition is in row stages as the most of works depend on off-the-shelf tools (MFCC based tools), which reduce the opportunities to investigate different speech features as well as reduce the opportunity to present innovative ideas (i.e. featuring new methods).

TABLE III. PREVIOUS STUDIES EMPLOYING MFCC

Isolated speech (digits or control command)		
Reference	Year	Features
[15]	2001	LPCC
[16]	2003	MFCC
[17]	2003	MFCC
[18]	2006	MFCC
[19]	2007	MFCC
[5]	2007	MFCC
[19]	2008	MFCC
[21]	2008	MFCC
[22]	2009	MFCC
Continuous speech		
Reference	Year	Features
[23]	2007	MFCC
[24]	2008	MFCC
[25]	2010	MFCC
[26]	2011	MFCC
[27]	2011	MFCC
[28]	2012	MFCC
[29]	2012	MFCC
[30]	2017	MFCC

5. Conclusion

This study demonstrates the MFCC speech features extraction method as one of the most commonly used in ASR systems. Compared to other speech features extraction methods, MFCC is standard choice for front-end feature in state-of-the-art ASR systems. According to our best knowledge and the review that we performed on the previous studies of Arabic ASR, we found that MFCC dominates the works in this field. We employed the HTK system to demonstrate the extraction process of MFCC speech feature vectors of a simple speech file. As a future work, it is worth to continue this work by conducting a practical research to compare MFCC with other methods such as LPCC and PLP.

References

[1] https://cmusphinx.github.io/wiki/faq/ Accessed on 31 March 2017.

[2] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

[3] Povey, Daniel, et al. "The Kaldi speech recognition toolkit." IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[4] Hermansky, Hynek. "Perceptual linear predictive (PLP) analysis of speech." the Journal of the Acoustical Society of America 87.4 (1990): 1738-1752.

[5] Haraty, Ramzi A., and Omar El Ariss. "CASRA+: a colloquial Arabic speech recognition application." American Journal of Applied Sciences 4.1 (2007): 23-32.

[6] Sharma, Davinder Pal, and Jamin Atkins. "Automatic speech recognition systems: challenges and recent implementation trends." International Journal of Signal and Imaging Systems Engineering 7.4 (2014): 220-234.

[7] Molau, Sirko, et al. "Computing mel-frequency cepstral coefficients on the power spectrum." Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on. Vol. 1. IEEE, 2001.

[8] Benzeghiba, Mohamed, et al. "Automatic speech recognition and speech variability: A review." Speech communication 49.10 (2007): 763-786.

[9] Ramsay, Allan. "HOW DO SPEECH RECOGNISERS WORK?" A presentation. Kuwait Univeristy (2016).

[10] http://www.thefreedictionary.com/ Accessed on 31 March 2017.

[11] Jurafsky, Dan. Speech & language processing. Pearson Education India, 2000.

[12] Forsberg, Markus. "Why is speech recognition difficult." Chalmers University of Technology (2003).

[13] Alcaraz Meseguer, Noelia. Speech analysis for automatic speech recognition. MS thesis. Institutt for elektronikk og telekommunikasjon, 2009.

[14] Huang, Xuedong, et al. Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR, 2001.

[15] Bahi, Halima, and Mokhtar Sellami. "Combination of vector quantization and hidden Markov models for Arabic speech recognition." *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*. IEEE, 2001.

[16] Elmisery, F. A., et al. "A FPGA-based HMM for a discrete Arabic speech recognition system." *Microelectronics, 2003. ICM 2003. Proceedings of the 15th International Conference on*. IEEE, 2003.

[17] Amrouche, Abderrahmane, and J. Michel Rouvaen. "Arabic isolated word recognition using general regression neural network." *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*. Vol. 2. IEEE, 2003.

[18] Bourouba, H., et al. "New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition." *Information and Communication Technologies, 2006. ICTTA'06. 2nd*. Vol. 1. IEEE, 2006.

[19] Satori, Hassan, Mostafa Harti, and Nouredine Chenfour. "Introduction to Arabic speech recognition using CMUSphinx system." *arXiv preprint arXiv:0704.2083* (2007).

[20] Essa, E. M., A. S. Tolba, and S. Elmougy. "A comparison of combined classifier architectures for Arabic Speech Recognition." *Computer Engineering & Systems, 2008. ICCES 2008. International Conference on*. IEEE, 2008.

[21] Azmi, M., et al. "Syllable-based automatic arabic speech recognition in noisy-telephone channel." *WSEAS Transactions on Signal Processing* 4.4 (2008): 211-220.

[22] Satori, Hassan, et al. "Investigation arabic speech recognition using CMU sphinx system." *Int. Arab J. Inf. Technol.* 6.2 (2009): 186-190.

[23] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." *International Journal of Speech Technology* 10.4 (2007): 183-195.

[24] Alotaibi, YousefAjami, Sid-Ahmed Selouani, and Douglas O'shaughnessy. "Experiments on automatic recognition of nonnative Arabic speech." *EURASIP Journal on Audio, Speech, and Music Processing* 2008.1 (2008): 679831.

[25] Selouani, Sid Ahmed, and Malika Boudraa. "Algerian Arabic speech database (ALGASD): corpus design and automatic speech recognition application." Arabian Journal for Science and Engineering 35.2C (2010): 158.

[26] AbuZeina, Dia, et al. "Toward enhanced Arabic speech recognition using part of speech tagging." *International Journal of Speech Technology* 14.4 (2011): 419-426.

[27] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." *International Journal of Speech Technology* 14.3 (2011): 227-236.

[28] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: a direct data-driven approach." *International Journal of Speech Technology* 15.2 (2012): 65-75.

[29] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." *International Arab Journal of Information Technology (IAJIT)* 9.1 (2012): 84-93.

[30] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology(2017): 1-9.

CHAPTER 8:

LANGUAGE MODELING TOOLKITS FOR ARABIC TEXT

1. Introduction

Language models (LMs) are of significant contribution to the performance of natural language processing (NLP) systems such as automatic speech recognition (ASR) and machine translation. LMs have been successfully applied in different linguistic applications such as Part-of-Speech (PoS) tagging, parsing, information retrieval, spell correction, summarization, etc. In particular, LM is a critical component in linguistic applications that produce sequences of words as output. In the last decades, extensive research has been devoted to promote new techniques to compile LMs as well as to address some challenges such as missing some of n-grams. In speech recognition, the ASR decoder uses the information that is provided in the LM to find the best possible word sequence of the testing speech for transcription purpose. In general, it is extremely important for the language applications to have the ability to predict the next word given the previous word(s), or the history.

LMs can be either probabilistic or non-probabilistic. The probabilistic LMs are known as statistical LMs, such as N-grams, while the non-probabilistic is known as “any-word” grammar. Any-word grammar does not use probabilities for words. It is unconstraint grammar that leads to very poor accuracy in continuous ASR systems. That is, any-word grammar relies entirely on the acoustic model. On the other hand, statistical language model is based on computing the probabilities of all word combinations (i.e. all possible word sequences) in the training source text. Statistical language models are generally demonstrated using ARPA format textual files that include the statistical estimation of the desired N-grams, typical up to 3-grams. No doubt, compiling a statistical language model requires a large number of words from different textual resources. In addition, the data should not be too specific to a particular domain; otherwise, it will not generalize well to the sentences in question.

During ASR decoding, the recognizer employs the language model to transcribe speech files using the acoustic model and the dictionary (vocabulary). Hence, the most likely hypothesis for each testing speech file is generated as an output. Employing language models reinforces the speech recognition accuracy as the more you can constrain the range of possible utterances, the more accurate the recognizer will be. Based on the information that is provided in the N-grams, the probability of a words sequence is computed using the following formula [1]:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_1^{k-1})$$

Where n is limited to include the words' history. Hence, the Chain Rule applied to compute joint probability of words in sentence. LMs utilize Markov assumption to simplify data estimation. For instance, for *n*=2, the bigram is calculated for the words sequence as follows:

$$P(w_1\, w_2) = P(w_2\,|w_1)P(w_1\,)$$

This work aims at demonstrating the process of computing the N-grams for small Arabic text corpus that contains five sentences using two well-known toolkits. The toolkits are the Carnegie Mellon University (CMU) -Cambridge language modeling toolkit [2] and the Cambridge University Hidden Markov Model Toolkit (HTK) language modeling tools [3]. Of course, there are other popular tools such as the SRI Language Modeling Toolkit (SRILM) [4]. For clarification, we demonstrate all intermediate steps that are required to produce the language models, which include model estimation using the training data. This work also demonstrates some grammars that are mainly used for isolated words such as digits and control commands. The implementation requires to run the command line under UNIX. In this work we used Cygwin which is a Unix-like environment for Microsoft Windows.

In next section, we present the literate review. In section 3, we present the grammars followed by the background of N-gram language models in section 4. In section 5, we present HTK toolkits followed by CMU-Cambridge toolkits in section 6. Finally, we conclude in section 7.

2. Literature Review

The literature shows that LMs have been exploited in many linguistic applications. For instances, Reference [5] used CMU-Cambridge language modeling toolkit LM for continuous Arabic ASR. Reference [6] augmented naive Bayes models with statistical N-gram language models to address short-comings of the standard naive Bayes text classifier. Reference [7] proposed a model, called

emoticon smoothed language model for Twitter sentiment analysis. Reference [8] investigated the document categorization task with statistical language models. Reference [9] reported the benefits of largescale statistical language modeling in machine translation. Reference [10] proposed to use a new statistical language model that is based on a continuous representation of the words in the vocabulary for machine translation. Reference [11] employs statistical language models in temporal action detection. Reference [12] explores recent advances in recurrent neural networks for large scale language modeling. Reference [13] employs statistical LMs for large statistical machine translation task. Reference [14] explores the application of neural LMs to machine translation. Reference [15] employs LMs for text summarization.

3. Grammars

In this section, we demonstrate the grammars language models. Such grammars are simple that do not have probabilities and are designed according to the information that is provided in the corresponding application. That is, grammars mainly contain isolated words such as commands, control words, and digits. However, grammars might allow sequences of words. Fig. 1 shows a simple grammar for ten digits that can be used in continuous speech recognition to choose one or more words from the list. The grammar is written using JSGF format as shown in Fig. 1.

```
#JSGF V1.0;
grammar myGrammar;
public <command> = <word>* ;
<word> =( One | Two | Three | … | Ten);
```

Another form of the ten digits grammar

```
$WORD = (One | Two | Three | … | Ten);
($WORD)
```

Fig. 5. A part of any-word grammar for ten digits

Grammars are usually written by hand or it can be generated using a program. Despite most of grammars do not use probabilities, however, some elements might be weighted. In fact, grammars are rarely used in ASR systems since the probabilistic models of a language is more useful than the hard models (i.e. grammars) of the legal sentences in the languages.

4. N-grams Language Models

Statistical N-grams language models are the most widely used language model in speech recognition. That is, the goal of LM is to compute the probability of a sentence or sequence of words. The most likely sequence of words is estimated, given the speech feature vectors, is given by [1]:

$$\widehat{w} = \underset{W \in L}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} = \underset{W \in L}{\operatorname{argmax}} P(O|W)P(W)$$

Where $\widehat{w}$ is the most likely recognized words, $P(O|W)$ is the probability of the feature vectors, given a sequence of words that is computed using the acoustic model, $P(W)$ is the probability of the words sequence that is computed using the language model. $P(O)$ is the probability of the acoustic observation sequence and can be ignored. Hence, the statistical language model has to be computed at the first place in order to decode the testing speech files in ASR systems. The statistical N-grams language model is trained by counting N-grams occurrences in a large transcription corpus to be then smoothed and normalized. N-gram models can be trained by counting and normalizing. The following formula is used to estimate the N-grams parameter [1]:

$$P(w_n | w_{n-N+1}^{n-1}) = \frac{Count(w_{n-N+1}^{n-1} w_n)}{Count(w_{n-N+1}^{n-1})}$$

One major problem in LMs is unseen words or n-grams that are found in the testing set while, at the same time, out of vocabulary. Accordingly, a probability of 0.0 is given to the items that are not seen in the training data. That is, not all n-grams will be present (i.e. not observed) in the training data. One solution is smoothing by assigning non-zero or small probabilities to unseen n-grams in which all word sequences can occur with some probability. Hence, smoothing provides a better way of estimating the probability of zero frequency n-grams which never occur in order to produce generalized LMs. Smoothing is also called discounting. When creating a language model, it is more efficient to use log probabilities rather than actual probabilities due to the risk of numerical underflow especially in very long strings. It is also efficient in ASR decoding algorithm such as Viterbi algorithm.

Creating an N-grams language model follows three main steps that are: compute the word unigram counts, convert the word unigram counts into a vocabulary list, and generate bigram and trigram (or more) tables based on this vocabulary. As a preprocessing step, it is must to include special words such as <s> to indicate for the “start of sentence” and the </s> to indicate for the “end of

the sentence". CMU-Cambridge toolkits uses <UNK> token to indicate for unknown words whereas HTK toolkits uses !!UNK for the same purpose. In order to demonstrate the process to create a statistical language model for Arabic text, we prepared a small set that contains five sentences to demonstrate the required steps. The training data set is shown in Fig. 2. The sentences include 22 unique words.

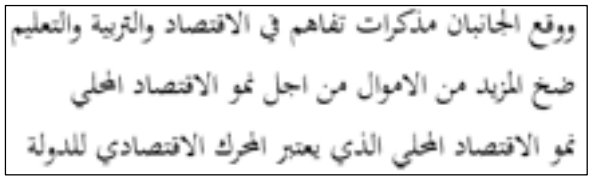


Fig. 6. A small corpus tha contains three sentences

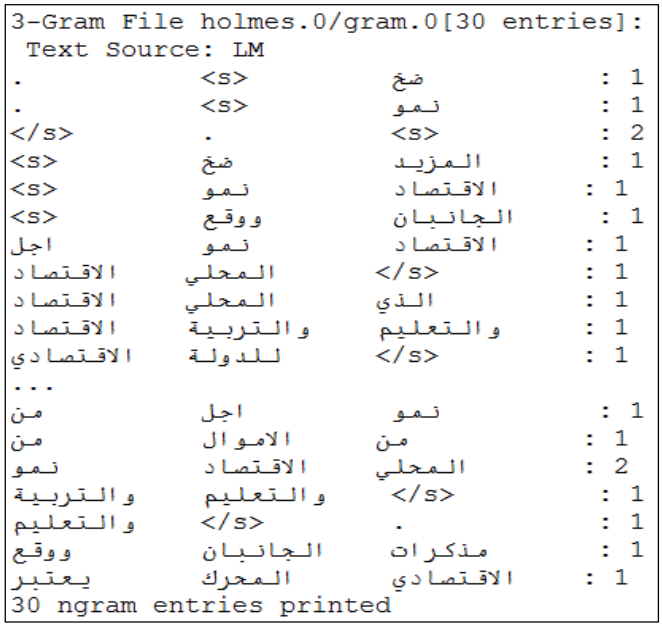
5. The HTK Toolkits

HTK provides two approaches to generate N-grams. The first method employs ***HLStats*** function, which is used exclusively to computer bigram language model (i.e., 2-grams). The words probabilities is computed using the following formula [16]:

$$p(i,j) = \begin{cases} (N(i,j) - D)/N(i) & if\ N(i,j) > t \\ b(i)p(j) & otherwise \end{cases}$$

Where $N(i,j)$ is the number of times word j follows word i and $N(i)$ is the number of times that word i appears. D is the discount constant that has a default value 0.5. t is a threshold that is used to ensure that all bigram probabilities for a given history sum to one. The LMs that is generated using ***HLStats*** use probabilities as base-10 logs.

The second method implements a series of functions to computer N-grams. In this work, we implement the second method that is demonstrated by HTK toolkit in [16] to compute N-grams for general N-grams. That is, it is not restricted for bigram as the case in ***HLStats*** method. However, the second method is not one command as the case in ***HLStats***, but it requires several steps to compute N-grams as the following [16]: ***LNewMap*** to prepare an initial empty word map file. This can help to add future n-gram files without having to rebuild existing ones. The created word map file contains just a header and no words. ***LGPrep*** to process the training text data to produce all newly encountered words and the identifiers that the tool has assigned them. ***LGList*** contains the collected N-grams (3-grams in our case) as shown in Fig. 3.



```
3-Gram File holmes.0/gram.0[30 entries]:
 Text Source: LM
.              <s>            ضخ             : 1
.              <s>            نمو            : 1
</s>           .              <s>            : 2
<s>            ضخ             المزيد         : 1
<s>            نمو            الاقتصاد       : 1
<s>            ووقع           الجانبان       : 1
اجل            نمو            الاقتصاد       : 1
الاقتصاد       المحلي         </s>           : 1
الاقتصاد       المحلي         الذي           : 1
الاقتصاد       والتربية       والتعليم       : 1
الاقتصادي      للدولة         </s>           : 1
...
من             اجل            نمو            : 1
من             الاموال        من             : 1
نمو            الاقتصاد       المحلي         : 2
والتربية       والتعليم       </s>           : 1
والتعليم       </s>           .              : 1
ووقع           الجانبان       مذكرات         : 1
يعتبر          المحرك         الاقتصادي      : 1
30 ngram entries printed
```

Fig. 7. A part of the the 3-grams in the triaing data

LGCopy is employed to derive a sequenced set of N-grams files. The word list should be supplied in a spate file that contains the system's vocabulary. The unknown word symbol defaults to !!UNK. ***LGList*** saves the new word map containing the new class symbols (!!UNK in this case) and only words in the vocabulary. ***LFoF*** produces a frequency of frequency (FoF) table for the chosen vocabulary list. Finally, ***LBuild*** builds the actual language model. Fig. 4 shows a part of the generated 3-grams language model using ARPA format.

The 3-grams shows in Fig. 4 shows the contents of the N-grams language model along with the corresponding probabilities of the 1-grams cases (23 unigrams), 2-grams cases (contains 4 bigrams), and 3-grams cases (2 trigrams). The ARPA format shown in Fig. 4 is the language model form that is generally used in in speech recognition systems. In Fig. 4, the probabilities (in term of $\log_{10}$) are stored on the left of the n-grams while the back-off weight (in terms of $\log_{10}$) is stored on the right of the word(s).

```
\data\
ngram 1=23
ngram 2=4
ngram 3=2
\1-grams:
-1.1614 !!UNK    -2.0000
-1.1614 </s>     -1.9690
-99.9900         <s>
-1.4624 اجل
-0.9853 الاقتصاد -0.4375
-1.4624 الاقتصادي
...
-1.1614 نمو      -1.9526
-1.4624 والتربية
-1.4624 والتعليم
-1.4624 ووقع
-1.4624 يعتبر
\2-grams:
-0.0044 !!UNK <s>
-0.0044 </s> !!UNK       +0.0000
-0.1805 المحلي الاقتصاد
-0.0044 الاقتصاد نمو     -1.5315
\3-grams:
-0.0044 </s> !!UNK <s>
-0.0044 المحلي الاقتصاد نمو
\end\
```

Fig. 8. A part of the 3-grams using HTK toolkit

6. The CMU-Cambridge Toolkits

We used the CMU-Cambridge toolkit to compute the N-grams for the dataset that is described in section 4. In particular, we computer the 1-grams, 2-grams, and the 3-grams. The CMU-Cambridge toolkit uses the following five commands to produce the LM dump file: ***text2wfreq***, ***wfreq2vocab***, ***text2idngram*** , ***idngram2lm***, ***lm3g2dmp*** [17]. The outputs of the previous commads are shown in Fig. 5.

text2wfreq	wfreq2vocab	text2idngram

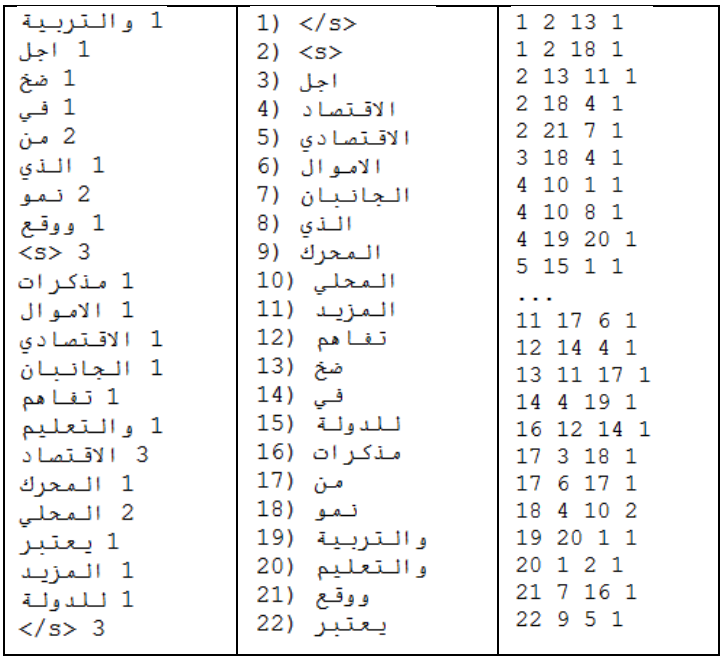


Fig. 9. The outpus of three commands to generat LM

The descriptions of the CMU-Cambridge toolkits command is as follows: ***text2wfreq***: list of every word which occurred in the text, along with its number of occurrences. ***wfreq2vocab***: a vocabulary file. ***text2idngram***: list of every id n-gram which occurred in the text, along with its number of occurrences. ***idngram2lm*** : a language model, in either binary format, or in ARPA format. ***lm3g2dmp***: convert the language model file from ARPA format to DMP format. Fig. 6 shows the outputs of the previous commands. This is a 3-gram language model, based on a vocabulary of 22 words. We highlight that the output of the ***text2idngram*** command is the 3-grams occurred in the source text. For instance, the sequence 18 4 10 reprents the trigram "نمو الاقتصاد المحلي" and it occurs two times as indicated in the Fig. 5, the right colmun.

The ARPA-standard format LM that is shown in Fig. 6 contains 23 items of the 1-grams, 26 items of 2-gram, and 28 item of the 3-grams. For each n-gram, the value on the left represents the actual probability while the value on the right represents the back-off weights. According to the LM in Fig. 6, the actual probability of the word "الاقتصاد" is -0.9853 (in $\log_{10}$ format). The probability in standard from is $10^{-0.9853}$=0.101. The LM also shows that the probability of the 3-gram " نمو الاقتصاد المحلي" is greater than the 3-gram "ووقع الجانبان مذكرات" as the actual probability of the first 3-gram is ($10^{-0.1761}$) 0.6666 while the prbability of the second 3-gram is almost zero ($10^{-99.9990}$).

```
\data\
ngram 1=23
ngram 2=26
ngram 3=28
\1-grams:
-1.5168 <UNK>    0.0000
-1.1614 </s>     -0.4297
-0.9853 <s>      0.0604
-1.5168 اجل      0.0310
-0.9853 الاقتصاد -0.4317
...
-1.5168 ووقع     0.0134
-1.5168 يعتبر    0.0134
\2-grams:
-0.1761 </s> <s> 0.0000
-99.9990 <s> ضخ 0.0000
...
-99.9990 والتعليم </s> 0.4771
-99.9990 الجانبان ووقع 0.0000
-99.9990 المحرك يعتبر 0.0000
\3-grams:
-99.9990 </s> <s> ضخ
-99.9990 </s> <s> نمو
-99.9990 <s> المزيد ضخ
...
-0.1761 المحلي الاقتصاد نمو
-99.9990 والتعليم والتربية </s>
-99.9990 والتعليم </s> <s>
-99.9990 مذكرات الجانبان ووقع
-99.9990 الاقتصادي المحرك يعتبر
\end\
```

Fig. 10. A part of the the 3-grams using CMU-Cambridge toolkit

Perplexity are the most common metrics used to evaluate N-grams LM. Perplexity is defined in terms of the inverse of the average log likelihood per word. It is an indication of the average number of words that can follow a given word, a measure of the predictive power of the language model. It is a way to measure the quality of a model independent of any NLP system. The lower perplexity system is considered better than one of higher perplexity. The perplexity formula is:

$$PP(W) = N\sqrt{\frac{1}{P(w1, w2, \dots, w_N)}}$$

Where PP is the perplexity, P is the probability of the word set to be tested W=w1, w2, … , w_N, and N is the total number of words in the testing set. Since we used a very small data set, we did not evaluate the perplexity in this work.

7. Conclusion

This study demonstrates two well-known LMs toolkits; the CMU-Cambridge and HTK toolkits. We used a small data set to demonstrate the steps to compute N-grams. We also highlight the basic

concepts of the grammars language models that is rarely used in NLP systems. As a future work, we recommend to investigate new research directions to run NLP systems while skipping the language models. That is, evaluating the performance if we employ an ASR system that entirely use acoustic models.

References

[1] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. Pearson, 2014.

[2] Available: http://www.speech.cs.cmu.edu/SLM/toolkit.html

[3] Available: http://htk.eng.cam.ac.uk/download.shtml

[4] Available: http://www.speech.sri.com/projects/srilm/

[5] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[6] Liu, Kun-Lin, Wu-Jun Li, and Minyi Guo. "Emoticon smoothed language models for twitter sentiment analysis." AAAI. 2012.

[7] Peng, Fuchun, Dale Schuurmans, and Shaojun Wang. "Augmenting naive bayes classifiers with statistical language models." Information Retrieval 7.3 (2004): 317-345.

[8] Tantug, Ahmet Cüneyd. "Document categorization with modified statistical language models for agglutinative languages." International Journal of Computational Intelligence Systems 3.5 (2010): 632-645.

[9] Brants, Thorsten, et al. "Large language models in machine translation." In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007.

[10] Schwenk, Holger, Daniel Dchelotte, and Jean-Luc Gauvain. "Continuous space language models for statistical machine translation." Proceedings of the COLING/ACL on Main conference poster sessions. Association for Computational Linguistics, 2006.

[11] Richard, Alexander, and Juergen Gall. "Temporal action detection using a statistical language model." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[12] Jozefowicz, Rafal, et al. "Exploring the limits of language modeling." arXiv preprint arXiv:1602.02410 (2016).

[13] Schwenk, Holger, Anthony Rousseau, and Mohammed Attik. "Large, pruned or continuous space language models on a gpu for statistical machine translation." Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT. Association for Computational Linguistics, 2012.

[14] Vaswani, Ashish, et al. "Decoding with Large-Scale Neural Language Models Improves Translation." EMNLP. 2013.

[15] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[16] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

[17] http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

CHAPTER 9:
MARKOV CHAIN MODELS IN LINGUISTICS

1. Introduction

Markov chains theory is increasingly being adopted in real-world computing applications since it provides a convenient way for modeling temporal, time-series data. At each clock tick, the system moves into a new state that can be the same as the previous one. A Markov chain model is a mathematical tool that capture the patterns dependencies in pattern recognition systems. For this reason, Markov chain theory is appropriate in natural langue processing (NLP) where it is naturally characterized by dependencies between patterns such as characters or words.

Markov chains are directed graphs (a graphical model) that are generally used with relatively long data sequences for data-mining tasks. Such tasks include prediction, classification, clustering, pattern discovery, software testing, multimedia analysis, networks, etc. Reference [1] indicated that there are two reasons of Markov chains popularity; very rich in mathematical structure and work well in practice for several important applications. Hidden Markov models (HMM) is an extension of Markov chains that used to find the hidden system's states based on the observations.

In order to facilitate the research in this direction, this study provides a survey of this so popular data modeling technique. However, because of the wide range of the research domains that use this technique. We specifically focus on the linguistics related applications. Reference [2] list some domains that utilize Markov chains theory which include: physics, chemistry, testing, speech recognition, information sciences, queueing theory, internet applications, statistics, economics and finance, social sciences, mathematical biology, genetics, games, music, baseball, Markov text generators, bioinformatics. Reference [3] lists the five greatest applications of Markov chains that include Scherr's application to computer performance evaluation, Brin and Page's application to PageRank and Web Search, Baum's application to HMM, Shannon's application to information theory, and Markov's application to Eugeny Onegin.

This study is organized as follows. The next section presents a background of Markov chains theory. Section 3 highlights the main concepts of HMM followed by a literature review of Markov chains and HMM in section 4. Finally, we conclude in section 5.

2. Markov Chains

Markov chains are quite useful in modeling computational linguistics. A Markov chain is a memoryless stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [4], a random process is said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram is used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e. states). In the figure, each arc represents the probability value for transition from one state to another.

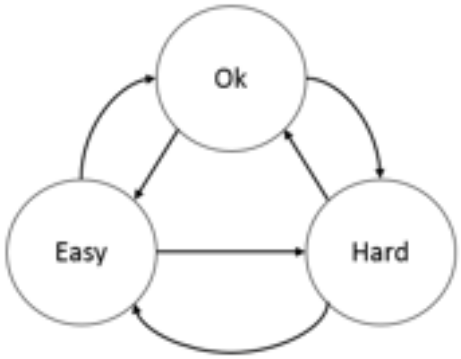


Figure 1. A Simple Markov chain with three states

The Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g. the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

Table 2. A state transition matrix of three states

State		Next Exam		
		Easy (E)	**Ok (O)**	**Hard (H)**
Previous Exam	**Easy (E)**	P(E|E)	P(O|E)	P(H|E)
	Ok (O)	P(E|O)	P(O|O)	P(H|O)
	Hard (H)	P(E|H)	P(O|H)	P(H|H)

In Table 1, the sum of the probability values at each row is 1 as the the sum of the probabilities coming out of each node should be 1. Hence, P(E|E)+P(O|E)+P(P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time, time-reversed, reversible, and irreducible Markov chains. The case shown in Figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.To illustrate a simple Markov chain data model, a small data set contains two English sentences used to create a transition matrix based on the neighbouring characters sequences. The sentences are inspirational English quotes picked from [5]:

(1) Power perceived is power achieved. (2) If you come to a fork in the road, take it.

Figure 2 shows the transition matrix of these quotes by counting the total number of occurrences of the adjacent two character sequences. It is a 19 × 19 matrix where the value 19 is the total number of unique characters appeared in the sentences (i.e the two quotes). In this example, creating transition matrix is case insensitive where D is same as d, as an example. In addition, a space between two words discarded and not considered in the transition matrix. Figure 2 also shows that the maximum number in the matrix's entries is 3 (a highlighted underlined value) which means that moving from character e to r (e➔r) is the most frequently sequence appeared in this small corpus. The words that contains this sequence are :{ Power (two times) and perceived}.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	2	0	0	0	1	0	0	0	0	0	**3**	0	0	0	1	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	2	0
p	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
r	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 2. A transition matrix of two characters sequences

Based on the information provided in the transition matrix shown in Figure 2. It is possible to answer some questions related to the given data collection. Among inquires, what is the total number of the two characters sequences appeared in the given data set? What are the two characters sequences that did not appear in the data collection? What is the least frequently two characters sequences in the data set? Accordingly, Markov chains are used as prediction systems such as weather forecasting. Therefore, it is possible to predict the tomorrow's weather according to the today's weather. For example, if we have two states (Sunny, Rainy), and the requirement is to find the probability P(Sunny|Rainy), Markov chains make it possible based on the information provided in the probability transition matrix. Another example of the using Markov chains is banking industry. A big portfolio of banks is based on loans. Therefore, Markov chains are used to classify loans to different states such as Good, Risky, and Bad loans.

For simplicity, the information presented in Figure 2 shows the transition matrix based on total number of occurrences. Figure 3 shows the same information but using probabilities instead of the number of occurrences. That is, it contains the probability of moving from one character to another. As previously indicated, the sum of entries at each row is equal 1. In Figure 3, any matrix entry that has 0 means that there is no transition at that case. Similarly, if the matrix entry is 1, it means

that there is only one possible output of that state. For example, the character “o” comes after “y”, and this is the only possible arc of the state “y”.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	0.33	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	0.33	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0.29	0	0	0	0.14	0	0	0	0	0	0.43	0	0	0	0.14	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0.17	0.17	0	0	0	0	0.17	0	0	0	0.17	0.17	0	0.17	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	0.17	0	0	0	0	0	0	0	0.17	0	0	0	0.17	0	0	0.17	0	0.33	0
p	0	0	0	0.33	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
r	0	0.33	0	0	0	0	0	0.33	0	0	0.33	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0.33	0	0	0	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 3. A probability transition matrix of two characters sequences

3. Hidden Markov Models

Hidden Markov models (HMM) is an extension to Markov chains models as both used for temporal data modeling. However, the difference is that the states in Markov chain models are directly observed while they are hidden in the case of HMM. We explain the concept of HMM based on Figure 1 that shows a three exam’s states Markov diagram. As a very simple example, supposed that a student’s parents want to know the levels (i.e the difficulty) of their son’s exams, naturally, it is possible to recognize the exam as Easy or Ok if the son feels Fine. Similarly, it is possible to recognize the exam as Hard if the son looks Scared. From the parents’ point of view, the required states (i.e. Easy, Ok, or Hard) are hidden. However, they directly observe the student’s reaction or feeling. Hence, the parents might use the observed reaction as an indication to know the hidden states. HMM is described using three matrices: the initial probability matrix, the observation probability matrix, and the state transition matrix. Figure 4 shows a HMM diagram that shows the

states and the observations. In the figure, each arc represents the probability between the states and between the states and the observations.

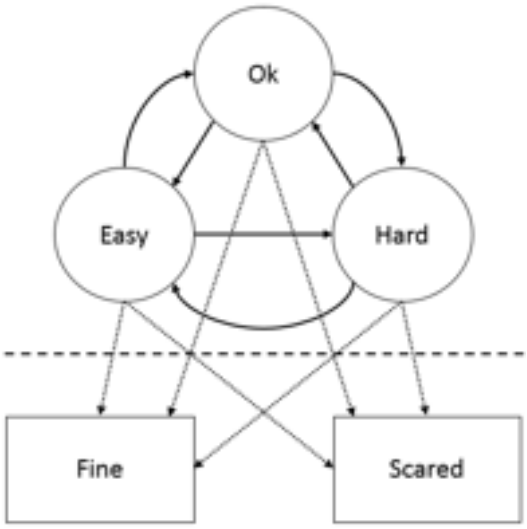


Figure 4. A HMM diagram with the transition and the observation arcs

Based on the information provided in the matrices, either Baum-Welch (also called any path) or Viterbi (also called best path) algorithms used to find the probability scores during recognition phase. Figure 5 shows the trellis diagram for exam states HMM. While Baum-Welch algorithm is used to compute the recognitin probability of a sequence, Viterbi is used to find the best-state sequence associated with the given observtatin, this procoss is also known as back-trakcing. Hence, after computing the observations sequence probability and finding the maximum probability (supposed the star in Figure 5), the Viterbi algorithm leads the process back to identify the states (sources) from which the observations sequence have been emitted. In Figure 5, the maximum probalities supposed to be achieved at the states shown using the dotted lines: Ok, Easy, Hard, respectively.

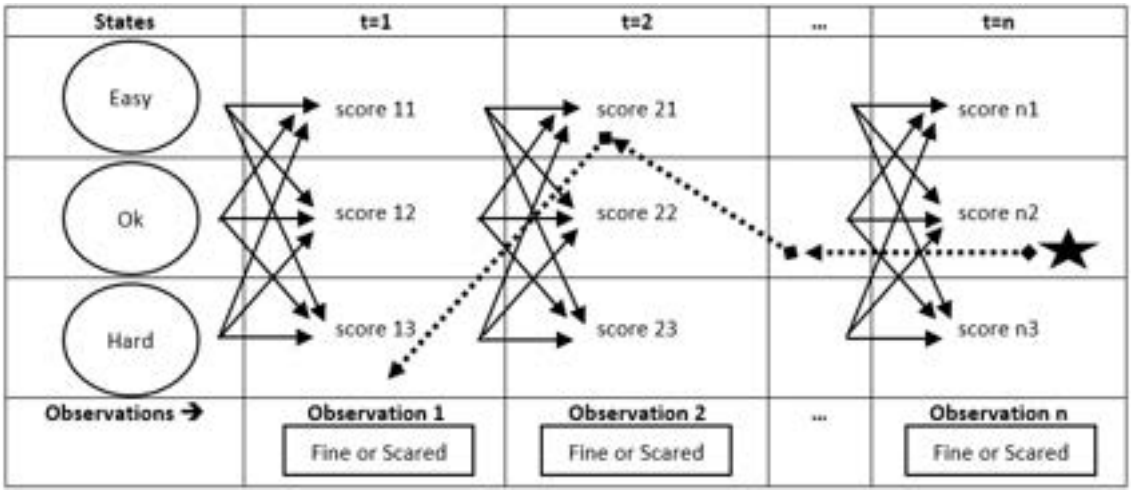


Figure 5. Trellis diagram of three states HMM

4. Linguistic Applications

In the literature, there are quite many works on modeling content dependencies for linguistics applications. Markov chain models and HMMs are of great interest to linguistic scholar who primarily work on data sequences. Even though this study focuses on linguistic applications, however, Markov chains used to model a variety of phenomena in different fields. The following are some of studies employed Markov chains. We intentionally ignored the references as the literature has too many studies employed Markov chains:

image processing, text and image compression, video segmentation ,forecasting, networking, signal processing, communications, software testing, genetics, bioinformatics, genome structure recognition, anomaly detection, tumour classification, water quality,

The following two subsections include some of the linguistic studies that utilized Markov chain theory. Linguistic applications topics mainly include (but not limited) speech recognition, speech emotion recognition, part-of-speech tagging, machine translation, text classification, text summarization, optical character recognition (OCR), named entity recognition, question answering, authorship attribution, etc. For the reader who interested in NLP, Reference [6] is a good reference as it demonstrates a thorough study of NLP (Almost) from Scratch.

4.1. Markov chains based research

The literature has a large number of studies that employ Markov chains for NLP applications. The following are some linguistic related applications. Reference [7] proposed a word-dividing algorithm based on statistical language models and Markov chain theory for Chinese speech processing. Reference [8] presented a semantic indexing Markov chains algorithm that uses both audio and visual information for event detection in soccer programs. Reference [9] investigated the use of Markov Chains and sequence kernels for the task of authorship attribution. Reference [10] implemented a probabilistic framework for support vector machine (SVM) that allows for automatic tuning of the penalty coefficient parameters and the kernel parameters via Markov chain for web searching via text categorization. Reference [11] demonstrated an automatic video annotation using multimodal Dirichlet process mixture model by collecting samples from the corresponding Markov chain. Reference [12] used a linguistic steganography detection method based on Markov chain models. Reference [13] showed how probabilistic Markov chain models can be used to detect topical structure in large text corpora.

Reference [14] proposed a method of recognizing location names from Chinese texts based on Max-Margin Markov Network. Reference [15] utilized Markov chain and statistical language models in a linguistic steganography detection algorithm. Reference [16] proposed a Markov chain based algorithm for Chinese word segmentation. Reference [17] presented two new textual feature selection methods based on Markov chains rank aggregation techniques. Reference [18] proposed a Markov chain model for radical descriptors in Arabic Text Mining. Reference [19] presented statistical Markov chain models for the distributions of words in text lines. Reference [20] proposed a method for handwritten Chinese/Japanese text (character string) recognition based on semi-Markov conditional random fields (semi-CRFs). Reference [21] presented a Markov chain method to find authorship attribution on relational data between function words. Reference [22] utilized a probabilistic Markov chain model to infer the location of Twitter users. Reference [23] proposed a Markov chain based technique to determine the number of clusters of a corpus of short-text documents. Reference [24] proposed a Markov chain based method for digital document authentication. Reference [25] used Markov chain for authorship attribution in Arabic poetry.

4.2. Hidden Markov models based research

Linguistic HMM based research has been for long an active research area due to the rapid development in NLP applications. The literature has many studies as follows. Reference [26] proposed to extract acronyms and their meaning from unstructured text as a stochastic process using HMM. Reference [27] proposed a morphological segmentation method with HMM method for Mongolian. Reference [28] employed HMM for Arabic handwritten word recognition based on HMM. Reference [29] presented a scheme for off-line recognition of large-set handwritten characters in the framework of the first-order HMMs. Reference [30] proposed the use of hybrid HMM/Artificial Neural Network (ANN) models for recognizing unconstrained offline handwritten texts. Reference [31] used HMMs for recognizing Farsi handwritten words.

Reference [32] describes recent advances in HMM based OCR for machine-printed Arabic documents. Reference [33] proposed a HMM based method for named entity recognition. Reference [34] combined text classification and HMM techniques for structuring randomized clinical trial abstracts. Reference [35] employed HMM for medical text classification. Reference [36] propose text (sequences of pages) categorization architecture based on HMM. Reference [37] described a model for machine translation based on first-order HMM. Reference [38] introduced speech emotion recognition by use of HMM. Reference [39] presented a HMM based method for speech emotion recognition. Reference [40] discussed the role of HMM in speech recognition. Reference [41] indicated that almost all present day large vocabulary continuous speech recognition (LVCSR) systems based on HMMs. Reference [42] presented a text summarization method based on HMM. Reference [43] presented a method for summarizing speech documents using HMM. Reference [44] used HMM for part-of-speech tagging task. Reference [45] presented a second-order approximation of HMM for part-of-speech tagging task.

5. Conclusions

This work demonstrates the potential and the size of Markov chains research. The study reveals that the Markov chain and HMM is of high important for linguistic applications. Similarly, Markov chains are also widely used in many other applications. For future work, it worthy to explore the power of Markov chain in new linguistic and scientific directions with more details.

References

[1] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[2] Markov_chain. (2016, August). Retrieved from https://en.wikipedia.org/wiki/Markov_chain

[3] Von Hilgers, Philipp, and Amy N. Langville. "The five greatest applications of Markov Chains." Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA. 2006.

[4] Leon-Garcia, Alberto, and Alberto. Leon-Garcia. Probability, statistics, and random processes for electrical engineering. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.

[5] California Indian Education. (2016, August). Retrieved from http://www.californiaindianeducation.org/inspire/world/

[6] Collobert, Ronan, et al. "Natural language processing (almost) from scratch."Journal of Machine Learning Research 12.Aug (2011): 2493-2537.

[7] Bin, Tian, et al. "A Chinese word dividing algorithm based on statistical language models." Signal Processing, 1996., 3rd International Conference on. Vol. 1. IEEE, 1996.

[8] Leonardi, Riccardo, Pierangelo Migliorati, and Maria Prandini. "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains." IEEE Transactions on Circuits and Systems for Video Technology 14.5 (2004): 634-643.

[9] Sanderson, Conrad, and Simon Guenter. "On authorship attribution via Markov chains and sequence kernels." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.

[10] Lim, Bresley Pin Cheong, et al. "Web search with text categorization using probabilistic framework of SVM." 2006 IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. IEEE, 2006.

[11] Velivelli, Atulya, and Thomas S. Huang. "Automatic video annotation using multimodal Dirichlet process mixture model." Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on. IEEE, 2008.

[12] Chen, Zhi-li, et al. "Effective linguistic steganography detection." Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on. IEEE, 2008.

[13] Dowman, Mike, et al. "A probabilistic model of meetings that combines words and discourse features." IEEE Transactions on Audio, Speech, and Language Processing 16.7 (2008): 1238-1248.

[14] Li, Lishuang, Zhuoye Ding, and Degen Huang. "Recognizing location names from Chinese texts based on max-margin markov network." Natural Language Processing and Knowledge Engineering, 2008. NLP-KE'08. International Conference on. IEEE, 2008.

[15] Meng, Peng, et al. "Linguistic steganography detection algorithm using statistical language model." Information Technology and Computer Science, 2009. ITCS 2009. International Conference on. Vol. 2. IEEE, 2009.

[16] Baomao, Pang, and Shi Haoshan. "Research on improved algorithm for Chinese word segmentation based on Markov chain." Information Assurance and Security, 2009. IAS'09. Fifth International Conference on. Vol. 1. IEEE, 2009.

[17] Wu, Ou, et al. "Rank aggregation based text feature selection." Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on. Vol. 1. IET, 2009.

[18] El Hassani, Ibtissam, Abdelaziz Kriouile, and Youssef BenGhabrit. "Measure of fuzzy presence of descriptors on Arabic Text Mining." 2012 Colloquium in Information Science and Technology. IEEE, 2012.

[19] Haji, Mehdi, et al. "Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms." Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on. IEEE, 2012.

[20] Zhou, Xiang-Dong, et al. "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields." IEEE transactions on pattern analysis and machine intelligence 35.10 (2013): 2413-2426.

[21] Segarra, Santiago, Mark Eisen, and Alejandro Ribeiro. "Authorship attribution using function words adjacency networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

[22] Rodrigues, Erica, et al. "Uncovering the location of Twitter users." Intelligent Systems (BRACIS), 2013 Brazilian Conference on. IEEE, 2013.

[23] Goyal, Anil, Mukesh K. Jadon, and Arun K. Pujari. "Spectral approach to find number of clusters of short-text documents." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on. IEEE, 2013.

[24] Shen, Jau Ji, and Ken Tzu Liu. "A Novel Approach by Applying Image Authentication Technique on a Digital Document." Computer, Consumer and Control (IS3C), 2014 International Symposium on. IEEE, 2014.

[25] Ahmed, Al-Falahi, et al. "Authorship attribution in Arabic poetry." 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA). IEEE, 2015.

[26] Osiek, Bruno Adam, Geraldo Xexéo, and Luis Alfredo Vidal de Carvalho. "A language-independent acronym extraction from biomedical texts with hidden Markov models." IEEE Transactions on Biomedical Engineering 57.11 (2010): 2677-2688.

[27] He, Miantao, Miao Li, and Lei Chen. "Mongolian Morphological Segmentation with Hidden Markov Model." Asian Language Processing (IALP), 2012 International Conference on. IEEE, 2012.

[28] Alma'adeed, Somaya, Colin Higgens, and Dave Elliman. "Recognition of off-line handwritten Arabic words using hidden Markov model approach." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 3. IEEE, 2002.

[29] Park, Hee-Seon, and Seong-Whan Lee. "Off-line recognition of large-set handwritten characters with multiple hidden Markov models." Pattern Recognition 29.2 (1996): 231-244.

[30] Espana-Boquera, Salvador, et al. "Improving offline handwritten text recognition with hybrid HMM/ANN models." IEEE transactions on pattern analysis and machine intelligence 33.4 (2011): 767-779.

[31] Imani, Zahra, et al. "offline Handwritten Farsi cursive text recognition using Hidden Markov Models." Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on. IEEE, 2013.

[32] Prasad, Rohit, et al. "Improvements in hidden Markov model based Arabic OCR." Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.

[33] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.

[34] Xu, Rong, et al. "Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts." AMIA. 2006.

[35] Yi, Kwan, and Jamshid Beheshti. "A hidden Markov model-based text classification of medical documents." Journal of Information Science (2008).

[36] Frasconi, Paolo, Giovanni Soda, and Alessandro Vullo. "Hidden markov models for text categorization in multi-page documents." Journal of Intelligent Information Systems 18.2-3 (2002): 195-217.

[37] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. "HMM-based word alignment in statistical translation." Proceedings of the 16th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, 1996.

[38] Schuller, Björn, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 2. IEEE, 2003.

[39] Nwe, Tin Lay, Say Wei Foo, and Liyanage C. De Silva. "Speech emotion recognition using hidden Markov models." Speech communication 41.4 (2003): 603-623.

[40] Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." Technometrics 33.3 (1991): 251-272.

[41] Gales, Mark, and Steve Young. "The application of hidden Markov models in speech recognition." Foundations and trends in signal processing 1.3 (2008): 195-304.

[42] Conroy, John M., and Dianne P. O'leary. "Text summarization via hidden markov models." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.

[43] Maskey, Sameer, and Julia Hirschberg. "Summarizing speech without text using hidden markov models." Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006.

[44] Kupiec, Julian. "Robust part-of-speech tagging using a hidden Markov model." Computer Speech & Language 6.3 (1992): 225-242.

[45] Thede, Scott M., and Mary P. Harper. "A second-order hidden Markov model for part-of-speech tagging." Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. Association for Computational Linguistics, 1999.

CHAPTER 10:
PERFORMANCE EVALUATION OF SPOKEN ARABIC LANGUAGE SPEECH RECOGNIZERS

1. Introduction

Arabic is the most widely spoken Semitic language today that recently has received significant attention for automatic speech recognition (ASR). ASR is a component of the natural language processing (NLP), which is used to automate the communication process between human and machine, i.e., the man-machine interaction. In this regard, much research has been devoted to introducing innovative technologies in dialogue systems for automation purposes (e.g., banking services, cars, control machines, etc.). However, employing ASR technology in Arabic NLP applications is still limited due to various challenges about within the Arabic language itself. For instance, it is difficult to obtain corpora for dialects that are spoken rather than written, i.e., there is no common writing standard, difficulty in obtaining a sizable diacritized text as Arabic allows writing without diacritics, and an enormous number of word forms due to the morphology richness of Arabic. In fact, one of the most difficult tasks in Arabic ASR is preparing a large diacritized text for ASR systems, which is a time-consuming preprocessing stage. In order to promote research on the Arabic ASR, we considered the corpora availability problem by producing a manually diacritized large-vocabulary speaker-independent continuous speech corpus for Modern Standard Arabic (MSA). The contents of the prepared corpus belong to general broadcast news.

In this work, we used the prepared corpus for an experimental evaluation of two off-the-shelf open source speech recognition toolkits, namely the Carnegie Mellon University (CMU) Sphinx [1] and the Hidden Markov Model Toolkit (HTK) [2]. In fact, it is important to find the performance of the popular speech engines using an identical speech collection because it reveals the unique characteristics of each engine for further understanding of their behavior in NLP and ASR applications. With the growing interest in ASR technology, it becomes more important to evaluate recent ASR systems in order to find the best-suited system for the tasks in question. For instance, the study in [3] demonstrated a large-scale evaluation of

open-source speech recognition toolkits that include Sphinx, HTK, and Kaldi [4]. The study in [3] indicated that Kaldi is better than Sphinx and HTK in terms of results and training recipes (for the German and English languages).

The work with the Arabic ASR considers one of two textual formats, either Arabic characters or Roman characters. Accordingly, there are two main approaches to address the training with Arabic text. The first approach considers using Arabic alphabet characters such as the Sphinx recognizer; the second approach uses Roman characters such as the HTK recognizer. It is sometimes required to convert Arabic characters into Roman characters, especially when recognizers assume that ASCII is used to write the training textual files rather than Unicode. HTK expects Roman characters, which means that Romanization of the Arabic ASR tasks must be completed. From the ASR point of view, Romanization means that the recognizer is trained on the Romanized transcriptions of the data. In general, choosing a recognizer on the basis of accepting Arabic characters should have no difference in performance since the core of the algorithms of training and decoding are the same in the different ASR engines. One purpose of this work is to evaluate the performance using the Sphinx and HTK recognizers. Therefore, we choose Buckwalter (BW) transliteration for the Romanization process in the HTK recognizer. There are many character options for Romanization; however, BW transliteration [5] is a good option for Romanization as it has the following two advantages: namely, it is popular, so the data can be easily made available to others as well as making it possible to use the data of other people.

In this study, we demonstrate an experimental evaluation of two cases; the Arabic character-based recognizer is the CMU Sphinx (PocketSphinx decoder), and the Roman character-based recognizer is the HTK version 3.4.1 (HVite decoder). However, the HTK toolkit has another famous decoder, which is ‘HDecode’, but in this work, we used HVite. As future work, we will aim at implementing HDecode for the Arabic ASR. The CMU Sphinx toolkit includes the latest available releases as follows: ‘Sphinxbase – 5Prealpha’, ‘PocketSphinx - 5prealpha’, and ‘SphinxTrain - 5prealpha’. For the experiments, we used a new “in-house” continuous speech corpus that contains 15.94 hours of MSA speech. The training set contains 12.74 hours (1,611 speech files), while the testing set contains 3.19 hours (403 speech files). This study also presents the intermediate steps to train the models as well

as the Romanization process. For fairness in the comparison, we emphasized that the training set and the testing set are identical in both systems (i.e., the same wave files, the same diacritized text, and the same phonemes set but with different symbols). Based on our best knowledge, this is the first attempt to experimentally compare the Sphinx and HTK recognizers for continuous Arabic speech.

This study is organized as follows: Section 2 presents the literature review. In Section 3, we present the phonemes and the pronunciation dictionary. The language models are explained in Section 4. The implementation and the results of the PocketSphinx and HTK recognizers are presented in Section 5, followed by the conclusion in Section 6.

2. Literature Review

Textual training data are an essential part of any speech corpus as they represent speech transcription. However, not all open-source recognizers can handle Arabic characters, which may lead to complications in ASR implementation. In fact, most of the current recognizers are designed for the English language, which is sometimes not compatible with the Arabic language due to the character representations. The literature shows that two types of textual data are used according to the employed recognizer. If the used recognizer supports Arabic characters, then it is straightforward to use the Arabic characters. However, if the recognizer cannot handle the Arabic characters, then the choice is to consider Romanization. In general, Arabic ASR researchers use either Sphinx or HTK for the recognition task. However, no single work presents a comparison between both systems, which is the motivation of this work. Based on the literature, most Arabic ASR studies employ the CMU sphinx engine because it is compatible with Arabic characters. However, few works have used the HTK engine based on Romanization. For instance, previous studies that used the CMU Sphinx are as follows: Hyassat and Abu Zitar in [6] employed Sphinx tools for the Holy Quran speech recognition, Alghamdi et al. in [7] used Sphinx tools for an MSA ASR system, AbuZeina et al. in [8] used the Sphinx tool for crossword Arabic pronunciation variation modeling for ASR, AbuZeina et al. in [9] used the Sphinx tool for within-word Arabic pronunciation variation modeling for ASR, Abushariah et al. in [10] used Sphinx tools for an Arabic ASR system based on a phonetically rich and balanced speech corpus, and Al-Anzi and AbuZeina

reported the most recent Arabic ASR work [11] that is based on Arabic characters. They used the CMU PocketSphinx to evaluate the impact of phonological rules on the Arabic ASR.

The literature shows that employing Roman characters for the Arabic ASR is less than what was observed for the Arabic characters. The following are results from previous studies. Kirchhoff et al. in [12] proposed the use of the Romanization method for the transcription of the speech corpus. Vergyri et al. in [13] used Romanized transcriptions to train the speech recognizer. Kirchhoff et al. in [14] used Romanized transcription in language modeling for large-vocabulary conversational Arabic speech recognition. Satori et al. in [15] introduced an Arabic voice recognition system where both the training and recognizing processes use the Romanized characters. Elmahdy et al. in [16] used Sphinx tools in addition to the SAMPA Romanization method for dialectal Arabic speech recognition. Al-Qatab et al. in [17] employed HTK for a small corpus of continuous speech and isolated words; however, they did not indicate anything regarding Romanization. The study in [18] used BW Arabic transliteration for Tunisian dialect dialogue systems. A recent work using HTK was reported by Merad-Boudia et al. in [19], and they employed the HTK engine for a small corpus of isolated and connected words. In summary, the literature shows that no single work has conducted practical research to evaluate the performance using an identical speech corpus, which indicates the importance and originality of this work.

3. Phonemes Set and Pronunciation Dictionaries

In order to train the models for ASR tasks, a speech corpus is required. This contains a set of speech files and corresponding textual transcription. The textual transcription is used to find the phonetic transcriptions through the phonemes set. Therefore, defining the phonemes set is the initial step in ASR work. A phoneme is the basic unit of sound that ASR uses for classification. While it is possible to use words and syllables for ASR classification, the phoneme approach is the most widely used in recent ASRs. On the other hand, the number of phonemes for Arabic is still a debated matter. For instance, Alghamdi et al. in [7] used 46 phonemes. Al-Qatab et al. in [17] used 34 phonemes (28 consonants and 6 vowels). Elmahdy et al. in [16] indicated that MSA consists of 38 phonemes, where 28 are original consonants, 4 are foreign and rare consonants, and 6 are vowels. Haraty et al. in [20] indicated that Arabic

has at least 112 phonemes, as they considered that every letter has four diacritics and therefore four phonemes. Alotaibi in [21] used 37 MSA phonemes as given by the Language Data Consortium (LDC).

Hence, the defined phonemes set in addition to the corpus transcription are used to generate the pronunciation dictionary that contains the phonetic transcription. In this work, the employed ASR recognizers have the following two options to handle the Arabic characters: the Sphinx recognizer allows the use of Arabic characters. Therefore, we do not need to consider Romanization. However, HTK expects Roman characters. That is, if the corpus textual transcription is stored using the standard Arabic characters set, then it needs to be transcribed into something that HTK can handle. For the Arabic ASR, BW transliteration has a distinguished attribute in which it uses one Roman character for each Arabic character, making it reasonable to approximate that each Arabic character corresponds to a single phoneme. For instance, the phonetic transcription of the word "kataba," which means “he wrote,” is "k a t a b a". However, BW uses a number of non-alphabetic characters to consider in the conversion process (e.g., BW uses “$” for the Arabic letter ‘Sheen’ “ش”, which is a special character in some recognizers such as HTK). Hence, the conversion of the Arabic text to BW transliteration requires a further step to swap the non-alphabetic characters with some other arbitrary characters. Of course, it is straightforward to convert the Romanized text back to the Arabic transcription in the case of the need to output in Arabic. Table 1 shows the Arabic and Romanized characters using BW transliteration. In the table, we indicate some replacement cases.

TABLE 1. Buckwalter (BW) transliteration with some replacements

#	Arabic character	Phoneme (BW)	#	Arabic character	Phoneme (BW)
1	ا	A	23	ك	k
2	أ	> (replaced to) O	24	ل	l
3	ب	b	25	م	m
4	ت	t	26	ن	n
5	ث	v	27	و	w
6	ج	j	28	ى	Y
7	ح	H	29	ه	h

8	خ	x	30	ي	y
9	د	d	31	آ	| (replaced with) U
10	ذ	* (replaced to) J	32	ء	G
11	ر	r	33	ؤ	& (replaced with) W
12	ز	z	34	ئ	} (replaced with) Q
13	س	s	35	ة	p
14	ش	$ (replaced to) C	36	ً	F
15	ص	S	37	ٌ	N
16	ض	D	38	ٍ	K
17	ط	T	39	َ	a
18	ظ	Z	40	ُ	u
19	ع	E	41	ِ	i
20	غ	g	42	ّ	~ (duplicate previous)
21	ف	f	43	ْ	o (not used)
22	ق	q	44	ٱ	> (replaced with) I

In fact, using either Roman or Arabic characters is a non-issue since there is no real difference between both types of transcription. It is just a matter of handling the textual characters. Therefore, the performance of the ASR recognizer based on Arabic characters should have no significant difference from one based on Roman characters. However, Romanization is harder to read for the native Arabic speaker. Of course, it is straightforward to do the transliteration in both directions in case of the need to output in the Arabic transcription.

In addition to the phonemes set shown in Table 1, we use three more phonemes to represent the Fatha that proceeds Alif (ا) ➔ ـَا as a single phoneme that is (aA), the Damma that proceeds Waw (و) ➔ ـُو as a single phoneme that is (uw), and the Kasra that proceeds Ya (ي) ➔ ـِي as (iy). The reason for handling these cases as a single phoneme is that the pronunciation of the short vowels is different when it proceeds the long vowels. Hence, it will be correctly transcribed as single phonemes. Therefore, we used 46 phonemes for the HTK system, 42 phonemes in Table 1 (Shadda (ّ<>~) and Sukun (ْ<>o) are discarded) in

addition to the three phonemes (aA, iy, and uw) and the sil phonemes to indicate silence. In the Sphinx system, decoding fails when the phonemes set contains small letter and capital letter symbols. For instance, it fails when the phonemes set has a phoneme b and another phoneme B. However, this problem has not been observed in the HTK system. Hence, we used a new phonemes set for Sphinx that contains 45 phonemes (ّ is not counted since it is duplicated) as shown in Table 2. The phonemes set contains 46 phonemes as the set that is used for the HTK system. Hence, both systems have the same phonemes set.

TABLE 2. Proposed phonemes set for the Sphinx recognizer

#	Letter	Phoneme	#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	17	ر	R	33	ه	H
2	آ	AA	18	ز	Z	34	و	W
3	أ	O	19	س	S	35	ى	AY
4	ؤ	EW	20	ش	SH	36	ي	Y
5	إ	I	21	ص	SS	37	ً	AU
6	ئ	EY	22	ض	DD	38	ّ	AWW
7	ا	A	23	ط	TT	39	ٍ	AIY
8	ب	B	24	ظ	ZZ	40	َ	AU
9	ة	P	25	ع	AE	41	ُ	AW
10	ت	T	26	غ	GH	42	ِ	AI
11	ث	TH	27	ف	F	43	ّ	duplicate
12	ج	J	28	ق	Q	44	ـا	AUA
13	ح	HH	29	ك	K	45	ـُو	AWW
14	خ	KH	30	ل	L	46	ـِي	AIY
15	د	D	31	م	M			
16	ذ	DH	32	ن	N			

The pronunciation dictionary is an ASR component that contains the phonetic transcription of each word. That is, each word that appears in the training text is listed in the dictionary in terms of phonemes. Nevertheless, generating a pronunciation dictionary is a hard task due to the different acoustic cases of pronunciations. In the case of a large number of words (i.e., a

sizable vocabulary), it is quite time-consuming. In this work, we used a Python program to generate the pronunciation dictionaries for both the Sphinx and HTK recognizers. In the literature, there are various studies that describe the rules to generate a pronunciation dictionary for MSA. For instance, Ramsay et al. [22] described a knowledge-based approach to generate the phonetic transcription for MSA. Ali et al. in [23] presented a tool for generating phonetic dictionaries for MSA.

Figure 1 shows entries of the dictionaries that are used in this work. The figure shows the phonemes sequence of each word that appears in the training textual data. The pronunciation of each word is used to model the acoustic model during the training phase. For instance, the word "الوَزَارَة" is mapped to the sequence "A L W AU Z AUA R AU P". Hence, wherever this word appears, it is replaced with the phonemes sequence in order to produce the phonetic transcription of the speech file. Each phoneme is then used to train a part of the acoustic signal that corresponds to a phoneme name. This process (i.e., the training stage) is performed using the Baum-Welch algorithm to create a single hidden Markov Model (HMM) for each phoneme in the phoneme list. The figure also shows two sets of phonemes: the phonemes set for Sphinx (on the left of the figure), which is proposed in this work, and the HTK set (on the right of the figure), which is based on the BW transliteration. Creating a single HMM model for each phoneme is called the 'context-independent' phase (CI). After the CI stage, the training stage continues to perform the untied 'context-dependent' phase (CD) for creating triphones; finally, the tied context-dependent phase is used for tying some HMM states.

Arabic-based dictionary (for Sphinx)	*Roman character-based dictionary (for HTK)*
…	…
الوَزَارَة A L W AU Z AUA R AU P	AlwazaArap A l w a z aA r a p
الوَزِير A L W AU Z AIY R	Alwaziyr A l w a z iy r
الوَزيرَ A L W AU Z AIY R AU	Alwaziyra A l w a z iy r a
الوَزِيرَة A L W AU Z AIY R AU P	Alwaziyrap A l w a z iy r a p
الوَزِيرُ A L W AU Z AIY R AW	Alwaziyru A l w a z iy r u
الوَزِيرِ A L W AU Z AIY R AI	Alwaziyri A l w a z iy r i
الوَزِيرِة A L W AU Z AIY R AI P	Alwaziyrip A l w a z iy r i p

…	…

FIGURE 1. Entries of the pronunciation dictionaries

In both implemented systems, we consider the pronunciation case of Shadda as follows: for the Shadda case, we first remove the germination marker "ّ" and then duplicate the previous consonant. Another case where an unnecessary symbol is deleted is the 'Sukon' "ْ". Ramsay et al. in [22] described pre-processing cases that should be considered for phonetic transcription. For illustration, Figure 1 shows some cases where the semi-vowels y, w and A are written with a preceding short vowel, so it is assumed that iy, uw and aA are two-character names for the relevant vowels. The same is found in the Sphinx dictionary (i.e., AUA, AWW, and AIY).

4. Language Models

(LMs) are considered to be a significant contribution to the performance of NLP systems such as ASR and machine translation. LMs have been successfully applied in different linguistic applications such as Part-of-Speech (PoS) tagging, parsing, information retrieval, spell correction, summarization, etc. In particular, LM is a critical component in linguistic applications producing sequences of words as output. In the last few decades, extensive research has been devoted to promoting new techniques to compile LMs as well as to address challenges such as missing n-grams. In speech recognition, the ASR decoder uses the information provided by the LM to find the best possible word sequence of the testing speech for transcription purposes. In general, it is extremely important for language applications to have the ability to predict the next word given by the previous word(s), or the history.

LMs can be either probabilistic or non-probabilistic. The probabilistic LMs are known as statistical LMs such as n-grams, while non-probabilistic LMs are known as 'any-word' grammar. Any-word grammar does not use probabilities of words. It is unconstrained

grammar that leads to very poor accuracy in continuous ASR systems. That is, any-word grammar relies entirely on the acoustic model. On the other hand, statistical language models are based on computing the probabilities of all word combinations (i.e., all possible word sequences) in the training source text. Statistical LMs are generally demonstrated using ARPA format textual files that include the statistical estimation of the desired n-grams, typically up to 3-grams. Compiling a statistical language model requires a large number of words from different textual resources. In addition, the data should not be too specific to a particular domain; otherwise, it will not generalize well to the sentences in question.

During ASR decoding, the recognizer employs the language model to transcribe speech files using the acoustic model and the pronunciation dictionary (i.e., the vocabulary). Hence, the most likely hypothesis for each testing speech file is generated as an output. Employing language models reinforces the speech recognition accuracy, as the more you can constrain the range of possible utterances, the more accurate the recognizer will be. Based on the information provided in the n-grams, the probability of a word sequence is computed using the following formula [25]:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_1^{k-1}) \qquad (1)$$

where 'n' is limited to include the history of the word. Hence, the Chain Rule is applied to compute the joint probability of words in a sentence. LMs utilize the Markov assumption to simplify data estimation. For instance, for n=2, the bigram is calculated for the word sequence as follows:

$$P(w_1\, w_2) = P(w_2\,|w_1)P(w_1\,) \quad (2)$$

For speech features, Mel Frequency Cepstral Coefficients (MFCC) are the classical front-end analyses used in speech recognition to produce the sequence of real-valued numbers that represent feature vectors based on the input signal. Since 1980, it has dominated the ASR feature extraction method due to its good performance. The success of MFCC makes it the standard choice in the state-of-the-art speech recognizers such as the CMU Sphinx, HTK,

and Kaldi speech recognizers. Reference [24] has some details of MFCC. Given the speech feature vectors, the most likely sequence of words is estimated by the following [25]:

$$\widehat{w} = \underset{W \in L}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} = \underset{W \in L}{\operatorname{argmax}} P(O|W)P(W) \qquad (3)$$

Where $\widehat{w}$ is the most likely recognized words, $P(O|W)$ is the probability of the feature vectors, given a sequence of words that is computed using the acoustic model, and $P(W)$ is the probability of the words sequence that is computed using the language model. $P(O)$ is the probability of the acoustic observation sequence and can be ignored. Hence, the statistical LM has to be computed first to decode the testing speech files in ASR systems. The statistical n-grams LM is trained by counting n-grams occurrences in a large transcription corpus to then be smoothed and normalized. Counting and normalizing can train the n-gram models. The following formula is used to estimate the n-grams parameter [25]:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{Count(w_{n-N+1}^{n-1} w_n)}{Count(w_{n-N+1}^{n-1})} \quad (4)$$

One major problem in LMs is unseen words or n-grams that are found in the testing set while out of vocabulary. Accordingly, a probability of 0.0 is given to the items that are not seen in the training data. That is, not all n-grams will be present (i.e., not observed) in the training data. One solution is smoothing by assigning non-zero or small probabilities to the unseen n-grams in which all word sequences can occur with some probability. Hence, smoothing provides a better way of estimating the probability of zero frequency n-grams that never occur in order to produce generalized LMs. Smoothing is also called discounting. When creating a language model, it is more efficient to use log probabilities rather than actual probabilities due to the risk of numerical underflow, especially for very long strings. It is also efficient in ASR decoding algorithms such as the Viterbi algorithm.

Creating an n-gram LM entails the following three main steps: compute the word unigram counts, convert the word unigram counts into a vocabulary list, and generate bigram and trigram (or more) tables based on this vocabulary. As a preprocessing step, it is essential to include special words such as <s> in order to indicate the “start of sentence” and the </s> to

indicate the “end of the sentence”. The CMU Cambridge University toolkits [26] use the <UNK> token to indicate unknown words, whereas HTK toolkits [2] use !!UNK for the same purpose.

Sometimes ASR recognizers employ the grammar of LMs for small applications with limited isolated words. Such grammar is simple and does not have probabilities; they are designed according to the information that is provided in the corresponding application. That is, grammar mainly contains isolated words such as commands, control words, and digits. However, grammar might allow sequences of words. Figure 2 shows a simple grammar for ten digits that can be used in continuous speech recognition to choose one word or more from the list. The grammar is written using JSGF format, as shown in Figure 2. The star (*) in Figure 2 means zero or more words. This is helpful when this grammar is used in continuous speech in order to recognize a sentence that has many words.

Ten digit grammar (for Sphinx)
#JSGF V1.0; grammar myGrammar; public <command> = <word>* ; <word> =(One | Two | Three | … | Ten);
Ten digit grammar (for HTK)
$WORD = (One \| Two \| Three \| … \| Ten); ($WORD)

FIGURE 2. Any-word grammar for ten digits

Grammar is usually written by hand, or it can be generated using a program. Most grammar does not use probabilities; however, some elements might be weighted. In fact, grammar is rarely used in ASR systems since the probabilistic models of a language are more useful than the hard models (i.e., grammar) for legal sentences in various languages.

There are a number of toolkits that are used to compile LMs such as the CMU-Cambridge LM toolkit [26], the Cambridge University HTK language modeling tools [2], and the SRI Language Modeling Toolkit (SRILM) [27]. The CMU-Cambridge toolkit uses the following five commands to produce the LM dump file: text2wfreq, wfreq2vocab, text2idngram,

idngram2lm, and lm3g2dmp [28]. HTK provides two approaches to generate n-grams. The first method employs the HLStats function, which is used exclusively to compute the bigram language model (i.e., 2-grams). The second method implements a series of functions to compute n-grams. In this work, we used the first approach (i.e., bigram statistical LM). In addition to HLStats, HBuild is used to create the word network that describes the allowable word sequences, of course, with the corresponding probabilities. Figure 3 shows a simple 3-gram LM for three sentences as shown in the figure. It was created using the CMU-Cambridge toolkit.

A small corpus of three sentences

<s> ووقع الجانبان مذكرات تفاهم في الاقتصاد والتربية والتعليم </s>
<s> ضخ المزيد من الاموال من اجل نمو الاقتصاد المحلي </s>
<s> نمو الاقتصاد المحلي الذي يعتبر المحرك الاقتصادي للدولة </s>

```
\data\
ngram 1=23
ngram 2=26
ngram 3=28
```

```
\1-grams:
-1.5168 <UNK>    0.0000
-1.1614 </s>     -0.4297
-0.9853 <s>      0.0604
-1.5168 اجل      0.0310
-0.9853 الاقتصاد -0.4317
...
-1.5168 ووقع     0.0134
-1.5168 يعتبر    0.0134
```

```
\2-grams:
-0.1761 </s> <s> 0.0000
-99.9990 <s> 0.0000 ضخ
...
-99.9990 والتعليم </s> 0.4771
-99.9990 0.0000 ووقع الجانبان
-99.9990 0.0000 يعتبر المحرك
```

```
\3-grams:
-99.9990 </s> <s> ضخ
-99.9990 </s> <s> نمو
-99.9990 <s> ضخ المزيد
...
-0.1761 نمو الاقتصاد المحلي
-99.9990 والتربية والتعليم </s>
-99.9990 والتعليم </s> <s>
-99.9990 ووقع الجانبان مذكرات
-99.9990 يعتبر المحرك الاقتصادي
\end\
```

FIGURE 3. Three sentences with the 3-grams LM using the CMU-Cambridge tool

For this work, we used 2-grams statistical language models as shown in Figure 4 for both systems. In fact, the 3-grams language model is more commonly used for ASR tasks; however, we used the 2-grams language model in this work due to the restriction of HVite. HVite uses 2-grams, while HTK HDecode can use 3-grams. PocketSphinx can use either 2-grams or 3-grams language models.

CMU-Cambridge tool	*HTK tool*

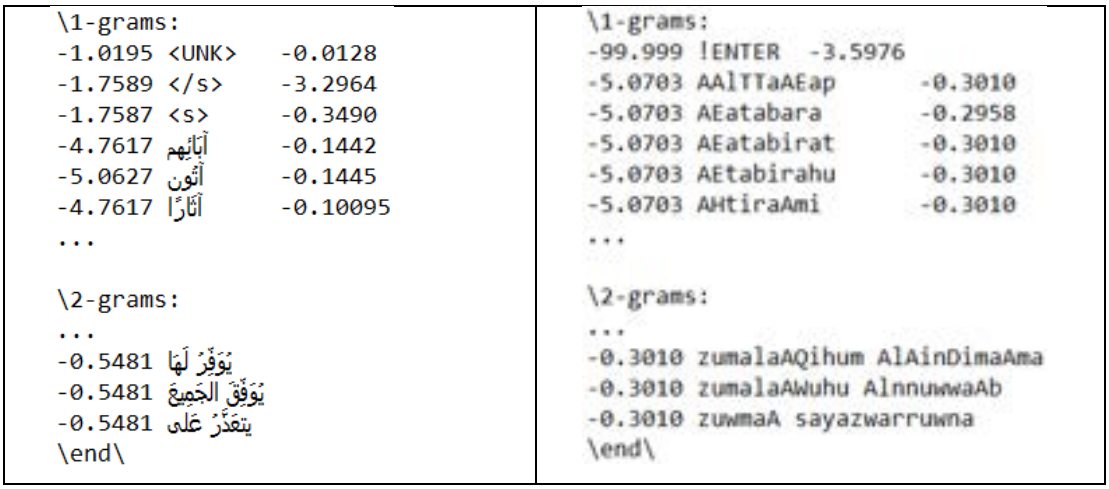


FIGURE 4. Parts of the 2-grams LM that are used in this work

5. Implementation of the Sphinx and HTK Methods

The Sphinx and HTK methods used Cygwin, which is a Unix-like environment for Windows. However, it is preferred to run the command line in a UNIX-based system rather than Cygwin that is installed for the Windows environment. Implementing Sphinx for Arabic ASRs includes the steps described in [29]. The first step includes creating the directory where the files live. The files include the training and testing speech files, the transcription of the entire speech collection, and other necessary files that are used for training and decoding. In particular, the speech files are stored in the wav directory, while the ***etc*** directory has the following files: the pronunciation dictionary, the phonemes file, the list of fillers, the list of files for training, the transcription for training, the list of files for testing, and the transcription for testing. Of course, the language model also lives in the ***etc*** directory. The following are the main commands (for task1, for instance) that are used in Sphinx:

- $ mkdir task1 (create the main directory)
- $ sphinxtrain -t task1 setup (create the structure of the main directory)
- $ sphinxtrain run (start training, once done, it provides the word error rate (WER))

For HTK implementation, reference [30] presents comprehensive details for training and decoding as the following steps:

Step 1 - the Task Grammar, Step 2 - the Dictionary, Step 3 - Recording the Data, Step 4 - Creating the Transcription Files, Step 5 - Coding the Data, Step 6 - Creating Flat Start Monophones, Step 7 - Fixing the Silence Models, Step 8 - Realigning the Training Data, Step 9 - Making Triphones from Monophones, Step 10 - Making Tied-State Triphones, and finally, Step 11 - Recognizing the Test Data. In the previous sections, we demonstrated some of the steps such as the language model and pronunciation dictionary. To train a model, a further set of files is needed such as the following:

- words.mlf: this is just a rearranged version of the training textual files,
- train.scp: a list of the training speech file names,
- phones0.mlf: the phonetic transcriptions, obtained by substituting the entries in the pronunciation dictionary for the words in the textual transcriptions,
- monophones0: list of the phones that appear in phones0.mlf,
- codetrain.scp: pairs linking .wav files to .mfc files (the MFCC speech features),
- proto.txt: the "flat start" file for the hidden Markov models (HMMs),
- config.txt that contains some parameters related to the speech features.

After creating the necessary files in the same directory where the speech files and the speech features (.mfc) reside, it is possible to start training. This produces trained models that will be used for decoding. During training, many functions are executed, such as the HLEd and HERest functions. For decoding, HVite is used.

To investigate the performance, we split the speech corpus (15.94 hours) into two parts including the training set that contains 12.74 hours (1,611 speech files) and the testing set that contains 3.19 hours (403 speech files). That is, the testing set is 20% of the overall speech corpus. The speech files were prepared to have a fixed length between 10-40 seconds, mono, and sampled at 16 kHz. The average length of the textual files is 55 words. The total number of speakers in the corpus is 29 native Arabic speakers (19 males and 10 females). In this work, we used three emitting states of HMMs that corresponded to the subphones at the beginning, middle, and end of the phones. The acoustic models were calculated using context-dependent HMM triphones. Regarding Sphinx recognizers, the acoustic models are trained using the SphinxTrain for the phonetic tied-mixture (PTM). However, other acoustic model types can be used such as semi-continuous and fully continuous models. The

performance is measured based on different parameters such as the number of Senones and the number of Gaussian densities. Table 3 shows the WER for different cases.

TABLE 3. The performance of the Sphinx recognizer

Experiment	*Densities*	*Senones*	*WER (%)*	*Accuracy (%)*
1	*8*	*500*	*22.6*	*77.4*
2	*8*	*1000*	*22.2*	*77.8*
3	*8*	*2000*	*21.5*	*75.5*
4	*16*	*500*	*21.8*	*78.2*
5	*16*	*1000*	*21.1*	*78.9*
6	***16***	***2000***	***20.7***	***79.3***
7	*32*	*500*	*21.8*	*78.2*
8	*32*	*1000*	*21.3*	*78.7*
9	*32*	*2000*	*21.3*	*78.7*
10	*64*	*500*	*21.9*	*78.1*
11	*64*	*1000*	*21.9*	*78.1*
12	*64*	*2000*	*21.9*	*78.1*
13	*128*	*500*	*21.7*	*78.3*
14	*128*	*1000*	*22.6*	*77.4*
15	*128*	*2000*	*22.1*	*77.9*
16	*256*	*500*	*21.8*	*78.2*
17	*256*	*1000*	*22.2*	*77.8*

Table 3 shows that the best (lowest) WER was found to be 20.7% using 16 Gaussian densities and 2000 senones. We emphasize that these results are based on the language model that contains both the training and testing transcriptions. However, if the langue model contains only the training transcription, the results will be less than what we scored in Table 3. The reason for using the training and the testing sets to create the language model is that the language model requires a large amount of data that is not available in our work. During experiments, we considered speeding up the execution time using the PocketSphinx configuration (i.e., number of parts to run Forward-Backward estimation ➔ $CFG_NPART = 10; and how many pieces to split decode in➔ $DEC_CFG_NPART = 10;). The number 10 is just an optional number that can be fixed based on the user preferences. This option is helpful since it clearly reduces the execution time by utilizing a number of processors in multicore machines. For the HTK recognizer, we found the performance to be less than what

we achieved using the Sphinx recognizer. The WER is (100%-65.31%=34.69%). However, the lowest WER in the Sphinx is 20.7%. Figure 5 shows the HTK overall results.

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 18 19:01:41 2017
  Ref : words.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=403, N=403]
WORD: %Corr=67.50, Acc=65.31 [H=15582, D=1044, S=6459, I=506, N=23085]
====================================================================
```

FIGURE 5. Performance of the HTK recognizer

Regarding the output, Sphinx gives the results using Arabic characters. However, the HTK gives Romanized characters. Figure 6 shows an example of the output of the HTK recognizer. In the figure, Arabic characters (the upper part in Figure 6) represent the transcription of a speech file after recognition, and the lower English characters are the Romanized text as recognized by the HTK HVite decoder.

مَازَالَ رُدُودُ الأَفْعَالِ الدَّوْلِيَّةِ تَتَوَالَى عَلَى تَطَوُّرَاتِ الأَحْدَاثِ فِي لِيبْيَا فَقَدْ قَامَتْ فَرَنْسَا بِطَرْدِ أَرْبَعَةَ عَشَرَ دِبْلُومَاسِيًّا لِيبْيًّا لِمُنَاصَرَتِهِمْ نِظَامَ الْعَقِيدِ مُعَمَّرِ الْقَذَّافِي
الَّذِي لَمْ تَقْدِ بَارِيسُ تَشْرِيفًا شَرْعِيًّا بَعْدَ إِعْتِرَافِهَا بِالْمَجْلِسِ الْوَطَنِيِّ الإِنْتِقَالِيِّ وَمِنْ جِهَتِهَا صَنَّفَتِ الوِلاَيَاتُ الْمُتَّحِدَةُ الأَمْرِيكِيَّةُ ضُغُوطَهَا عَلَى نِظَامِ الْعَقِيدِ مُعَمَّرِ
الْقَذَّافِي وَخَفَّضَتْ ثَلاَثَ شِرْكَاتٍ مَمْلُوكَةٍ لِنِظَامِ وَأَكَّدَتْ وَزِيرَةُ الخَارِجِيَّةِ الأَمْرِيكِيَّةِ هِيلاَرِي كِلِينْتُونْ أَنَّ الإِدَارَةَ الأَمْرِيكِيَّةَ قَرَّرَتْ إِصْدَارَ قَانُونٍ يَسْمَحُ
بِاسْتِغْلاَلِ جُزْءٍ مِنَ الأَمْوَالِ الخَاصَّةِ بِالْقَذَّافِي وَنِظَامِهِ فِي أَمْرِيكَا لِمُسَاعَدَةِ الشَّعْبِ اللِّيبِيِّ (1323)

```
File: mfc/1323.mfc
!ENTER maAtazaAlu ruduwdu OafEaAli kuwiyataA waAlEaql quwwapi AlOaHdaAvi fiy liybyaA
faqad qaAmat faransaA biTardi OarbaEapa EaCra dibluwmaAsiyyaA liybiyFA limunaASartihim
niZaAm AlEaqiyd muEammar AlqaJAfiyy taHta AllaJiy lam taEud baAriys taEtabiruhu
CarEiyyFA baEda tawfiyra biAlmajlis AlwaTaniyyi AintiqaAliyyap wa min jihap yaSEubu
Daruwrapi AlmuttaHidapu AlOamriykiyyap DuguwTahaA EalaY niZaAm AlEaqiyd muEammar
AlqaJJaAfiy wajammadat valaAva CirkaAtK mamluwkapK linniZaAm waOakkadat waziyrapu
AlxaArij AlOamriykiyyap hiylaAriy kliyntuwn Oanna AlIidaArapa AlOamriykiyyap qarrarat
IiSdaAra qaAnuwnK yasmaHu tasliyTu sinna AlOawwal AlxaAS biAlqaJJaAfiy waniZaAmihi fiy
OamriykaA limusaAEadapi CaEbi Allliybiyyap !EXIT  == [3545 frames] -66.0275
[Ac=-232137.1 LM=-1930.4] (Act=300763.6)
```

FIGURE 6. The recognition output of a speech file using the HTK

In this work, we used the default settings of HTK. The default settings include the thresholds for outlier removal (RO=100) and the tree branch threshold (TB=350). TB is used for the decision of tree clustering of states. Both RO and TB affect the degree of tying and therefore the number of states output in the clustered system [30]. In future work, it is worth evaluating the performance using different values of RO and TB. It is also worth employing the HDecode, which is an HTK extension decoder released on a restricted basis.

6. Conclusion.

This work discusses the implementation of two well-known speech recognizers; the CMU Sphinx, and the HTK. It includes a comparative study of both recognizers using a continuous speech corpus of MSA. The results show that Sphinx outperforms the HTK recognizer. Sphinx is also better in some issues such as handling long speech files, since some of the long speech files were discarded due to failure execution using the HTK (i.e., the training fails using long speech files). Sphinx is also better in terms of execution time as it takes less training and decoding time compared to the HTK. Finally, we have found that it is easier to perform an ASR task with Sphinx than HTK. The only issue with Sphinx is that it fails when the phonemes set has capital and small letters. For instance, if we use the character <B> to indicate a specific phoneme and, at the same time, use the character <b> to indicate another phoneme, then we get an error during training. On the other hand, this error did not appear in the HTK system. We also have found that HTK is better documented than Sphinx. In conclusion, more research is required to understand the reasons for the performance difference between both systems.

REFERENCES

[1] Available: https://cmusphinx.github.io/

[2] Available: http://htk.eng.cam.ac.uk/

[3] Gaida, Christian, et al. "Comparing open-source speech recognition toolkits." Tech. Rep., DHBW Stuttgart (2014).

[4] Available: http://kaldi-asr.org/doc/index.html

[5] Available: http://www.qamus.org/transliteration.htm

[6] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal

of Speech Technology 9.3-4 (2006): 133-150.

[7] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[8] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[9] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[10] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[11] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology(2017): 1-9.

[12] Kirchhoff, Katrin, et al. "Novel approaches to Arabic speech recognition: report from the 2002 Johns-Hopkins summer workshop." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 1. IEEE, 2003.

[13] Vergyri, Dimitra, et al. "Morphology-based language modeling for arabic speech recognition." INTERSPEECH. Vol. 4. 2004.

[14] Kirchhoff, Katrin, et al. "Morphology-based language modeling for conversational Arabic speech recognition." Computer Speech & Language 20.4 (2006): 589-608.

[15] Satori, Hassan, Mostafa Harti, and Nouredine Chenfour. "Introduction to Arabic speech recognition using CMUSphinx system." arXiv preprint arXiv:0704.2083 (2007).

[16] Elmahdy, Mohamed, et al. "Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition." Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on. IEEE, 2009.

[17] Al-Qatab, Bassam AQ, and Raja N. Ainon. "Arabic speech recognition using hidden Markov model toolkit (HTK)." Information Technology (ITSim), 2010 International Symposium in. Vol. 2. IEEE, 2010.

[18] Graja, Marwa, Maher Jaoua, and L. Hadrich Belguith. "Lexical study of a spoken dialogue corpus in tunisian dialect." The international arab conference on information technology (acit), benghazi–libya. 2010.

[19] Merad-Boudia, N., Benyettou, A., Rubio Ayuso, A., Arabic Speech Recognition for Connected Words Using HTK: Triphones Expanded to Gmm Based Quran Recognition, (2016) *International Review on Computers and Software (IRECOS)*, 11 (12), pp. 1209-1216.

[20] Haraty, Ramzi A., and Omar El Ariss. "CASRA+: A colloquial Arabic speech recognition application." American Journal of Applied Sciences 4.1 (2007): 23-32.

[21] Alotaibi, Yousef Ajami. "Comparative study of ANN and HMM to Arabic digits recognition systems." Journal of King Abdulaziz University: Engineering Sciences 19.1 (2008): 43-59.

[22] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern

standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[23] Ali, Mohamed, et al. "Arabic phonetic dictionaries for speech recognition." Journal of Information Technology Research (JITR) 2.4 (2009): 67-80.

[24] Al-Anzi, Fawaz S., and Dia AbuZeina. "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", World Academy of Science Engineering and Technology, International Journal of Computer and Information Engineering, Vol:11, No:10, 2017

[25] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. Pearson, 2014.

[26] Available: http://www.speech.cs.cmu.edu/SLM/toolkit.html

[27] Available: http://www.speech.sri.com/projects/srilm/

[28] Available: http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

[29] Available: https://cmusphinx.github.io/wiki/tutorialam/

[30] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

Appendix:
Journal & Conference Papers

Conference Papers:

1. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina ,"Arabic Speech Recognition: A Survey of the Literature", The 10th International Conference on Informatics and Systems Cairo, Egypt, May 9-11, 2016.
2. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Utilizing Long Distance Word Dependencies for Automatic Speech Recognition", The International Conference on Innovations in Information Technology (IIT'16) , United Arab Emirates University,28 - 30 November 2016.
3. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "An Empirical Study of Arabic Continuous Speech Recognition Performance", The International Conference on Computer Applications & Technology, ICCAT' 2017, from 28-29 January, in Cairo, Egypt.
4. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Effect of diacritization on Arabic Speech Recognition" The 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT). Oct 11 - 13, 2017.
5. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Phonetic Tied-Mixture PTM Acoustic Model for Arabic Continuous Speech Recognition", The 18th International Arab Conference on Information Technology (ACIT'2017), Yassmine Hammamet, Tunisia.
6. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", ICNMLKD 2017 : 19th International Conference on Network, Machine Learning and Knowledge Discovery, Bangkok, Thailand, October 26 - 27, 2017.
7. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Exploring the Language Modeling Toolkits for Arabic Text", The International Conference on Electrical and Computing Technologies and Applications, 2017 (ICECTA'2017), November, 21-23, 2017, AURAK, UAE.

8. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "A Survey of Markov Chain Models In Linguistics Applications", Fifth International Conference on Data Mining & Knowledge Management Process (CDKP 2016) , November 12-13, 2016, Dubai, UAE
9. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina ," A Literature Survey of Arabic Speech Recognition", Second International Conference on Computing Sciences and Engineering (ICCSE 2018), March 11th to 13th 2018 - Kuwait University, Kuwait.

Journal Papers:

6. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Statistical Markovian Data Modeling for Natural Language Processing", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.7, No.1, January 2017. (Extended Version of Conference Paper 8).
   http://aircconline.com/ijdkp/V7N1/7117ijdkp03.pdf
7. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Impact of Phonological Rules on Arabic Speech Recognition", International Journal of Speech Technology.
   https://link.springer.com/article/10.1007/s10772-017-9440-2
8. [**Accepted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:11, No:10, 2017. (Extended Version of Conference Paper 6).
   https://waset.org/journal/Computer
9. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina, "Theoretical and practical models for Arabic speech recognition", Submitted to International Journal of Applied Mathematics and Computer Science.
10. [**Submitted**] Al-Anzi, Fawaz S., and Dia AbuZeina, " Performance Evaluation of Sphinx and HTK Speech Recognizers for Spoken Arabic Language", Submitted to Engineering Applications of Artificial Intelligence.

Published Paper(s)

STATISTICAL MARKOVIAN DATA MODELING FOR NATURAL LANGUAGE PROCESSING

Fawaz S. Al-Anzi and DiaAbuZeina

Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait

ABSTRACT

Markov chain theory is a popular statistical tool in applied probability that is quite useful in modelling real-world computing applications. Over the past years; there has been grown interest to employ Markov chain theory in statistical learning of temporal (i.e. time series) data. A wide range of applications found to utilize Markov concepts; such applications include computational linguists, image processing, communications, bioinformatics, finance systems, etc .In fact, Markov processes based research applied with great success in many of the most efficient natural language processing (NLP) tools. Hence, this paper explores the Markov chain theory and its extension hidden Markov models (HMM) in (NLP) applications. This paper also presents some aspects related to Markov chains and HMM such as creating transition and observation matrices, calculating data sequence probabilities, extracting the hidden states, and profile HMM.

KEYWORDS

Markov chains, hidden Markov models, profile hidden Markov Models, natural language processing

1. INTRODUCTION

Markov chains theory is increasingly being adopted in real-world computing applications since it provides a convenient way for modelling temporal, time-series data. At each clock tick, the system moves into a new state that can be the same as the previous one. A Markov chain model is a statistical tool that captures the patterns dependencies in pattern recognition systems. For this reason, Markov chain theory is appropriate in natural langue processing (NLP) where it naturally characterized by dependencies between patterns such as characters or words. Reference [1] demonstrated the limitations of using hand-written rules in NLP applications, and the reasons why to move toward statistical approaches.

Markov chains are directed graphs (a graphical model) that generally used with relatively long data sequences for data-mining tasks. Such tasks include prediction, classification, clustering, pattern discovery, software testing, multimedia analysis, networks, etc. Reference [2] indicated that there are two reasons of Markov chains popularity; very rich in mathematical structure and work well in practice for several important applications. Hidden Markov models (HMM) are an extension of Markov chains that used to find the hidden system's states based on the observations. Consequently, the conventional HMM described as follows. Given a sequence of observations, based on the trained model that fit the training data best, find the hidden states that most likely have generated the observations.

In order to facilitate the research in this direction, this paper provides a survey of this so popular data modelling technique. However, because of the wide range of the research domains that uses this technique. We specifically focus on the NLP related applications. Reference [3] lists some

DOI: 10.5121/ijdkp.2017.7103

domains that utilize Markov chains theory which include: physics, chemistry, testing, speech recognition, information sciences, queuing theory, internet applications, statistics, economics and finance, social sciences, mathematical biology, genetics, games, music, baseball, text generators, bioinformatics. Reference [4] lists the five greatest applications of Markov chains that include Scherr's application to computer performance evaluation, Brin and Page's application to Page Rank and Web Search, Baum's application to HMM, Shannon's application to information theory, and Markov's application to Eugeny Onegin.

This paper organized as follows. The next section presents a background of Markov chains theory. Section 3 highlights the main concepts of HMM followed by description of profile HMM in section 4. In section 5, we present the literature review of both Markov chains and HMM. Finally, we conclude in section 5.

2. MARKOV CHAINS

In the early of twentieth century, Andrei Markov used his name to indicate for the theory he proposed, [5].Markov chains are quite popular in computational linguistics for data modelling. A Markov chain is a memory less stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [6], a random process said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e. states). In the figure, each arc represents the probability value for transition from one state to another.

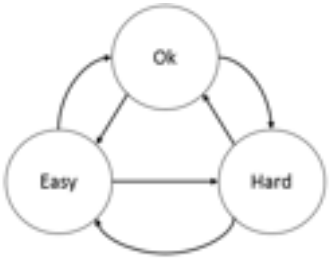


Figure 1. A Simple Markov chain with three states

Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g. the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

Table 1. A state transition matrix of three states

State		Next Exam		
		Easy (E)	Ok (O)	Hard (H)
Previous Exam	Easy (E)	P(E|E)	P(O|E)	P(H|E)
	Ok (O)	P(E|O)	P(O|O)	P(H|O)
	Hard (H)	P(E|H)	P(O|H)	P(H|H)

In Table 1, the sum of the probability values at each row is 1 as the sum of the probabilities coming out of each node should be 1. Hence, P(E|E)+P(O|E)+P(P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time, time-reversed, reversible, and irreducible Markov chains. The case shown in Figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.
To illustrate a simple Markov chain data model, a small data set contains two English sentences used to create a transition matrix based on the neighbouring characters sequences. The sentences are inspirational English quotes picked from [7]:

(1) Power perceived is power achieved. (2) If you come to a fork in the road, take it.

Figure 2 shows the transition matrix of these two quotes by counting the total number of occurrences of the adjacent two character sequences. It is a 19 × 19 matrix where the number19 is the total number of unique characters appeared in the sentences (i.e the upper mentioned quotes). In this example, creating a transition matrix is case insensitive where D is same as d, as an example. In addition, a space between two words discarded and not considered in the transition matrix. Figure 2 shows that the maximum number in the matrix's entries is 3 (a highlighted underlined value) which means that moving from character e to r (e➔r) is the most frequently appeared sequence in this small corpus. The words that contain this sequence are :{ Power (two times) and perceived}.

Figure 2. A transition matrix of two characters sequences

Based on the information obtained in the transition matrix shown in Figure 2. It is possible to answer some questions related to the given data collection. Among questions, what is the total unique number of the two characters sequences appeared in the given data set? What are the two characters sequences that did not appear in the data collection? Whatare the least frequently two characters sequences in the data set? Accordingly, Markov chains used as prediction systems such as weather forecasting. Therefore, it is possible to predict the tomorrow's weather according to the today's weather. For example, if we have two states (Sunny, Rainy), and the requirement is

to find the probability P(Sunny|Rainy), Markov chains make it possible based on the information provided in the probability transition matrix. Another example of the using Markov chains is banking industry. A big portfolio of banks based on loans. Therefore, Markov chains used to classify loans to different states such as Good, Risky, and Bad loans.

For simplicity, the information presented in Figure 2 shows the transition matrix based on total number of occurrences. Figure 3 shows the same information but using probabilities instead of the number of occurrences. That is, it contains the probability of moving from one character to another. As previously indicated, the sum of entries at each row is equal 1. In Figure 3, any matrix entry that has 0 means that there is no transition at that case. Similarly, if the matrix entry is 1, it means that there is only one possible output of that state. For example, the character “o” comes after “y”, and this is the only possible arc of the state “y”.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	0.33	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	0.33	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0.29	0	0	0	0.14	0	0	0	0	0	0.43	0	0	0	0.14	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0.17	0.17	0	0	0	0	0.17	0	0	0	0.17	0.17	0	0.17	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	0.17	0	0	0	0	0	0	0	0.17	0	0	0	0.17	0	0	0.17	0	0.33	0
p	0	0	0	0.33	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
r	0	0.33	0	0	0	0	0	0.33	0	0	0.33	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0.33	0	0	0	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 3. A probability transition matrix of two characters sequences

3. HIDDEN MARKOV MODELS

HMMis an extension to Markov chains models as both used fortemporal datamodeling.The theory of HMM introduced by Baum and his colleagues in 1960s [8]. Reference [9] indicated that most current learning research in NLP employs particular statistical techniques inspired by research in speech recognition, such as HMM and probabilistic context-free grammars (PCFGs).However, the difference is that we observe the outputs in Markov chain, but the system states are hidden in HMM. Of course, the numbers of states and the observations have to be fixed and known. In this section, we explain the concept of HMM based on example provided in Figure 1 that shows a three exam’s states Markov diagram. As a simple example, supposed that a student’s parents want to know the levels (i.e the difficulty) of their son’s exams, naturally, it is possible to recognize the exam as Easy or Ok if the son feels Fine. Similarly, it is possible to recognize the exam as Hard if the son looks Scared. From the parents’ point of view, the required states (i.e. Easy, Ok, or Hard) are hidden. However, they directly observe the student’s reaction or feeling. Hence, the parents might use the observed reactions an indication to know the hidden states. HMM is described using three matrices: the initial probability matrix, the observation probability matrix, and the state transition matrix. Figure 4 shows a HMM diagram that shows the states and the observations. In the figure, each arc represents the probability between the states and between the states and the observations.

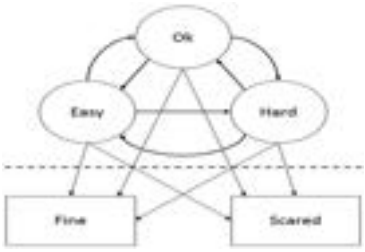


Figure 4. A HMM diagram with states and observations

Based on the information provided in the matrices, one can use either forward (also called any path) or Viterbi (also called best path) algorithms to find the probability scores during recognition phase. Figure 5 shows the trellis diagram for exam states HMM. While Forward algorithm used to compute the recognition probability of a sequence, Viterbi used to find the best-state sequence associated with the given observations, this process is also known as back-tracking. Hence, after computing the observations sequence probability and finding the maximum probability (supposed the star in Figure 5), the Viterbi algorithm leads the process back to identify the states (sources) from which the observations sequence have been emitted. In Figure 5, the maximum probabilities supposed to occur at the states shown using the dotted lines as follows; (starting at t=1): Hard, Easy, and Ok. Hence, the parents might consider the exams were Hard, Easy, and Ok, respectively.

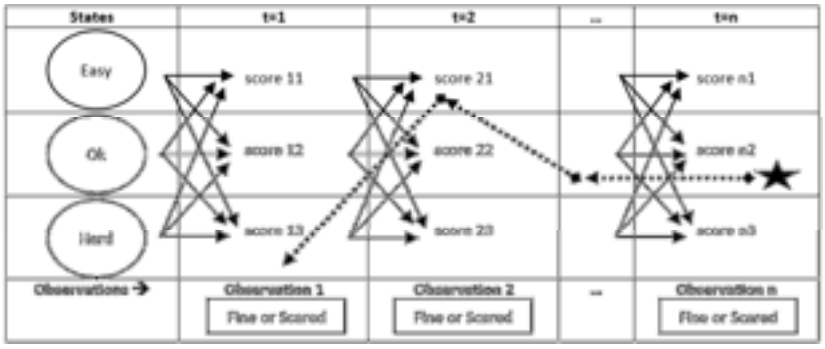


Figure 5.Trellis diagram of three states HMM

To illustrate how HMM employed to extract hidden states, Viterbi algorithm used to fine the probability at each time t. Reference [10] and other many references describe Viterbi algorithm. The example is regarding exam's states and the reactions of the student as explained in the previous sections. Figure 6 shows the HMM parameters and a question regarding the exam's states. As shown in the figure, at t=5, the values are small and they will continue decreasing as the observations increase that might lead to underflow problem. Reference [11] proposed solutions to the floating-point underflow problem that appear in the context of extremely small probability values when applying the Viterbi or forward algorithm to along sequences. The solution is to log all probability values and then add values instead of multiply for Viterbi algorithm. Regarding, Forward algorithm, the solution is to use scaling coefficients that keep the probability values in the dynamic range of the machine.

<table>
<tr><th>HMM parameters</th><th>A question and the answer</th></tr>
<tr><td>The transition matrix:
<table>
<tr><td></td><td colspan="3">next state</td></tr>
<tr><td></td><td>Easy</td><td>Ok</td><td>Hard</td></tr>
<tr><td>Easy</td><td>0.1</td><td>0.3</td><td>0.6</td></tr>
<tr><td>Ok</td><td>0.3</td><td>0.3</td><td>0.4</td></tr>
<tr><td>Hard</td><td>0.7</td><td>0.2</td><td>0.1</td></tr>
</table>
The observation matrix:
<table>
<tr><td></td><td>Easy</td><td>Ok</td><td>Hard</td></tr>
<tr><td>Fine</td><td>0.8</td><td>0.5</td><td>0.1</td></tr>
<tr><td>Scared</td><td>0.2</td><td>0.5</td><td>0.9</td></tr>
</table>
Theinitial matrix:
Π=[1/3,1/3,1/3]’</td><td>Given the following observation {Scared, Find, Fine, Scared, and Fine}; what are the most likely exam’s states. The following table shows the information obtained when implementing Viterbi algorithm.
<table>
<tr><td>States</td><td>α1(t=1)</td><td>α2(t=2)</td><td>α3(t=3)</td><td>α4(t=4)</td><td>α5(t=5)</td></tr>
<tr><td>Easy</td><td>0.0667</td><td>0.1680</td><td>0.0134</td><td>0.0015</td><td>0.0051</td></tr>
<tr><td>Ok</td><td>0.1667</td><td>0.0300</td><td>0.0252</td><td>0.0038</td><td>0.0009</td></tr>
<tr><td>Hard</td><td>0.3000</td><td>0.0067</td><td>0.0101</td><td>0.0091</td><td>0.0002</td></tr>
<tr><td>Observation</td><td>Scared</td><td>Fine</td><td>Fine</td><td>Scared</td><td>Fine</td></tr>
</table>
Where α is a variable that describe the probability at each time t.The shaded numbers indicated the max probability at time t. Accordingly, the exam’s states are :{ Hard, Easy, Ok, Hard, Easy}.Examples of some calculations: α1 (t=1, Scared) = 1/3*0.2 =0.0667. α1 (t=2, Fine) = max (0.0667*0.1, 0.1667*0.3, 0.3*0.7)*0.8 = max (0.006, 0.05, 0.21) * 0.8=0.168.</td></tr>
</table>

Figure 6.An example ofHMM and Viterbi calculations

3. PROFILE HIDDEN MARKOV MODELS

Even though HMM has been successfully used in linguistics such as speech recognition, however, it currently being used in modeling molecular biology sequences (e.g. genes and proteins) through what is called profile HMM. A profile HMM is a certain type of HMM that allowsposition dependent gap penalties. Hence, a profile HMM generally used for protein classification by creating a profile for each family through a sequence alignment process. The motivation of profile HMM is that it treats protein-spelling complexities in a systematic way. Figure 7show a profile HMM. As shown, profile HMM is a special type of Left-Right HMM (i.e. one direction) contains three stats: match, insert, and delete. In classification systems, the Baum-Welch (Forward-Backward) algorithms used for training and the Forward (any-path) algorithm used for scoring. Viterbi algorithm also used in profile HMM training and classification. Hence, profile HMM used to build an individual profile for each family and then find the max probability (i.e. the most likely family) of a molecular sequence, in question, given the model.

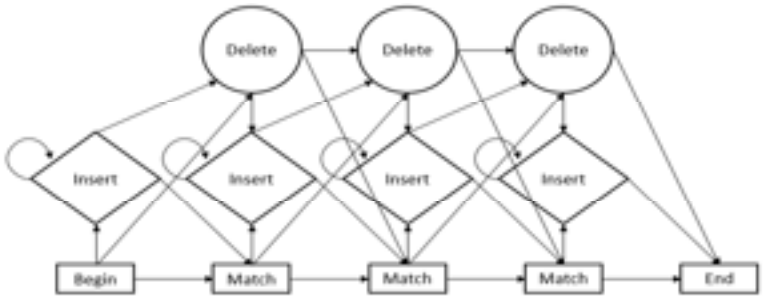


Figure 7. A profile HMM

Figure 8 shows examples of match insert, delete cases of some of molecular sequences. The match case in Figure 8 (a) shows the most conserved states (shaded area➔ACTAGT). After selecting the main states, the insert states specified as shown in Figure 8 (b). Hence, the match state characterized by high frequently observed symbols while insert states are the little observed one. Case (c) shows the delete states. As showing Figure 8 (c), the third sequence has a gap in the column 4 whereas this location has previously considered as a main states. Therefore, the case represented as a delete case. Similarly, the fourth sequence has a delete case.

(a) Match cases	(b) Insert cases	(c) Delete cases
AC-TA-GT	AC-TA-GT	AC-TA-GT
AC-TACGT	AC-TACGT	AC-TACGT
ACG-A-GT	ACG-A-GT	ACG-A-GT
A--TACGT	A--TACGT	A--TACGT
ACGTA-GT	ACGTA-GT	ACGTA-GT

Figure 8.Match, Insert, and Delete cases

4. LINGUISTIC APPLICATIONS

In the literature, there are quite many studies on modelling content dependencies for linguistics applications. Markov chain models and HMM are of great interest to linguistic scholars who primarily work on data sequences. Even though this study focuses on linguistic applications, however, Markov chains used to model a variety of phenomena in different fields. Figure 9 showssome of applications employed Markov chains. We intentionally ignored the references as the literature hastoo many studies employed Markov chains:

image processing, text and image compression, video segmentation ,forecasting, networking, signal processing, communications, software testing, genetics, bioinformatics, genome structure recognition, anomaly detection, tumour classification, water quality, epidemic spread, wind power, malicious and cyber-attack detection, traffic management, physics, chemistry, mathematical biology, games, music, multimedia processing, business activities, frauds detection.

Figure 9.Some of Markov chains and HMM applications

The following two subsections include some of the linguistic studies that utilized Markov chain theory. Linguistic applications topics mainly include (but not limited) speech recognition, speech emotion recognition, part-of-speech tagging, machine translation, text classification, text summarization, optical character recognition (OCR), named entity recognition, question answering, authorship attribution, etc. For the reader who interested in NLP, Reference [12] is a good reference as it demonstrates a thorough study of NLP (Almost) from Scratch.

4.1. NLP Markov Chains Based Research

The literature has a large number of studies that employ Markov chains for NLP applications. The following are some linguistic related applications. Reference [13] proposed a word-dividing algorithm based on statistical language models and Markov chain theory for Chinese speech processing. Reference [14] presented a semantic indexing Markov chains algorithm that uses both audio and visual information for event detection in soccer programs. Reference [15] investigated the use of Markov Chains and sequence kernels for the task of authorship attribution. Reference [16] implemented a probabilistic framework for support vector machine (SVM) that allows for automatic tuning of the penalty coefficient parameters and the kernel parameters via Markov chain for web searching via text categorization. Reference [17] demonstrated an automatic video annotation using multimodal Dirichlet process mixture model by collecting samples from the corresponding Markov chain. Reference [18] used a linguistic steganography detection method based on Markov chain models. Reference [19] showed how probabilistic Markov chain models used to detect topical structure in large text corpora.

Reference [20] proposed a method of recognizing location names from Chinese texts based on Max-Margin Markov network. Reference [21] utilized Markov chain and statistical language models in a linguistic steganography detection algorithm. Reference [22] proposed a Markov chain based algorithm for Chinese word segmentation. Reference [23] presented two new textual feature selection methods based on Markov chains rank aggregation techniques. Reference [24] proposed a Markov chain model for radical descriptors in Arabic text mining. Reference [25] presented statistical Markov chain models for the distributions of words in text lines. Reference [26] proposed a method for handwritten Chinese/Japanese text (character string) recognition based on semi-Markov conditional random fields (semi-CRFs). Reference [27] presented a Markov chain method to find authorship attribution on relational data between function words. Reference [28] utilized a probabilistic Markov chain model to infer the location of Twitter users. Reference [29] proposed a Markov chain based technique to determine the number of clusters of a corpus of short-text documents. Reference [30] proposed a Markov chain based method for digital document authentication. Reference [31] used Markov chain for authorship attribution in Arabic poetry. Reference [32] investigated the application of mixed-memory Markov models (MMMs) to automatic language identification. MMMs used to approximate standard statistical n-gram models ($n > 2$) by a mixture of bigram models.

4.2. NLP Hidden Markov Models Based Research

HMM based research has been for long an active research area due to the rapid development in NLP applications. The literature has many studies as follows. Reference [33] proposed to extract acronyms and their meaning from unstructured text as a stochastic process using HMM. Reference [34] proposed a morphological segmentation method with HMM method for Mongolian. Reference [35] employed HMM for Arabic handwritten word recognition based on HMM. Reference [36] presented a scheme for off-line recognition of large-set handwritten characters in the framework of the first-order HMM. Reference [37] proposed the use of hybrid HMM/Artificial Neural Network (ANN) models for recognizing unconstrained offline handwritten texts. Reference [38] used HMM for recognizing Farsi handwritten words. Reference [39] describes recent advances in HMM based OCR for machine-printed Arabic documents. Reference [40] proposed a HMM based method for named entity recognition. Reference [41] combined text classification and HMM techniques for structuring randomized clinical trial abstracts. Reference [42] employed HMM for medical text classification. Reference [43] proposes text (sequences of pages) categorization architecture based on HMM. Reference [44] described a model for machine translation based on first-order HMM. Reference [45] introduced speech emotion recognition by use of HMM. Reference [46] presented a HMM based method for speech emotion recognition. Reference [47] discussed the role of HMM in speech recognition. Reference [48] indicated that almost all present day large vocabulary continuous speech recognition (LVCSR) systems based on HMM. Reference [49] presented a text summarization method based on HMM. Reference [50] presented a method for summarizing speech documents using HMM. Reference [51] used HMM for part-of-speech tagging task. Reference [52] presented a second-order approximation of HMM for part-of-speech tagging task.

4.3. Profile Markov Models Based Research

Up to the date of writing this paper, no profile HMM based research found to serve NLP. Most of the works related to molecular applications.

5. CONCLUSIONS

In this paper, we presented Markov chains and HMM as standard models in language modelling. In the last decades, utilizing Markov and Hidden Markov based concepts have been steadily increasing in linguistic applications such as speech recognition, part of speech tagging, and noun-phrase chunking. This work discussed the potential and the size of Markov and Hidden Markov based research particularly related to NLP applications. For future work, it is worthy to compare the capabilities of HMM with other machine learning tools such as deep neural networks in building automatic speech recognition (ASR) systems.

ACKNOWLEDGEMENTS

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

REFERENCES

[1] Nadkarni, Prakash M., LucilaOhno-Machado, and Wendy W. Chapman. "Natural language processing: an introduction." Journal of the American Medical Informatics Association 18.5 (2011): 544-551.
[2] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.
[3] Markov_chain. (2016, November). Retrieved from https://en.wikipedia.org/wiki/Markov_chain
[4] Von Hilgers, Philipp, and Amy N. Langville. "The five greatest applications of Markov Chains." Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA. 2006.
[5] Bikel, Daniel M., Richard Schwartz, and Ralph M. Weischedel. "An algorithm that learns what's in a name." Machine learning 34.1-3 (1999): 211-231.
[6] Leon-Garcia, Alberto, and Alberto. Leon-Garcia. Probability, statistics, and random processes for electrical engineering. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.
[7] California Indian Education. (2016, November). Retrieved from http://www.californiaindianeducation.org/inspire/world/
[8] L. Baum et. al. A maximization technique occuring in the statistical analysis of probabilistic functions of markov chains. Annals of Mathematical Statistics, 41:164–171, 1970.
[9] Cardie, Claire, and Raymond J. Mooney. "Guest editors' introduction: Machine learning and natural language." Machine Learning 34.1 (1999): 5-9.
[10] Marsland, Stephen. Machine learning: an algorithmic perspective. CRC press, 2015.
[11] Blunsom, Phil. "Hidden markov models." Lecture notes, August 15 (2004): 18-19.
[12] Collobert, Ronan, et al. "Natural language processing (almost) from scratch."Journal of Machine Learning Research 12.Aug (2011): 2493-2537.
[13] Bin, Tian, et al. "A Chinese word dividing algorithm based on statistical language models." Signal Processing, 1996., 3rd International Conference on. Vol. 1. IEEE, 1996.
[14] Leonardi, Riccardo, PierangeloMigliorati, and Maria Prandini. "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains." IEEE Transactions on Circuits and Systems for Video Technology 14.5 (2004): 634-643.
[15] Sanderson, Conrad, and Simon Guenter. "On authorship attribution via Markov chains and sequence kernels." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.
[16] Lim, Bresley Pin Cheong, et al. "Web search with text categorization using probabilistic framework of SVM." 2006 IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. IEEE, 2006.
[17] Velivelli, Atulya, and Thomas S. Huang. "Automatic video annotation using multimodal Dirichlet process mixture model." Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on. IEEE, 2008.

[18] Chen, Zhi-li, et al. "Effective linguistic steganography detection." Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on. IEEE, 2008.
[19] Dowman, Mike, et al. "A probabilistic model of meetings that combines words and discourse features." IEEE Transactions on Audio, Speech, and Language Processing 16.7 (2008): 1238-1248.
[20] Li, Lishuang, Zhuoye Ding, and Degen Huang. "Recognizing location names from Chinese texts based on max-margin markov network." Natural Language Processing and Knowledge Engineering, 2008. NLP-KE'08. International Conference on. IEEE, 2008.
[21] Meng, Peng, et al. "Linguistic steganography detection algorithm using statistical language model." Information Technology and Computer Science, 2009. ITCS 2009. International Conference on. Vol. 2. IEEE, 2009.
[22] Baomao, Pang, and Shi Haoshan. "Research on improved algorithm for Chinese word segmentation based on Markov chain." Information Assurance and Security, 2009. IAS'09. Fifth International Conference on. Vol. 1. IEEE, 2009.
[23] Wu, Ou, et al. "Rank aggregation based text feature selection." Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on. Vol. 1. IET, 2009.
[24] El Hassani, Ibtissam, AbdelazizKriouile, and Youssef BenGhabrit. "Measure of fuzzy presence of descriptors on Arabic Text Mining." 2012 Colloquium in Information Science and Technology. IEEE, 2012.
[25] Haji, Mehdi, et al. "Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms." Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on. IEEE, 2012.
[26] Zhou, Xiang-Dong, et al. "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields." IEEE transactions on pattern analysis and machine intelligence 35.10 (2013): 2413-2426.
[27] Segarra, Santiago, Mark Eisen, and Alejandro Ribeiro. "Authorship attribution using function words adjacency networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.
[28] Rodrigues, Erica, et al. "Uncovering the location of Twitter users." Intelligent Systems (BRACIS), 2013 Brazilian Conference on. IEEE, 2013.
[29] Goyal, Anil, Mukesh K. Jadon, and Arun K. Pujari. "Spectral approach to find number of clusters of short-text documents." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on. IEEE, 2013.
[30] Shen, Jau Ji, and Ken Tzu Liu. "A Novel Approach by Applying Image Authentication Technique on a Digital Document." Computer, Consumer and Control (IS3C), 2014 International Symposium on. IEEE, 2014.
[31] Ahmed, Al-Falahi, et al. "Authorship attribution in Arabic poetry." 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA). IEEE, 2015.
[32] Kirchhoff, Katrin, Sonia Parandekar, and Jeff Bilmes. "Mixed-memory Markov models for automatic language identification." Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on. Vol. 1. IEEE, 2002.
[33] Osiek, Bruno Adam, Geraldo Xexéo, and Luis Alfredo Vidal de Carvalho. "A language-independent acronym extraction from biomedical texts with hidden Markov models." IEEE Transactions on Biomedical Engineering 57.11 (2010): 2677-2688.
[34] He, Miantao, Miao Li, and Lei Chen. "Mongolian Morphological Segmentation with Hidden Markov Model." Asian Language Processing (IALP), 2012 International Conference on. IEEE, 2012.
[35] Alma'adeed, Somaya, Colin Higgens, and Dave Elliman. "Recognition of off-line handwritten Arabic words using hidden Markov model approach." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 3. IEEE, 2002.
[36] Park, Hee-Seon, and Seong-Whan Lee. "Off-line recognition of large-set handwritten characters with multiple hidden Markov models." Pattern Recognition 29.2 (1996): 231-244.
[37] Espana-Boquera, Salvador, et al. "Improving offline handwritten text recognition with hybrid HMM/ANN models." IEEE transactions on pattern analysis and machine intelligence 33.4 (2011): 767-779.
[38] Imani, Zahra, et al. "offline Handwritten Farsi cursive text recognition using Hidden Markov Models." Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on. IEEE, 2013.

[39] Prasad, Rohit, et al. "Improvements in hidden Markov model based Arabic OCR." Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.
[40] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.
[41] Xu, Rong, et al. "Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts." AMIA. 2006.
[42] Yi, Kwan, and JamshidBeheshti. "A hidden Markov model-based text classification of medical documents." Journal of Information Science (2008).
[43] Frasconi, Paolo, Giovanni Soda, and Alessandro Vullo. "Hidden markov models for text categorization in multi-page documents." Journal of Intelligent Information Systems 18.2-3 (2002): 195-217.
[44] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. "HMM-based word alignment in statistical translation." Proceedings of the 16th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, 1996.
[45] Schuller, Björn, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 2. IEEE, 2003.
[46] Nwe, Tin Lay, Say Wei Foo, and Liyanage C. De Silva. "Speech emotion recognition using hidden Markov models." Speech communication 41.4 (2003): 603-623.
[47] Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." Technometrics 33.3 (1991): 251-272.
[48] Gales, Mark, and Steve Young. "The application of hidden Markov models in speech recognition." Foundations and trends in signal processing 1.3 (2008): 195-304.
[49] Conroy, John M., and Dianne P. O'leary. "Text summarization via hidden markov models." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.
[50] Maskey, Sameer, and Julia Hirschberg. "Summarizing speech without text using hidden markov models." Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006.
[51] Kupiec, Julian. "Robust part-of-speech tagging using a hidden Markov model." Computer Speech & Language 6.3 (1992): 225-242.
[52] Thede, Scott M., and Mary P. Harper. "A second-order hidden Markov model for part-of-speech tagging." Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. Association for Computational Linguistics, 1999.

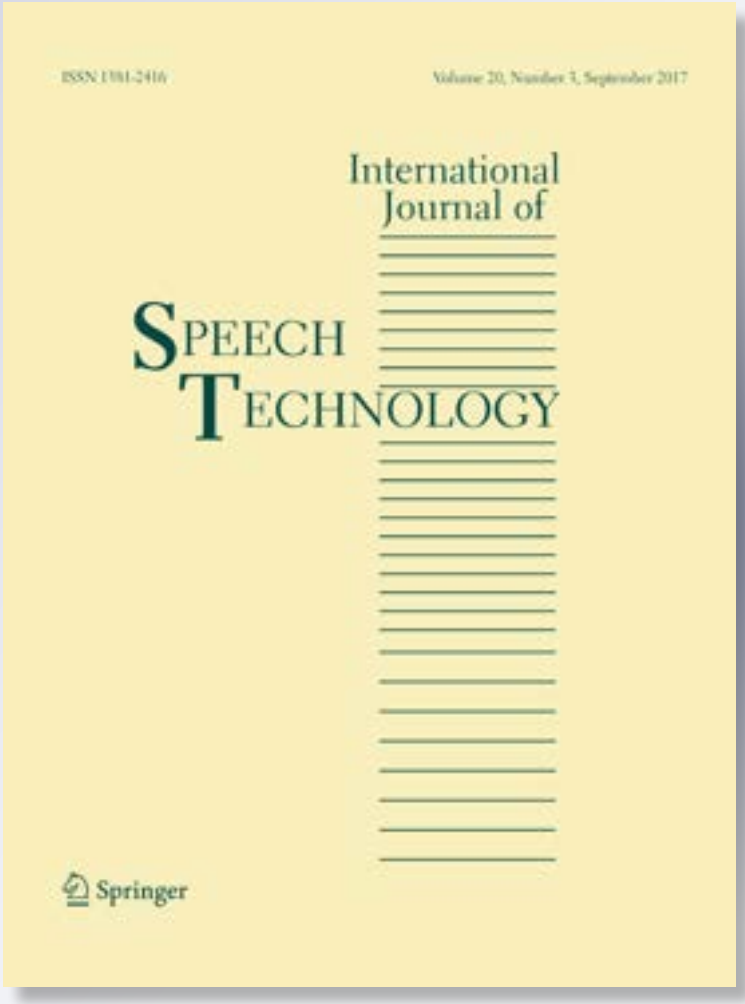
The impact of phonological rules on Arabic speech recognition

Fawaz S. Al-Anzi & Dia AbuZeina

International Journal of Speech Technology

ISSN 1381-2416
Volume 20
Number 3

Int J Speech Technol (2017) 20:715-723
DOI 10.1007/s10772-017-9440-2



Springer

Your article is protected by copyright and all rights are held exclusively by Springer Science+Business Media, LLC. This e-offprint is for personal use only and shall not be self-archived in electronic repositories. If you wish to self-archive your article, please use the accepted manuscript version for posting on your own website. You may further deposit the accepted manuscript version in any repository, provided it is only made publicly available 12 months after official publication or later and provided acknowledgement is given to the original source of publication and a link is inserted to the published article on Springer's website. The link must be accompanied by the following text: "The final publication is available at link.springer.com".

Author's personal copy

Int J Speech Technol (2017) 20:715–723
DOI 10.1007/s10772-017-9440-2

The impact of phonological rules on Arabic speech recognition

Fawaz S. Al-Anzi[1] · **Dia AbuZeina**[1]

Received: 29 April 2017 / Accepted: 15 July 2017 / Published online: 24 July 2017
© Springer Science+Business Media, LLC 2017

Abstract The pronunciation variation is a well-known phenomenon that has been widely investigated for automatic speech recognition (ASR). The knowledge-based phonological rules are generally used to capture the accurate phonetic realization in order to minimize the mismatch between the ASR dictionary and the actual phonetic representation of the speech signal. For the Arabic ASR, there are a number of studies that employ these rules on Arabic ASR systems; however, little research has been devoted to measure the precise performance of each rule. In this paper, we aim at finding the exact effect of each rule as well as the rules that have no influence. We used the Carnegie Mellon University PocketSphinx speech recognizer with a new "in-house" modern standard Arabic speech corpus that contains 19 h for training and 3.7 h for testing. We evaluated the effect of three famous rules (Shadda, Tanween, and the solar letters). The experimental results do not show clear evidence that using phonological rules for ASR dictionary adaptation can enhance the performance for within-word pronunciation variation. The obtained results might be an indication to rethink or use other ASR performance aspects, such as cross-word pronunciation variation and the optimal phonemes set of the Arabic language.

Keywords Arabic · Speech recognition · Phonological rules · Sphinx

✉ Fawaz S. Al-Anzi
FAWAZ.ALANZI@KU.EDU.KW

Dia AbuZeina
DIA.ABUZEINA@KU.EDU.KW

[1] Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait

1 Introduction

Automatic speech recognition (ASR) is of particular interest in different fields, such as human computer interface (HCI) and the natural language processing (NLP). Recently, the Arabic large-vocabulary speaker-independent continuous speech recognition system has received significant attention in the NLP research community. However, Arabic ASR poses some challenges, such as the difficulty to obtain corpora for dialects that are spoken rather than written (i.e. there is no common standard for writing), difficulty in obtaining a large diacritized text as the Arabic allows writing without diacritics, and the enormous number of word forms due to the morphology richness of Arabic. In addition to the previous difficulties, the pronunciation variation phenomenon adds further challenges to ASR systems. That is, the continuous speech naturally has some acoustic variations that not accounted for in the pronunciation dictionary, which can lead to less than optimal performance. Due to the pronunciation variation problem, it is almost impossible to consider all possible variants in the pronunciation. No doubt that the mismatch between the acoustic features of the speech signal and the phonetic transcription in the ASR dictionary is a source of errors. In fact, it is extremely important that the phonemes of the pronunciation dictionary to adequately represent the actual contents of the training speech files. Pronunciation variations modeling is an active research area for robust ASR as well as the other related applications, such as text-to-speech systems to generate speech that is more natural.

One approach to tackle the pronunciation variations is through the language's phonological rules that consider the phonetic mismatch through ASR pronunciation dictionaries. For instance, (Ramsay et al. 2014) indicates that the performance of ASR is improved by shrinking the

Author's personal copy

mismatch between the speech and the text used in training the acoustic model. Employing phonological rules for the ASR dictionary adaptation is classified as a knowledge based approach, however, data-driven is another option for the pronunciation variation. Hence, both approaches introduce some variants to generate phonetically rich dictionary pronunciation that might alleviate the acoustic changes on the performance. Modeling pronunciation variation includes two types, the within-word and the crossword pronunciation variation. In this work, we consider a knowledge based approach for within-word pronunciation variation. For comparison purposes and to evaluate the phonological rules, this work considers two testing cases: the baseline case and the dictionary adaptation case. The baseline case uses the phonemes set without any adaptation while the dictionary adaption case considers some rules. For each adaptation case, the ASR performance is evaluated to separately measure the impact of the corresponding rule.

In this paper, we employed the latest Carnegie Mellon University (CMU) PocketSphinx ASR engine (CMU Sphinx Downloads 2017) for exploring the effect of the Arabic phonological rule on the Arabic ASR performance. PocketSphinx includes the latest available releases as follows: sphinxbase—5prealpha, PocketSphinx—5prealpha, SphinxTrain—5prealpha. In the experiments, we used a new "in house" continuous speech corpus that contains 19 h for training and 3.7 h for testing. The speech is of broadcast news using the modern standard Arabic (MSA). The speech transcription is manually diacritized. This study also presents the intermediate steps for training and decoding, such as the proposed and used phonemes set, the pronunciation dictionary, the acoustic model, and the language model. We emphasize that this work is a preliminary step toward further research using the newly created corpus. This corpus has been fully supported by Kuwait University. The size of the corpus in this work is 22.7 h; however, we aim at increasing the size to about 30 h.

In the next section, we present the motivation of pronunciation variation for the Arabic ASR. In Sect. 3, we present the literature review followed by the phonemes set in Sect. 4. Section 5 presents the Arabic phonological rules followed by the baseline system in Sect. 6. Section 7 presents the proposed method and the experimental results in Sect. 8. We present diacritization in Sect. 9 and, finally, the conclusion and the future work are presented in Sect. 10.

2 Motivation

The acoustic properties of speech signals introduce some pronunciation variations, which is the major source of errors in ASR. Hence, employing phonological rules in ASR might enhance the supposed match between the transcription of the speech files and the actual acoustic features in the training process. In the case of training, without considering the phonological rules, many of the phonetic segments might lose suitable representation in the acoustic model. The differences between the actual speech signal and the phonetic spelling of the ASR dictionary leads to out-of-vocabulary word forms and, therefore, reduces the performance. The variation comes into the form of insertions, deletions, or substitutions of phoneme(s) beyond their listed forms in the ASR dictionary. (Benzeghiba and De Mori 2007) lists the major sources of errors in ASR, which include foreign and regional accents, speaker physiology, speaking style and spontaneous speech, rate of speech, children's speech, emotional state, and more. In order to handle the phonetic mismatch cases, some variants are generally added to the ASR dictionary (that is also called the lexical adaptation). For the Arabic ASR, little research has been devoted to find the exact contribution of each phonological rule on the overall performance. The motivation of this work is to explore the performance using some of the well-known rules. Based on our best knowledge, this is the first attempt to explore the effect of these rules using a continuous speech corpus. In fact, the Arabic ASR research is in need of exhaustive practical studies to define the most influential phonological rules. The precise evaluation of the most influential rules might lead to generating a phonetic transcription that is a reasonably approximation to reality. In addition, this study aims at finding the pronunciation rules that have no effect or even degrade the performance, if any.

3 Literature review

The literature shows that employing phonetically rich dictionaries will perform better than standard dictionaries that have no variants. For instance, (Fosler-Lussier et al. 1999) showed that the mismatch between the phonetics recognized and the word's phonetic transcription in the dictionary increases word error rate (WER) and degrades performance. (Fosler-Lussier et al. 1999) showed that the ASR performance will be highly improved if there is a closer match between the phonetic sequence recognized by the decoder and the phonetic transcription in the dictionary. Phonological rules have been utilized in ASR systems for different languages. For instance, (Tajchman et al. 1995) and (Finke and Waibel 1997) used a set of US English rules to generate pronunciation variants. (Wester 2003) and (Kessens et al. 1999) used a set of Dutch phonological rules to model pronunciation variations. (Kyong-Nim and Minhwa 2007) and (Jeon et al. 1998) used a set of Korean phonological rules to generate pronunciation variants. (Liu and Fung 2003)

Author's personal copy

applied phonological rules to produce variants for Cantonese accented Mandarin speech. The knowledge-based approach was also implemented by (Seman and Jusoff 2008) for spontaneous Standard Malay.

For the Arabic language, (Ali et al. 2009) developed a software tool to generate pronunciation dictionaries for Arabic texts using Arabic pronunciation rules. This tool was later used in other works, such as (AbuZeina et al. 2011, 2012). However, the tool that was developed by (Ali et al. 2009) demonstrated the performance of the overall performance without the precise evaluation of each rule. (Alghamdi et al. 2007) demonstrates a phonetically rich ASR dictionary for a news transcription system for MSA. (Ramsay et al. 2014) presents a comprehensive system for generating a phonetic transcription based on a set of (language-dependent) pronunciation rules that convert the fully Arabic text into the actual sounds. The experimental results in (Abushariah et al. 2012) show that the non-diacritized case slightly outperforms the diacritized text case for a phonetically rich and balanced Arabic speech corpus. The research in (Vergyri et al. 2008) found that the diacritized text improved the acoustic model more than undiacritized orthography. Most of the previous works were performed using relatively small corpora; however, we used a larger corpus to explore the effect of phonological rules on Arabic ASR. (Masmoudi et al. 2014) employed a set of pronunciation rules (80 rules) for creating a phonetic dictionary for the Tunisian Arabic. (Biadsy et al. 2009) shows that using linguistically motivated pronunciation rules can significantly improves the ASR performance. (Al-Haj et al. 2009) demonstrated the knowledge-based approach to add variants to dictionary. They worked on the Iraqi-Arabic speech and focused on short vowels. The literature shows many studies have discussed the phonological rules, however, no study explores the impact of these rules separately.

4 The phonemes set

The phoneme is the basic unit of speech that represents a distinctive sound of the language's phonology. Hence, a change of a particular phoneme in a word makes a change in the meaning of the word. Phonemes play a vital role in the performance of ASR and text to speech systems. In this work, we propose a phoneme set that is used to evaluate the recognition performance of the prepared corpus. The pronunciation dictionary is prepared using the proposed phonemes set by a mapping process between the Arabic letters (the language's vowels and consonants) and their corresponding phonemes. However, in some cases, morphologically driven rules are used for a phonetic rich dictionary. In addition, some pronunciation exceptions might be manually processed for better acoustic representation. (Ali et al. 2009) and (Ramsay et al. 2014) elaborate on Arabic phonemes and the pronunciation rules.

In general, creating a dictionary of a particular language requires linguistic experts and a deep knowledge of the language sounds. However, the choice of the phoneme in the phonemes set is not straightforward as it has some constraints. For instances, it should represent the relevant information of the language sounds, it should consider the surrounding context between the letters, and it should carefully estimate the starting and the ending of the letters. No doubt, the phonemes that are used to represent the training

Table 1 The Arabic letters and the phonemes set

#	Letter	Phoneme	#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	17	ر	R	33	ه	H
2	آ	AA	18	ز	Z	34	و	W
3	أ	O	19	س	S	35	ى	AY
4	ؤ	EW	20	ش	SH	36	ي	Y
5	إ	I	21	ص	SS	37	ً	N
6	ئ	EY	22	ض	DD	38	ٌ	N
7	ا	A	23	ط	TT	39	ٍ	N
8	ب	B	24	ظ	ZZ	40	َ	AU
9	ة	P	25	ع	AE	41	ُ	AW
10	ت	T	26	غ	GH	42	ِ	AI
11	ث	TH	27	ف	F	43	ّ	~
12	ج	J	28	ق	Q	44	ـا	AUA
13	ح	HH	29	ك	K	45	ـو	AWW
14	خ	KH	30	ل	L	46	ـي	AIY
15	د	D	31	م	M			
16	ذ	DH	32	ن	N			

Author's personal copy

words characterize the quality of the acoustic models and, therefore, the overall performance. Table 1 shows the phonemes set used in this work. It contains 46 phonemes. In addition to the Arabic letters, the table includes the short vowels that are Fatha (َ), Damma (ُ), and Kasra (ِ). As shown in the table, the Shadda (ّ) is represented using the symbol (~). We also used three phonemes to represent the Fatha that proceeds Alif (ا) ➔ ـآ as a single phoneme that is (AUA), the Damma that proceeds Waw (و) ➔ ـُو as a single phoneme that is (AWW), and the Kasra that proceeds Ya (ي) ➔ ـِي as (AIY). The reason for handling these cases as a single phoneme is that the pronunciation of the short vowels is different when it proceeds the long vowels. Hence, it would be correctly transcribed as single phonemes. For instance, "AW", "W" would be short vowel /AW/, consonant /W/, which is different in pronunciation from long /AWW/ as a single vowel, and likewise for the others. Consider the English name of the country "Kuwait". That would be correctly transcribed as /K AW W Y T/, because the /AW/ and /W/ are separate phonemes. But that's different from when the Arabic character "ـُو" is used as the long vowel /AWW/ such as in the word "مَشرُوعَات" which means "projects". Hence, AUA, AWW and AIY has a better representation of the actual sounds that reflects the actual pronunciation. In this work, the transliteration of Arabic will be presented using the phonemes that are shown in Table 1. Table 1 has no symbol for Sukon (ْ) that does not correspond to any sound.

The selection of a phoneme symbol is an optional and it does not matter what phoneme symbol is used for an individual letter. For instance, (Ali et al. 2009) used (UW) for (ـُو) while we used (AWW). In the training stage, each phoneme is modelled using a sequence of a hidden Markov model (HMM) that is stated for computing the acoustic model. In the decoding stage, the phoneme is initially recognized and then used to find the most likely spoken words based on the best-matched phonemes between the speech file in question (the observations) and the trained HMMs of the acoustic model.

5 The Arabic phonological rules

In this work, we employ knowledge based phonological rules to model the pronunciation variations in an Arabic ASR for MSA. That is, a set of rules (defined by the experience of language experts) are used to adapt the phonetic dictionary in order to account for some variations that naturally occur in the Arabic pronunciation. (Elshafei 1991) is a good reference for the Arabic sounds. The essence of this work is to replace the standard phonetic representation to the expected actual pronunciations to, hopefully, perform better in the training and the decoding process. The rules convert the phonetic transcription in the dictionary to a "better" phonetic form that is close to the actual sounds based on the neighboring phonemes. Hence, the phonological rules could predict the variation within a word in order to control its representation in the dictionary. The rules introduced in this study include Shadda (الشدة), Nunation or Tanween (التنوين), and Assimilation (الادغام) using the sun letters also called solar letters (الحروف الشمسية).

To clarify the pronunciation differences due to the phonological rules, Table 2 shows the used rules along with examples. The Shadda (ّ) rule is a double or repeat of the previous consonant (also called the gemination mark). Nunation also called Tanween is a doubling of short vowels that includes (ُ: u, َ: a, ِ: i). Hence, Tanween includes any case of Dammatan (two consecutive short Damma), Fathatan (two consecutive short Fatha), or Kasratan (two consecutive short Kasra). Each Tanween is symbolized as (ٌ ، ً ، ٍ). Assimilation is a merging of the sounds of two consecutive consonants (it could be within a single word or between two separated words) to produce a single geminated sound, so that the two sounds become alike or even identical. In this work, we used assimilation using the solar letters: {ت:T , ث:TH , د:D , ذ:DH , ر:R , ز:Z , س:S , ش:SH , ص:SS , ض:DD , ط:TT , ظ:ZZ , ل:L , ن:N }. The L "ل" that proceeds any of the solar consonants is assimilated with the consonant.

6 The baseline system

The goal of preparing the baseline system is to compare the performance when employing the phonological rules. Creating a continuous speech corpus was the first step in this research. We got the raw MSA speech files form (Al-Sabah TV 2007) in Kuwait. The speech contents belong to broadcast news. We performed the preprocessing step that includes segmenting the long speech files into short segments of 30–60 s. The produced segmented speech files cover different news stories and it sums up to 22.7 h of 29 speakers (19 male speakers and the rest are for female speakers). The speech files were sampled at 16 KHz mono. A silence of 0.1 s was used at the beginning and at the end of each speech file. We collected 2160 speech files that were transcribed and manually diacritized. The speech files were divided into two parts: the training set that contained 1802 speech files (19 h) and the testing set that contained 358 speech files (3.7 h). We emphasize that creating a continuous speech corpus is a time-consuming task.

The training stage of an ASR system consists of building an acoustic model that is a major component of ASR engines. Acoustic models statistically represent the relationships between the speech signals and the language phonemes. It has been long observed that the HMM based

Author's personal copy

Table 2 The popular Arabic phonological rules

No.	Rule	Examples	Phonetic transcription	Actual pronunciation	Meaning
1	Shadda	التَّنمِيَة	A L T ~ AU N M AI Y AU P	التتنمِيَة	Development
		التَّنَسِيق	A L T ~ AU N AU S AI Y Q	التتَنَسِيق	Formatting
		مُشَرَّف	M AW SH AU R ~ AI F	مُشَررف	Honorable
2	Tanween	مُعضِلَة	M AW AE DD AI L AU P WW	مُعضِلتن	Problem
		مُعَتبِرةً	M AW AE AU T B AI R P UU	مُعَتبِرتن	Consider
		مُفعَمَةٍ	M AW F AE AU M AU P II	مُفعَمَتن	Replete
3	Assimilation (solar letters)	التابِع	A L T A B AI AE AI	اتابِع	Dependent
		الثراءِ	A L TH AU R A E AI	اثراءِ	Get rich
		الداعِمَة	A L D A AE AI M AU P	اداعِمَة	Support
		الذهَب	A L DH AU H AU B AI	اذهَب	Gold
		الرَّئيس	A L R ~ AU EY Y S AI	ارَّئيس	President
		الزّجَاج	A L Z ~ AW J AU A J AI	ازّجَاج	Glass
		السّاحِل	A L S ~ A HH AI L AI	اسّاحِل	Coast
		الشَبَابُ	A L SH AU B AU A B AW	اشَبَابُ	Young
		الصَّادِر	A L SS ~ AU A D AI R	اصَّادِر	Issued
		الضَّبط	A L DD ~ AU B TT	اضَّبط	Settings
		الطاقَة	A L TT ~ AU A Q AU P	اطاقَة	Energy
		الظن	A L ZZ ~ AU N	اظن	Suspicion
		اللّجنَة	A L L AU ~ J N AU P	الّجنَة	Committee
		النَائِب	A L N AU A EY AI B	انَائِب	Deputy

acoustic models have been successfully implemented in the state of the art speech recognizers. CMU Sphinx speech engines support three types for HMM based acoustic modeling. For instance, the CMU Sphinx configuration file "Sphinx_train.cfg" has the commands to enable or disable the desired acoustic model. The types of acoustic models include the traditional fully continuous, the semi-continuous, and the phonetic tied-mixture (PTM) models. Despite the common implementation of fully continuous and semi-continuous in the Arabic ASR, however, PTM is a recent method that is compromised between important factors, such as speed and performance. It is also characterized by fast decoding as well as its ability to handle large amounts of speech collections. In this work, we used the PTM based acoustic models.

The pronunciation dictionary was generated using a Python based program based on the proposed phonemes set. The total number of unique words in the training set is 37,158. The corpus vocabulary and the size of the speech corpus determines some training parameters, such as the number of Senones (tied-state) and the number of Gaussians. Table 3 shows the approximation number of Senones and the Gaussian densities according to the vocabulary and the size of some English speech corpora (Training Acoustic Model for CMUSphinx 2017). For the language model, we used the CMU language toolkit (Building Language Model 2017) to calculate the statistical N-grams (i.e. 1, 2, and 3-g) based on the entire corpus transcription.

In addition to the previous steps, the SphinxTrain performs some internal tasks, such as computing features from audio files, training context independent models, training context dependent models, build trees, prune trees, lattice generation, lattice pruning, and finally decoding using the trained models. Once having the trained acoustic model, the PocketSphinx is used for decoding by utilizing other components, such as the pronunciation dictionary and the

Table 3 The approximate number of senones and Gaussian densities

Vocabulary	Hours	Senones	Densities	Example
20	5	200	8	Tidigits digits recognition
100	20	2000	8	RM1 command and control
5000	30	4000	16	WSJ1 5k small dictation
20,000	80	4000	32	WSJ1 20k big dictation
60,000	200	6000	16	HUB4 broadcast news
60,000	2000	12,000	64	Fisher rich telephone transcription

Author's personal copy

language model. In ASR, the training phase is time-consuming. Hence, we considered speeding up the execution time using an option in the PocketSphinx. We used the configuration file that is called "sphinx_train.cfg". This file has an option for the multiprocessing mode. The two options that can be used for reducing the training and the decoding time are as follows. $CFG_NPART = 10 ➔ the number of parts to run forward–backward estimation; and $DEC_CFG_NPART = 10 ➔ how many pieces to split decoding. The number ten is specified by the user according to the desired factor to reduce the execution time. The default value of these two parameters is one. This option is helpful since it clearly reduces the execution time by utilizing a number of processors in multicore machines.

7 The proposed method

The proposed method includes the dictionary adaptation (also called lexicon adaptation) process to change the phonetic transcription in the pronunciation dictionary according to the phonological rule effect. We have three cases that are included: Shadda, Tanween, and Assimilation. For the Shadda rule, we investigated two cases. The first case is to discard the Shadda. For instance, "T ~" becomes "T". The second case includes a replacement of the Shadda (~) to the proceeding consonant. For instance, "T ~" becomes "T T". Table 4 shows examples of the replacement process for the two cases. Of course, the replacement will occur for the dictionary's words that have Shadda. When implementing this rule, the phonemes set is also adapted to remove the Shadda phoneme (or the Shadda symbol) from the phonemes list since it is not used anymore. No change is performed on the language model. After the dictionary adaptation, the acoustic model is trained to generate the modified acoustic model.

The same process is repeated for the Tanween rule. For all the dictionary's entries that include any of the symbol {(WW :ٌ),(UU :ً),(II :ٍ)} they will be replaced to N. Hence, the Tanween rule is appended as N instead of the Tanween symbols. Table 5 shows some examples.

For the solar letters, the transformation includes an assimilation process of the phoneme (L) with the following solar consonant. (Akesson 2010) phonetically explained that the letter (L) and the solar letters have a close articulation area and they all originate from between the teeth to the lower part of the palate. Table 6 shows the dictionary adaptation for this rule.

Table 4 The Shadda rule transformation process

The Shadda rule	
Case 1: discarding Shadda	Case 2: duplicate the proceeding consonant
Before dictionary adaptation	Before dictionary adaptation
… أَجَّلَ O AU J ~ AU L AU …	… أَجَّلَ O AU J ~ AU L AU …
After dictionary adaptation	After dictionary adaptation
… أَجَّلَ O AU J AU L AU …	… أَجَّلَ O AU J J AU L AU …

Table 5 The Tanween rule transformation process

The Tanween rule		
Case 1: replace (ٌ) by (N)	Case 2: replace (ً) by (N)	Case 3: replace (ٍ) by (N)
Before dictionary adaptation	Before dictionary adaptation	Before dictionary adaptation
… أَعدَادٌ O AU AE D AU A D WW …	… أَعضَاءً O AU AE DD AU A E UU …	… أَهدَافٍ O AU H D AU A F II …
After dictionary adaptation	After dictionary adaptation	After dictionary adaptation
… أَعدَادٌ O AU AE D AU A D N …	… أَعضَاءً O AU AE DD AU A E N …	… أَهدَافٍ O AU H D AU A F N …

Author's personal copy

Table 6 The Solar letters transformation process

The Solar letters rule {ت:T , ث:TH , د:D , ذ:DH , ر:R , ز:Z , س:S , ش:SH , ص:SS , ض:DD , ط:TT , ظ:ZZ , ل:L , ن:N}		
Case 1: replace (التـ) by (اتـ)	Case 2 … Case 13 (for all Solar letters)	Case 14: replace (الن) by (انـ)
Before dictionary adaptation	Before dictionary adaptation	Before dictionary adaptation
… التابِع A L T A B AI AE AI …	(ث)➔اثراء, (د)➔ادعم, (ذ)➔اذهب, (ر)➔ الشرق, (ش)➔السودان, (س)➔الرسالة, (ص)➔الصحافة, (ض)➔الضخمة, (ط)➔ اللائحة ➔(ل) ,الظروف ➔(ظ) ,الطبي	… النَائِب A L N AU A EY AI B …
After dictionary adaptation	After dictionary adaptation	After dictionary adaptation
… التابِع A T A B AI AE AI …	(ث)➔اثراء, (د)➔ادعم, (ذ)➔اذهب, (ر)➔ ارسالة, (س)➔اسودان, (ش)➔اشرق, (ص)➔ اصحافة, (ض)➔اضخمة, (ط)➔اطبي, (ظ)➔ الائحة ➔(ل) ,اظروف	… النَائِب A N AU A EY AI B …

التَّرشِح A L T ~ AU R AU SH ~ AI HH
التَّرَقُبْ A L T ~ AU R AU Q ~ AW B AW
التَّرَقُبِ A L T ~ AU R AU Q ~ AW B AI
التَّرَيُّثْ A L T ~ AU R AU Y ~ AW TH
التَّرَيُّثْ A L T ~ AU R AU Y ~ AW TH AW
التَّرْكِيَة A L T ~ AU Z K AI Y AU P AW
التَّزوير A L T ~ AU Z W AI Y R
التَّسجِيل A L T ~ AU S J AI Y L
التَّسجِيلْ A L T ~ AU S J AI Y L AU
التَّسجِيلْ A L T ~ AU S J AI Y L AW

Fig. 1 Some entries of the baseline dictionary

8 Experimental results

This section presents the experimental results based on the introduced MSA speech corpus. In this work, we used three emitting states of HMMs that corresponds to the subphones at the beginning, middle, and end of the phones. The acoustic models were calculated using context-dependent HMM triphones. Our acoustic models are all trained using the SphinxTrain for the phonetic tied-mixture (PTM) Pocket-Sphinx. The performance is measured based on different parameters, such as the number of Senones and the number of Gaussian densities. Word Error Rate (WER) was used to evaluate the ASR performance in investigating the different cases. We initially evaluated the performance using the phonemes set presented in Table 1. Regarding the phonemes set, we evaluated the performance for two cases (43 phonemes and 46 phonemes). Figure 1 shows some entries of the baseline dictionary.

We initially conducted the experiments without employing the phonemes (ـَا:AUA, ـُو:AWW, ـِي:AIY). In fact, we wanted to measure the impact of these phonemes on the overall performance. Then, for the best performing case, we repeated an experiment with employing the phonemes (ـَا:AUA, ـُو:AWW, ـِي:AIY) as shown in Table 7. Hence, there is a slight performance difference when combining the short vowels with the long vowels. It seems that the difference (0.5%) is small but not significant. For investigating the phonological rules, we used the baseline dictionary that has no (ـَا:AUA, ـُو:AWW, ـِي:AIY) phonemes. It is worthy to mention that the Phonemes set also includes one more phoneme, which is SIL to handle the silent cases at the beginning and the end of speech files.

This relatively low accuracy is reasonable since we used a small size corpus. Ideally, ASR requires 200–300 h speech corpus. The language models also require huge

Table 7 The baseline system performance

Experiment	Densities	Senones	WER (%)	Accuracy (%)
43 Phonemes (without AUA, AWW, and AIY)				
1	64	500	31.2	68.8
2	128	500	31.2	68.8
3	256	500	31.2	68.8
4	64	1000	31.0	69.0
5	128	1000	30.9	69.1
6	256	1000	31.3	68.7
7	64	2000	31.1	68.9
8	128	2000	31.1	68.9
9	256	2000	31.5	68.5
46 Phonemes (with AUA, AWW, and AIY)				
10	128	1000	30.4	69.6

Table 8 Rough WERs for a number of English corpora

Speech collection	Vocabulary	WER %
TI Digits	11 (zero-nine, oh)	0.5
Wall Street Journal read speech	5000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64,000+	10
Conversational telephone speech	64,000+	20

Author's personal copy

textual data (gigabytes of text) for reasonable performance. It is reported in the Training Acoustic Model for CMUSphinx (2017) that the WER for 10-h task should be around 10%. For a large task, it could be around 30%. Table 8 shows the WER for some ASR systems using different English speech corpora (Jurafsky and Martin 2009).

One more reason for the obtained relatively low accuracy is that the used corpus has no filler dictionary. The filler dictionary generally contains noise and inhalation speech that are appropriately handled during the training phase. The fillers require indicating the noises and inhalations in the transcription of the speech files, which is an extremely difficult task for our corpus. The output of this work is demonstrated in Table 9. Based on the obtained results, no clear evidence that employed phonological rules is of significant performance enhancement in the case of a within-word pronunciation variation model. We emphasize that this work is based on replacing the standard phonetic transcription in the baseline dictionary by the proposed phonetic transcription, as some well-known method is based on adding the variants while, at the same time, keeping the standard phonetic transcription.

9 The effect of diacritization

Diacritization is the process of marking the letters using optional orthographic symbols that are called diacritics (i.e. the short vowels). The Arabic formal text is generally written without diacritics, which produces different pronunciation forms. That is, the Arabic writing system allows discarding short vowels and, hence, forcing the reader to use the prior knowledge and the words context to infer the missing diacritics. For the Arabic ASR, the problem of short vowels is that they are generally pronounced, but almost never written, which adds more challenges to the learning process. The missing of short vowels may increase the ambiguity in the acoustic model and, hence, produces less than optimal performance. The study in (Vergyri and Krichhoff 2004) indicates that the non-diacritized text leads to problems for both acoustic and language modeling and therefore may lead to a loss in recognition accuracy. Similarly, it is reported in (Kirchhoff et al. 2002) that the missing of short vowels leads to a significant increase in both the language model perplexity and the word error rate.

The importance of diacritization is that it enhances the supposed match between the phonetic transcription of the training textual files and the corresponding speech files. In fact, it is extremely important that the phonemes of the pronunciation dictionary adequately represent the actual training speech. In the case of training using non-diacritized text, many of phonetic segments will be lost because the short vowels are not there. Despite short vowels that help the reader to realize the meaning of a particular word, not using fully diacritized text might lead to ambiguity as the same word might have several meanings. For instance, the word "جنة: J N P" has three different meanings based on the short vowels (u:ُ,a:َ,i:ِ) on the first letter: (جُنة, جَنة, جِنة) (J U N P, J A N P, J I N P) so it can mean protection, paradise, and jinn, respectively. More on Arabic diacritization and some other related challenges are found in (Al-Anzi and AbuZeina 2015). On other hand, obtaining a sizable diacritized text for ASR and NLP applications is extremely difficult as well as a time-consuming task.

In this section, we present the performance using non-diacritized text. The experimental results show that the non-diacritized text system scored 81.2% while the diacritized text based system scored 69.1%. The dictionary size in case of non-diacritized is 23,481 unique words, however, the dictionary size of the base line is 37,158. Even the diacritized case has less accuracy due to the slight differences in diacritics; however, the non-diacritized case might be adequate and faultless for the Arabic native speakers. Regarding the execution time of both the training and the decoding stages, we found that the non-diacritized case required less execution time due to the reduced vocabulary.

10 Conclusion and future work

This paper presents an experimental ASR performance evaluation using a set of Arabic phonological rules. We investigated three well-known rules for within-word

Table 9 The Performance of the phonological rules

Experiment	The phonetic change	Densities	Senones	WER (%)	Accuracy (%)
The Shadda rule					
1	Remove ~	128	1000	30.4	69.6
2	Duplicate the proceeding	128	1000	31.2	68.8
The Tanween Rule					
3	Change (ً,ٌ,ٍ) to N	128	1000	31.0	69.0
The Solar letter rule					
4	Remove L before Solar letters	128	1000	30.7	69.3

Author's personal copy

pronunciation variations. We conducted the experiments using a new continuous speech corpus that contains about 22.7 h of news transcription. The corpus was manually diacritized. The experimental results reveal that employing the phonological rules does not clearly enhance the ASR performance. We investigated three rules that include Shadda, Tanween, and the solar letters. We emphasize that we replaced, and did not add, the phonetic transcription according to the phonological rules. Accordingly, the output of this work pushes to rethink the importance of phonological rules in ASR (i.e. for within-word pronunciation variations modeling). Hence, we recommend devoting ASR research for the cross-word pronunciation variations modeling as well as for finding the optimal phonemes set of the Arabic. In addition to the phonological rules, this paper presents an experimental evaluation of diacritized and non-diacritized based text. The experimental results show that the non-diacritized based system outperforms the diacritized based system even with a smaller vocabulary. However, the diacritized based system gives vowelized text output, which is not obtained by a non-diacritized based system.

Acknowledgements This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

References

Abushariah, M. A.-A. M., et al. (2012). Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. *International Arab Journal of Information Technology, 9*(1), 84–93.

AbuZeina, D., et al. (2011). Toward enhanced Arabic speech recognition using part of speech tagging. *International Journal of Speech Technology, 14*(4), 419–426.

AbuZeina, D., et al. (2011). Cross-word Arabic pronunciation variation modeling for speech recognition. *International Journal of Speech Technology, 14*(3), 227–236.

AbuZeina, D., et al. (2012) Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach. *International Journal of Speech Technology, 15*(2), 65–75.

Akesson, J. (2010). *A study of the assimilation and substitution in Arabic*. Lund: Pallas Athena Distribution.

Al-Anzi, F. S., & AbuZeina, D. (2015). Stemming impact on Arabic text categorization performance: A survey. In *Proceedings of the 2015 5th international conference on information & communication technology and accessibility (ICTA)*, IEEE.

Alghamdi, M., Elshafei, M., & Al-Muhtaseb, H. (2007). Arabic broadcast news transcription system. *International Journal of Speech Technology, 10*(4), 183–195.

Al-Haj, H., Hsiao, R., Lane, I., Black, A., & Waibel, A. (2009) Pronunciation modeling for dialectal Arabic speech recognition, ASRU 2009: IEEE workshop, Italy.

Ali, M., Elshafei, M., Alghamdi, M., Almuhtaseb, H., & Alnajjar, A. (2009). Arabic phonetic dictionaries 236 for speech recognition. *Journal of Information Technology Research, 2*(4), 67–80.

Al-Sabah TV. (2017). http://www.alsabahpress.com/.

Benzeghiba, M., & De Mori, R. et al. (2007). Automatic speech recognition and speech variability: A review. *Speech Communication, 49*(10–11), 763–786.

Biadsy, F., Habash, N., & Hirschberg, J. (2009). Improving the Arabic pronunciation dictionary for phone and word recognition with linguistically-based pronunciation rules. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics.

Building Language Model. (2017). http://cmusphinx.sourceforge.net/wiki/tutoriallm.

CMU Sphinx Downloads. (2017). http://cmusphinx.sourceforge.net/wiki/download.

Elshafei, M.A. (1991) Toward an Arabic text-to-speech system. *The Arabian Journal for Science and Engineering, 16*(4B), 565–583.

Finke, M., & Waibel, A. (1997). Speaking mode dependent pronunciation modeling in large vocabulary conversational speech recognition. In *Proceedings of EuroSpeech-97* (pp. 2379–2382), Rhodes.

Fosler-Lussier, E., Greenberg, S., & Morgan, N. (1999) Incorporating contextual phonetics into automatic speech recognition. In *Proceedings of the international congress on phonetic sciences*, (pp 611–614).

Jeon, J., Cha, S., Chung, M., Park, J., & Hwang, K. (1998). Automatic generation of Korean pronunciation variants by multistage applications of phonological rules. In *ICSLP-1998* (paper 0675).

Jurafsky, D., Martin, J. (2009). *Speech and language processing*, 2nd edn. Hoboken: Pearson.

Kessens, J. M., Wester, M., et al. (1999). Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation. *Speech Communication, 29*(2–4), 193–207.

Kirchhoff, K., et al. (2002) Novel approaches to Arabic speech recognition-final report from the JHU summer workshop 2002. Technical Reports, John-Hopkins University.

Kyong-Nim, L. & Minhwa, C. (2007). Morpheme-based modeling of pronunciation variation for large vocabulary continuous speech recognition in Korean, *IEICE Transactions on Information and Systems, 90*(7), 1063–1072.

Liu, Y., & Fung, P. (2003). Modeling partial pronunciation variations for spontaneous Mandarin speech recognition. *Computer Speech and Language, 17*, 357–379.

Masmoudi, A., et al. (2014) A Corpus and Phonetic Dictionary for Tunisian Arabic Speech Recognition. In *LREC*.

Ramsay, A., Alsharhan, I., Ahmed H. (2014). Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model. *Computer Speech & Language, 28*(4), 959–978.

Seman, N., & Jusoff, K. (2008). Acoustic pronunciation variations modeling for standard Malay speech recognition. *Computer and Information Science, 1*(4), 112.

Tajchman, G., Foster, E., Jurafsky, D. (1995) Building multiple pronunciation models for novel words using exploratory computational phonology. In *EUROSPEECH-1995* (pp. 2247–2250).

Training Acoustic Model for CMUSphinx. (2017). http://cmusphinx.sourceforge.net/wiki/tutorialam.

Vergyri, D., et al. (2008) Development of the SRI/nightingale Arabic ASR system. Interspeech.

Vergyri, D., & Kirchhoff, K. (2004). Automatic diacritization of Arabic for acoustic modeling in speech recognition. In *Proceedings of the workshop on computational approaches to Arabic script-based languages*, Association for Computational Linguistics.

Wester, M. (2003). Pronunciation modeling for ASR: Knowledge-based and data-derived methods. *Computer Speech & Language, 17*, 69–85.

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:10, 2017

The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition

Fawaz S. Al-Anzi, Dia AbuZeina

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/Publication/10008047

***Abstract*—**Speech recognition is of an important contribution in promoting new technologies in human computer interaction. Today, there is a growing need to employ speech technology in daily life and business activities. However, speech recognition is a challenging task that requires different stages before obtaining the desired output. Among automatic speech recognition (ASR) components is the feature extraction process, which parameterizes the speech signal to produce the corresponding feature vectors. Feature extraction process aims at approximating the linguistic content that is conveyed by the input speech signal. In speech processing field, there are several methods to extract speech features, however, Mel Frequency Cepstral Coefficients (MFCC) is the popular technique. It has been long observed that the MFCC is dominantly used in the well-known recognizers such as the Carnegie Mellon University (CMU) Sphinx and the Markov Model Toolkit (HTK). Hence, this paper focuses on the MFCC method as the standard choice to identify the different speech segments in order to obtain the language phonemes for further training and decoding steps. Due to MFCC good performance, the previous studies show that the MFCC dominates the Arabic ASR research. In this paper, we demonstrate MFCC as well as the intermediate steps that are performed to get these coefficients using the HTK toolkit.

***Keywords*—**Speech recognition, acoustic features, Mel Frequency Cepstral Coefficients.

I. Introduction

ASR is an attractive user-friendly technology to felicitate human computer interface (HCI) in different domains. In the last years, there has been a growing interest to reinforce natural man-machine communication through speech technology. In this regard, much research has been devoted to introduce innovative ideas in the industry for automation purpose (e.g. banking services, cars, control machines, etc.). In general, sound is made out of vibrations of an object to generate a type of energy. The energy causes a movement in the air particles that propagate as audible waves. The air particles movement keeps going until they run out of energy. Humans can hear sound waves with frequencies between about 20 Hz (cycles per second) and 20 kHz. However, the most sensitive limit of human hearing is in the 2000 - 5000 Hz frequency range. In general, machine-learning systems perform feature extraction process at the first place in order to produce the feature values based on the input patterns, these speech features are then pass to an ASR system.

MFCC is the classical front-end analysis in speech recognition to produce the sequence of real-valued numbers that represent feature vectors based on the input signal. Since 1980, it has dominated the ASR feature extraction methods due to its good performance. The success of MFCC makes it the standard choice in the state-of-the-art speech recognizers such as the CMU Sphinx [1], the HTK [2], and the Kaldi speech recognizer [3]. The literature shows that there is a variety of feature extraction methods; however, it is clearly observed that MFCC is extensively used in the most speech classification tasks. An example of another feature extraction method is Perceptual Linear Prediction (PLP) [4]. In fact, previous studies show that MFCC is an appropriate choice to maximize the recognition performance as reported by [5]. It indicates that the MFCC is characterized by better performance and ability of the frequency domain to model adequately the sound. Reference [6] indicated the MFCC and the relative spectral analysis PLP are the most commonly used due to their ability to provide more robust features in adverse conditions. Similarly, Reference [7] demonstrated that the most of today's ASR systems are based on some types of MFCC, which have proven to be effective and robust under various conditions.

The rest of this paper is organized as follows. In the next section, we present some of the challenges of speech features. In Section III, we present the background of MFCC technique followed by the literate review in Section IV. Finally, we conclude in Section V.

II. Speech Features Challenges

Due to the difficulties of handling speech features, it has been long observed that ASR researches employ off-the-shelf toolboxes for features extraction. It is clear that employing MFFC, or even other speech features, for speech applications is not a straightforward task since some of the intermediate functions are difficult for non-specialist researchers. For instance, writing a program for fast Fourier transform (FFT), which is the heart of computing MFCC, requires highly qualified scientists or engineers who have a solid background in complex mathematics, and then, can understand and write FFT program from scratch. No doubt, conducting valuable research that includes speech processing (e.g. speech recognition or speech synthesis) requires deep understanding of signal processing. Speech features pose some challenges in terms of the nature of the data. For instance, textual data or even images features are constant, which remain fixed wherever they appear. To clarify, the features of an article (i.e. the words or the roots are always the same for a particular text; however, speech features are not constant as they are continuously changed according to different aspects such as

Fawaz S. Al-Anzi and Dia AbuZeina are with the Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait (e-mail: fawaz.alanzi@ku.edu.kw, dia.abuzeina@ku.edu.kw).

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/Publication/10008047

gender, accent, and age, etc. Simply, it is hard to directly compare speech features due to the (small) differences in vibrations that lead to completely different sounds. The speech-recording environment might have noise such as background music, a second speaker, unwanted breathing, and be affected by the quality of the microphone, or the health and psychological state of person. Reference [8] has a thorough study of the pronunciation variations sources that degrade the performance of ASR systems. In fact, humans can easily interpret signals by extracting relevant information; however, this task is more complex when performed using signal processing and machine learning algorithms. More problems can be observed regarding the speech context. Sounds are quite substantially changed by the surrounding context. The vocal tract goes through different stages getting from 't' to 'a' and getting from 'r' to 'a', and the parameters during the transition will be different as indicated in [9]. Moreover, sounds can last different amounts of time. Deciding where one ends and the next one starts is hard. Moreover, the speech extraction process is a tricky task that requires care and skill. The input waveform is sliced up into frames (usually of 20~30 milliseconds) to generate speech spectrum, which is the distribution of energy as a function of frequency for a particular sound source [10]. Therefore, the waveform is transformed into spectral features (i.e. acoustic feature vectors), as shown in Fig. 1. The figure is obtained from reference [11], which has more details of speech and language processing. For general overview of the difficulty to handle speech recognition, reference [12] elaborates on some of the difficulties with ASR.

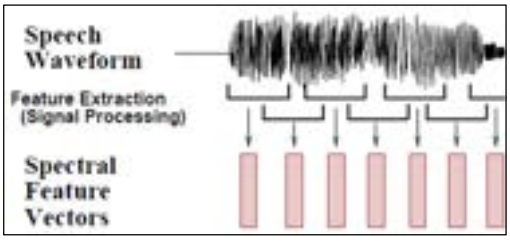


Fig. 1 Extracting features by dividing the signal into frames

III. MFCC Background

To compute the MFCC, the time domain representation of the input speech signal is used to produce the spectral properties, as the patterns are more evident in the frequency domain. The MFCC consists of a set (39 coefficients) that represents the speech signal by dividing it to a set of overlapping short segments called frames. In particular, MFCC coefficients represent the spectral envelope of the speech signal on the Mel-frequency scale. Fig. 2 shows the steps to extract the MFCC of a speech signal. For better performance, the temporal properties might be considered to obtain the first and the second derivative (named respectively ΔMFCC and ΔΔMFCC) of the first order 13 coefficients. We emphasize that the first step, which is sampling and quantization, is performed by the sound card (i.e. a hardware related issue) and is not a part of the MFCC process. However, it is shown in the figure as an indication of the nature of the input data for the Pre-emphasis stage. The goal of the sampling and quantization (also called digitization) step is to convert the analog signal to digital forms for further processing. The sampling rate is the number of samples taken per second, while quantization is the process of representing real-valued numbers as integers. It is worthy to indicate that the MFCC process is not invertible; it is impossible to get the signal back from the set of MFCCs.

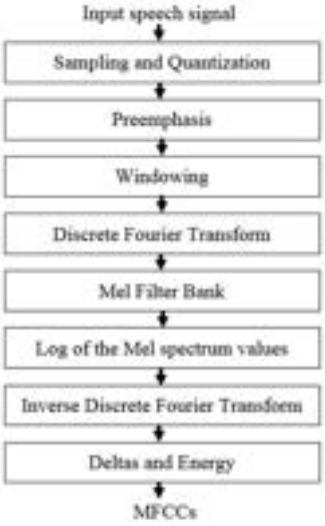


Fig. 2 Extracting features using MFCC algorithm

Reference [13] highlighted some reasons of MFCC popularity in parametric representation of the spectrum as follows. First, the calculation of these parameters leads to a source-filter separation. Second, the parameters have an analytically tractable model. Third, experience proves that these parameters work well in recognition applications. The following is a brief description of the tasks to extract the speech features:

Pre-emphasis: Pre-emphasis is performed after the digitization step. It aims at increasing the amplitude of high frequency bands and decreases the amplitudes of lower bands. That is, this stage is to attain the high frequency formants that carry the relevant information. Without Pre-emphasis, it might be difficult for the receiver to interpret the signal due to the suppression during the sound production mechanism. Hence, the purpose of Pre-emphasis is to apply to the signal with the proper weight sometimes called alpha. The Pre-emphasis is also considered as noise reduction module as it leaves the desired signal untouched, but reduces the noise power considerably.

Windowing: The pre-emphasized speech signal is subjected to the short-time Fourier transform analysis with frame durations of 20-30 ms, frame shifts overlap of around 10 ms. In this stage, the speech signal is analyzed to extract the stationary portion of speech using a window function, which can be characterized by minimizing the discontinuities of the signal.

Discrete Fourier Transform: This stage is the basis of spectral analysis to extract the speech features based on magnitude spectrum computation. It is performed by decomposing an N point time domain signal to obtain the

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/Publication/10008047

magnitude frequency response of each frame. That is, it calculates the N frequency spectra corresponding to the N time domain signals

Mel Filter-bank: Computing the Mel frequency spectrum is performed after the discrete Fourier transform by passing the spectrum through Mel filters to obtain Mel spectrum. To produce the filter-bank energies, a number of triangular filters are used that are uniformly spaced on the Mel scale between the lower and upper frequency. It is used to approximate the frequency resolution of the human ear. That is, the Mel scale approximates the sensitivity of the human ear. The Mel frequency scale is linear up to 1000 Hz and logarithmic thereafter.

Log of the Mel spectrum values: Mel filter-bank is used to generate a range of natural logarithm values and replacing the original values by this range. It is an approximation of the spectrum to the Gaussian statistical distribution.

Inverse Discrete Fourier Transform: A set of low order coefficients is compressed in this transform and used to convert spectral information. This is called the Mel cepstrum representation.

Deltas and Energy: The previous step provides the 12 cepstral coefficient for each frame. This step is to add the 13th feature: the energy from the frame. It is useful to identify phone identity.

To explain the output of the MFCC algorithm, we used a small speech file that contains a single Arabic word "*as'hum*" that means "stocks". The speech waveform of this word is shown in Fig. 3. In addition, the spectrogram of this word is shown in Fig. 4. The spectrogram is a visualization tool that is used to understand the information in the signal using time and frequency. Acoustic phones and their properties are better observed in spectrogram. The spectrogram representation of the speech signal is based on short-time Fourier analysis. In the spectrogram, if gray scale is used, the higher the amplitude (the energy), the darker the corresponding region; however, if a color scale is used, the blue represents the low energy, while the red parts represent high energy [14].

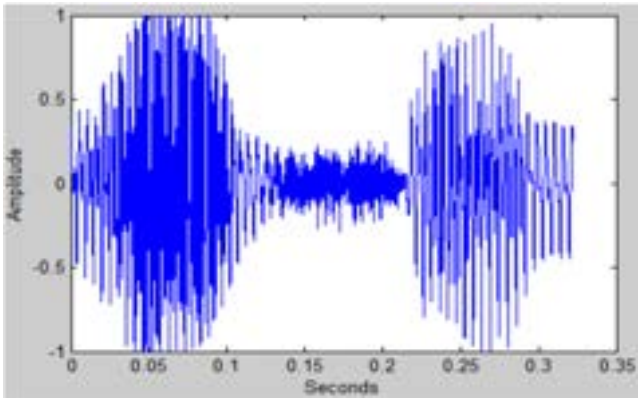


Fig. 3 A speech signal waveform of the Arabic single word "*as'hum*"

The HTK system was used to extract the MFCC speech features of the single word speech file that is represented in Fig. 3. The speech file is of length 0.323 seconds and uses a sampling rate of 16 kHz with 16-bit quantization for each sample. Table I shows the first 12-order of the MFCC coefficients after completing the feature extraction process. Each column represents the 13 features (the 13th feature is the energy from the frame) of a 25 milliseconds frame.

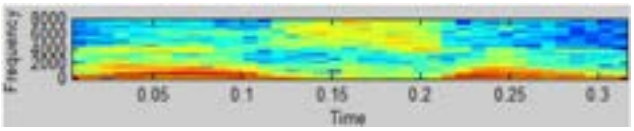


Fig. 4 The spectrogram of a single Arabic word speech file "*as'hum*"

TABLE I
MFCC OF A SINGLE WORD SPEECH FILE

	Frame				
Feat.	1	2	3	4➔29	30
1	-1.81965	-3.15548	-3.76447	…	1.40033
2	-3.10861	-7.92128	-9.68467	…	1.19619
3	1.95010	-2.75036	-4.08556	…	2.44150
4	-12.21996	-14.16043	-15.84370	…	-8.50239
5	-8.21085	-10.14035	-13.00282	…	-11.45462
6	-14.98533	-13.45490	-17.62610	…	-0.93472
7	-22.24395	-23.61402	-13.59459	…	-10.85606
8	2.53291	-1.30409	10.01444	…	-4.10245
9	-8.75291	-14.95345	-3.51132	…	-15.78435
10	-7.62615	-4.33472	-2.60953	…	-13.07826
11	-4.17761	-6.58369	-8.04277	…	3.56489
12	-8.05171	-7.96873	-10.30831	…	-7.61198
13	80.21140	83.59767	86.78981	…	76.75751

TABLE II
TYPICAL HTK SETTINGS – CONFIGURATION FILE

Coding parameters	Comments
SOURCEFORMAT = WAV	The format of the source file
TARGETKIND = MFCC_0	Cepstral C_0 coefficient appended
TARGETRATE = 100000.0	10ms frame rate
SAVECOMPRESSED = T	Save the output file in compressed form
SAVEWITHCRC = T	Attach a checksum to output parameter file
WINDOWSIZE = 250000.0	25ms window
USEHAMMING = T	Use a Hamming window
PREEMCOEF = 0.97	Set pre-emphasis coefficient
NUMCHANS = 26	Number of filter-bank channels
CEPLIFTER = 22	Cepstral filtering coefficient
NUMCEPS = 12	Number of cepstral parameters

The extracted MFCC speech features shown in Table I were extracted using the HTK-HCopy command and the default parameters [2], as shown in Table II. A configuration file (generally called config) is needed which specifies all of the conversion parameters. The HCopy command is used as the following, supposing that the input speech file is "sample.wav":

```
HCopy –C config.txt sample.wav sample.mfc
```

However, the HCopy command creates a binary file (special format non-text file) that contains the MFCC data. Therefore, another option to obtain the MFCC data in textual form is by using HTK-HList command as the following:

```
HList –C config.txt –r sample.wav
```

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/Publication/10008047

IV. Literature Review

Based on a thorough review of Arabic speech recognition literature, it is observed that MFCC is extensively used in most studies of Arabic ASR. Table III shows some of the previous studies. However, some of the studies employ other feature extraction methods such as the first work in Table III, in which the LPCC is the shorthand of linear prediction spectrum coefficients, which is one of the famous speech features extraction method. As illustrated, the information in the table belongs to two main categories of speech recognition; isolated and continuous speech recognition. Table III also reveals that Arabic speech recognition is in row stages as most of works depend on off-the-shelf tools (MFCC-based tools), which reduce the opportunities to investigate different speech features as well as reduce the opportunity to present innovative ideas (i.e. featuring new methods).

TABLE III
Previous Studies Employing MFCC

Isolated speech (digits or control command)		
Reference	**Year**	**Features**
[15]	2001	LPCC
[16]	2003	MFCC
[17]	2003	MFCC
[18]	2006	MFCC
[19]	2007	MFCC
[5]	2007	MFCC
[20]	2008	MFCC
[21]	2008	MFCC
[22]	2009	MFCC
Continuous speech		
Reference	**Year**	**Features**
[23]	2007	MFCC
[24]	2008	MFCC
[25]	2010	MFCC
[26]	2011	MFCC
[27]	2011	MFCC
[28]	2012	MFCC
[29]	2012	MFCC
[30]	2017	MFCC

V. Conclusion

This paper demonstrates the MFCC speech features extraction method as one of the most commonly used in ASR systems. Compared to other speech features extraction methods, MFCC is the standard choice for front-end features in state-of-the-art ASR systems. According to our best knowledge and the review that we performed on the previous studies of Arabic ASR, we found that MFCC dominates the works in this field. We employed the HTK system to demonstrate the extraction process of MFCC speech feature vectors of a simple speech file. As a future work, it is worth to continue this work by conducting a practical research to compare MFCC with other methods such as LPCC and PLP.

References

[1] https://cmusphinx.github.io/wiki/faq/ Accessed on 31 March 2017.
[2] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.
[3] Povey, Daniel, et al. "The Kaldi speech recognition toolkit."IEEE 2011 workshop on automatic speech recognition and understanding. No. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.
[4] Hermansky, Hynek. "Perceptual linear predictive (PLP) analysis of speech." The Journal of the Acoustical Society of America 87.4 (1990): 1738-1752.
[5] Haraty, Ramzi A., and Omar El Ariss. "CASRA+: a colloquial Arabic speech recognition application." American Journal of Applied Sciences 4.1 (2007): 23-32.
[6] Sharma, Davinder Pal, and Jamin Atkins. "Automatic speech recognition systems: challenges and recent implementation trends." International Journal of Signal and Imaging Systems Engineering 7.4 (2014): 220-234.
[7] Molau, Sirko, et al. "Computing mel-frequency cepstral coefficients on the power spectrum." Acoustics, Speech, and Signal Processing, 2001. Proceedings. (ICASSP'01). 2001 IEEE International Conference on. Vol. 1. IEEE, 2001.
[8] Benzeghiba, Mohamed, et al. "Automatic speech recognition and speech variability: A review." Speech communication 49.10 (2007): 763-786.
[9] Ramsay, Allan. "How Do Speech Recognisers Work?" A presentation. Kuwait Univeristy (2016).
[10] http://www.thefreedictionary.com/ Accessed on 31 March 2017.
[11] Jurafsky, Dan. Speech & language processing. Pearson Education India, 2000.
[12] Forsberg, Markus. "Why is speech recognition difficult." Chalmers University of Technology (2003).
[13] Alcaraz Meseguer, Noelia. Speech analysis for automatic speech recognition. MS thesis. Institutt for elektronikk og telekommunikasjon, 2009.
[14] Huang, Xuedong, et al. Spoken language processing: A guide to theory, algorithm, and system development. Prentice hall PTR, 2001.
[15] Bahi, Halima, and Mokhtar Sellami. "Combination of vector quantization and hidden Markov models for Arabic speech recognition." *Computer Systems and Applications, ACS/IEEE International Conference on. 2001*. IEEE, 2001.
[16] Elmisery, F. A., et al. "A FPGA-based HMM for a discrete Arabic speech recognition system." *Microelectronics, 2003. ICM 2003. Proceedings of the 15th International Conference on*. IEEE, 2003.
[17] Amrouche, Abderrahmane, and J. Michel Rouvaen. "Arabic isolated word recognition using general regression neural network." *Circuits and Systems, 2003 IEEE 46th Midwest Symposium on*. Vol. 2. IEEE, 2003.
[18] Bourouba, H., et al. "New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition." *Information and Communication Technologies, 2006. ICTTA'06. 2nd*. Vol. 1. IEEE, 2006.
[19] Satori, Hassan, Mostafa Harti, and Nouredine Chenfour. "Introduction to Arabic speech recognition using CMUSphinx system." *arXiv preprint arXiv:0704.2083* (2007).
[20] Essa, E. M., A. S. Tolba, and S. Elmougy. "A comparison of combined classifier architectures for Arabic Speech Recognition." *Computer Engineering & Systems, 2008. ICCES 2008. International Conference on*. IEEE, 2008.
[21] Azmi, M., et al. "Syllable-based automatic arabic speech recognition in noisy-telephone channel." *WSEAS Transactions on Signal Processing* 4.4 (2008): 211-220.
[22] Satori, Hassan, et al. "Investigation arabic speech recognition using CMU sphinx system." *Int. Arab J. Inf. Technol.* 6.2 (2009): 186-190.
[23] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." *International Journal of Speech Technology* 10.4 (2007): 183-195.
[24] Alotaibi, YousefAjami, Sid-Ahmed Selouani, and Douglas O'shaughnessy. "Experiments on automatic recognition of nonnative Arabic speech." *EURASIP Journal on Audio, Speech, and Music Processing* 2008.1 (2008): 679831.
[25] Selouani, Sid Ahmed, and Malika Boudraa. "Algerian Arabic speech database (ALGASD): corpus design and automatic speech recognition application." Arabian Journal for Science and Engineering 35.2C (2010): 158.
[26] Abu Zeina, Dia, et al. "Toward enhanced Arabic speech recognition using part of speech tagging. "*International Journal of Speech Technology* 14.4 (2011): 419-426.
[27] Abu Zeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition. "*International Journal of Speech Technology* 14.3 (2011): 227-236.

[28] Abu Zeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: a direct data-driven approach." *International Journal of Speech Technology* 15.2 (2012): 65-75.
[29] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." *International Arab Journal of Information Technology (IAJIT)* 9.1 (2012): 84-93.
[30] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology (2017): 1-9.

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/Publication/10008047

Submitted Paper(s)

AMCS: Paper submission

A

AMCS <amcs@uz.zgora.pl>

Reply all

Today, 12:09

Fawaz Alanzi;
Jozef Korbicz <J.Korbicz@issi.uz.zgora.pl>

Inbox

Dear Author: Fawaz Al-Anzi,

your paper:

First name: Fawaz
Last name: Al-Anzi
Co-authors: Dia AbuZeina
Street: Kuwait University
Organisation: Computer Engineering
Department: PO Box 5969 Safat
ZIP/Post code: 13060
City: Kuwait City
Country: KUWAIT
Phone number:
Fax:
E-mail: Fawaz.alanzi@ku.edu.kw
Paper title: Theoretical and Practical Models for Arabic Speech Recognition
Abstract: Large-vocabulary speaker-independent continuous speech recognition systems have recently received significant attention. Although considerable research has been devoted to English speech recognition, less attention has been paid to the Arabic speech recognition. This paper aims to highlight the achievements that have been made during the last several decades on Arabic speech recognition. The paper also discusses speech recognition components, such as corpora, phonemes, language models, acoustic models, and performance evaluation. Hidden Markov models (HMMs) and the Viterbi algorithm are also discussed. For empirical evaluation of Arabic speech recognition, the free, off-the-shelf Mac Soundflower tool was employed to evaluate the recognition performance using a continuous speech corpus that contains 2.63 hours (by 10 male and 10 female speakers) of modern standard Arabic (MSA) broadcast news. The experimental results show that the recognition accuracy is 54.02%, and the accuracies for the male and female speakers are almost the same.

---

has been submitted. Thank you.

The paper will be assessed and you will be hearing from me regarding its possible publication in the International Journal of Applied Mathematics and Computer Science (AMCS) in due course.

You can track the status of your paper through the paper information page: https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.amcs.uz.zgora.pl%2Fpaper.php%3Fpaper%3Deo2wS7eqX5a5486dbdf26cXYrBeEzZ&data=02%7C01%7CFawaz.alanzi%40ku.edu.kw%7Cd9c348ebc3874584a71008d55740bc20%7Cf9258092e3624609bea875884d326920%7C0%7C1%7C636510857913868993&sdata=0cZQinccJm%2FuiYN801o8awPM1zzkuB%2BCcaXok4pbEJs%3D&reserved=0

Yours sincerely,

Józef Korbicz
Professor, Editor-in-Chief

--
International Journal of Applied Mathematics & Computer Science
https://emea01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fwww.amcs.uz.zgora.pl&data=02%7C01%7CFawaz.alanzi%40ku.edu.kw%7Cd9c348ebc3874584a71008d55740bc20%7Cf9258092e3624609bea875884d326920%7C0%7C1%7C636510857913868993&sdata=rrcGH4SjZMTwE78jZq%2FBGfQhlFnjEHMqYQNfCgwoyNA%3D&reserved=0
PRESENT YOUR RESEARCH WITH US!

PLEASE RETAIN THIS E-MAIL FOR FUTURE REFERENCE!

Theoretical and Practical Models for Arabic Speech Recognition

Fawaz S. Al-Anzi and Dia AbuZeina

Department of Computer Engineering, Kuwait University

fawaz.alanzi@ku.edu.kw; abuzeina@ku.edu.kw

Abstract: Large-vocabulary speaker-independent continuous speech recognition systems have recently received significant attention. Although considerable research has been devoted to English speech recognition, less attention has been paid to the Arabic speech recognition. This paper aims to highlight the achievements that have been made during the last several decades on Arabic speech recognition. The paper also discusses speech recognition components, such as corpora, phonemes, language models, acoustic models, and performance evaluation. Hidden Markov models (HMMs) and the Viterbi algorithm are also discussed. For empirical evaluation of Arabic speech recognition, the free, off-the-shelf Mac Soundflower tool was employed to evaluate the recognition performance using a continuous speech corpus that contains 2.63 hours (by 10 male and 10 female speakers) of modern standard Arabic (MSA) broadcast news. The experimental results show that the recognition accuracy is 54.02%, and the accuracies for the male and female speakers are almost the same.

Keywords: Arabic; speech recognition; phoneme; language model; acoustic model; hidden Markov models

1 Introduction

Automatic speech recognition (ASR) is the process of converting spoken language (i.e., acoustic speech signal) into a machine-readable text. Hence, the interpretation of human speech by a machine-learning algorithm is the goal of ASR systems. Currently, the fast growth of technology is making man-machine interfaces increasingly useful and pervasive. However, speech recognition is not an easy task and there is a long way to go to efficiently utilize speech recognition to fulfil people needs. Despite the notable ASR research, perfect speech-to-text conversion is still an unrealized vision. Speech processing is much complicated than other pattern recognition problems such as text classification and image recognition. While great success has been achieved in finding a particular word in a textual collection or classifying an image in a database, locating a particular speech segment in a speech file is still an active research area. Speech recognition is a multidisciplinary field, which includes machine learning, phonetics, linguistics, signal processing, and statistics. Accordingly, significant integration is required for positive outcomes.

Although significant research has been devoted to English ASR, less attention has been paid to Arabic ASR. The Arabic language is one of the most commonly used languages worldwide and is in need of precise speech-to-text converters. However, Arabic ASR research is still behind compared to ASR research in other languages such as English. In addition to the intrinsic challenges of Arabic, other logistical challenges are encountered,

Copyright © 201x *xxxxxxxx xxxxxxxx Ltd*.

such as resource availability. The absence of a unified large continuous speech corpus is an obstacle that might limit the research in this promising domain. Almost all Arabic speech recognition studies are conducted using in-house small corpora. In contrast, English ASR research has many common large corpora, such as North America business (NAB) and Broadcast News switchboard (Rabiner and Schafer, 2007). Working on a common corpus saves time and leads to gradual research enhancement since the outputs are compared and improved. It is known in the speech recognition community that creating a large speech corpus is a time-consuming and extremely expensive task. Consequently, it is difficult and inconvenient for individuals to perform this task. Furthermore, (Elmahdy, 2009) indicated that preparing large training corpora for dialectal Arabic acoustic modeling is more difficult compared to Modern Standard Arabic (MSA).

Although isolated-word speech recognition is an important task for certain applications, continuous and conversational speech recognition is of more interest. Continuous and conversational ASR technology would make it possible to reduce the amount of human-to-human communication and, therefore, reduce the number of employees. For several years, two well-known ASR engines have been used for speech-to-text conversion tasks: Carnegie Mellon University (CMU) Sphinx and the Cambridge University Hidden Markov Model Toolkit (HTK). Both are statistic-based engines that are based on Hidden Markov Models (HMMs). Developing a speech recognition engine, starting from scratch, is a complex task and requires qualified programming experts; therefore, most researchers use free (i.e., black box) research ASR engines, such as Sphinx and HTK. However, several other techniques have been used in the literature, such as artificial neural networks (ANNs) and support vector machines (SVMs). The Sphinx engine supports Arabic scripts. However, the HTK accepts as input only Roman letters. Therefore, for Arabic ASR, a transliteration process, such as that available in (ARABIC TRANSLITERATION,2016), is required. Fig. 1 shows the architecture of a typical ASR engine, which includes three knowledge databases: the acoustic model, which contains the trained HMM; the language model, which represents the statistical word co-occurrences; and the dictionary (also called pronunciation dictionary or vocabulary), which contains the pronunciation of each word in terms of phonemes, which are the basic distinct units of sounds. The pronunciation of a word is also called the phonetic transcription.

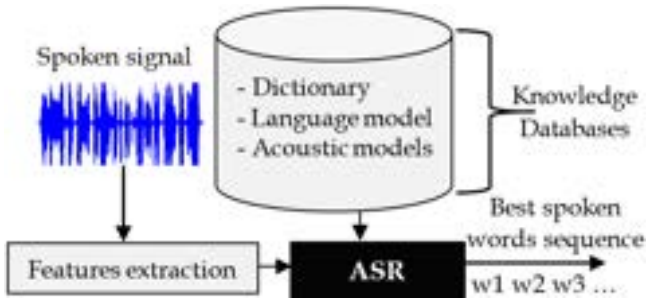


Fig. 1. Automatic Speech Recognition (ASR) Architecture

The goal of this paper is to present the recent advances in Arabic ASR, with a particular focus on the essential components of a typical ASR. The topics include the corpora that are used for both isolated words and continuous speech. The corpus information includes the

vocabulary size, nature of the data, topics, and numbers of male and female speakers and their ages. The paper also explores other topics, such as feature extraction methods, classification approaches, phoneme sets, pronunciation dictionaries, language models, and acoustic models. ASR mainly uses the Viterbi algorithm to map between acoustic model parameters and the language phoneme set. Then, it uses this information to find the most likely phoneme sequences, words, and sentence. An explanation of how the Viterbi algorithm is employed, along with HMM matrices, is provided. An empirical study is also presented for Arabic ASR using the free, off-the-shelf Soundflower tool.

The rest of this paper is organized as follows: The next section presents some speech recognition applications, followed by the Arabic speech corpora in section 3. Section 4 presents the Arabic phoneme sets. In section 5, we present language models, followed by acoustic models in section 6. Section 7 presents performance evaluation of ASR and an empirical study in section 8. Finally, we present the study's conclusions in section 9.

2 Speech Recognition Applications

Commercial speech recognizers have provided successful business interactive solutions in various industrial sectors, such as healthcare, telecommunication, banking and finance, retail and mall management, education, hospitality, governmental institutions, and travel (Emerging Technologies, 2016). In the last decade, great effort has been made to include ASR in new technologies such as search engines, voice maps, and communications. Recently, speech recognition has been used in security and protection fields for authentication purposes (also called voiceprints). Findbiometrics lists five unique applications of voiceprints: targeting of developers, hands-free interfaces, call center authentication, proof of life, and multi-factor logical access control (Voice Month, 2016). Nuance lists some benefits of using voiceprints, which include simpler authentication, fraud prevention, and almost-instant return on investment (Nuance, 2016). Although the utilization of voiceprints in the banking industry is not new, this technology has only recently been transferred to Arab countries. For example, Kuwait Finance House (KFH) uses a speech recognition platform for user authentication. This service was initially only available to VIP customers and has recently been made available to KFH's entire customer base, (Speech Recognition Case Study, 2016). Abu Dhabi Commercial Bank (ADCB) has also adopted voiceprints (ADCB, 2016). Remarkably, the communication infrastructure that already exists helps to spread the voiceprint biometric. Installing a new machine, such as image scanner for reading fingerprints to be sent for authentication, is not necessary. Primarily, main goal of ASR is to allow natural interaction between the user and the system through a series of queries and responses.

The utilization of speech recognition in natural language processing (NLP) and linguistic applications has been globally widespread. Arabic is no exception. There have been efforts to utilize this technology for the Arabic language, which has more than 380 million speakers (Mubarak et al., 2014). Importantly, the Holy Quran, which was revealed in Arabic, reinforces the interest in ASR research. The Holy Quran must be read in Arabic, which makes this technology very important for serving millions of Holy Quran learners. (Mohammed et al., 2015) presented a speech recognition technique for verification of Quranic recitation in sound files and media. (Jamaliah et al., 2013) provided a structural overview of a speech recognition system for Quranic verse recitation recognition with the Tajweed rule-checking function. (El Amrani et al., 2016) investigated the use of a

simplified set of Arabic phonemes in the application of an Arabic Speech Recognition system to the Holy Quran. The CMU Sphinx 4 was used to train and evaluate a language model for the Hafs narration of the Holy Quran.

3 Arabic Speech Corpora

Conducting ASR research requires the identification of the type of speech as either isolated words or continuous speech. Isolated-word speech (also called discrete-word speech) contains pauses between digits or words; this constraint is not imposed on continuous or conversational speech. Isolated-word speech recognition is characterized by straightforward implementation compared to continuous speech recognition, which suffers from the co-articulation phenomenon, which is the critical factor for continuous speech performance. In general, other factors are known to reduce ASR performance, such as varying acoustic conditions, pronunciation variations, dialects, accents, and ages, which must be considered when preparing a continuous speech corpus. (Strik and Cucchiarini, 1999) presented various factors that affect the way in which words are pronounced, such as assimilation, co-articulation, reduction, deletion, and insertion. The Arabic language has even more challenges, such as morphological complexity and diacritization, wherein short vowels are usually missed in formal writings. (Al-Anzi and AbuZeina, 2015) presented some Arabic language challenges.

3.1 Isolated-word speech recognition

In this subsection, we present some isolated-word speech corpora, speech features, and classifiers. In the literature, isolated-word size is generally represented using the number of recorded speech files, while a continuous-speech size is represented using the number of hours (the sum of the lengths of all recorded speech files). Table 1 summarizes some isolated-word ASR research. It contains information regarding corpora, Mel-Frequency Cepstral Coefficient (MFCC) and Linear Predictive Cepstral Coefficient (LPCC) features, and HMM-based classifiers.

Table 1 Isolated-word ASR research

Reference	Corpus Information, Features, Classifier/s
(Bahi and Sellami,2001)	The training set is composed of 50 speakers, each of whom uttered ten digits three times. The test set is comprised of two groups of 30 speakers and 10 speakers. LPCC features are used.
(Alimi and Jemaa, 2002)	Corpus information is not available. They used fuzzy neural networks for recognition of isolated words.
(Elmisery et al., 2003)	Speech corpus consists of the 10 isolated digits, with 20 repetitions for each digit, by a single male speaker. LPCC and MFCC features are used.
(Amrouche et al., 2003)	The corpus contains of a total 1800 digits, which are pronounced by 60 speakers (30 males and 30 females). For testing, they used 1000 digits, which are pronounced by 50

	other speakers (25 males and 25 females). An ANN classifier was used.
(Bourouba et al., 2006)	The corpus consists of 92 speakers (46 male and 46 female), who pronounce each word twice, where 20/92 of the corpus is used for learning. (HMM, SVM) classifiers were used.
(Hyassat and Abu Zitar, 2006)	The corpus contains approximately 1.5 hours of commands and less than 1 hour of digits.
(Satori et al., 2007)	Training set: 300 tokens (10 digits * 5 repetitions * 6 Moroccan speakers). Testing set: 30 tokens of different individuals.
(Haraty and El Ariss, 2007)	128 words for training and another 7 words for testing. Dynamic time warping (DTW) is used as a similarity measure for classification.
(Essa et al., 2008)	The corpus consists of 600 utterances (10 speakers, 10 words, 6 repetitions), which are split into 300 utterances for training and 300 utterances for testing. Neural networks are used for recognition.
(Azmi et al., 2008)	The corpus contains 59 Egyptian men (33 speakers for training and 26 for testing). Speakers are asked to utter 16 sentences of proverbs.
(Alotaibi, 2008)	The training set contains 340 tokens (17 speakers × 2 repetitions × 10 digits). For testing, 1,700 tokens (17 speakers × 10 repetitions × 10 digits) were used. ANN was used for classification.
(Satori et al., 2009)	The corpus was created from all 10 Arabic digits. Sixty Moroccan speakers (35 males and 25 females) were asked to utter all digits 5 times.
(Al-Qatab et al., 2010)	The corpus contains 3650 speech files, which were recorded by 13 speakers. The training set contains 3000 speech files, of which 650 were used for testing.

3.2 Continuous speech recognition

Less progress has been made on Arabic continuous speech recognition than in the isolated-word case because of the previously discussed difficulty regarding obtaining a continuous speech corpus. However, considerable work has been initiated by the Linguistic Data Consortium (LDC). The LDC is an open consortium of universities, libraries, corporations and government research laboratories (About LDC, 2016). More information about LDC Arabic speech corpora can be found in the LDC catalogue (Linguistic Data Consortium, 2016), which contains hundreds of (not free) holdings. One important contribution Arabic speech recognition research is the work of IBM in the Gale project, which used LDC corpora. The Gale project has many phases that gradually improve the performance. The Gale acoustic training set is composed of approximately 1800 hours of transcribed Arabic broadcasts and is provided by LDC. The published works that describe the phases include the IBM 2006 Gale Arabic ASR system (Soltau et al. 2007), the IBM 2009 GALE Arabic speech transcription system (Kingsbury et al., 2011), and the IBM 2011 GALE Arabic

speech transcription system (Mangu et al., 2011). LDC also produced the CallHome (CH) corpus of Egyptian colloquial Arabic (ECA) (CALLHOME, 2016). This corpus is a collection of informal phone conversations between close friends or family members. This corpus contains many sets, as shown in Table 2 (Kirchhoff et al. 2006).

Table 2 LDC ECA CallHome datasets

collection	# of conversations	# of words	# of hours
train	80	146,298	14
dev	20	32,148	3.5
eval96	20	15,584	1.5
eval97	20	17,670	1.8
h5_new	20	16,752	1.8
eval03	10	11,015	1.9

In speech recognition systems, it is highly recommended and more accurate to use large speech collections. Reference (Jurafsky and Martin 2000) explains the meaning of large vocabulary speaker-independent continuous speech recognition as follows: Large vocabulary means that the corpus has vocabulary (unique words) of approximately 20,000 to 60,000 words. Speaker-independent indicates the classifier's ability to recognize the speech of people whose speech to which it has never exposed. Continuous means that the speech is recorded according to the human natural language.

One of the earliest attempts to develop a speech corpus was made by (Siemund et al. 2002), who describe the OrienTel speech dataset. They indicated that the OrienTel dataset represents the first effort to collect speech data on a large scale. The participants of OrienTel collected standard and colloquial varieties of Arabic in Saudi Arabia, the UAE, Egypt, Palestine, Tunisia and Morocco. The GlobalPhone project produced a read speech corpus that was designed for the development and evaluation of large continuous speech recognition systems (Schultz, 2002). (Alghamdi et al., 2008) presented a Saudi-accented Arabic telephone speech database. It contains 96 hours of speech, which were collected on a telephone network during 2002 and 2003 using 1033 native speakers (51% males, 49% females). Reference (Kirchhoff et al., 2006) used the CallHome corpus for morphology-based language models at different stages of Arabic ASR. New versions were used by others; for example, (Emami et al., 2007) used a new LDC test set called Dev07, which consists of 2.5 hours of speech (18186 words). The research work of (Emami et al., 2007) consists of using neural network language models for Arabic ASR.

(Alghamdi et al., 2007) developed an MSA broadcast news speech recognition system. The system trained on 7.0 hours of 7.5 hours of recorded speech and tested on the remaining half hour. The corpus contains 235 news items, 41 of which cover sport news and the rest of the items cover mainly economic news. Among the items, 88 were produced by female speakers. (Hyassat and Abu Zitar, 2006) presented a Holy Quran corpus of approximately 18.5 hours in length. The main challenge is to develop a broadcast news corpus, since the Holy Quran recordings already available. (Hyassat and Abu Zitar, 2006) indicated that it

took approximately 732 working hours to build their Holy Quran corpus. (Elmahdy, 2009) used MSA acoustic models as multilingual models to decode Egyptian dialect. They chose the Nemlar broadcast news speech corpus for building the acoustic models. The corpus consists of 40 hours of MSA news broadcasts. The total number of speakers is 259, with a lexicon of 62,000 words. (Selouani et al. 2010) presented an MSA continuous speech corpus of 200 sentences, which are pronounced by 300 Algerian native speakers from eleven regions of Algeria. (Abushariah et al., 2012) developed an Arabic ASR system based on a phonetically rich and balanced speech corpus. That work was based on 8,043 utterances, which were gathered from eight (five male and three female) speakers and resulted in approximately 8 hours of speech. The round-robin testing approach was applied.

The procedures that are applied to produce a continuous speech corpus are summarized as follows: Multiple speech files are recorded (under the same conditions) from radio and/or TV broadcast news. It is highly recommended to cooperate with local or international radio and TV stations to obtain pre-recorded speech collections (e.g., parts of their archives). This cooperation avoids the labor and cost of manually generating the required speech corpora. If the lengths of the speech files are too long, the speech files split into smaller speech files of 10-30 seconds. There is no problem with files being longer than 30 seconds, but the speech recognizer might fail to align the recordings with their phonetic transcriptions during the training process. Hence, the process will be more productive and efficient if the recordings are short. However, if the recordings are long, e.g., 10 minutes, initial alignment might be fail, thereby causing a problem during the training process. While recording to create a corpus, the following parameters are set, as indicated by Sphinx (Frequently Asked Questions, 2016): Sampling Rate = 16 kHz (or 8 kHz, depending on the training data), No. of Bits = 16 bits, No. of Channels = Mono (= single channel), and File Format = ".wav". Once the recordings have been obtained, the spoken words in the speech files are transcribed and diacritized. Based on the diacritized text, the pronunciation dictionary is produced. After this step, two models are created: the acoustic model and the language model. The three knowledge bases (acoustic models, pronunciation dictionary, and language model) will be ready for setting up and testing a continuous speech recognition system.

4 Arabic Phoneme Set

A speech recognition task is generally performed using one of the three approaches based on the basic units of classification. The units include words, syllables, and phonemes. Word-based recognition has the drawback that it requires a large data set for training. A syllable, which is a single unit of a written or spoken word, has a relatively smaller number of units and syllable-based recognition runs faster than word-based recognition. However, the recognizer, which depends on the distinctive unit of sound (i.e., phoneme), is a widely used approach since it is easy to train, (Azmi et al., 2008). In research that uses isolated words or continuous words speech, the phoneme set defined before the training stage if phoneme method is used. Linguistics scholars generally define the phoneme set of a particular language after studying and carefully classifying the speech voices. In this section, we summarize the phoneme-based research that has been reported in the literature.

Speech recognition mainly consists of two stages: training and decoding. The training stage requires two datasets: a set of speech files and a set of files that contain the phonetic transcriptions of the speech files. There are various ways of obtaining phonetic

transcriptions. The easiest is to use a phonetic dictionary in combination with a transcription of the training text. Some ASR engines such as HTK have a tool for doing this, or it can be prepared manually. Writing a phonetic dictionary is difficult, and if the vocabulary has many words, it is time-consuming. For Arabic, it is reasonable to approximate each character as a single phoneme. For instance, the phonetic transcription of "kataba" is assumed to be "k a t a b a" (ARABIC TRANSLITERATION, 2016). This method of transcription has two advantages: everyone uses it, so that data can easily be made available to other people and it allows the researchers to use other people's data; and it uses one Roman character for each Arabic character, unlike most of the other options. However, there is a problem: it uses multiple non-alphabetic characters that have an assigned meaning in some ASR engines. Another option for representing words in the phonetic dictionary is to use Arabic characters such as "كَتَبَ" with phonemes "K AE T AE B AE", as an example. Reference (Ali et al., 2009) provides more information on how to generate phonemes automatically for Arabic words (i.e., the phonetic dictionary).

Many studies have indicated that the Arabic language has 34 phonemes (28 consonants and 6 vowels), such as (Essa et al. 2008), (Al-Qatab et al., 2010), and (Alghamdi et al., 2007). However, the number of Arabic phonemes is debated by researchers. For example, (Ali et al., 2009) used a phoneme set that contains 46 phonemes to develop a tool for creating pronunciation dictionaries. The tool that was generated by (Ali et al., 2009) was later used in other research, such as (Alghamdi et al., 2007) and (AbuZeina et al., 2011). (Haraty and El Ariss, 2007) indicated that Arabic has at least one hundred twelve phonemes, as they considered that every letter has four diacritics and, therefore, four phonemes. (Alotaibi, 2008) used 37 MSA phonemes, as specified by LDC. (Elmahdy 2009) stated that MSA consists of 38 phonemes, of which 28 are original consonants, 4 are foreign and rare consonants and 6 are vowels. Examples of entries from the pronunciation dictionary that was produced by (Ali et al., 2009) are shown in Fig. 2 (a). Each word is listed with its phonetic transcription (phonemes). The dictionary in Fig. 2 (a) is suitable for CMU Sphinx. Fig. 2 (b) shows dictionary entries that are generated using Buckwalter transcription, which is suitable for the HTK engine.

(a) Arabic-word-based dictionary	(b) Buckwalter-transcription-based dictionary
... يُـوَائِم Y UH W AE: E IH M يُـوَاجِه Y UH W AE: JH IH H يُـوَازِي Y UH W AE: Z IY يُـوَاصِلُ Y UH W AE: SS IH L UH يُـوَاكِب Y UH W AE: K IH B يُـوَقِّعُ Y UH W AE Q IX AI UH ...	... AiEtibaArFA A i E t i b aA r F A AiEtidaAli A i E t i d aA l i AiHtifaAlAt A i H t i f aA l A t AimtanaEa A i m t a n a E a AinxafaDat A i n x a f a D a t ...

Fig. 2. Example entries of a pronunciation dictionary (words and phonemes)

5 Language Models

In addition to constructing the pronunciation dictionary, the ASR training stage involves declaring language models, which are also called grammars or probabilistic N-Grams. Language models are mainly used to estimate the most likely co-occurring word sequence. They are used with the acoustic model to recognize spoken words given the speech feature vectors. Hence, the choice of language model is key to the performance of the ASR

recognizer. The more constraints that are imposed on the range of possible utterances, the more accurate the recognizer will be. There are many types of language models that can be used. In general, three language model types are extracted from a set of training textual transcriptions: In the first type, the target utterance may be an arbitrary sequence of words that are drawn from the training textual transcription (which is called the "any word" language model), as shown in Fig. 3. The language model in Fig. 3 contains Arabic words that have been converted to English using the transliteration process that is described in (Arabic Transliteration, 2016). In the second type, the target utterance must be one of the training textual transcriptions, as shown in Fig. 3. In the figure, *sentence 1* is shorthand for a complete sentence; for simplicity, we used this format. The language model example that is shown in Fig. 3 is related to the HTK ASR system.

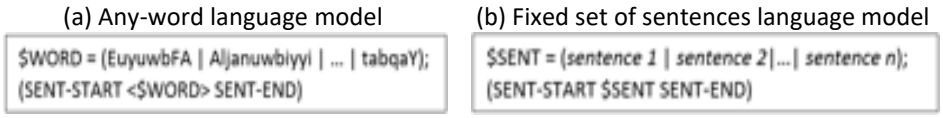


Fig. 3. Examples of language models

The first type is almost completely unconstrained and leads to very poor accuracy. (However, it allows researchers to experiment with the effects of different transcriptions because it relies entirely on the acoustic model.) The other is very tightly constrained and often leads to 100% accuracy. The third type of language model includes word co-occurrences, which are called probabilistic N-grams, as shown in Fig. 4. The language model in Fig. 4 is generated using the CMU statistical language tool (Building Language Model, 2016) of the corpus that is described in Reference (Alghamdi et al., 2007). The numbers beside the n-grams in Fig. 4 are related to the probabilities of the cases.

\1-grams:	\2-grams:	\3-grams:
...	...	...
-4.5936 الإيجَار -0.053	-0.9394 نَتَائِج أسفَرَت 0.0	-1.4602 بَاب إغلاق مَوعِدُ
-4.5936 الإيجَارِي -0.052	-0.9394 النَّمَار أسلحَة 0.	-1.4602 عَلَى آسِيَا مَوقِع
-4.2924 الإيذَاع -0.249	-0.9394 كِبَار أسمَاء 0.00	-1.4602 لِلكَتْرُونِيّ مَوقِعِهَا
...	...	...

Fig. 4. Example entries of a language model

The main purpose of the language model is to restrict the recognizer to choosing only valid word sequences. (Al-Qatab 2010) demonstrated that one benefit of the language model is to force the recognition process to follow certain rules to ensure better accuracy of the recognition output. (Alotaibi, 2008) indicated that the absence of diacritics in Arabic text decreases predictability in the language model. In general, much larger amounts of text lead to the development of more powerful language models and enhance their performance. In speech recognition, the N-gram language model is used to determine a contiguous sequence of n items, which are called unigram, bigrams, or trigrams of the language text. In the case of isolated-word speech recognition, context-free grammar is used. While language models are mainly used in large continuous speech recognition systems, context-

free grammar is used in speech recognition to predict subsequent words in small corpora or in isolated-word speech recognition.

We use language models and grammars to limit the possible words that could be considered to be the next word. However, (Building Language Model, 2016) indicated that statistical language models generally describe complex types of language, whereas grammars describe very simple types of language for command and control and they are usually written by hand. It was also indicated in this reference that grammars usually do not have associated probabilities for word sequences, but some elements might be weighed. The perplexity is the common way of evaluating language models. It indicates the average number of words that can follow a given word. (Selouani et al., 2010) used a bigram language model for continuous Arabic speech recognition. (Alghamdi et al., 2007) used both bigrams and trigrams for the language model. (Abushariah et al., 2012) used different language models (bigram and context-free grammar) for a continuous Arabic speech recognition system. (Hyassat and Abu Zitar, 2006) used a tool that is available in SPHINX-IV to generate the N-gram language model. (Kirchhoff et al. 2006) used n-gram models of up to order n = 6 to improve the perplexity of language models. (Emami et al., 2007) indicated that neural network language models for Arabic broadcast news and broadcast conversations outperform the 4-gram-based language model. (Al-Qatab et al., 2010) represented an Arabic grammar file that contains words and commands in their isolated word speech system. Reference (Vergyri et al., 2004) investigated the use of a morphology-based language model at different stages in a speech recognition system for conversational Arabic. (Satori et al., 2007) used a tool that is available in CMUSphinx to specify the grammars of a spoken Arabic digit recognition system.

6 Acoustic Models

Acoustic models are important in ASR and might influence the performance of speech recognizers. The main contribution of acoustic models is finding the most likely sentence given a speech file, with the help of the language model. During training, a sufficiently large speech corpus is used along with the corresponding phonetic transcriptions to generate a statistical representation (based on the MFCC acoustic feature vectors) of each phoneme. The overall statistical representations of the phonemes are called the acoustic model. Hence, the speech collection used to train the acoustic model. The statistical representations are commonly constructed using the HMM approach with Gaussian mixtures (typically ranging from two to 64 Gaussian mixture distributions). The HMM approach is a machine-learning tool that is employed in ASR training in two cases: The first is within a single phoneme, using three states: beginning, middle, and end. The second between three phonemes, which are called a triphone. It represents the statistical co-occurrences of left and right phonemes. For example, a well-known acoustic model is 3-emitting-state HMM (within-phoneme) for triphone-based HMM (to capture the context between neighboring phonemes). However, triphones that are below a certain frequency are excluded. Fig. 5 shows an example of the phoneme (SH:ش) with some surrounding phonemes. Intuitively, the size of the training speech corpus affects the quality of the produced acoustic model. However, preparing a large speech corpus is a difficult task, especially for dialectical Arabic, since it difficult to produce the textual form of dialectical speech. In addition, the number of speakers is an important factor for adequate ASR performance. In general, ASR toolkits such as Sphinx and HTK have modules for training

the acoustic models. Some studies indicated that there are performance differences between using acoustic models that are based on vowelized (i.e., diacritized) and non-vowelized transcripts.

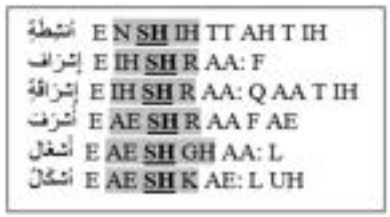


Fig. 5. Examples of triphones of the phoneme SH

MFCC is the commonly used acoustic feature extraction method, which was introduced by (Davis and Mermelstein, 1980). However, there are other speech feature extraction methods, such as linear predictive cepstral coefficients (LPCCs) and perceptual linear predictive (PLP) coefficients. Hence, HMM parameters are trained using statistical representations that are obtained from MFCCs. MFCCs are based on dividing a signal into overlapping short frames that are represented using a 12-cepstral coefficient for each frame. The 13th feature is the energy from the frame, which is useful for determining phone identity. Fig. 6 (a) shows the steps for extracting the MFCCs of a speech signal [Rabiner 2004]. Fig. 6 (b) shows the feature vectors of a speech file after completing the feature extraction process. Each column represents the 13 features of a 25.6-millisecond frame. However, despite the popularity of MFCC, (Xiao and Qin, 2010) demonstrated that MFCC does not fully reflect speech information because of noise effects. They presented a new method for noise-robust speech recognition based on a hybrid model of HMM and Wavelet Neural Network (WNN).

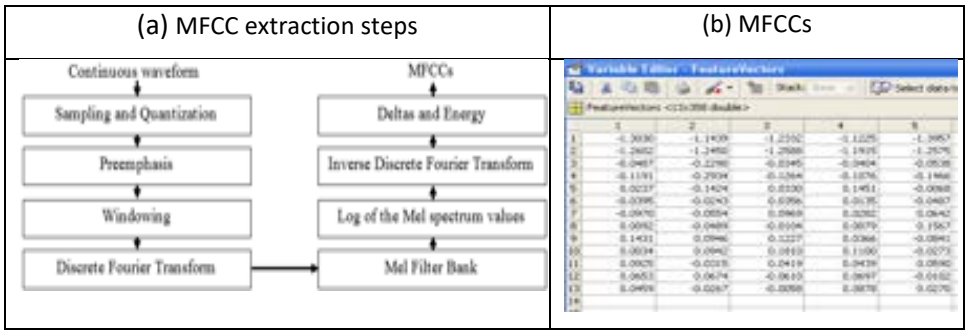


Fig. 6. MFCC extraction and MFCCs

As HMM is extremely important in ASR, we provide a simple example to demonstrate how an HMM model is utilized to compute the most likely hidden state sequence given some observations. In general, an HMM model is described by the following parameters: the number of states N, the state transition probabilities, the observation probability, and the initial probabilities. Suppose that we have a system that describes student states. The

states are {Studying, Working, and Sleeping} and the observations are {Fine, Scared, and Tired}. Fig. 7 shows the parameters of the student HMM. In general, HMM parameters are estimated using the Baum-Welch algorithm, which estimates the parameters of the acoustic model.

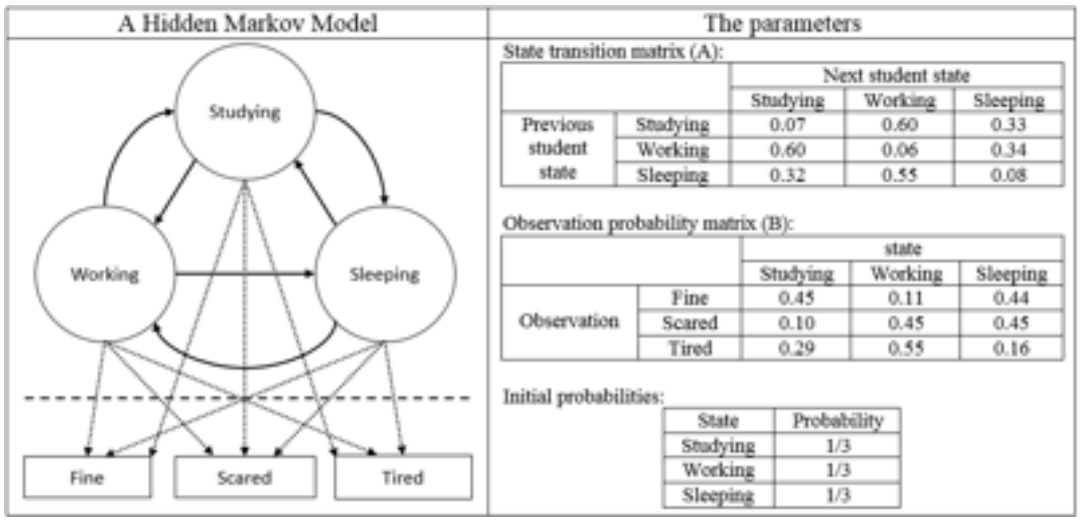


Fig. 7. HMM with its parameters

Now, we need to answer the following question: "What are the most likely states if we observe the student to be: Tired, Scared, and Fine?" This is similar to the ASR problem: "What is the most likely sentence out of all sentences in the language L given some acoustic input observations O?" The Viterbi algorithm used to find the most likely hidden states. It also includes backtracking function that keeps track of which states seem to be the best at each stage. The example is illustrated in Fig. 8. For illustration, at t=1, an initialization step is performed as follows: α1 (Studying) = initial probability * probability of Fine observation:1/3*0.45=0.1500; α1 (Working) = 1/3*0.11=0.0367; and α1 (Sleeping) = 1/3*0.44=0.1467, where α is a variable that describes the probability at time t. The maximum value of this variable is used at each subsequent t, as we will see next. At t=2, α2 of Studying state for Tired observation is calculated as follows: α2 (Studying) = max (α1 (Studying)* the probability of (Studying | Studying), α1 (Working)* the probability of (Studying | Working), α1 (Sleeping)* the probability of (Studying | Sleeping))* probability (Tired) = max (0.15*0.07, 0.0367*0.6, 0.1467*0.32)*0.29 = max (0.0105,0.022,0.0469)*0.29 = 0.0136. We calculated one additional variable at t=3, namely, α3: α3 (Sleeping) = max (α2 (Studying)* the probability of (Sleeping | Studying), α2 (Working)* the probability of (Sleeping | Working), α2 (Sleeping)* the probability of (Sleeping | Sleeping))* probability (Fine) = max (0.0136*0.33, 0.0495*0.34, 0.0079*0.08)*0.29 = max (0.0044,0.016,0.00063)*0.44 = 0.0076. The reset of calculations is performed in the same way. Once the end of observations has been reached, the best sequence is identified by backtracking to determine the sequence with the highest probability. Hence, the student was {Studying, Sleeping, Working, and studying} for the

observations {Fine, Tired, Scared, and Fine}. Fig. 8 shows the backtracking process, which is indicated by stars.

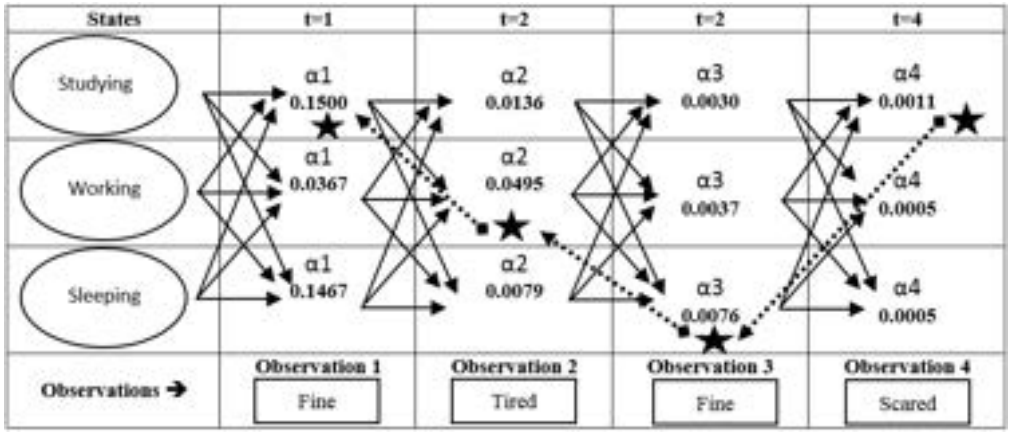


Fig. 8. Viterbi trellis for three states and four observations

7 Performance Evaluation

The performance of isolated-word speech recognition is generally measured using the recognition accuracy rate, which is the percentage of correctly recognized patterns, such as words or digits. However, in continuous speech, the word error rate (WER) is the common metric for measuring ASR performance. WER is computed using the following formula: WER=(S+D+I)/N, where S is the number of substitution word errors, D is the number of the deletion word errors, I is the number of the insertion word errors, and N is the number of words in the testing set. The word accuracy can be measured using the WER formula: Word Accuracy = 1 – WER.

According to (Hyassat and Abu Zitar, 2006), the WER of MSA is in the range of 15–20%. However, WER also depends on other parameters, such as the size of the training corpus, the number of words in the dictionary, and the perplexity of the language model. We emphasize that measuring performance requires showing whether the achieved performance is statistically significant. It is noted that most Arabic speech recognition studies do not conduct the appropriate statistically significant tests to prove their performance. Therefore, a baseline system should be developed and tested, and the output of the proposed method should be statistically compared with the baseline results to determine the statistically significant enhancement. If the result does not show statistically significant enhancement, then the enhancement is considered accidental and the proposed method is not regarded as a strong method for enhancing the classification performance. Reference (Plötz, 2005) describes how to perform statistical significance testing.

In this section, we present some performance evaluations for continuous speech recognition, since it is of greater interest and is more challenging than isolated-word speech recognition. (Kirchhoff et al. 2006) performed performance evaluation of morphology-based language models for conversational ASR. WER improved by 1.8% and 1.5% on 2 different test sets. (Emami et al., 2007) achieved WER improvements of 0.8% and 3.8%

for 2 different configurations of neural probabilistic models. (Alghamdi et al., 2007) achieved a WER of 13.66% on a broadcast news corpus. (Hyassat and Abu Zitar, 2006) achieved a WER of 46.182% using the Holy Quran corpus. (Elmahdy, 2009) reported a recognition accuracy of 99.34% for Egyptian Colloquial Arabic. (Selouani et al. 2010) presented an accuracy rate of 91.65% for an MSA continuous speech corpus. Reference (Abushariah et al., 2012) demonstrated WERs of 11.27% and 10.07% with and without diacritical marks, respectively, for an MSA continuous speech corpus. Table 3 shows WER results for multiple systems on different English speech corpora (Jurafsky and Martin 2000).

Table 3 WERs for ASR on different English corpora

Speech Corpus	Vocabulary	WER %
TI Digits	11 zero-nine, oh	0.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech (CST)	64000+	20

8 Empirical Study

In this section, we evaluate the recognition performance of Arabic continuous speech using the Soundflower Mac free utility. Soundflower is employed as a speaker-independent continuous speech recognition system to evaluate the performance of Arabic ASR. The speech corpus that is used in this work is an in-house corpus that contains 275 speech files, which were recorded by 20 native Arabic speakers (10 male and 10 female). Each male speaker uttered 15-speech items, while some of female speakers uttered less than 15 speech items, as shown in Table 4. The speech files mainly contain MSA local and international news and were recorded from the Al-Sabah TV channel in Kuwait. The speech files were prepared in fixed lengths of 30-60 seconds. The speech items were sampled at 16 kHz and sum up to 2.63 hours of speech. The training textual transcriptions of the speech files were prepared by transcribing the speech files according to speakers' utterances. Table 4 presents the corpus information.

Table 4 Corpus information

#	Gender	# of Speakers	# of Speech Files	Length (hour)	# of Unique Words
1	Male	10	150	1.53	5,149
2	Female	10	104	1.10	3,738
	Total	20	254	2.63	8,887*

*the total number of unique words in the corpus is 7,386 because of common words

Fig. 9 shows an example of the Soundflower output of a speech file after the recognition process. This textual output is aligned with the actual transcription to find D, S, I, and N, which are used to calculate the WER through a previously outlined method. There are some

recognition errors in the outputs in Fig. 9 (e.g., the word "لمنع" instead of "لميناء"). As in any other classification or pattern recognition system, it is natural to have a nonzero misclassification rate.

Transcription of a speech file

في ظل النهضة البارزة التي تشهدها الكويت في شتى المجالات العلمية والاقتصادية والعمرانية شمل حضرة صاحب السمو أمير البلاد الشيخ صباح الأحمد برعايته وحضوره وضع حجر الأساس لميناء مبارك الكبير بحضور كل من ولي العهد سمو الشيخ نواف الأحمد ورئيس مجلس الوزراء المكلف سمو الشيخ ناصر المحمد ورئيس مجلس الأمة جاسم الخرافي ووزير الدولة لشؤون البلدية الدكتور فاضل صفر بالإضافة إلى مجموعة من الشيوخ والنواب والوزراء السابقين

Soundflower output

Fig. 9. Example of Soundflower output

We evaluated the performances for three cases: male-only, female-only and mixed (male and female) speech files. In the first case, the performance was measured using the male speech files. That is, Soundflower was employed to measure the accuracies of the 150 speech files that correspond to the 10 male speakers. Table 5 shows the achieved results of each speaker. The table also shows the accuracy range, which is [42.26%, 70.39%]. The differences in the scored accuracies were related to several factors, such as the anatomy of the speaker's vocal tract, the speed of the speech, and the accent. Table 5 also shows that the WER is 44.67% (= 100% - 55.33%).

Table 5 Accuracies for male-only speech

#	Male Speaker	# of Speech Files	Length (min:sec)	Accuracy (%)
1	Speaker 1	15	9:29	70.39
2	Speaker 2	15	10:12	48.80
3	Speaker 3	15	9:47	64.93
4	Speaker 4	15	8:46	59.07
5	Speaker 5	15	8:59	42.26
6	Speaker 6	15	9:55	54.67
7	Speaker 7	15	8:12	57.87
8	Speaker 8	15	8:30	55.16

9	Speaker 9	15	9:36	44.58
10	Speaker 10	15	8:54	55.66
	Total	150	92:20	Avg =55.33%

For female speakers, 104 speech files were used to evaluate the accuracy. Table 6 shows the accuracy of each of the 10 female speakers. The accuracy range was [46.52%, 68.73%]. This range is close to what we achieved for male speakers. This reveals that the performances of male and female speech recognition are very close when the Soundflower tool is used. This result calls for more research to determine the effect of acoustic differences between male and female speakers on Arabic speech recognition. Table 6 also shows that the WER is 43.03% (= 100% - 56.97%).

Table 6 Accuracies for female-only speech

#	Female Speaker	# of Speech Files	Length (min:sec)	Accuracy (%)
1	Speaker 1	3	2:00	60.51
2	Speaker 2	15	9:42	68.73
3	Speaker 3	15	10:34	57.19
4	Speaker 4	7	5:12	52.07
5	Speaker 5	15	8:00	56.53
6	Speaker 6	15	9:15	50.85
7	Speaker 7	15	8:45	46.89
8	Speaker 8	2	1:27	62.83
9	Speaker 9	2	1:29	67.63
10	Speaker 10	15	9:56	46.52
	Total	104	66:20	Avg=56.97%

The average accuracies for the previous two cases indicate that female speech recognition outperforms the male speech recognition. The third case separates the corpus into male and female speech to find the accuracies separately. Finally, we evaluated the mixed male and female case for all speech files. Table 7 shows the results of the mixed case.

Table 7 Accuracies for mixed speech (male and female)

Gender	# of Speakers	# of Speech Files	Length (min:sec)	Accuracy (%)
Male	10	150	92:20	54.66
Female	10	104	66:20	55.17
Male & Female	20	254	158:40	54.02

Although gender is an important factor that must be considered in speech recognition, the experimental evaluation did not show clear performance differences between using the prepared corpus and the Soundflower tool. Despite this, we expect to obtain lower accuracy in the case of female speech, as reported in the literature; for example, (Vergin et al. 1996) found that female speech recognition performance is superior to male speech recognition

performance. Nevertheless, we conducted our experiments on a small corpus. Thus, our results are not generalizable to other cases of Arabic speech recognition. However, this result motivates future research on investigating and enhancing the performance of Arabic speech recognition.

9 Conclusions and Future Work

The review of the Arabic speech recognition literature shows that the research is still in the raw stage, especially for continuous speech. Most of the research is focused on isolated-word speech recognition. However, there are some research activities on continuous speech recognition. The major obstacle is corpus availability. Hence, reinforcing Arabic speech recognition requires the professionality production of large Arabic continuous speech collections (i.e., corpora). The study also presents some ASR components such as pronunciation dictionaries, language models, and acoustic models. The following are some research topics for investigation. Although there are some studies regarding pronunciation variations, such as (AbuZeina et al. 2011), more studies are needed to tackle this phenomenon. The research could also investigate using long-distance word relationships in language models. The traditional N-gram language models assume that a word is only influenced by a few (typically one or two) proceeding words. However, it is much more accurate to account for longer-distance constraints. Consideration of semantic relationships is one option for enhancing ASR performance. A study of semantic and syntactic relationships (e.g., WordNet) was presented in (Ruiz-Casado et al. 2007). The deep-neural-network hidden Markov model (DNN-HMM) hybrid architecture has been employed for Arabic (Dahl et al. 2012). Further research is needed to clarify the realistic Arabic phoneme set. (Nahar et al. 2013) performed some preliminary work in this direction. However, an method is needed for automatic extraction of Arabic phonemes using data-driven approaches and clustering methods. Currently, research is being conducted on training a recognizer using synthesized data. If this becomes possible, we could obtain as much data as we want in a preferred domain or with a new, large vocabulary (Prof Allan Ramsay, 2016). Other research topics include diphone-based speech synthesis, measuring the disfluency of speech-impaired patients (in particular, patients with Parkinson's disease), and using deep neural networks to map acoustic events to phonemes rather than the more traditional Gaussian-mixture models.

Acknowledgements

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

References

[1] Emerging Technologies: available: *http://www.em-t.com/*. Accessed: Nov 26, 2016.

[2] Voice Month: 5 Unique Applications of Voice Biometrics - FindBiometrics: Nov 26, 2016. *http://findbiometrics.com/voice-month-5-unique-applications-of-voice-biometrics-22186/*. Accessed: Nov 26, 2016.

[3] Nuance | Nuance - PDF, Customer Service, HIM, and Speech Recognition Solutions: available: *http://www.nuance.com/*. Accessed: Nov 26, 2016.

[4] Speech Recognition Case Study Kuwait Finance House (KFH): available: *http://*www.*em-t.com/content/speech-recognition-case-study-kuwait-finance-house-kfh*. Accessed: Nov 26, 2016.

[5] ADCB: available: *http://www.adcb.com/*. Accessed: Nov 26, 2016.

[6] Mubarak, Hamdy, and Kareem Darwish. 2014.Using Twitter to collect a multi-dialectal corpus of Arabic. *ANLP* (2014): 1.

[7] Mohammed, A., Sunar, M. and Hj Salam, M. 2015. Quranic Verses Verification using Speech Recognition Techniques. *Jurnal Teknologi*. 73, 2 (2015).

[8] Jamaliah Ibrahim, N., Yamani Idna Idris, M., Razak, Z. and Naemah Abdul Rahman, N. 2013. Automated tajweed checking rules engine for Quranic learning. *Multicultural Education & Technology Journal*. 7, 4 (2013), 275-287.

[9] Strik, H. and Cucchiarini, C. 1999. Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*. 29, 2-4 (1999), 225-246.

[10] Rabiner, L. and Schafer, R. 2007. Introduction to Digital Speech Processing. *FNT in Signal Processing*. 1, 1â€“2 (2007), 1-194.

[11] Elmahdy, Mohamed, et al. 2009. Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition. *Natural Language Processing*. *SNLP'09. Eighth International Symposium* on. IEEE, 2009.

[12] ARABIC TRANSLITERATION, available: http://www.qamus.org/transliteration.htm, accessed Nov 26, 2016.

[13] Bahi H, Sellami M. 2001. Combination of vector quantization and hidden Markov models for Arabic speech recognition. *ACS/IEEE international conference on computer systems and applications*, 2001

[14] Alimi AM, Ben Jemaa M. 2002. Beta fuzzy neural network application in recognition of spoken isolated Arabic words. *Int J Contr Intell Syst* 30(2), Special issue on speech processing techniques and applications

[15] Elmisery FA, Khalil AH et al.2003. A FPGA-based HMM for a discrete Arabic speech recognition system. In: Proceedings of the *15th international conference on microelectronics, 2003. ICM 2003*

[16] Amrouche, Abderrahmane, and Jean Michel Rouvaen. 2003. Arabic isolated word recognition using general regression neural network. *Circuits and Systems, 2003 IEEE 46th Midwest Symposium* on. Vol. 2. IEEE, 2003

[17] Bourouba H, Djemili R et al. 2006. New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition. *2nd Information and Communication Technologies*, 2006. ICTTA'06

[18] Satori H, Harti M, Chenfour N.2007. Introduction to Arabic speech recognition using CMU Sphinx system. *Information and communication technologies international symposium proceeding ICTIS07*, 2007

[19] Haraty, R. and El Ariss, O. 2007. CASRA+: A Colloquial Arabic Speech Recognition Application. *American Journal of Applied Sciences*. 4, 1 (2007), 23-32.

[20] Essa EM, Tolba AS et al. 2008. A comparison of combined classifier architectures for Arabic speech recognition. *International conference on computer engineering and systems*, 2008. ICCES 2008

[21] Hyassat, H. and Abu Zitar, R. 2006. Arabic speech recognition using SPHINX engine. *Int J Speech Technol*. 9, 3-4 (2006), 133-150.

[22] Azmi M, Tolba H,Mahdy S, FashalM.2008. Syllable-based automatic Arabic speech recognition in noisy-telephone channel. In: *WSEAS transactions on signal processing proceedings, World Scientific and Engineering Academy and Society (WSEAS),* vol 4, issue 4, pp 211–220

[23] Alotaibi, Y. 2008. Comparative Study of ANN and HMM to Arabic Digits Recognition Systems. *eng*. 19, 1 (2008), 43-60.

[24] Satori, Hassan, et al. 2009. Investigation Arabic speech recognition using CMU sphinx system. *Int. Arab J. Inf. Technol*. 6.2 (2009): 186-190.

[25] Al-Qatab, Bassam AQ, and Raja N. Ainon. 2010. Arabic speech recognition using hidden Markov model toolkit (HTK). *Information Technology (ITSim), 2010 International Symposium* in. Vol. 2. IEEE, 2010.

[26] About LDC | Linguistic Data Consortium: available: *https://www.ldc.upenn.edu/about*. Accessed: Nov 26, 2016.

[27] Linguistic Data Consortium - Linguistic Data Consortium: available: *https://catalog.ldc.upenn.edu/*. Accessed: Nov 26, 2016.

[28] Soltau, Hagen, et al. 2007. The IBM 2006 Gale Arabic ASR system. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference* on. Vol. 4. IEEE, 2007.

[29] Kingsbury, Brian, et al.2011. The IBM 2009 GALE Arabic speech transcription system. *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on. IEEE, 2011.

[30] Mangu, Lidia, et al. 2011. The IBM 2011 GALE Arabic speech transcription system. *Automatic Speech Recognition and Understanding (ASRU)*, 2011 IEEE Workshop on. IEEE, 2011.

[31] CALLHOME Egyptian Arabic Speech - Linguistic Data Consortium. Available: *https://catalog.ldc.upenn.edu/LDC97S45*. Accessed Nov 26, 2016.

[32] Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K. and Stolcke, A. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*. 20, 4 (2006), 589-608.

[33] Jurafsky, D. and Martin, J. 2000. Speech and language processing. Prentice Hall.

[34] Siemund, Rainer, et al. 2002. OrienTel—Arabic speech resources for the IT market. *LREC 2002 Arabic Workshop*. 2002.

[35] Schultz, Tanja. 2002. Globalphone: a multilingual speech and text database developed at karlsruhe university. *INTERSPEECH*. 2002.

[36] Alghamdi, M., Alhargan, F., Alkanhal, M., Alkhairy, A., Eldesouki, M. and Alenazi, A. 2008. Saudi Accented Arabic Voice Bank. *Journal of King Saud University - Computer and Information Sciences*. 20, (2008), 45-64.

[37] Emami, Ahmad, and Lidia Mangu. 2007. Empirical study of neural network language models for Arabic speech recognition. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop* on. IEEE, 2007.

[38] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. *Int J Speech Technol*. 10, 4 (2007), 183-195.

[39] Selouani, Sid Ahmed, and Malika Boudraa. 2010. Algerian Arabic speech database (ALGASD): corpus design and automatic speech recognition application. *Arabian Journal for Science and Engineering* 35.2C (2010): 158.

[40] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. 2012. Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. *International Arab Journal of Information Technology (IAJIT)* 9.1 (2012): 84-93.

[41] Frequently Asked Questions (FAQ) [CMUSphinx Wiki]: Avilable: *http://cmusphinx.sourceforge.net/wiki/faq*. Accessed: November 26,2016.

[42] Ali, M., Elshafei, M., Al-Ghamdi, M. and Al-Muhtaseb, H. 2009. Arabic Phonetic Dictionaries for Speech Recognition. *Journal of Information Technology Research*. 2, 4 (2009), 67-80.

[43] AbuZeina, D., Al-Khatib, W., Elshafei, M. and Al-Muhtaseb, H. 2011. Cross-word Arabic pronunciation variation modeling for speech recognition. *Int J Speech Technol*. 14, 3 (2011), 227-236.

[44] Building Language Model [CMUSphinx Wiki]: 2015. *http://cmusphinx.sourceforge.net/wiki/tutoriallm*. Accessed: Nov 26, 2016.

[45] Vergyri D., Kirchhoff K., Duh K., and Stolcke A. 2004. Morphology Based Language Modeling for Arabic Speech Recognition. *Proceedings of Interspeech, Germany*, pp. 2245-2248, 2004.

[46] Plötz T. 2005. Advanced stochastic protein sequence analysis, Ph.D. thesis, *Bielefeld University*

[47] Ruiz-Casado, M., Alfonseca, E. and Castells, "Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia." *Data & Knowledge Engineering*. 61, 3 (2007), 484-499.

[48] Dahl, G., Dong Yu, Li Deng, and Acero, A. 2012. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*. 20, 1 (2012), 30-42.

[49] Nahar, Khalid MO, et al. "Data-driven Arabic phoneme recognition using varying number of HMM states." Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on. IEEE, 2013.

[50] Prof Allan Ramsay (BSc, MSc, PhD) research profile - research | The University of Manchester: available: *http://www.manchester.ac.uk/research/Allan.ramsay/research*. Accessed: Nov 26, 2016.

[51] El Amrani, Mohamed Yassine, et al. "Building CMU Sphinx language model for the Holy Quran using simplified Arabic phonemes." Egyptian Informatics Journal (2016).

[52] Vergin, Rivarol, Azarshid Farhat, and Douglas O'Shaughnessy. "Robust gender-dependent acoustic-phonetic modelling in continuous speech recognition based on a new automatic male/female classification." Spoken Language, 1996. ICSLP 96. Proceedings of Fourth International Conference on. Vol. 2. IEEE, 1996.

[53] Rabiner, L. R. and Juang, B. H., Statistical Methods for the Recognition and Understanding of Speech, Encyclopedia of Language and Linguistics, 2004

[54] Davis, Steven, and Paul Mermelstein. "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences." IEEE transactions on acoustics, speech, and signal processing, 28.4 (1980): 357-366.

[55] Al-Anzi, Fawaz S., and Dia AbuZeina. "Stemming impact on Arabic text categorization performance: A survey." 2015 5th International Conference on Information & Communication Technology and Accessibility (ICTA). IEEE, 2015.

[56] Xiao Y, Qin, "A Noise robust speech recognition based on improved hidden Markov model and wavelet neural network." Computer Engineering Applications (2010), 46(22): pp 162–164, 235.

A manuscript number has been assigned

E

Engineering Applications of Artificial Intelligence

Reply all

Mon 11/12/2017, 17:35

Fawaz Alanzi;
alanzif@eng.kuniv.edu.kw

Inbox

Dear Professor Fawaz S. Al-Anzi,

Your submission entitled "Performance Evaluation of Sphinx and HTK Speech Recognizers for Spoken Arabic Language" has been been assigned the following manuscript number: EAAI-17-2181.

You will be able to check on the progress of your paper by logging on to Elsevier Editorial System as an author.
The URL is https://ees.elsevier.com/eaai/.

Thank you for submitting your work to this journal.

Kind regards,

Ajith Abraham, PhD
Editor In Chief.
Engineering Applications of Artificial Intelligence

For further assistance, please visit our customer support site at http://help.elsevier.com/app/answers/list/p/7923. Here you can search for solutions on a range of topics, find answers to frequently asked questions and learn more about EES via interactive tutorials. You will also find our 24/7 support contact details should you need any further assistance from one of our customer support representatives

Performance Evaluation of Sphinx and HTK Speech Recognizers for Spoken Arabic Language

Fawaz S. Al-Anzi, Dia AbuZeina
Department of Computer Engineering
Kuwait University, Kuwait City, Kuwait
{fawaz.alanzi; dia.abuzeina}@ku.edu.kw

ABSTRACT: *Automatic speech recognition (ASR) has recently received significant attention with respect to more convenient human-computer interaction. Despite the successful implementation of ASR technology in different languages, employing this technology in Arabic natural language processing (NLP) applications is limited and constrained to a small vocabulary of isolated words such as digits and control commands. Therefore, particular attention has been paid to promoting research in this field to automate man-machine communication. We aim to examine the performance of two popular ASR engines for identical Arabic speech collection. The ASR engines include the Carnegie Mellon University (CMU) PocketSphinx and the Hidden Markov Model Toolkit (HTK). In fact, performing an ASR task using different recognizers will increase researcher knowledge regarding which engine is the best fit for particular target applications, as well as enhancing research in this field. This paper includes an experimental evaluation of both PocketSphinx and HTK recognizers using a new "in-house" Arabic continuous speech corpus that contains 15.94 hours (12.74 hours for training and 3.19 hours for testing). The vocabulary contains 30,986 words. In these experiments, we used two text formats, Arabic characters for CMU Sphinx (PocketSphinx decoder) and Roman characters for HTK (HVite decoder) because HTK expects Roman characters. The experimental comparison shows that PocketSphinx outperforms (even in a shorter time) HTK. In addition, this study shows the intermediate steps for training the necessary models such as the acoustic model and the language models.*

Keywords: Arabic speech recognition, PocketSphinx, HTK, HVite, Buckwalter, language model, phonetic dictionary.

1. **Introduction.** Arabic is the most widely spoken Semitic language today that recently has received significant attention for automatic speech recognition (ASR). ASR is a component of the natural language processing (NLP), which is used to automate the communication process between human and machine, i.e., the man-machine interaction. In this regard, much research has been devoted to introducing innovative technologies in dialogue systems for automation purposes (e.g., banking services, cars, control machines, etc.). However, employing ASR technology in Arabic NLP applications is still limited due to various challenges about within the Arabic language itself. For instance, it is difficult to obtain corpora for dialects that are spoken rather than written, i.e., there is no common writing standard, difficulty in obtaining a sizable diacritized text as Arabic allows writing without diacritics, and an enormous number of word forms due to the morphology richness of Arabic. In fact, one of the most difficult tasks in Arabic ASR is preparing a large diacritized text for ASR systems, which is a time-consuming preprocessing stage. In order to promote research on the Arabic ASR, we considered the corpora availability problem by producing a manually diacritized large-vocabulary speaker-independent continuous speech corpus for Modern Standard Arabic (MSA). The contents of the prepared corpus belong to general broadcast

news.

In this work, we used the prepared corpus for an experimental evaluation of two off-the-shelf open source speech recognition toolkits, namely the Carnegie Mellon University (CMU) Sphinx [1] and the Hidden Markov Model Toolkit (HTK) [2]. In fact, it is important to find the performance of the popular speech engines using an identical speech collection because it reveals the unique characteristics of each engine for further understanding of their behavior in NLP and ASR applications. With the growing interest in ASR technology, it becomes more important to evaluate recent ASR systems in order to find the best-suited system for the tasks in question. For instance, the study in [3] demonstrated a large-scale evaluation of open-source speech recognition toolkits that include Sphinx, HTK, and Kaldi [4]. The study in [3] indicated that Kaldi is better than Sphinx and HTK in terms of results and training recipes (for the German and English languages).

The work with the Arabic ASR considers one of two textual formats, either Arabic characters or Roman characters. Accordingly, there are two main approaches to address the training with Arabic text. The first approach considers using Arabic alphabet characters such as the Sphinx recognizer; the second approach uses Roman characters such as the HTK recognizer. It is sometimes required to convert Arabic characters into Roman characters, especially when recognizers assume that ASCII is used to write the training textual files rather than Unicode. HTK expects Roman characters, which means that Romanization of the Arabic ASR tasks must be completed. From the ASR point of view, Romanization means that the recognizer is trained on the Romanized transcriptions of the data. In general, choosing a recognizer on the basis of accepting Arabic characters should have no difference in performance since the core of the algorithms of training and decoding are the same in the different ASR engines. One purpose of this work is to evaluate the performance using the Sphinx and HTK recognizers. Therefore, we choose Buckwalter (BW) transliteration for the Romanization process in the HTK recognizer. There are many character options for Romanization; however, BW transliteration [5] is a good option for Romanization as it has the following two advantages: namely, it is popular, so the data can be easily made available to others as well as making it possible to use the data of other people.

In this paper, we demonstrate an experimental evaluation of two cases; the Arabic character-based recognizer is the CMU Sphinx (PocketSphinx decoder), and the Roman character-based recognizer is the HTK version 3.4.1 (HVite decoder). However, the HTK toolkit has another famous decoder, which is ‘HDecode’, but in this work, we used HVite. As future work, we will aim at implementing HDecode for the Arabic ASR. The CMU Sphinx toolkit includes the latest available releases as follows: ‘Sphinxbase – 5Prealpha’, ‘PocketSphinx - 5prealpha’, and ‘SphinxTrain - 5prealpha’. For the experiments, we used a new “in-house” continuous speech corpus that contains 15.94 hours of MSA speech. The training set contains 12.74 hours (1,611 speech files), while the testing set contains 3.19 hours (403 speech files). This study also presents the intermediate steps to train the models as well as the Romanization process. For fairness in the comparison, we emphasized that the training set and the testing set are identical in both systems (i.e., the same wave files, the same diacritized text, and the same phonemes set but with different symbols). Based on our best knowledge, this is the first attempt to experimentally compare the Sphinx and HTK recognizers for continuous Arabic speech.

This paper is organized as follows: Section 2 presents the literature review. In Section 3, we present the phonemes and the pronunciation dictionary. The language models are explained in Section 4. The implementation and the results of the PocketSphinx and HTK recognizers are presented in Section 5, followed by the conclusion in Section 6.

2. **Literature Review**. Textual training data are an essential part of any speech corpus as they represent speech transcription. However, not all open-source recognizers can handle Arabic characters, which may lead to complications in ASR implementation. In fact, most of the current recognizers are designed for the English language, which is sometimes not compatible with the Arabic language due to the character representations. The literature shows that two types of textual data are used according to the employed recognizer. If the used recognizer supports Arabic characters, then it is straightforward to use the Arabic characters. However, if the recognizer cannot handle the Arabic characters, then the choice is to consider Romanization. In general, Arabic ASR researchers use either Sphinx or HTK for the recognition task. However, no single work presents a comparison between both systems, which is the motivation of this work. Based on the literature, most Arabic ASR studies employ the CMU sphinx engine because it is compatible with Arabic characters. However, few works have used the HTK engine based on Romanization. For instance, previous studies that used the CMU Sphinx are as follows: Hyassat and Abu Zitar in [6] employed Sphinx tools for the Holy Quran speech recognition, Alghamdi et al. in [7] used Sphinx tools for an MSA ASR system, AbuZeina et al. in [8] used the Sphinx tool for crossword Arabic pronunciation variation modeling for ASR, AbuZeina et al. in [9] used the Sphinx tool for within-word Arabic pronunciation variation modeling for ASR, Abushariah et al. in [10] used Sphinx tools for an Arabic ASR system based on a phonetically rich and balanced speech corpus, and Al-Anzi and AbuZeina reported the most recent Arabic ASR work [11] that is based on Arabic characters. They used the CMU PocketSphinx to evaluate the impact of phonological rules on the Arabic ASR.

The literature shows that employing Roman characters for the Arabic ASR is less than what was observed for the Arabic characters. The following are results from previous studies. Kirchhoff et al. in [12] proposed the use of the Romanization method for the transcription of the speech corpus. Vergyri et al. in [13] used Romanized transcriptions to train the speech recognizer. Kirchhoff et al. in [14] used Romanized transcription in language modeling for large-vocabulary conversational Arabic speech recognition. Satori et al. in [15] introduced an Arabic voice recognition system where both the training and recognizing processes use the Romanized characters. Elmahdy et al. in [16] used Sphinx tools in addition to the SAMPA Romanization method for dialectal Arabic speech recognition. Al-Qatab et al. in [17] employed HTK for a small corpus of continuous speech and isolated words; however, they did not indicate anything regarding Romanization. The study in [18] used BW Arabic transliteration for Tunisian dialect dialogue systems. A recent work using HTK was reported by Merad-Boudia et al. in [19], and they employed the HTK engine for a small corpus of isolated and connected words. In summary, the literature shows that no single work has conducted practical research to evaluate the performance using an identical speech corpus, which indicates the importance and originality of this work.

3. **Phonemes Set and Pronunciation Dictionaries.** In order to train the models for ASR tasks, a speech corpus is required. This contains a set of speech files and corresponding textual transcription. The textual transcription is used to find the phonetic transcriptions through the phonemes set. Therefore, defining the phonemes set is the initial step in ASR work. A phoneme is the basic unit of sound that ASR uses for classification. While it is possible to use words and syllables for ASR classification, the phoneme approach is the most widely used in recent ASRs. On the other hand, the number of phonemes for Arabic is still a debated matter. For instance, Alghamdi et al. in [7] used 46 phonemes. Al-Qatab et al. in [17] used 34 phonemes (28 consonants and 6 vowels). Elmahdy et al. in [16] indicated that

MSA consists of 38 phonemes, where 28 are original consonants, 4 are foreign and rare consonants, and 6 are vowels. Haraty et al. in [20] indicated that Arabic has at least 112 phonemes, as they considered that every letter has four diacritics and therefore four phonemes. Alotaibi in [21] used 37 MSA phonemes as given by the Language Data Consortium (LDC).

Hence, the defined phonemes set in addition to the corpus transcription are used to generate the pronunciation dictionary that contains the phonetic transcription. In this work, the employed ASR recognizers have the following two options to handle the Arabic characters: the Sphinx recognizer allows the use of Arabic characters. Therefore, we do not need to consider Romanization. However, HTK expects Roman characters. That is, if the corpus textual transcription is stored using the standard Arabic characters set, then it needs to be transcribed into something that HTK can handle. For the Arabic ASR, BW transliteration has a distinguished attribute in which it uses one Roman character for each Arabic character, making it reasonable to approximate that each Arabic character corresponds to a single phoneme. For instance, the phonetic transcription of the word "kataba," which means “he wrote,” is "k a t a b a". However, BW uses a number of non-alphabetic characters to consider in the conversion process (e.g., BW uses “$” for the Arabic letter ‘Sheen’ “ش” , which is a special character in some recognizers such as HTK). Hence, the conversion of the Arabic text to BW transliteration requires a further step to swap the non-alphabetic characters with some other arbitrary characters. Of course, it is straightforward to convert the Romanized text back to the Arabic transcription in the case of the need to output in Arabic. Table 1 shows the Arabic and Romanized characters using BW transliteration. In the table, we indicate some replacement cases.

TABLE 1. Buckwalter (BW) transliteration with some replacements

#	*Arabic character*	*Phoneme (BW)*	#	*Arabic character*	*Phoneme (BW)*
1	ا	*A*	*23*	ك	*k*
2	أ	*> (replaced to) O*	*24*	ل	*l*
3	ب	*b*	*25*	م	*m*
4	ت	*t*	*26*	ن	*n*
5	ث	*v*	*27*	و	*w*
6	ج	*j*	*28*	ى	*Y*
7	ح	*H*	*29*	ه	*h*
8	خ	*x*	*30*	ي	*y*
9	د	*d*	*31*	آ	*| (replaced with) U*
10	ذ	** (replaced to) J*	*32*	ء	*G*
11	ر	*r*	*33*	ؤ	*& (replaced with) W*
12	ز	*z*	*34*	ئ	*} (replaced with) Q*
13	س	*s*	*35*	ة	*p*
14	ش	*$ (replaced to) C*	*36*	ً	*F*
15	ص	*S*	*37*	ّ	*N*
16	ض	*D*	*38*	ٍ	*K*
17	ط	*T*	*39*	َ	*a*
18	ظ	*Z*	*40*	ُ	*u*
19	ع	*E*	*41*	ِ	*i*
20	غ	*g*	*42*	ّ	*~ (duplicate previous)*
21	ف	*f*	*43*	ْ	*o (not used)*
22	ق	*q*	*44*	إ	*> (replaced with) I*

In fact, using either Roman or Arabic characters is a non-issue since there is no real difference between both types of transcription. It is just a matter of handling the textual characters. Therefore, the performance of the ASR recognizer based on Arabic characters should have no significant difference from one based on Roman characters. However, Romanization is harder to read for the native Arabic speaker. Of course, it is straightforward to do the transliteration in both directions in case of the need to output in the Arabic transcription.

In addition to the phonemes set shown in Table 1, we use three more phonemes to represent the Fatha that proceeds Alif (ا) ➔ ـا as a single phoneme that is (aA), the Damma that proceeds Waw (و) ➔ ـُو as a single phoneme that is (uw), and the Kasra that proceeds Ya (ي) ➔ ـِي as (iy). The reason for handling these cases as a single phoneme is that the pronunciation of the short vowels is different when it proceeds the long vowels. Hence, it will be correctly transcribed as single phonemes. Therefore, we used 46 phonemes for the HTK system, 42 phonemes in Table 1 (Shadda (ّ<>~) and Sukun (ْ<>o) are discarded) in addition to the three phonemes (aA, iy, and uw) and the sil phonemes to indicate silence. In the Sphinx system, decoding fails when the phonemes set contains small letter and capital letter symbols. For instance, it fails when the phonemes set has a phoneme b and another phoneme B. However, this problem has not been observed in the HTK system. Hence, we used a new phonemes set for Sphinx that contains 45 phonemes (ّ is not counted since it is duplicated) as shown in Table 2. The phonemes set contains 46 phonemes as the set that is used for the HTK system. Hence, both systems have the same phonemes set.

TABLE 2. Proposed phonemes set for the Sphinx recognizer

#	*Letter*	*Phoneme*	#	*Letter*	*Phoneme*	#	*Letter*	*Phoneme*
1	ء	*E*	*17*	ر	*R*	*33*	ه	*H*
2	آ	*AA*	*18*	ز	*Z*	*34*	و	*W*
3	أ	*O*	*19*	س	*S*	*35*	ى	*AY*
4	ؤ	*EW*	*20*	ش	*SH*	*36*	ي	*Y*
5	إ	*I*	*21*	ص	*SS*	*37*	ً	*AU*
6	ئ	*EY*	*22*	ض	*DD*	*38*	ٌ	*AWW*
7	ا	*A*	*23*	ط	*TT*	*39*	ٍ	*AIY*
8	ب	*B*	*24*	ظ	*ZZ*	*40*	َ	*AU*
9	ة	*P*	*25*	ع	*AE*	*41*	ُ	*AW*
10	ت	*T*	*26*	غ	*GH*	*42*	ِ	*AI*
11	ث	*TH*	*27*	ف	*F*	*43*	ّ	*duplicate*
12	ج	*J*	*28*	ق	*Q*	*44*	ـا	*AUA*
13	ح	*HH*	*29*	ك	*K*	*45*	ـُو	*AWW*
14	خ	*KH*	*30*	ل	*L*	*46*	ـِي	*AIY*
15	د	*D*	*31*	م	*M*			
16	ذ	*DH*	*32*	ن	*N*			

The pronunciation dictionary is an ASR component that contains the phonetic transcription of each word. That is, each word that appears in the training text is listed in the dictionary in terms of phonemes. Nevertheless, generating a pronunciation dictionary is a hard task due to the different acoustic cases of pronunciations. In the case of a large number of words (i.e., a sizable vocabulary), it is quite time-consuming. In this work, we used a Python program to generate the pronunciation dictionaries for both the Sphinx and HTK recognizers. In the literature, there are various studies that

describe the rules to generate a pronunciation dictionary for MSA. For instance, Ramsay et al. [22] described a knowledge-based approach to generate the phonetic transcription for MSA. Ali et al. in [23] presented a tool for generating phonetic dictionaries for MSA.

Figure 1 shows entries of the dictionaries that are used in this work. The figure shows the phonemes sequence of each word that appears in the training textual data. The pronunciation of each word is used to model the acoustic model during the training phase. For instance, the word "الوَزَارَة" is mapped to the sequence "A L W AU Z AUA R AU P". Hence, wherever this word appears, it is replaced with the phonemes sequence in order to produce the phonetic transcription of the speech file. Each phoneme is then used to train a part of the acoustic signal that corresponds to a phoneme name. This process (i.e., the training stage) is performed using the Baum-Welch algorithm to create a single hidden Markov Model (HMM) for each phoneme in the phoneme list. The figure also shows two sets of phonemes: the phonemes set for Sphinx (on the left of the figure), which is proposed in this work, and the HTK set (on the right of the figure), which is based on the BW transliteration. Creating a single HMM model for each phoneme is called the 'context-independent' phase (CI). After the CI stage, the training stage continues to perform the untied 'context-dependent' phase (CD) for creating triphones; finally, the tied context-dependent phase is used for tying some HMM states.

Arabic-based dictionary (for Sphinx)	*Roman character-based dictionary (for HTK)*
...	...
الوَزَارَة A L W AU Z AUA R AU P	AlwazaArap A l w a z aA r a p
الوَزِير A L W AU Z AIY R	Alwaziyr A l w a z iy r
الوَزِيرَ A L W AU Z AIY R AU	Alwaziyra A l w a z iy r a
الوَزِيرَة A L W AU Z AIY R AU P	Alwaziyrap A l w a z iy r a p
الوَزِيرُ A L W AU Z AIY R AW	Alwaziyru A l w a z iy r u
الوَزِيرِ A L W AU Z AIY R AI	Alwaziyri A l w a z iy r i
الوَزِيرة A L W AU Z AIY R AI P	Alwaziyrip A l w a z iy r i p
...	...

FIGURE 1. Entries of the pronunciation dictionaries

In both implemented systems, we consider the pronunciation case of Shadda as follows: for the Shadda case, we first remove the germination marker "ّ" and then duplicate the previous consonant. Another case where an unnecessary symbol is deleted is the 'Sukon' "ْ". Ramsay et al. in [22] described pre-processing cases that should be considered for phonetic transcription. For illustration, Figure 1 shows some cases where the semi-vowels y, w and A are written with a preceding short vowel, so it is assumed that iy, uw and aA are two-character names for the relevant vowels. The same is found in the Sphinx dictionary (i.e., AUA, AWW, and AIY).

4. Language Models. (LMs) are considered to be a significant contribution to the performance of NLP systems such as ASR and machine translation. LMs have been successfully applied in different linguistic applications such as Part-of-Speech (PoS) tagging, parsing, information retrieval, spell correction, summarization, etc. In particular, LM is a critical component in linguistic applications producing sequences of words as output. In the last few decades, extensive research has been devoted to promoting new techniques to compile LMs as well as to address challenges such as missing n-grams. In speech recognition, the ASR decoder uses the information provided by the LM to find the best possible word sequence of the testing

speech for transcription purposes. In general, it is extremely important for language applications to have the ability to predict the next word given by the previous word(s), or the history.

LMs can be either probabilistic or non-probabilistic. The probabilistic LMs are known as statistical LMs such as n-grams, while non-probabilistic LMs are known as 'any-word' grammar. Any-word grammar does not use probabilities of words. It is unconstrained grammar that leads to very poor accuracy in continuous ASR systems. That is, any-word grammar relies entirely on the acoustic model. On the other hand, statistical language models are based on computing the probabilities of all word combinations (i.e., all possible word sequences) in the training source text. Statistical LMs are generally demonstrated using ARPA format textual files that include the statistical estimation of the desired n-grams, typically up to 3-grams. Compiling a statistical language model requires a large number of words from different textual resources. In addition, the data should not be too specific to a particular domain; otherwise, it will not generalize well to the sentences in question.

During ASR decoding, the recognizer employs the language model to transcribe speech files using the acoustic model and the pronunciation dictionary (i.e., the vocabulary). Hence, the most likely hypothesis for each testing speech file is generated as an output. Employing language models reinforces the speech recognition accuracy, as the more you can constrain the range of possible utterances, the more accurate the recognizer will be. Based on the information provided in the n-grams, the probability of a word sequence is computed using the following formula [25]:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_1^{k-1}) \qquad (1)$$

where 'n' is limited to include the history of the word. Hence, the Chain Rule is applied to compute the joint probability of words in a sentence. LMs utilize the Markov assumption to simplify data estimation. For instance, for n=2, the bigram is calculated for the word sequence as follows:

$$P(w_1\, w_2) = P(w_2\,|w_1)P(w_1\,) \qquad (2)$$

For speech features, Mel Frequency Cepstral Coefficients (MFCC) are the classical front-end analyses used in speech recognition to produce the sequence of real-valued numbers that represent feature vectors based on the input signal. Since 1980, it has dominated the ASR feature extraction method due to its good performance. The success of MFCC makes it the standard choice in the state-of-the-art speech recognizers such as the CMU Sphinx, HTK, and Kaldi speech recognizers. Reference [24] has some details of MFCC. Given the speech feature vectors, the most likely sequence of words is estimated by the following [25]:

$$\widehat{w} = \text{argmax}_{W \in L} \frac{P(O|W)P(W)}{P(O)} = \text{argmax}_{W \in L}\, P(O|W)P(W) \qquad (3)$$

Where $\widehat{w}$ is the most likely recognized words, $P(O|W)$ is the probability of the feature vectors, given a sequence of words that is computed using the acoustic model, and $P(W)$ is the probability of the words sequence that is computed using the language model. $P(O)$ is the probability of the acoustic observation sequence and can be ignored. Hence, the statistical LM has to be computed first to decode the testing speech files in ASR systems. The statistical n-grams LM is trained by counting n-grams occurrences in a large transcription corpus to then be smoothed and normalized. Counting and normalizing can train the n-gram models. The following formula is used to estimate the n-grams parameter [25]:

$$P(w_n \big| w_{n-N+1}^{n-1}) = \frac{Count(w_{n-N+1}^{n-1} w_n)}{Count(w_{n-N+1}^{n-1})} \qquad (4)$$

One major problem in LMs is unseen words or n-grams that are found in the testing set while out of vocabulary. Accordingly,

a probability of 0.0 is given to the items that are not seen in the training data. That is, not all n-grams will be present (i.e., not observed) in the training data. One solution is smoothing by assigning non-zero or small probabilities to the unseen n-grams in which all word sequences can occur with some probability. Hence, smoothing provides a better way of estimating the probability of zero frequency n-grams that never occur in order to produce generalized LMs. Smoothing is also called discounting. When creating a language model, it is more efficient to use log probabilities rather than actual probabilities due to the risk of numerical underflow, especially for very long strings. It is also efficient in ASR decoding algorithms such as the Viterbi algorithm.

Creating an n-gram LM entails the following three main steps: compute the word unigram counts, convert the word unigram counts into a vocabulary list, and generate bigram and trigram (or more) tables based on this vocabulary. As a preprocessing step, it is essential to include special words such as <s> in order to indicate the "start of sentence" and the </s> to indicate the "end of the sentence". The CMU Cambridge University toolkits [26] use the <UNK> token to indicate unknown words, whereas HTK toolkits [2] use !!UNK for the same purpose.

Sometimes ASR recognizers employ the grammar of LMs for small applications with limited isolated words. Such grammar is simple and does not have probabilities; they are designed according to the information that is provided in the corresponding application. That is, grammar mainly contains isolated words such as commands, control words, and digits. However, grammar might allow sequences of words. Figure 2 shows a simple grammar for ten digits that can be used in continuous speech recognition to choose one word or more from the list. The grammar is written using JSGF format, as shown in Figure 2. The star (*) in Figure 2 means zero or more words. This is helpful when this grammar is used in continuous speech in order to recognize a sentence that has many words.

Ten digit grammar (for Sphinx)
#JSGF V1.0; grammar myGrammar; public <command> = <word>* ; <word> =(One | Two | Three | … | Ten);
Ten digit grammar (for HTK)
$WORD = (One \| Two \| Three \| … \| Ten); ($WORD)

FIGURE 2. Any-word grammar for ten digits

Grammar is usually written by hand, or it can be generated using a program. Most grammar does not use probabilities; however, some elements might be weighted. In fact, grammar is rarely used in ASR systems since the probabilistic models of a language are more useful than the hard models (i.e., grammar) for legal sentences in various languages.

There are a number of toolkits that are used to compile LMs such as the CMU-Cambridge LM toolkit [26], the Cambridge University HTK language modeling tools [2], and the SRI Language Modeling Toolkit (SRILM) [27]. The CMU-Cambridge toolkit uses the following five commands to produce the LM dump file: text2wfreq, wfreq2vocab, text2idngram, idngram2lm, and lm3g2dmp [28]. HTK provides two approaches to generate n-grams. The first method employs the HLStats function, which is used exclusively to compute the bigram language model (i.e., 2-grams). The second method implements a series of functions to compute n-grams. In this work, we used the first approach (i.e., bigram statistical LM). In addition to HLStats, HBuild is used to create the word

network that describes the allowable word sequences, of course, with the corresponding probabilities. Figure 3 shows a simple 3-gram LM for three sentences as shown in the figure. It was created using the CMU-Cambridge toolkit.

A small corpus of three sentences

<s> ووقع الحانيان منكرات تقاهم في الاقتصاد والتربية والتعليم </s>
<s> ضخ المزيد من الاموال من اجل نمو الاقتصاد المحلي </s>
<s> نمو الاقتصاد المحلي الذي يعتبر المحرك الاقتصادي للدولة </s>

```
\data\
ngram 1=23
ngram 2=26
ngram 3=28

\1-grams:
-1.5168 <UNK>    0.0000
-1.1614 </s>     -0.4297
-0.9853 <s>      0.0604
-1.5168 اجل      0.0310
-0.9853 الاقتصاد -0.4317
...
-1.5168 ووقع     0.0134
-1.5168 يعتبر    0.0134

\2-grams:
-0.1761 </s> <s> 0.0000
-99.9990 <s> 0.0000 ضخ
...
-99.9990 والتعليم </s> 0.4771
-99.9990 0.0000 ووقع الحانيان
-99.9990 0.0000 يعتبر المحرك

\3-grams:
-99.9990 </s> <s> ضخ
-99.9990 </s> <s> نمو
-99.9990 <s> ضخ المزيد
...
-0.1761 نمو الاقتصاد المحلي
-99.9990 والتربية والتعليم </s>
-99.9990 والتعليم </s> <s>
-99.9990 ووقع الحانيان منكرات
-99.9990 يعتبر المحرك الاقتصادي
\end\
```

FIGURE 3. Three sentences with the 3-grams LM using the CMU-Cambridge tool

For this work, we used 2-grams statistical language models as shown in Figure 4 for both systems. In fact, the 3-grams language model is more commonly used for ASR tasks; however, we used the 2-grams language model in this work due to the restriction of HVite. HVite uses 2-grams, while HTK HDecode can use 3-grams. PocketSphinx can use either 2-grams or 3-grams language models.

CMU-Cambridge tool

```
\1-grams:
-1.0195 <UNK>    -0.0128
-1.7589 </s>     -3.2964
-1.7587 <s>      -0.3490
-4.7617 أبنائهم  -0.1442
-5.0627 أكون     -0.1445
-4.7617 آثارًا   -0.10095
...

\2-grams:
...
-0.5481 يوفر لها
-0.5481 يوافق الجميع
-0.5481 يتفوق على
\end\
```

HTK tool

```
\1-grams:
-99.999 !ENTER  -3.5976
-5.0703 AAlTTaAEap      -0.3010
-5.0703 AEatabara       -0.2958
-5.0703 AEatabirat      -0.3010
-5.0703 AEtabirahu      -0.3010
-5.0703 AHtiraAmi       -0.3010
...

\2-grams:
...
-0.3010 zumalaAQihum AlAinDimaAma
-0.3010 zumalaAwuhu AlnnuwwaAb
-0.3010 zuwmaA sayazwarruwna
\end\
```

FIGURE 4. Parts of the 2-grams LM that are used in this work

5. **Implementation of the Sphinx and HTK methods.** The Sphinx and HTK methods used Cygwin, which is a Unix-like environment for Windows. However, it is preferred to run the command line in a UNIX-based system rather than Cygwin that is installed for the Windows environment. Implementing Sphinx for Arabic ASRs includes the steps described in [29]. The first step includes creating the directory where the files live. The files include the training and testing speech files, the transcription of the entire speech collection, and other necessary files that are used for training and decoding. In particular, the speech files are stored in the wav directory, while the ***etc*** directory has the following files: the pronunciation dictionary, the phonemes file, the list of fillers, the list of

files for training, the transcription for training, the list of files for testing, and the transcription for testing. Of course, the language model also lives in the ***etc*** directory. The following are the main commands (for task1, for instance) that are used in Sphinx:

- $ mkdir task1 (create the main directory)
- $ sphinxtrain -t task1 setup (create the structure of the main directory)
- $ sphinxtrain run (start training, once done, it provides the word error rate (WER))

For HTK implementation, reference [30] presents comprehensive details for training and decoding as the following steps:
Step 1 - the Task Grammar, Step 2 - the Dictionary, Step 3 - Recording the Data, Step 4 - Creating the Transcription Files, Step 5 - Coding the Data, Step 6 - Creating Flat Start Monophones, Step 7 - Fixing the Silence Models, Step 8 - Realigning the Training Data, Step 9 - Making Triphones from Monophones, Step 10 - Making Tied-State Triphones, and finally, Step 11 - Recognizing the Test Data. In the previous sections, we demonstrated some of the steps such as the language model and pronunciation dictionary. To train a model, a further set of files is needed such as the following:

- words.mlf: this is just a rearranged version of the training textual files,
- train.scp: a list of the training speech file names,
- phones0.mlf: the phonetic transcriptions, obtained by substituting the entries in the pronunciation dictionary for the words in the textual transcriptions,
- monophones0: list of the phones that appear in phones0.mlf,
- codetrain.scp: pairs linking .wav files to .mfc files (the MFCC speech features),
- proto.txt: the "flat start" file for the hidden Markov models (HMMs),
- config.txt that contains some parameters related to the speech features.

After creating the necessary files in the same directory where the speech files and the speech features (.mfc) reside, it is possible to start training. This produces trained models that will be used for decoding. During training, many functions are executed, such as the HLEd and HERest functions. For decoding, HVite is used.

To investigate the performance, we split the speech corpus (15.94 hours) into two parts including the training set that contains 12.74 hours (1,611 speech files) and the testing set that contains 3.19 hours (403 speech files). That is, the testing set is 20% of the overall speech corpus. The speech files were prepared to have a fixed length between 10-40 seconds, mono, and sampled at 16 kHz. The average length of the textual files is 55 words. The total number of speakers in the corpus is 29 native Arabic speakers (19 males and 10 females). In this work, we used three emitting states of HMMs that corresponded to the subphones at the beginning, middle, and end of the phones. The acoustic models were calculated using context-dependent HMM triphones. Regarding Sphinx recognizers, the acoustic models are trained using the SphinxTrain for the phonetic tied-mixture (PTM). However, other acoustic model types can be used such as semi-continuous and fully continuous models. The performance is measured based on different parameters such as the number of Senones and the number of Gaussian densities. Table 3 shows the WER for different cases.

TABLE 3. The performance of the Sphinx recognizer

Experiment	*Densities*	*Senones*	*WER (%)*	*Accuracy (%)*
1	*8*	*500*	*22.6*	*77.4*
2	*8*	*1000*	*22.2*	*77.8*
3	*8*	*2000*	*21.5*	*75.5*
4	*16*	*500*	*21.8*	*78.2*

5	*16*	*1000*	*21.1*	*78.9*
6	***16***	***2000***	***20.7***	***79.3***
7	*32*	*500*	*21.8*	*78.2*
8	*32*	*1000*	*21.3*	*78.7*
9	*32*	*2000*	*21.3*	*78.7*
10	*64*	*500*	*21.9*	*78.1*
11	*64*	*1000*	*21.9*	*78.1*
12	*64*	*2000*	*21.9*	*78.1*
13	*128*	*500*	*21.7*	*78.3*
14	*128*	*1000*	*22.6*	*77.4*
15	*128*	*2000*	*22.1*	*77.9*
16	*256*	*500*	*21.8*	*78.2*
17	*256*	*1000*	*22.2*	*77.8*

Table 3 shows that the best (lowest) WER was found to be 20.7% using 16 Gaussian densities and 2000 senones. We emphasize that these results are based on the language model that contains both the training and testing transcriptions. However, if the langue model contains only the training transcription, the results will be less than what we scored in Table 3. The reason for using the training and the testing sets to create the language model is that the language model requires a large amount of data that is not available in our work. During experiments, we considered speeding up the execution time using the PocketSphinx configuration (i.e., number of parts to run Forward-Backward estimation ➔ $CFG_NPART = 10; and how many pieces to split decode in➔ $DEC_CFG_NPART = 10;). The number 10 is just an optional number that can be fixed based on the user preferences. This option is helpful since it clearly reduces the execution time by utilizing a number of processors in multicore machines. For the HTK recognizer, we found the performance to be less than what we achieved using the Sphinx recognizer. The WER is (100%-65.31%=34.69%). However, the lowest WER in the Sphinx is 20.7%. Figure 5 shows the HTK overall results.

```
====================== HTK Results Analysis =======================
  Date: Sun Jun 18 19:01:41 2017
  Ref : words.mlf
  Rec : recout.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=0.00 [H=0, S=403, N=403]
WORD: %Corr=67.50, Acc=65.31 [H=15582, D=1044, S=6459, I=506, N=23085]
===================================================================
```

FIGURE 5. Performance of the HTK recognizer

Regarding the output, Sphinx gives the results using Arabic characters. However, the HTK gives Romanized characters. Figure 6 shows an example of the output of the HTK recognizer. In the figure, Arabic characters (the upper part in Figure 6) represent the transcription of a speech file after recognition, and the lower English characters are the Romanized text as recognized by the HTK HVite decoder.

FIGURE 6. The recognition output of a speech file using the HTK

In this work, we used the default settings of HTK. The default settings include the thresholds for outlier removal (RO=100) and the tree branch threshold (TB=350). TB is used for the decision of tree clustering of states. Both RO and TB affect the degree of tying and therefore the number of states output in the clustered system [30]. In future work, it is worth evaluating the performance using different values of RO and TB. It is also worth employing the HDecode, which is an HTK extension decoder released on a restricted basis.

6. **Conclusion.** This work discusses the implementation of two well-known speech recognizers; the CMU Sphinx, and the HTK. It includes a comparative study of both recognizers using a continuous speech corpus of MSA. The results show that Sphinx outperforms the HTK recognizer. Sphinx is also better in some issues such as handling long speech files, since some of the long speech files were discarded due to failure execution using the HTK (i.e., the training fails using long speech files). Sphinx is also better in terms of execution time as it takes less training and decoding time compared to the HTK. Finally, we have found that it is easier to perform an ASR task with Sphinx than HTK. The only issue with Sphinx is that it fails when the phonemes set has capital and small letters. For instance, if we use the character <B> to indicate a specific phoneme and, at the same time, use the character <b> to indicate another phoneme, then we get an error during training. On the other hand, this error did not appear in the HTK system. We also have found that HTK is better documented than Sphinx. In conclusion, more research is required to understand the reasons for the performance difference between both systems.

Acknowledgment. This work was supported by the Kuwait University Research Administration Research, Project Number EO06/12.

REFERENCES

[1] Available: https://cmusphinx.github.io/
[2] Available: http://htk.eng.cam.ac.uk/
[3] Gaida, Christian, et al. "Comparing open-source speech recognition toolkits." Tech. Rep., DHBW Stuttgart (2014).
[4] Available: http://kaldi-asr.org/doc/index.html
[5] Available: http://www.qamus.org/transliteration.htm
[6] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.
[7] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.
[8] AbuZeina, Dia, et al. "Cross-word Arabic

pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[9] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[10] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[11] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology(2017): 1-9.

[12] Kirchhoff, Katrin, et al. "Novel approaches to Arabic speech recognition: report from the 2002 Johns-Hopkins summer workshop." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 1. IEEE, 2003.

[13] Vergyri, Dimitra, et al. "Morphology-based language modeling for arabic speech recognition." INTERSPEECH. Vol. 4. 2004.

[14] Kirchhoff, Katrin, et al. "Morphology-based language modeling for conversational Arabic speech recognition." Computer Speech & Language 20.4 (2006): 589-608.

[15] Satori, Hassan, Mostafa Harti, and Nouredine Chenfour. "Introduction to Arabic speech recognition using CMUSphinx system." arXiv preprint arXiv:0704.2083 (2007).

[16] Elmahdy, Mohamed, et al. "Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition." Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on. IEEE, 2009.

[17] Al-Qatab, Bassam AQ, and Raja N. Ainon. "Arabic speech recognition using hidden Markov model toolkit (HTK)." Information Technology (ITSim), 2010 International Symposium in. Vol. 2. IEEE, 2010.

[18] Graja, Marwa, Maher Jaoua, and L. Hadrich Belguith. "Lexical study of a spoken dialogue corpus in tunisian dialect." The international arab conference on information technology (acit), benghazi–libya. 2010.

[19] Merad-Boudia, N., Benyettou, A., Rubio Ayuso, A., Arabic Speech Recognition for Connected Words Using HTK: Triphones Expanded to Gmm Based Quran Recognition, (2016) *International Review on Computers and Software (IRECOS)*, 11 (12), pp. 1209-1216.

[20] Haraty, Ramzi A., and Omar El Ariss. "CASRA+: A colloquial Arabic speech recognition application." American Journal of Applied Sciences 4.1 (2007): 23-32.

[21] Alotaibi, Yousef Ajami. "Comparative study of ANN and HMM to Arabic digits recognition systems." Journal of King Abdulaziz University: Engineering Sciences 19.1 (2008): 43-59.

[22] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[23] Ali, Mohamed, et al. "Arabic phonetic dictionaries for speech recognition." Journal of Information Technology Research (JITR) 2.4 (2009): 67-80.

[24] Al-Anzi, Fawaz S., and Dia AbuZeina. "The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition", World Academy of Science Engineering and Technology, International Journal of Computer and Information Engineering, Vol:11, No:10, 2017

[25] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. Pearson, 2014.

[26] Available: http://www.speech.cs.cmu.edu/SLM/toolkit.html

[27] Available: http://www.speech.sri.com/projects/srilm/

[28] Available: http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

[29] Available: https://cmusphinx.github.io/wiki/tutorialam/

[30] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

CONFERENCE PAPER(S)

From: infos2016@fci-cu.edu.eg
Date: 10 March 2016 at 10:16:24 pm GMT+3
To: Fawaz Al-Anzi <alanzif@yahoo.com>
Subject: [INFOS 2016] Your paper #1570241498 ('Arabic Speech Recognition: A Survey of the Literature')

Dear Prof. Fawaz Al-Anzi:

On behalf of the Technical Program Committee, I am pleased to inform you that your paper 1570241498: "Arabic Speech Recognition: A Survey of the Literature" submitted to Track INFOS'16-DSTA has been accepted for presentation at INFOs2016 in Cairo, Egypt.

The paper is accepted contingent upon:

1) your completion and submission of the camera-ready final version of the technical manuscript for the paper following ACM strict format,

2) receipt of the signed copyright form, and

3) your pre-registration for the conference (One full registration must be paid for paper publication. One registration can cover up to two papers for the same author).

All the above items must be completed by 31 March 2016, carefully following the instructions that will be published online.

We will notify you by email when the detailed registration and final paper submission procedures are posted. Until then, you may check our website http://infos2016.fci.cu.edu.eg/ for regular updates.

Additionally, INFOS 2016 requires that each accepted paper be presented in-person at the conference site according to the schedule published. It reserves the right to exclude from distribution on ACM Digital Library any paper not presented on-site. If none of the authors are able to attend, a qualified surrogate may present the paper, and registrations may be transferred free of charge.

The reviews of your paper are given below and can be found at http://edas.info/showPaper.php?m=1570241498.

Thank you for submitting your paper to INFOS2016, and we look forward to seeing you in Egypt next May!

Sincerely,

INFOS 2016 Program Chairs

Arabic Speech Recognition: A Survey of the Literature

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
Kuwait
0096524985827
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
Kuwait
0096595539196
abuzeina@ku.edu.kw

ABSTRACT
Large vocabulary speaker-independent continuous speech recognition systems have recently received significant attention. However, speech recognition poses some challenges such as varying acoustic conditions, dialects, and articulation at word's boundaries. This study demonstrates a survey of Arabic speech recognition that even has more challenges such as the optional diacritization of the Arabic script. Even though Arabic is a live language that is spreading widely throughout a large area, the research devoted to this technology still in the row stage compared to other languages such as English. In this paper, we highlight the progress made so far in Arabic speech recognition field that include corpora, phonemes, language models, acoustic models, and some promising research directions. This survey reveals that the shortage of freely available continuous speech corpora disserve the research in this domain; it also shows the need to compile large corpora as a key factor to promote the Arabic language research for effective human-computer interaction.

CCS Concepts
Computing methodologies → Artificial intelligence → Natural language processing → Speech recognition.

Keywords
Arabic; speech recognition; corpus; phoneme; language model

----- Forwarded Message -----
From: iit16@uaeu.ac.ae.edas.info
To: Fawaz Al-Anzi <Fawaz.AlAnzi@Ku.Edu.Kw>, Dia AbuZeina <dia.abuzeina@Ku.Edu.Kw>
Cc: Ezedin Barka <ebarka@uaeu.ac.ae>, Yacine Atif <yacine.atif@his.se>, Abderrahmane Lakas <alakas@uaeu.ac.ae>
Sent: Fri, 21 Oct 2016 16:14:52 +0300 (AST)
Subject: [IIT'16] Your paper #1570311911 has been accepted

Dear Prof. Fawaz Al-Anzi:

Congratulations - we are pleased to inform you that your paper #1570311911, entitled: 'Utilizing Long Distance Word Dependencies for Automatic Speech Recognition', has been accepted for presentation in 2016 12th International Conference on Innovations in Information Technology (IIT), and for publication in its proceedings.

Please ensure that you incorporate the reviewers' comments into your camera-ready manuscript, and that the paper conforms to the formatting requirements which are detailed at the conference web site http://www.it-innovations.ae. The camera-ready manuscript should be uploaded through EDAS system by November 15, 2016. Manuscripts uploaded after this date will not be included in the conference proceedings.

Please be advised that at least one of the authors must register for the conference before the camera-ready is uploaded. For information about registration kindly follow this link: http://conferences.uaeu.ac.ae/iit2016/en/registration.shtml.
Conference Online Registration Form - conferences.uaeu.ac.ae<http://conferences.uaeu.ac.ae/iit2016/en/registration.shtml>
conferences.uaeu.ac.ae
Registration. The author and attendee registration is to be done via EDAS using a credit card . In special circumstances, wire transfers are also allowed – email ...

Authors' instructions to prepare the camera-ready version are detailed at the conference website. Only papers presented at the conference will be included in the IEEE Xplore proceedings and considered for Journals submission; otherwise the paper may not be included in IEEE Xplore.

Thank you very much for your contribution to the conference. We look forward to welcoming you to the United Arab Emirates.

Kind regards,
Dr. Ezedin Barka
IIT'16 Program Chair
The 12th International Conference on Innovations in Information Technology (IIT'16)
http://www.it-innovations.ae/

Utilizing Long Distance Word Dependencies for Automatic Speech Recognition

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
abuzeina@ku.edu.kw

***Abstract*—Statistical language models have been widely used in natural language processing (NLP) applications. N-gram has long been proven as a useful words representation technique for language models. However, n-gram assumes that the probability of any word in a sequence of words depends only on the previous n-1 consecutive words. Therefore, investigating the performance of long distance dependencies (LDDs) is an important research area to consider the words' relationships beyond n-1 proceeding words. LDDs aims at finding the words co-occurrences while relaxing the consecutive constraint through a wider window rather than two or three previous words. That is, LDDs are a set of association rules that go beyond the scope of n-gram. One possible use of LDDs is for N-best hypotheses rescoring in automatic speech recognition (ASR) systems. In this paper, we used a textual part of a speech corpus that contains 6,145 short sentences (speech corpora usually have short sentences). The experimental results show that the predictive Apriori data-mining algorithm is a suitable candidate to generate the frequently appeared LDDs that also contains consecutive and nonconsecutive words' relationships. The study also reveals that extracting LDDs is a computation expensive task that requires high performance computing (HPC) environment.**

Keywords—Speech recognition, language model, n-gram, data mining, predictive Apriori.

I. INTRODUCTION

Statistical language models, also called n-gram models, are very popular and successfully used in different computational linguistic fields. For example, statistical language models are integral part of the state-of-the-art automatic speech recognizers (ASR) systems such as Sphinx [1] and HTK [2]. Trigram is the basis of the classical language models that assumes the next words is predicted based on the two immediately preceding words. Hence, n-gram model assumes that a word is only influenced by the (n-1) proceeding words, typically one or two words. It is indicated in [3] that due to memory and computation requirements, the value of (n) is restricted to two or three for bigram or trigram language models, respectively. In speech recognition community, language models also called grammars. Despite language models are known as probabilistic models, however, it could be non-probabilistic (i.e. "any word" grammar).

Language modeling research has long history of improvements as indicated by [4]. He indicated that there are many improvements over trigram simple model including caching, clustering, higher-order n-grams, skipping models, and sentence-mixture models. However, not all of the previous methods proved to be useful. For example, reference [4] showed that even using a very large corpus for n-gram model, very small improvements occurred, where n is larger than 5.

Even the statistical language models are quite success and have proven to be reasonable, however, longer contexts deserved for better language modeling representation. Such longer contexts (i.e. long words' relationships) go by the name of long-distance dependencies (LDDs) that obtained by looking at a wider window beyond n-gram. Discussing LDDs is not new, for example, it is indicated in [5] that the n-gram is weak in terms of capturing LDDs. Therefore, it could be better and more appropriate if the textual corpus used to extract long-distance words relationships. This research discusses a new knowledge-based method to mine the words co-occurrences among a data collection of textual sentences.

In fact, employing LDDs is extremely important since it is the key to the performance of many natural language processing (NLP) applications such as ASR systems, parsing, tagging, translation, etc. In this regard, the important aspect is how to find such words-relationships and how to exploit them in NLP systems. In this study, we propose using the predictive Apriori data-mining algorithm to extract the LDDs. We also propose an algorithm for scoring the N-best list of an ASR system. Intuitively, the proposed method is not a replacement of the celebrated conventional language models, but it is a complementary knowledge base for farther enhancement.

The predictive Apriori algorithm used in this work is available in Weka machine learning tool [6]. In fact, finding LDDs is computational expensive and is feasible for small textual corpora that have relatively short sentences. This is the reason why we chose to demonstrate this work for ASR corpora that usually have short recordings that corresponds to short sentences. The reason of such short recordings is that the speech recognizer should be able during training to align the recordings with their phonetic transcription; if the recordings are long then initial alignment might be fail and therefore causing a problem during training process.

In the next section, we present the language models limitations. Section III presents the literature review, followed by the experiments setup and results in section IV and the ASR

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

N-best rescoring using LDDs in section V. We conclude in section VI.

II. Language Models Limitations

The standard language model has some limitations and constrains as demonstrated by a number of researchers. It is indicated in [7] that words relevant to predicting the next word might lay in any position beyond the scope of a word trigram. The n-gram models are constrained in their inability to take advantage of dependencies longer than *n* in a sentence [3]. The work in [9] demonstrated two drawbacks when considering neighbor words with fixed window size: 1) the actual relation between words within the sentence is ignored, thus the long-distance context cannot be captured for training. 2) Word embeddings learned from flat context are not isomorphic among languages due to the order difference between diverse languages. The classic text representation methods seldom consider the role of the words order in the texts for the semantic representation, and it is supposed that the words are independent of each other [10]. In the same meaning, reference [5] demonstrated that the trigram model is unable to characterize word dependence beyond the span of three successive words.

The study in [8] used clustering as an alternative way of dealing with the data sparseness problem to explore the effectiveness of cluster-based higher-order n-gram models. Similarly, the presented study in [11] shows that the language models do not describe the constraint relationships between words and words or sentences and sentences. They also indicated that LDDs could be used in other domains such as text classification, text clustering, text summarization, and so on. For farther exploring the the deficiency of standard language models, the work in [12] demonstrated that the n-gram model works in terms of discrete units that have no inherent relationship to one another. The work in [13] presented that the local structure (n-gram) constraint is a key limitation in many tasks, since natural language contains a great deal of nonlocal structure (LDDs). The vector space model (VSM) that is widely used for text representation in information retrieval (IR) models assumes that words occur independently (i.e. bags-of-words), which is not exactly appropriate to natural language. It is demonstrated in [15] that many of the context dependencies in natural language occur beyond a three-word window.

Even the Arabic language is widely spoken by hundreds of millions (approximately 400 millions); still, there is little research to improve the the linguistic applications through LDDs or lexical databases. Most research studies found in the literature show that Arabic research focusses on the conventional standard language models that restrict n-gram to the neighboring words with fixed window size. That is, the research in the Arabic language mainly discusses local dependencies (i.e. short-distance model). For the above-mentioned deficiencies of standard language model, we have initiated this research.

III. Literature Review

Capturing long distance dependences was discussed in many studies as follows. An empirical study was presented in [6] on two techniques that used to generate LDDs; linguistically motivated word skipping and predictive clustering. They presented that the two techniques significantly outperform word trigram. The work in [3] used dynamic cache language models and context-free grammars to captures topic-related dependencies of words within and across sentences. Reference [10] proposed head-driven phrase structure grammar (HPSG) for LDDs. HPSG includes three aspects: surface oriented, constraint-oriented and strict lexicalism. A method for capturing the long distance dependency presented in [5] as word activation forces-based language model. A graph-based long-distance dependency method for LDDs language models presented in [11]. Reference [16] discussed the notion of using probabilistic context free grammars for modeling LDDs. The work in [13] showed how to account for the long distance structure with Gibbs sampling, a simple Monte Carlo method used to perform approximate inference in factored probabilistic models. Reference [14] presents topical n-grams, a topic model that discovers topics as well as topical phrases. Reference [15] proposed a language modeling approach to capture the preferred relationships between words over a short or long distance through the concept of mutual-information (MI)-Trigger pairs. A method based on extended the fixed surrounding words presented in [9], the approach based on learning distributed representations from dependency structure of a sentence that can capture long distance relations. A method for capturing the long distance common syntactical rules for the Holy Quran was presented in [24].

IV. Experiment Setup and Resluts

A. Data Set

In this section, we describe the corpus that used to generate the LDDs. The corpus contains 6145 short sentences that belong to sport and economy categories. The total number of unique words in the vocabulary is 11,576. This textual collection is a part of a speech corpus that contains 7.57 hours speech recordings for training and testing. Naturally, the audio files were not used in this study, as generating LDDs only requires the transcription (i.e. the textual forms) of the 6,145 audio files. The maximum sentence's length of the corpus is 30 words. The corpus was originally designed for Arabic continuous speech recognition systems as described in [17].

It is worthy to indicate that we initially started working using Reuters-21578 data set, but it was noticed that extracting LDDs for long sentences with huge vocabulary is infeasible. That is, we run the predictive Apriori algorithm to find LDDs, but we had no results even with so long time waiting, therefore, we stopped execution and moved towards a smaller corpus. Therefore, for efficient extracting LDDs, a high performance computing (HPC) environment is required especially for long sentences. Section VI explains how to exploit LDDs to enhance ASR systems

B. Predictive Apriori

To extract the LDDs, we implemented the predictive Apriori data-mining algorithm. In general, data mining algorithms are used in many areas such as health, marketing, communications, etc. For example, three algorithms (Apriori, Predictive Apriori and Tertius) presented in [19] for analyzing the information available on sick and healthy individuals and taking confidence as an indicator, females were seen to have less chance of coronary heart disease then males. For more information on association rules, reference [20] has a survey on such rules. The predictive Apriori algorithm is a Weka class implementing the predictive Apriori algorithm to mine association rules. It searches with an increasing support threshold for the best 'n' rules concerning a support-based corrected confidence value. The implementation of this algorithm in Weka follows the reference [21], as the rule is added if the expected predictive accuracy of this rule is among the 'n' best and it is not subsumed by a rule with at least the same expected predictive accuracy.

C. Implementation

The corpus described in the previous section was used to obtain the best association rules. The Weka machine-learning tool used to implement predictive Apriori algorithm. The experiments were implemented using a relatively high-speed machine with the following specifications: Intel(R) i7, CPU 3.4GHz, and 16.0 GB of RAM. However, for large data, HPC is required. In our execution environment, it took about 130 processing hours to produce the best 300 rules as shown in Fig. 1. Nevertheless, it took few hours to generate the rules using Apriori algorithm, as the n best rules are not sorted. The results included two types of LDDs relations. The consecutive and the nonconsecutive words relations. The following subsections demonstrated examples of such relationships. Fig. 1 shows the algorithm output starting from the highest accuracies. We chose to extract the best 300 rules. In the figure, x is the shorthand of word, so x5 means the word at the position 5.

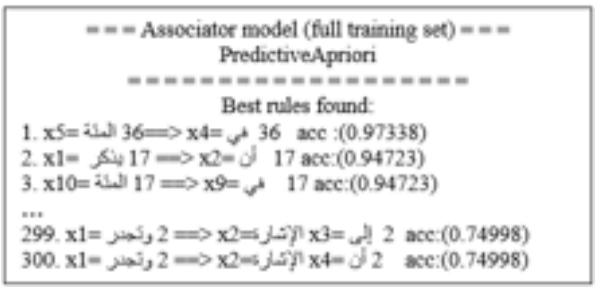


Fig. 1. 300 Best rules Using Predictive Apriori

Fig. 1 shows a part of the best 300 rules, even the rules shown in Fig. 1 are related to consecutive rules; however, there are some other rules that are related to nonconsecutive words as shown in the following subsections. Despite we focus on nonconsecutive words relationships; nevertheless, the consecutive words relationships are also important for NLP applications.

1) Consecutive Words Relations

The standard language model represents words relationships of consecutive words. The predictive Apriori algorithm provides similar relationships sorted based on common rules found in the corpus. The only difference is that the standard language models generate plain words sequences with the corresponding probabilities, while the predictive Apriori algorithm generate the association rules based on words co-occurrences with the corresponding accuracies. Table I presented some examples of such rules. The table also shows the accuracies associated with each rule.

TABLE I. EXAMPLE OF CONSECUTIVE WORDS RELATIONS

Rule type	Example
2 consecutive words	x5=المنة 36==> x4=في 36 acc:(0.97338) The rule indicates that the occurrence of word (“المنة” =”percent”) at the fifth position in the sentence occurred with the word (“في” = preposition means “in, at, on”) 36 times in the entire corpus. The rule also shows that the accuracy of this rule is 0.97. This rule is the highest rule among the obtained rules.
3 consecutive words	x2=ارتفاع x4=النفط 6 ==> x3=أسعار 6 acc:(0.87495) The rules indicates that the word (“ارتفاع” =”rise”) and the word (“النفط” = “oil”) associate with the word (“أسعار”= “prices”). This rule appears 6 times in the entire corpus with accuracy equal to 0.87.
4 consecutive words	x1=المزيد x4=في 6 ==> x2=من x3=التفاصيل 6 acc:(0.87495) This rule shows the relation between four words.

2) Nonconsecutive Words Relations

The nonconsecutive words includes associative rules between two, three, and four words. Table II presented some examples of such relations. The table also shows the accuracies associated with each rule.

TABLE II. EXAMPLE OF NONCONSECUTIVE WORDS RELATIONS

Rule type	Example
2 nonconsecutive words	x1=البالغ 4==> x4=مليون 4 acc:(0.8333) This rule shows the occurrence of a word at position # 1 and a word at the position # 4 among the corpus sentences. These nonconsecutive words relation appears 4 times with accuracy of 0.833.
3 nonconsecutive words	x1=وأربعة 3 ==> x3=سنتا x4=للبرميل 3 acc:(0.79997) This rule shows a rule between a word at the position #1 and a word at the position #3 and a word at the position #4.
4 nonconsecutive words	x2=النصف x3=الأول x6=العام 5==>x5=هذا5 acc:(0.8571) This rule shows a rule between a word at the position #2, a word at the position #3, a word at the position #5, and a word at the position #6.

Even the examples provided show the relations up to four words, however, more than 4 words relations can be extracted if such co-occurrences found in the corpus.

V. N-BEST LIST RESCORING

Speech recognition is the process of converting speech into machine-readable text. Three components are mainly used to perform recognition task; the acoustic models, the pronunciation dictionary, and the language model. An ASR generally has the option to produce the N-best list that contains the best recognition hypotheses. It is indicated in [21] that the Viterbi speech-decoding algorithm is an approximation algorithm. It actually computes an approximation of the most probable word sequence, instead of computing the most probable word sequence. The reason is that the pronunciation variants probabilities' mass is split up among different pronunciations. Therefore, the Viterbi algorithm ignores the correct word that has many-pronunciations and favor an incorrect word with only one pronunciation path. Hence, performance could be enhanced using N-best hypotheses rescoring. Hence, we propose to exploit the extracted LDDs for N-best list rescoring as shown in Fig. 2. The figure shows that the ASR decoder (supposing) recognize a speech file as Sentence 1. However, it might be better if the N-best list is rescored using LDDs as a double check step to obtain better results (it is supposed Sentence 4). The rescoring could be based on counting the number of association rules in each hypothesis. We have not empirically investigated the performance using rescoring step as the number of the obtained rules relatively low. However, it is a future research direction to implement this method for large corpus with very rich rules.

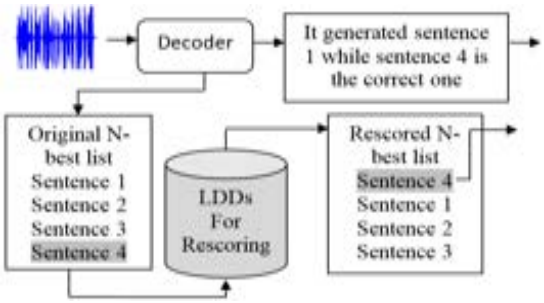


Fig. 2. Rescoring N-best list

Rescoring N-best list is not new. For example, Reference [22] demonstrated the usefulness N-best rescoring using syntactic trigrams. Reference [23] compared the efficacy of a variety of language models for rescoring word graphs and N-best lists generated by a large vocabulary continuous speech recognizer. Hence, the main contribution of this work is to develop an intelligent method that can raise the hypothesis number four (as an example in the top part of Fig. 3) to be the first choice as shown in the lower part of Fig. 3.

Fig. 3. N-best list befor and after rescoring

VI. CONCLUSION

This study presents a new method to obtain the LDDs for ASR N-best rescoring. The method based on predictive Apriori algorithm that generate best association rules. The study shows that generating LDDs is computational expensive and require high-speed machines. It is a future research to utilize LDDs in the implementation of speech recognition systems and other NLP applications.

REFERENCES

[1] Available :http://www.speech.cs.cmu.edu/sphinx/doc/sphinx-FAQ.html
[2] Available :http://htk.eng.cam.ac.uk/
[3] Iyer, Rukmini M., and Mari Ostendorf. "Modeling long distance dependence in language: Topic mixtures versus dynamic cache models." Speech and Audio Processing, IEEE Transactions on 7.1 (1999): 30-39.
[4] Goodman, Joshua T. "A bit of progress in language modeling." Computer Speech & Language 15.4 (2001): 403-434.
[5] Qin, Min, et al. "Word Activation Forces-Based Language Modeling and Smoothing." Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2013 5th International Conference on. Vol. 1. IEEE, 2013.
[6] Available :http://www.cs.waikato.ac.nz/ml/weka/
[7] Gao, Jianfeng, and Hisami Suzuki. "Long distance dependency in language modeling: an empirical study." Natural Language Processing–IJCNLP 2004. Springer Berlin Heidelberg, 2004. 396-405.
[8] Gao, Jianfeng, Joshua Goodman, and Jiangbo Miao. "The use of clustering techniques for language modeling–application to Asian languages."Computational Linguistics and Chinese Language Processing 6.1 (2001): 27-60.
[9] Zhao, Yinggong, et al. "Learning word embeddings from dependency relations." Asian Language Processing (IALP), 2014 International Conference on. IEEE, 2014.
[10] Xu, ZhiHai, et al. "Research on language model of long-distance dependency." 2010 International Conference on Advances in Energy Engineering. 2010.
[11] Zhou, Faguo, and Xingang Yu. "Graph-Based Language Model of Long-Distance Dependency." Asian Language Processing (IALP), 2011 International Conference on. IEEE, 2011.
[12] Mikolov, Tomas, Wen-tau Yih, and Geoffrey Zweig. "Linguistic Regularities in Continuous Space Word Representations." HLT-NAACL. 2013.
[13] Finkel, Jenny Rose, Trond Grenager, and Christopher Manning. "Incorporating non-local information into information extraction systems by gibbs sampling." Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2005.
[14] Wang, Xuerui, Andrew McCallum, and Xing Wei. "Topical n-grams: Phrase and topic discovery, with an application to information retrieval." Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on. IEEE, 2007.
[15] GuoDong, Zhou, and Lua KimTeng. "Interpolation of n-gram and mutual-information based trigger pair language models for Mandarin speech recognition." Computer Speech & Language 13.2 (1999): 125-141.
[16] Manning, Christopher D., and Hinrich Schütze. Foundations of statistical natural language processing. Vol. 999. Cambridge: MIT press, 1999.
[17] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.
[18] Nahar, Jesmin, et al. "Association rule mining to detect factors which contribute to heart disease in males and females." Expert Systems with Applications 40.4 (2013): 1086-1093.
[19] Malik, Meenakshi, and R. P. Agarwal. "A Survey On Association Rule Mining." International Journal of Research in Engineering and Applied Sciences 5.6 (2015): 48-56.

[20] Tobias Scheffer, T. (2005). "Finding association rules that trade support optimally against confidence." Intell. Data Anal. 9(4): 381-3

[21] Jurafsky D, Martin J (2009) Speech and language processing, 2nd edn. Pearson, NJ

[22] Salgado-Garza, Luis R., and Richard M. Stern. "N-Best list rescoring using syntactic trigrams." Mexican International Conference on Artificial Intelligence. Springer Berlin Heidelberg, 2004.

[23] Wang, Wen, Yang Liu, and Mary P. Harper. "Rescoring effectiveness of language models using different levels of knowledge and their integration."Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on. Vol. 1. IEEE, 2002.

[24] AbuZeina, Dia, and Mahmoud Hasan Alsaheb. "Capturing the Common Syntactical Rules for the Holy Quran: A Data Mining Approach." Advances in Information Technology for the Holy Quran and Its Sciences (32519), 2013 Taibah University International Conference on. IEEE, 2013.

International Conference on Computer Applications & Technology ICCAT 2017

28-29 January, 2017 Cairo, Egypt

http://www.iccat.net/

Thursday, 20 October 2016

ACCEPTATION LETTER

Dear Author(s),

Fawaz Al-Anzi and Dia AbuZeina (Kuwait University, Kuwait)

The ICCAT'2017 program committee is pleased to inform you that your paper entitled:

An Empirical Study of Arabic Continuous Speech Recognition Perfomance

submitted for ICCAT'2016 under paper ID: **ICCAT-2017-EDAS-5-1570311915** has been accepted for oral presentation in the conference. Authors are encouraged to present their papers either in person or through web conference. Every paper must have at least one registered author. Please proceed for registration in order to add your paper to the conference program, and have a possibility of publication in one of our proposed journals. Registration and payment of the fees is as per instructions posted on our website: http://www.iccat.net/ before ***November 10th 2016***.

Sincerely yours,
Dr. Rachid Sammouda
ICCAT'2016 General Chair

The Editors of the following journals welcome the submission of extended versions of papers that ICCAT' 2015 program committee deems excellent.

Journal of Information Systems Impact Factor: 1.768 5-Year Impact Factor: 1.838 Imprint: ELSEVIER ISSN: 0306-4379

Journal of Engineering Applications of Artificial Intelligence Impact Factor: 1.625 5-Year Impact Factor: 1.947 Imprint: ELSEVIER ISSN: 0952-1976

Publication chances at ADVANCED ENGINEERING FORUM (ISSN: 2234-9898) Cambridge Scientific Abstracts (CSA) Inspec (IET, Institution of Engineering Technology) EBSCO www.ebsco.com. Thomson Reuters (WoS), all volumes are submitted and selected ones will be indexed.

An Empirical Study of Arabic Continuous Speech Recognition Perfomance

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
Kuwait City, Kuwait
abuzeina@ku.edu.kw

***Abstract*—although considerable research devoted to English speech recognition, rather less attention paid to Arabic speech recognition. The Arabic language is one of the most commonly used languages worldwide that is in need for accurate audio to text converters. In this paper, we evaluate the recognition performance of the Arabic continuous speech using Soundflower Mac utility. That is, Soundflower employed as a speaker-independent continuous speech recognition system to evaluate the word error rate (WER) and the accuracy of the Arabic speech. Hence, this work has no acoustic training phase since the goal is to investigate the trained models in an off the shelf tool. The study also contains a comparative study of the speech recognition performance for male and female native speakers. The experiments conducted using a broadcast news modern standard Arabic (MSA) speech corpus of 2.63 hours (10 male and 10 female speakers). The experimental results show that the accuracy is 54.02 %, and the accuracies of the male and female speakers are almost the same.**

Keywords—Arabic, Speech, Recognition, Corpus, Soundflower.

I. INTRODUCTION

Automatic Speech recognition (ASR) has recently received significant attention as one of successful trend in information retrieval (IR) and intelligent systems. Converting speech into text is an important since it felicitates deploying online audio contents and make it more accessible. However, developing high-quality speech recognition systems is a challenging task and still is a promising research area. Recently, there has been growing interest in speech recognition for the Arabic language as one of the most common languages worldwide. In fact, there is a real need for software tools to transcribe speech into text. Moreover, Arabic is the language of the holy writings of Islam (e.g. the holy Quran) that raises the demand for software to dictate such huge speech resources. Reference [1] indicated that ASR research is currently moving from mere speech-to-text systems towards "rich transcription" systems, which annotate recognized text with non-verbal information such as speaker identity, emotional state for customer care purposes.

Nevertheless, speech recognition is not a straightforward task, as it requires dynamic programming algorithms along with different stages for training and decoding. Reference [2] demonstrated why speech recognition is difficult; among the reasons include body language, noise, differences between spoken language and written language, etc. Therefore, obtaining an accurate freely available software is difficult to achieve. However, free and commercial software tools are available for Arabic speech recognition. In this paper, we consider Soundflower [3] Mac utility that is a free, open-source speech application. The goal is to employ this utility to evaluate the performance of the Arabic speech recognition in terms of word error rate (WER) and the recognition accuracy. The study also aims at comparing the recognition performance of male and female Arabic native speakers. Reference [4] indicated that the performance of speech recognizers for female speakers is usually worse than that obtained for male speakers. In fact, the research in speech recognition contains different sources of pronunciation variations such as continuous or isolated speech, age, gender, emotion, dialects, noise, different accents, etc. Reference [5] presented the main phonetic differences between the speech of male and female speakers. The previous studies on Arabic speech recognition has not considered speaker's gender on speech recognition. The little research in this domain motivates the authors to to take over this research to find the effect of gender in speech recognition.

We have organized the rest of this paper as follows. In the next section, we present the literate review. In section 3, we present speech recognition overview followed by the male and female speech recognition in section 4. The speech corpus information presented in section 5. The experimental results presented in section 6. Finally, conclusion and future work presented in section 7.

II. LITERATURE REVIEW

In this section, we survey the reported contributions of Arabic speech recognition. Because speech recognition is a wide multidiscipline topic that contains vast and diverse subtopics, this literature focuses on the software tools developed for dictating (i.e. audio-to-text) Arabic speech. Soundflower [3] is a free audio system extension that allows applications to pass audio to other applications as one usage option. However, it has another option that is a speech to text converter with the following characteristics based on [6]: Soundflower is a Mac system extension, easy to use, simply presents itself as an audio device, allowing any audio application to send and receive audio with no other support needed.

Sakhr software company developed a commercial ASR [7] engine that has some features such as noisy environments, speaker independent, high accuracy, supports different Arabic accents. The DARPA-funded Babylon project [8] contains Arabic speech recognition as a part of the developed speech-to-speech translation systems. Hidden Markov Model Toolkit

This work is supported by Kuwait University Research Administration Research Project Number EO06/12. We also thank Al-Sabah TV in Kuwait as a source of the the speech collection and Shatha Hassan as the Mac machine administrator.

(HTK) [9] is a portable toolkit for speech recognition research. However, the HTK assumes that the textual files written using ASCII rather than Unicode, so if the training input text is stored using the standard Arabic character set then it has to transcribe to something that the HTK can handle. The obvious thing to use is the Buckwalter transcription [10]. CMUSphinx toolkit [11] is another option in the research community that used to build speech recognition systems. CMUSphinx is an open source speech software from Carnegie Mellon University (CMU) [12]. For example, Reference [23] employed CMUSphinx for cross-word Arabic continuous speech recognition. Unlike HTK, CMUSphinx does support Arabic language that used directly within the CMUSphinx components such as phonetic dictionaries and the language models. Choosing either HTK or CMUSphinx depends on some aspects such as implementation structure, supporting mobile platform, programming language, etc. nevertheless, both well-known ASR engines share the theoretical background for training and decoding that should give relatively similar outputs.

As existing literature shows, little work devoted to serve the Arabic language compared to the English language. Dragon [13] is an example of software that used to convert audio text for English. The developer [13] claimed that Dragon is the fastest and most accurate way to interact with your computer. Gotranscript [14] provides speech recognition service for English. They listed some features of the product such as uncompromising quality, rates within the budget, highly accurate transcripts, timely and convenient delivery. Google [15] cloud speech application program interface (API) enables developers to convert audio to text by applying powerful neural network models in an easy to use API. Reference [16] lists the best 2016 voice recognition software for English. Reference [17] compared the performance of three commercially available continuous speech recognition software packages for the English language. The packages include the IBM software that was found to have the lowest mean error rate (7.0 to 9.1 percent) followed by the L&H software (13.4 to 15.1 percent) and then Dragon software (14.1 to 15.2 percent).

III. Speech Recognition Overview

Speech recognition mainly contains two stages, training and decoding. The training stage requires two datasets: a set of speech files and a set of files containing the phonetic transcriptions of the speech files. There are various ways of getting phonetic transcriptions. The easiest is to use phonetic dictionary in combination with the training textual transcription. Some ASR engines such as HTK have a tool for doing this, or it can be prepared manually. Writing a phonetic dictionary is hard, and if the vocabulary has many words then it will be quite time-consuming. For Arabic, it is reasonable to approximate each Arabic character to a single phoneme. So, for instance, assuming that the phonetic transcription of "kataba" is "k a t a b a", Buckwalter transcription [6]. This method of transcription has two advantages, namely that everyone uses it, so that data can easily be made available to other people and it let the researchers to use other people's data; and that it uses one Roman character for each Arabic character, which is helpful, and which most of the other options don't do. There is, however, a problem, which is that it uses a number of non-alphabetic characters that have a reserved meaning in some ASR engines. Another option to represent words in the phonetic dictionary is by using Arabic characters such as "كَتَبَ" with the the phonemes "K AE T AE B AE", as an example. Reference [18] has more information of how generate phonemes (could say the phonetic dictionary) for Arabic words. Of course, there are other ways to generate phonetic dictionary for better performance. Linguistic scholars and phonetic specialists might help to in this regards.

In addition to the phonetic dictionary, the training stage also contains declaring language models that also called grammars. There are all sorts of kinds of grammars to use. The choice of the grammar is, indeed, the key to the performance of the recognizer. The more of constrains in the range of possible utterances, the more accurate the recognizer will be. In general, one can extract two types of grammars from a set of training textual transcription. One says that the target utterance may be an arbitrary sequence of words drawn from the training textual transcription (in short "any word" grammar); the other says that it must be one of the training textual transcription. The first is almost entirely not constraining, and leads to very poor accuracy (but lets researchers experiment with the effects of different transcriptions, because it relies entirely on the acoustic model); the other is very tightly constraining, and often leads to 100% accuracy. Naturally, there are other options to write grammars such as probabilistic N-Grams, the well-known approach for language modeling.

Using the phonetic transcriptions of the textual versions of the training speech, the audio files, and the list of phonemes, we can start training phase using the desired machine-learning tool such as hidden Markov models (HMMs). The output of the training stages is the acoustic models that used for testing, also called decoding process. The grammars are required throughout testing process. The testing stage employs a dynamic programming algorithm such as Viterbi algorithm to find the most likely phonemes sequence to find the textual words sequence of the spoken words. In fact, speech recognition is a complicated process that needs to handle different aspects such as Gaussian mixtures model, speech features such as Mel-frequency cepstral coefficients (MFCCs), Baum–Welch algorithm, triphone, pruning, etc.

MacOS recently introduced dictation (speech-to-text) as a feature usable in any application that takes text as input [19]. Reference [19] presented some technical issues that help to run Soundflower application. Fig. 1 shows the Soundflower starting page.

Figure 1. A snapshot of the Soundflower speech application

IV. Male and Female Speakers

One goal of this work is to investigate the speech recognition performance of male and female Arabic speakers. The research

on Arabic speech recognition has tended to focus on mixed male-female speech recognition rather than on gender based speech recognition. That is, the training corpus usually has mixed male and female speech that ignore the acoustic differences between female and male voices. Vogt in reference [20] indicated that the differences in speech features for male and female speakers are a well-known problem and the gender-dependent emotion recognizers perform better than gender-independent ones. Reference [21] separated the training dataset based on the gender. This separation yielded gender dependent HMMs that found significantly improve the word recognition accuracy over the gender independent method. Reference [4] indicated that separating training corpora into male and female acoustic-phonetic models is a common solution to enhance the speech recognition performance.

V. THE SPEECH CORPUS

The speech corpus used in this work is an in-house corpus that contains of 275 audio files recorded by 20 Arabic native speakers (10 male and 10 female). Each male speaker utter 15-speech items, while some of female speakers utter less than 15-speech items (see Table III). The speech files mainly contains local and international news recorded from Al-Sabah TV channel in Kuwait. The modern standard Arabic (MSA) is the language used by all speakers. The speech file were prepared to have a fixed length between 30-60 seconds. The speech items sampled at 16 kHz and sum up to 2.63 hours of speech. The training textual transcription of the speech files were prepared by transcribing the audio files according to speakers' utterance. Table I composed of the corpus information. We emphasize that the corpus prepared in this work used for testing. That is, we have no training part in this work, as the goal is to investigate the trained models in Soundflower.

TABLE I. THE CORPUS INFORMATION

#	Gender	Number of Speakers	Number of speech files	Length (hour)	Number of Unique words
1	Male	10	150	1.53	5,149
2	Female	10	104	1.10	3,738
	Total	**20**	**254**	**2.63**	**8,887***

*the number of unique words in the entire corpus is 7,386 because of common words

VI. EXPERIMENTAL RESULTS

Using the speech corpus described in the previous section, we evaluated the performance for three cases; male only, female only and mixed case (male and female) speech files. The accuracy used to measure the accuracy that based on WER. The WER measured using the following formula [22]: WER= (D+S+I)/N, where D is the deletion errors, S is the substitution errors, I is the insertion errors, and N is the total number of labels (i.e words) in the reference (actual) transcriptions. The accuracy expressed as:

$$\text{Accuracy} = (1\text{-WER}) \times 100\%$$

Fig. 2 shows an example of the Soundflower output of a particular speech file after recognition process. This textual output aligned with the actual transcription to find D, S, I, N, to calculate the WER according to what we have recognized, either for a single speech file or for the entire speech files collection. Of course, there are some recognition errors in Fig. 2 outputs (e.g. the word "لمنع"). This is natural like any other classification or pattern recognition system to have some misclassification rate.

Figure 2. An example of Soundflower output

In the first case, the performance measured using the male speech files. That is, Soundflower employed to measure the accuracy of 150 speech files that belong to 10 male speakers. Table II shows the achieved results of each speaker. The table also shows the range of accuracy [42.26%, 70.39%]. The difference in the scored accuracy related to several factors such as speaker's anatomy of vocal tract, the speed of the speech, and the accent. Table II also shows that the WER is (100% - 55.33% = 44.67%). For the used wave files that is between 30-60 seconds length, the results presented in Table II and Table III show the range of accuracies starts at (almost) 40% up to 70%. Hence, it might be a future work to investigate the performance using less or more wave files lengths.

TABLE II. ACCURACY FOR MALE ONLY SPEECH

#	Male Speakers	Number of speech files	Length (min:sec)	Accuracy (%)
1	Speaker 1	15	9:29	70.39
2	Speaker 2	15	10:12	48.80
3	Speaker 3	15	9:47	64.93
4	Speaker 4	15	8:46	59.07
5	Speaker 5	15	8:59	42.26
6	Speaker 6	15	9:55	54.67
7	Speaker 7	15	8:12	57.87
8	Speaker 8	15	8:30	55.16
9	Speaker 9	15	9:36	44.58
10	Speaker 10	15	8:54	55.66
	Total	**150**	**92:20**	**Avgerage =55.33%**

For female speakers, 104 speech files used to evaluate the accuracy. Table III shows the accuracy of each person of 10 female speakers. The accuracy range was [46.52%, 68.73%]. This range is close to what we achieved for male speakers. This reveals that the male and female speech recognition is very close in the case of using Soundflower tool. This result calls for more research to find the effect of acoustic differences between male and female speakers on Arabic speech recognition. Table III also shows that the WER is (100% - 56.97% = 43.03%).

TABLE III. ACCURACY FOR FEMAL ONLY SPEECH

#	Female Speakers	Number of speech files	Length (min:sec)	Accuracy (%)
1	Speaker 1	3	2:00	60.51
2	Speaker 2	15	9:42	68.73
3	Speaker 3	15	10:34	57.19
4	Speaker 4	7	5:12	52.07

5	Speaker 5	15	8:00	56.53
6	Speaker 6	15	9:15	50.85
7	Speaker 7	15	8:45	46.89
8	Speaker 8	2	1:27	62.83
9	Speaker 9	2	1:29	67.63
10	Speaker 10	15	9:56	46.52
	Total	**104**	**66:20**	**Average=56.97%**

The average of accuracies for the previous two cases indicates that the female speech recognition outperforms the male speech recognition. The third case separates the corpus for male and female speech to find the accuracy separately. Finally, we evaluated for the mixed male and female case for all speech files combined. Table IV shows the results of the mixed case.

TABLE IV. ACCURACY FOR MIXED MALE AND FEMALE SPEECH

Gender	Total number of speakers	Number of speech files	Length (min:sec)	Accuracy (%)
Male	10	150	92:20	54.66
Female	10	104	66:20	55.17
Male & Female	20	254	158:40	54.02

Fig. 3 shows the information provided in Table IV as a bar chart graph. The figure shows that the accuracy for Arabic speech is relatively low as the maximum-scored accuracy was 54.02%. The WER of the mixed speech corpus found to be (100% - 54.02% = 45.98%). However, we conducted our experiments on a small corpus that is unacceptable to generalize for the overall Arabic speech recognition. This result motivates the research to investigate and to enhance the performance of Arabic speech recognition.

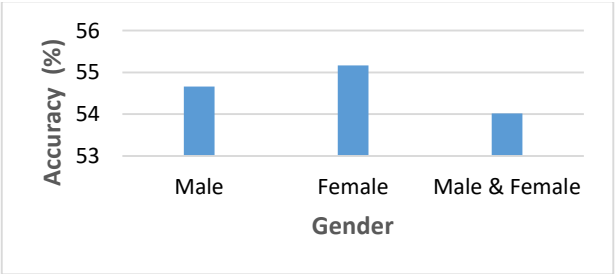


Figure 3. The accuracies of different testing cases

Even though gender is an important factor that has to consider in speech recognition. However, the experimental evaluation did not show clear performance differences using both the prepared corpus and the Soundflower tool. Despite we expect to have less accuracy in the case of female speech, as reported in some literature such as [4], it found that the female speech recognition performance outperforms the male speech recognition performance.

VII. CONCLUSISON AND FUTURE WORKS

The study demonstrated the performance of speaker independent Arabic continuous speech recognition. A free MAC software tool used to find the recognition accuracy. It found that the maximum-scored accuracy is 54.02% for mixed speech of male and female. The experimental results did not show obvious difference between the accuracies based on the gender. As a future work, we propose more investigation of the effect of gender on Arabic speech recognition.

REFERENCES

[1] Metze, Florian, et al. "Comparison of four approaches to age and gender recognition for telephone applications." Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on. Vol. 4. IEEE, 2007.

[2] Forsberg, Markus. "Why is speech recognition difficult." Chalmers University of Technology (2003).

[3] http://soundflower.en.softonic.com/mac

[4] R. Vergin, A. Farhat and D. O'Shaughnessy, "Robust gender-dependent acoustic-phonetic modelling in continuous speech recognition based on a new automatic male/female classification," Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on, Philadelphia, PA, 1996, pp. 1081-1084 vol.2.

[5] Simpson, Adrian P. "Phonetic differences between male and female speech." Language and Linguistics Compass 3.2 (2009): 621-640.

[6] https://code.google.com/archive/p/soundflower/

[7] http://www.sakhr.com/index.php/en/solutions/speech-technologies

[8] Waibel, Alex, et al. "Speechalator: two-way speech-to-speech translation in your hand." Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: Demonstrations-Volume 4. Association for Computational Linguistics, 2003.

[9] Available: http://htk.eng.cam.ac.uk/

[10] Available: http://www.qamus.org/transliteration.htm

[11] Available: http://cmusphinx.sourceforge.net/wiki/tutorialoverview

[12] Available: http://www.speech.cs.cmu.edu/

[13] Available: http://shop.nuance.co.uk/store/nuanceeu/en_GB/DisplayHomePage

[14] Available: https://gotranscript.com/

[15] Available: https://cloud.google.com/speech/

[16] Available: http://voice-recognition-software-review.toptenreviews.com/

[17] Devine, Eric G., Stephan A. Gaehde, and Arthur C. Curtis. "Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports." Journal of the American Medical Informatics Association 7.5 (2000): 462-468.

[18] Ali, M., Elshafei, M., Al-Ghamdi, M. and Al-Muhtaseb, H. 2009. Arabic Phonetic Dictionaries for Speech Recognition. Journal of Information Technology Research. 2, 4 (2009), 67-80.

[19] Available: http://telestreamblog.telestream.net/2013/12/using-dictation-to-turn-recorded-audio-to-text-2/

[20] Vogt, Thurid, and Elisabeth André. "Improving automatic emotion recognition from speech via gender differentiation." Proc. Language Resources and Evaluation Conference (LREC 2006), Genoa. 2006.

[21] Abdulla, W. H., N. K. Kasabov, and Dunedin–New Zealand. "Improving speech recognition performance through gender separation." changes 9 (2001): 10.

[22] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. *Int J Speech Technol*. 10, 4 (2007), 183-195.

[23] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

Your paper #1570358994 ('The Effect of Diacritization on Arabic Speech Recognition')

EM

EDAS Conference Manager <help@edas-help.com>
on behalf of
ghazi.sukkar@gmail.com

Reply all

Sat 08/07, 14:27
Fawaz Alanzi;
Ghazi M. AL Sukkar <ghazi.alsukkar@ju.edu.jo>;
Ali Maqousi <amaqousi@uop.edu.jo>
Inbox

You forwarded this message on 08/07/2017 15:25

Dear Prof. Fawaz Al-Anzi:

Congratulations! On behalf of the Conference Committee of the 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT 2017), we are happy to inform you that your paper #1570358994 entitled:

('The Effect of Diacritization on Arabic Speech Recognition')

has been accepted for presentation and inclusion in the Proceedings of AEECT 2017, published by IEEE.

We are proud to inform you that AEECT 2017 has received a large number of excellent submissions. Each submission was reviewed by several experts in the field and the committee chose a subset of these best submissions based on the reviews.

Please see the reviewers' comments below on your paper. These reviews can also be found at https://edas.info/showPaper.php?m=1570358994. These comments are intended to help you to improve your paper for final publication. The listed comments should be addressed, as final acceptance is conditional upon appropriate response to the requirements and comments. The conference committee retains a list of certain critical comments to be addressed by authors, and will control that these have been addressed in the camera-ready version.

AEECT 2017 implements IEEE's conference no show policy. AEECT 2017 reserves the right

to exclude a paper from distribution after the conference (e.g., removal from IEEE Xplore) if the paper is not presented at the conference.

All the information required to complete your paper and submit it for inclusion in the proceedings can be found through the following link:
http://ewh.ieee.org/conf/aeect/Accepted.html

Kindly pay special attention to the format guidelines detailed in the check list:
http://ewh.ieee.org/conf/aeect/CheckList.html

We hope to see you all in Amman, Jordan in October in what promises to be an excellent conference.

Sincerely,

Prof. Gheith Abandah, General Chair
Dr. Ghazi AL SUKKAR, TPC Chair
Dr. Issam Qaralleh, TPC Chair

======= Review 1 =======

> *** Presentation: How was the paper presentation?
Good (3)

> *** Organization: How do you consider the paper organization and flow of ideas?
Very Good (4)

> *** Originality: How do you rate the novelty and originality of this work?
An Interesting contribution (3)

> *** Relevance: AEECT 2017 is interested in applied research that addresses problems that face developing countries. How relevant is this paper?
Very relevant (4)

> *** Acceptance Score: This paper should be ...
Totally Accepted (4)

> *** Detailed Comments: Please provide your detailed Technical comments to Author(s).

The topic of the paper is very important and relevant.

Authors list

- There are two authors in the pdf file while in the EDAS record there is one author. The second author needs to be added to the EDAS record.

Abstract
- ASR received attention since a while and is not a very recent field, this does not mean it is not an important field, but the phrasing of its importance need to be more accurate

- "This ambiguity produces a less than optimal acoustic models" Either remove "a" or replace "models" with "model"

- Add short explanation of how the scores were computed. If you exceed the number of allowed words in the abstract, please add that in the results section.

I. Introduction
- Paragraph 2. Change "The important" to "The Importance"

- "More on Arabic diacritization and some other related challenges are found in [5].". List some of these challenges in this paper and add citation to [5].

- In the last paragraph I suggest to use the section numbers II, III .... VII instead of 2 and 3 ... 7.

III. Phonemes set
- Please justify why shadda is discarded
- please change "proceeds" to "preceeds"
- Please capitalize H in hidden and m in markov

IV. ACOUSTIC MODELS
- I think you mean "enable or disable" not "enable of disable"
- Add a reference for PTM method
- please replace "is used" by "used"
- please remove extra "the"

V. PROPOSED METHOD
- Please justify the training-testing splitting percentages

VI. EXPERIMENTAL RESULTS
- More information about MSA corpus is desired
- Justify why you are only reporting the best result not the average of the results (scores).

References
- More recent references would improve the recency of the paper

- References need to follow the IEEE format

> *** Detailed Format and Organizational Comments: Please provide your detailed Organizational comments to Author(s).

No formatting comments

======= Review 2 =======

> *** Presentation: How was the paper presentation?
Good (3)

> *** Organization: How do you consider the paper organization and flow of ideas?
Good (3)

> *** Originality: How do you rate the novelty and originality of this work?
A Slight modification of concepts (2)

> *** Relevance: AEECT 2017 is interested in applied research that addresses problems that face developing countries. How relevant is this paper?
Relevant (3)

> *** Acceptance Score: This paper should be ...
Marginaly Rejected (2)

> *** Detailed Comments: Please provide your detailed Technical comments to Author(s).

The paper tackles important question about the effect of diacritizing the input Arabic text on the accuracy of Arabic speech recognition. The paper is well written with good style. Moreover, the authors use good tools and are building a large data set of recorded modern standard Arabic for this purpose.

However, the paper draws a counter-intuitive conclusion. The main result is unexpected. The accuracy of using diacritized text is lower than the accuracy of using non-diacritized text. The common idea in this field is that non-diacritized Arabic text can be synthesized in multiple pronunciations. Therefore, There is a need for a pre-process to diacritize the Arabic text in order to correctly convert it to speech.

Most likely, the authors have some experimental errors that resulted in the low achieved accuracy and this strange conclusion. In the unlikely case that there are no errors, the authors, however, do not give sound justifications and explanations for this conclusion.

In case the paper is accepted due to favorable reviews by other reviewers, I do recommend that the final manuscript should include rechecked results and proper

explanations.

> *** Detailed Format and Organizational Comments: Please provide your detailed Organizational comments to Author(s).

* Review the use of punctuation marks, e.g., the period in Line 21 of the abstract.

The Effect of Diacritization on Arabic Speech Recognition

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
abuzeina@ku.edu.kw

***Abstract*—Arabic automatic speech recognition (ASR) is a successful application of natural language processing (NLP). However, Arabic formal text is generally written without diacritics, which produces different pronunciation forms. That is, the Arabic writing system allows discarding short vowels and, hence, forcing the reader to use the prior knowledge and the words context to infer the missing diacritics. For speech recognition, there are two options for textual training data; either diacritized (also called vowelized) or non-diacritized text. However, using non-diacritized text may introduce a challenge for Arabic ASR as missing the short vowels may lead to some confusion in the learning process. This ambiguity produces a less than optimal acoustic model that is one of the most important components of ASR systems. In this paper, we present the performance using diacritized and non-diacritized text. In the experiments, we used the Carnegie Mellon University (CMU) PocketSphinx speech recognizer. We also used a new "in house" modern standard Arabic (MSA) continuous speech corpus that contains 13.5 hours for training and 4.1 hours for testing. The text of the corpus was manually diacritized. For acoustic modelling, we used the phonetic tied-mixture (PTM). The experimental results show that the non-diacritized text system scored 76.4% (i.e. 1-word error rate (WER)) while the diacritized text based system scored 63.8%. Even the diacritized case has less accuracy due to the slight differences in diacritics; however, the non-diacritized case might be adequate and faultless for the Arabic native speakers.**

Keywords— Arabic; speech; recognition; diacritics; short vowels

I. INTRODUCTION

Automatic speech recognition (ASR) is of particular interest to human computer interface (HCI) and natural language processing (NLP). Recently, Arabic large-vocabulary speaker-independent continuous speech recognition system has recently received significant attention in the NLP research community. However, Arabic ASR poses some challenges such as the difficulty to obtain corpora for dialects that are spoken rather than written (i.e. no common standard for writing), difficulty in obtaining a large diacritized text as the Arabic allows writing without diacritics, and enormous number of word forms due to the morphology richness of the Arabic. Recently, there has been much interest in diacritization for better performance in ASR systems. Diacritization is the process of marking the letters using optional orthographic symbols that are called diacritics or short vowels. For Arabic ASR, the problem of short vowels is that they are generally pronounced, but almost never written. The study in [1] indicates that the non-diacritized text leads to problems for both acoustic and language modeling and therefore may lead to a loss in recognition accuracy. Similarly, it is reported in [2] that missing of short vowels leads to a significant increase in both the language model perplexity and the word error rate.

The importance of diacritization is that it enhances the supposed closely match between the training textual files and the corresponding speech files. In fact, it is extremely important that the phonemes of the pronunciation dictionary to adequately represent the actual training speech files. It is indicated in [3] that the performance of ASR is improved by shrinking the mismatch between the speech and the text used in training the acoustic model. In the case of training using non-diacritized text, many of phonetic segments will be lost because the short vowels are not there. Despite short vowels help the reader to realize the meaning of a particular word, however, not fully diacritized text might lead to ambiguity as the same word might have several meanings. For instance, the word "جنة: jnp" has three different meanings based on the short vowels (u:ُ,a:َ,i:ِ) on the first letter: (جِنة, جَنة, جُنة) (junp, janp, jinp) that means (protection, paradise, jinn), respectively. In the previous ward, the Buckwalter scheme was used for Arabic transliteration [4]. More on Arabic diacritization and some other related challenges are found in [5]. Reference [5] mainly discussed the differences in the pronunciation and the meaning of a particular word according to its diacritization. Nevertheless, obtaining a sizable diacritized text for ASR and NLP applications is extremely difficult as well as time-consuming task. This is the motivation of this work as we produced a manually diacritized continuous speech corpus for the modern standard Arabic (MSA).

In this paper, we employed the latest Carnegie Mellon University (CMU) PocketSphinx ASR engine [6] for exploring the Arabic ASR performance based on diacritized and non-diacritized text. PocketSphinx includes the latest available releases as follows: sphinxbase - 5prealpha, PocketSphinx - 5prealpha, SphinxTrain - 5prealpha. In the experiments, we used a new "in house" diacritized text corpus that contains 13.5 hours for training and 4.1 hours for testing. This study also presents the intermediate steps for training and decoding such as the proposed and used phonemes set, the pronunciation dictionary, the acoustic model, and the language model. We emphasize that this work is a preliminary step toward further research using the newly created corpus. This corpus has been fully supported by Kuwait University. The size of the corpus in

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

this work is 17.6 hours; however, we aim at increasing the size to about 30 hours.

In next section, we present the literate review. In section III, we present the phonemes set followed by a background of acoustic models in section IV. The proposed method is described in section V and the experiment results in section VI. Finally, we conclude in section VII.

II. Literature Review

Despite that Arabic is one of the popular languages. However, little research has been devoted to tackle the different ASR aspects such as the dialects, the diacritization, and the morphological complexity. In this literature, we focus on the studies that consider diacritization. The study in [7] demonstrates a news transcription system for MSA. It compares the performance using diacritized and non-diacritized text for broadcast news. The word recognition accuracy of the non-diacritized case outperforms the diacritized case. This is due to the errors that are introduced by missing of the short vowels in the diacritized case. However, this might be not a problem since the Arabic native speakers can infer the missing short vowels based on the prior knowledge and the words context. The studies in [8][9] presents methods for capturing the acoustic differences (pronunciation variations) at cross-words and within words for Arabic ASR systems.

The study in [2] demonstrates a comparison between script transcriptions (i.e. non-diacritized) and romanized transcription that is phonologically rich by vowels information. The romanized transcription case outperforms the standard Arabic script that has no diacritics. The work in [10] produces three different speech diacritized corpora that include a holy Qur'an corpus, a command corpus, and a digits corpus. The results were demonstrated based on the diacritized text. The experimental results in [11] show that the non-diacritized case slightly outperforms the diacritized text case for a phonetically rich and balanced Arabic speech corpus. The Sphinx tools along with SAMPA Romanization method were used in [12] for dialectal Arabic speech recognition. The research in [13] found that the diacritized text improved the acoustic model more than undiacritized orthography. Most of the previous works were performed using relatively small corpora; however, we used a larger corpus to explore the effect of diacritization on Arabic ASR.

III. Phonemes Set

The phoneme is the basic unit of speech that represents a distinctive sound of the language's phonology. Hence, a change of a particular phoneme in a word makes a change in the meaning of the word. Phonemes play a vital role in the performance of ASR and text to speech systems. In this work, we propose a phoneme set that is used to evaluate the recognition performance of the prepared corpus. The pronunciation dictionary is prepared using the proposed phonemes set by a mapping process between the Arabic letters (the language's vowels and consonants) and their corresponding phonemes. However, in some cases, morphologically driven rules are used for phonetic rich dictionary. In addition, some pronunciation exceptions might be manually processed for better acoustic representation. The studies [14] and [15] elaborate on Arabic phonemes and the pronunciation rules.

In general, creation a dictionary of a particular language requires linguistic experts and deep knowledge of the language sounds. However, the choice of the phoneme in the phonemes set is not straightforward as it has some constraints. For instances, it should represent the relevant information of the language sounds, it should consider the surrounding context between the letters, and it should carefully estimate the starting and the ending of the letters. No doubt, the phonemes that are used to represent the training words characterize the quality of the acoustic models and, therefore, the overall performance. Table I shows the phonemes set used in this work. It contains 46 phonemes. In addition to the Arabic letters, the table includes the short vowels that are Fatha (ً), Damma (ٌ), and Kasra (ٍ). In this work, we discarded the Shadda (ّ) as our experimental evaluation showed that it has no difference in the performance. We also used three phonemes to represent the Fatha that precedes Alif (ا) ➔ ىَ as a single phoneme that is (AUA), the Damma that precedes Waw (و) ➔ ىُ as a single phoneme that is (AWW), and the Kasra that precedes Ya (ي) ➔ ىِ as (AIY). The reason of handling these cases as a single phoneme is that the pronunciation of the short vowels is different when it precedes the long vowels. Hence, it would be correctly transcribed as single phonemes.

TABLE I. The Arabic letters and the phonemes set

#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	24	ظ	ZZ
2	آ	AA	25	ع	AE
3	أ	O	26	غ	GH
4	ؤ	EW	27	ف	F
5	إ	I	28	ق	Q
6	ئ	EY	29	ك	K
7	ا	A	30	ل	L
8	ب	B	31	م	M
9	ة	P	32	ن	N
10	ت	T	33	ه	H
11	ث	TH	34	و	W
12	ج	J	35	ى	AY
13	ح	HH	36	ي	Y
14	خ	KH	37	ْ	N
15	د	D	38	ّ	N
16	ذ	DH	39	ِ	N
17	ر	R	40	َ	AU
18	ز	Z	41	ُ	AW
19	س	S	42	ِ	AI
20	ش	SH	43	ً	ignored
21	ص	SS	44	ىَا	AUA
22	ض	DD	45	ىُو	AWW
23	ط	TT	46	ىِي	AIY

The selection of a phoneme symbol is an optional and it does not matter what phoneme symbol is used for an individual letter. In the training stage, each phoneme is modelled using a sequence of Hidden Markov Model (HMM) states for computing the acoustic model. In the decoding stage, the phoneme is initially recognized and then used to find the most likely spoken words based on the best-matched phonemes

between the speech file in question (the observations) and the trained HMMs of the acoustic model.

IV. ACOUSTIC MODELS

The training stage of an ASR system consists of building an acoustic model that is a major component of ASR engines. Acoustic models statistically represent the relationships between the speech signals and the language phonemes. These representations come into the form of probabilistic matrices that are known as three matrices: initial probability, transition probability, and the observation likelihoods or emission probability. There are different methods to train acoustic models; the most common method is HMM. It has been long observed that the HMM based acoustic models successfully implemented in the state of the art speech recognizers. However, there are other approaches such as artificial neural networks (ANN) [16] and support vector machine (SVM) [17].

CMU Sphinx speech engines support three types for acoustic modeling. For instances, the CMU Sphinx configuration file "Sphinx_train.cfg" has the commands to enable or disable the desired acoustic model. The types of acoustic models include the traditional fully continuous, the semi-continuous, and the phonetic tied-mixture (PTM) models [18]. Despite the common implementation of fully continuous and semi-continuous in Arabic ASR, however, PTM has less experimental studies for Arabic speech recognition. PTM is a recent method that compromises between important factors such as speed and performance. It is characterized by fast decoding as well as its ability to handle large amount of speech collections.

Hence, PTM might be good option if the decoding time is more important than the accuracy. There are some advantages of PTM based acoustic models. For instance, the PTM model consider pronunciation variations modeling such the work in [19]. Reference [19] proposes a state-dependent PTM model with variable codebook size to improve the coverage of phonetic variations while maintaining model discriminative ability. One reason of speed is that the PTM model used relatively low fixed Gaussians that speed up the recognition time.

In the decoding stage, the HMM states of each phoneme is compared with the query acoustic feature vectors to find the best-matched phonemes and, then, likely sequence of words. The HMMs parameters are estimated using special algorithms such as Baum-Welch re-estimation and expectation maximization (EM). The corpus vocabulary and the size of the speech corpus determines some training parameters such as the number of Senones (tied-state) and the number of Gaussians. Table II shows the approximation number of Senones and the densities according to the vocabulary and the size of some English speech corpora [20].

TABLE II. APPROXIMATION NUMBER OF SENONES AND DENSITIES

Vocabulary	Hours	Senones	Densities	Example
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

V. PROPOSED METHOD

We evaluated the Arabic ASR performance using the prepared corpus. We got the speech files that belong to MSA broadcast news form As-Sabah TV [21] in Kuwait. The first step was the preprocessing that includes segmenting the long speech files into short segments of 30-60 seconds. The produced speech files cover different news stories and it sums up to 17.6 hours of 29 speakers (19 male speakers and the rest are for female speakers). The speech files were sampled at 16 KHz mono. A silence of 0.1 seconds at the beginning and at the end of each speech file. We collected 1660 speech files that were transcribed and manually diacritized. We divided the speech files into two parts; the training set that contains 1,269 (13.5 hours) speech files and the testing that contains 391 speech files (4.1 hours). Hence, the testing part is 23% of the overall corpus, which follows the training-testing splitting percentage that is usually used in world of statistical classification (i.e. 20% ~ 25%). The vocabulary size of the training set is 29,843 words. The proposed method is summarized in the algorithm in Fig.1.

Part One

- *Prepare the speech collection.*
- *Prepare the transcription of the speech files.*
- *Diacritize the textual transcription*
- *Define the phonemes set.*
- *Create the pronunciation dictionary.*
- *Define the number of Gaussians.*
- *Define the senones (n_tied_states).*
- *Train the acoustic models.*
- *Prepare the language model.*
- *Run PocketSphinx for training and decoding.*

Part Two

- *Repeat the pervious steps without diacritization.*

Fig. 1. The proposed method

In addition to the previous steps, the CMU SphinxTrain performs some internal tasks such as computing features from audio files, training context independent models, training context dependent models, build trees, prune trees, lattice generation, lattice pruning, and finally decoding using the trained models. Once having the trained acoustic model, the PocketSphinx is used for decoding by use other components such as the pronunciation dictionary and the language model.

VI. EXPERIMENTAL RESULTS

This section presents the experimental results based on the introduced MSA speech corpus. We conducted the experiments for two cases, diacritized and non-diacritized text. In this work,

we used three emitting states of HMMs that corresponds to the subphones at the beginning, middle, and end of the phone. The acoustic models were calculated using context-dependent HMM triphones. Our acoustic models are all trained using SphinxTrain for PTM PocketSphinx. For language model, we used the CMU language toolkit [22] to calculate the statistical N-grams (i.e. 1-grams, 2-grams, and 3-grams) based on the corpus transcription. The pronunciation dictionary was generated using a Python based program. The total number unique words in the diacritized based system is 29,843 while it is 19,581 words in the non-diacritized case.

The word error rate (WER) was measured for different parameters such as the number of the Gaussian densities and the number of the Senones. The PocketSphinx configuration file indicates that the PTM based models have to use the same initial and final Gaussian densities, 256 Gaussians as indicated in [20]. However, we investigated different values as indicated in Table III. In Table III, we demonstrate the WER and the accuracy achieved for different parameter settings. The lowest WER is 36.2% and it achieved using 64 Gaussians densities and 1000 Senones. We investigated a wide range of parameters to clarify its effect on the performance as shown in the table.

TABLE III. THE DIACRITIZED BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	64	200	37.0	63.0
2	128	200	36.8	63.2
3	256	200	36.6	63.4
4	64	500	36.4	63.6
5	128	500	63.5	63.5
6	256	500	36.5	63.5
7	**64**	**1000**	**36.2**	**63.8**
8	256	1000	36.8	63.2
9	256	2000	36.8	63.2
10	256	3000	36.9	63.1
11	256	4000	37.1	62.9
12	256	5000	37.4	62.6
13	256	6000	37.5	62.5
14	256	7000	38.0	62.0
15	256	8000	38.2	61.8
16	256	9000	38.4	61.6
17	256	10000	38.7	61.3

This low accuracy is reasonable since we used a relatively small size corpus. Ideally, ASR requires 200-300 hours speech corpus. It is indicated in [23] that at least 1 gigabyte of texts for language models and 50 hours for acoustic models are required for reasonable performance. It is also reported in [20] that the WER for 10-hours task should be around 10%. For a large task, it could be around 30%. Table IV shows the WER for some ASR systems on different English speech corpora [24].

TABLE IV. ROUGH WERS FOR A NUMBER OF ENGLISH CORPORA

Speech collection	Vocabulary	WER %
TI Digits	11 (zero-nine, oh)	0.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech	64000+	20

One more reason for the obtained relatively low accuracy is that the used corpus has no filler dictionary. Filler dictionary generally contains noise and inhalation speech that are appropriately handled during the training phase. The fillers require indicating the noises and inhalations in the transcription of the speech files, which is an extremely difficult task for our corpus. It is worthy to point out to a recent study [25] of Arabic ASR that considers the impact of phonological rules on Arabic ASR performance.

For non-diacritized case, Table V shows the performance using different densities and Senones. The lowest WER is 23.6% that achieved using 128 Gaussians densities and 500 Senones as highlighted in Table V. Therefore, the information in Table V indicates that the best accuracy is achieved using 128 densities and 500 Senone, which might be the performance of the baseline for future work. Of course, this result belongs to the used corpus that contains 17.6 hours. Other Arabic speech corpora might have different results. Regarding the execution time of the training and the decoding stages for both diacritized and non-diacritized system, we found that the execution time of non-diacritized case was clearly less than the diacritized case.

TABLE V. THE NON-DIACRITIZED BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	8	200	27.8	72.2
2	16	200	25.7	74.3
3	32	200	24.8	75.2
4	64	200	24.3	75.7
5	128	200	24.1	75.9
6	256	200	23.9	76.1
1	8	500	26.9	73.1
2	16	500	24.8	75.2
3	32	500	24.2	75.8
4	64	500	23.9	76.1
5	**128**	**500**	**23.6**	**76.4**
6	256	500	23.7	76.3
1	8	1000	26.4	73.6
2	16	1000	24.7	75.7
3	32	1000	23.9	76.1
4	64	1000	23.9	76.1
5	128	1000	23.7	76.3
6	256	1000	23.8	76.2
1	8	5000	25.9	74.1
2	16	5000	24.9	75.1
3	32	5000	24.0	76.0
4	64	5000	24.2	75.8
5	128	5000	24.5	75.5
6	256	5000	25.6	74.4
1	128	10000	26.3	73.7
2	256	10000	28.0	72.0

In ASR, the training phase is time-consuming. Hence, we consider speeding up the execution time using an option in CMU PocketSphinx. The configuration file is called "sphinx_train.cfg". This file has an option for multiprocessing mode. The two options that can be used for reducing the training and the decoding time are as follows. $CFG_NPART = 10 ➔ the number of parts to run Forward-Backward

estimation; $DEC_CFG_NPART = 10 ➔ how many pieces to split decode in. The number 10 is specified by the user according to the desired factor to reduce the execution time. The default values of these two parameters is 1. This option is helpful since it clearly reduces the execution time by use a number of processors in multicore machines. We also conducted some experiments to compare the execution time of the continuous and PTM based acoustic models. We found that the PTM based acoustic model has less execution time. The PTM based acoustic model is three times faster than the continuous acoustic model.

VII. CONCLUSION

This paper presents an experimental evaluation of Arabic ASR performance using a new continuous speech manually diacritized corpus. In the experiments, we consider two cases of the Arabic text; diacritized and non-diacritized. The experimental results show that the non-diacritized based system outperforms the diacritized based system even with a smaller vocabulary. However, the diacritized based system gives vowelized text output, which is not produced by a non-diacritized based system. As a future work, it is worthy to reinvestigate and analysis the main conclusion of this work; which indicates that the accuracy of using diacritized text is lower than the accuracy of using non-diacritized text.

REFERENCES

[1] Vergyri, Dimitra, and Katrin Kirchhoff. "Automatic diacritization of Arabic for acoustic modeling in speech recognition." Proceedings of the workshop on computational approaches to Arabic script-based languages. Association for Computational Linguistics, 2004.

[2] Kirchhoff, Katrin, et al. "Novel approaches to Arabic speech recognition-final report from the JHU summer workshop 2002." John-Hopkins University, Tech. Rep (2002).

[3] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[4] http://www.qamus.org/transliteration.htm

[5] Al-Anzi, Fawaz S., and Dia AbuZeina. "Stemming impact on Arabic text categorization performance: A survey." Information & Communication Technology and Accessibility (ICTA), 2015 5th International Conference on. IEEE, 2015.

[6] http://cmusphinx.sourceforge.net/wiki/download

[7] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[8] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[9] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[10] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[11] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[12] Elmahdy, Mohamed, et al. "Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition." Natural Language Processing, 2009. SNLP'09. Eighth International Symposium on. IEEE, 2009.

[13] Vergyri, Dimitra, et al. "Development of the SRI/nightingale Arabic ASR system." Interspeech. 2008.

[14] Ali, Mohamed, et al. "Generation of Arabic phonetic dictionaries for speech recognition." Innovations in Information Technology, 2008. IIT 2008. International Conference on. IEEE, 2008.

[15] Ramsay, Allan, Iman Alsharhan, and Hanady Ahmed. "Generation of a phonetic transcription for modern standard Arabic: A knowledge-based model." Computer Speech & Language 28.4 (2014): 959-978.

[16] Wilinski, Piotr, et al. "Toward the border between neural and Markovian paradigms." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28.2 (1998): 146-159.

[17] Guo, Guodong, and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines." IEEE transactions on Neural Networks 14.1 (2003): 209-215.

[18] Lee, Akinobu, et al. "A new phonetic tied-mixture model for efficient decoding." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 3. IEEE, 2000.

[19] Liu, Yi, and Pascale Fung. "State-dependent phonetic tied mixtures with pronunciation modeling for spontaneous speech recognition." IEEE transactions on speech and audio processing 12.4 (2004): 351-364.

[20] Training Acoustic Model, Available:
http://cmusphinx.sourceforge.net/wiki/tutorialam

[21] AL-SABAH TV, Available:
http://www.alsabahpress.com/

[22] Building language model, Available:
http://cmusphinx.sourceforge.net/wiki/tutoriallm
CMU Sphinx Speech Recognition Toolkit, Available:
https://sourceforge.net/p/cmusphinx/discussion/help/thread/1f102f95/?limit=25#9d3d

[23] Jurafsky D, Martin J (2009) Speech and language processing, 2nd edn. Pearson, NJ

[24] Al-Anzi, Fawaz S., and Dia AbuZeina. "The impact of phonological rules on Arabic speech recognition." International Journal of Speech Technology(2017): 1-9.

[ACIT'2017] Your submission has been accepted!

A

ACIT <acit@ccis2k.org>

Reply all

Tue 17/10, 14:42

Fawaz Alanzi

Inbox

The message sender has requested a read receipt. To send a receipt, click here.

Dear Author,

On behalf of the ACIT 2017, We are pleased to inform you that your submission, titled: " Phonetic Tied-Mixture PTM Acoustic Model for Arabic Continuous Speech Recognition "

Has been accepted for presentation at the International Arab Conference on Information Technology (ACIT'2017) which will be organized by Sfax University, Tunisia, December 22-24, 2017. The decision was based on the reviewers' evaluation reports:

1.You must justify your paper, there are many word merging with other like implementedfor,contains13.5
2.The tables and figures must be as template in conference
3.there are no new REFERENCES for your paper, the REFERENCES is very old

The registration will be open as follow:

- Early bird registration: Oct. 16 2017 - Oct. 26, 2017.
- registration: Oct. 27, 2017- Nov. 25, 2017.

Kindly fill in the registration form available at
http://acit2k.org/ACIT/index.php/registration-and-accommodation-information2017

The registration fees include conference registration and proceedings in addition to (full board) hotel accommodation for 2 nights (from 22/12/2017 to 24/12/2017).

The filled registration form and the payment proof have to be sent to the conference secretary by emailacit@ccis2k.org before Nov. 25, 2017.

Please refer to ACIT Website at:
http://acit2k.org/ACIT/index.php/camera-ready2017
For more information and instructions in preparing your camera-ready version of your paper and for the sample paper that is prepared According to the instructions of IEEE template. Following this Template is mandatory for publication.

Thank you for submitting a paper to ACIT'2017 and congratulations on it's acceptance. We look forward to meeting you in Tunisia for a very Rewarding and professionally stimulating conference.

For authors who need a visa please note that according to Tunisia visa procedure, you should send the following:
A copy of relevant page of your passport to get an invitation letter in order to get a visa from Tunisia embassy in your country to acit@ccis2k.org.

Regards,

Program Committee, ACIT'2017

Best Regards
ACIT Secretariat
Zarqa University
P. O. Box 132222
Zarqa Code Number 13132
Jordan
Tel: +(962)-5-3821100 ext. 1451
Fax: +(962)-5-3821117
E-mail: acit@ccis2k.org
Web
Site: [../../../../../../Documents%20and%20Settings/acit2k/Application%20Data/Microsoft/Signatures/www.acit2k.org]www.acit2k.org

Phonetic Tied-Mixture PTM Acoustic Model for Arabic Continuous Speech Recognition

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
abuzeina@ku.edu.kw

***Abstract*—Speech recognition has recently received significant attention for better automation of customer service interfaces. However, compiling accurate acoustic models is still poses some challenges due to the variety of speaking styles and the pronunciation variations phenomenon. In the last decade, the traditional fully continuous and the semi-continuous acoustic models dominate the acoustic modeling work for speech recognition. However, the later phonetic tied-mixture (PTM) acoustic model has been less implemented for Arabic compared to the fully continuous and semi-continuous models. In this paper, we experimentally evaluated the speech recognition performance based on the PTM based acoustic models. We employed the Carnegie Mellon University (CMU) PocketSphinx speech recognizer using a modern standard Arabic (MSA) continuous speech corpus. The corpus contains 13.5 hours for training and 4.1 hours for testing. The experimental results show that the the PTM and semi-continuous models have almost same accuracy that is more than the fully continuous acoustic model. However, the PTM acoustic model has less execution time compared to the semi-continuous model. The experiments were conducted for different training setting parameters such as the number of Gaussians and the number of tied states (Senones).**

Keywords— Arabic; speech; recognition; phonetic tied-mixture; acoustic model

I. INTRODUCTION

Automatic speech recognition (ASR) has recently attracted much attention for more convenient in human-computer interaction (HCI). Speech recognition aims at converting the spoken language into machine-readable format. For this task, a speech recognizer generally employs several components such as pronunciation dictionary, language model, and acoustic model. The effective implementation of acoustic modeling is a critical process that aims at capturing the realistic acoustic features of the speech signal. The ASR training phase includes hidden Markov model (HMM) [1] parameters estimation that produces the learned acoustic models. That is, HMM is the popular method to implement acoustic models. The types of acoustic model include the traditional fully continuous, the semi-continuous, and the phonetic tied-mixture (PTM) models. Despite the common implementation of fully continuous and semi-continuous in Arabic ASR, however, PTM has less experimental studies for Arabic speech recognition. PTM is a recent method that compromises between important factors such as speed and performance. Regardless of the method used to implement the acoustic model, the training of ASR includes the optimal estimation of the acoustic parameters based on the input acoustic feature vectors, especially for large vocabulary continuous speech collections.

Fully continuous acoustic models use a large number of Gaussians to compute the score of each frame. In contrast, the semi-continuous models use extremely less Gaussians, of course, on the account of the accuracy. However, for insufficient training speech collections, semi continuous might give better performance than the fully continuous models. It is indicated in [2] that the use of limited number of mixture densities can not only improve the performance but also significantly reduce the amount of computation. Semi continuous is also good for memory constraint cases such as in mobile devices. For handling such constraints, PTM was released to enhance the performance in the semi-continuous models as well as reducing the heavy use of the computational resources in the fully continuous models. That is, PTM models are somewhere between semi-continuous and fully continuous models, offering the speed of the continuous models with the ability to effectively use large amounts of training data. That is, PTM model aims at reducing the number of parameters of continuous model while significantly reducing the execution time (i.e. the training and decoding time) [3].

This study aims at demonstrating an experimental study of modern standard Arabic (MSA) speech recognition performance based on different acoustic models. We prepared a corpus of Arabic continuous speech that contains 13.5 hours for training and 4.1 hours for testing. We used a pronunciation dictionary based on a proposed phonemes set that contains 44 phonemes. In this work, we used the latest CMU speech recognizer, the PocketSphinx. This tool includes the latest available releases as follows: Sphinxbase - 5prealpha, PocketSphinx - 5prealpha, Sphinxtrain - 5prealpha [4]. We demonstrate the results for different training settings such as different number Gaussians, tied states (Senones), and parts to run Forward-Backward estimation run training in parallel to fully load the machine processing units.

In next section, we present the literate review. In section 3, we present the phonemes set followed by a background of acoustic models in section 4. The proposed method is described in section 5 and the experiment results in section 6. Finally, we conclude in section 7.

II. LITERATURE REVIEW

There are quite studies in Arabic ASR employing semi-continuous and fully continuous acoustic models. For

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

instances, Reference [5] demonstrated an ASR study for the Arabic speech. It used five-state HMM for triphone continuous acoustic models, with 8 and 16 Gaussian mixture distributions. The acoustic model of Reference [5] was then used in other publications such as [6] and [7]. Reference [8] employed CMU Sphinx tools for three different acoustic models that belong to three different speech collections. Reference [9] used CMU Sphinx tool for Arabic speech recognition. They developed three acoustic models for three different speech collections. They demonstrated the performance using using different number of Gaussians. Reference [10] used continuous based acoustic models for a phonetically rich and balanced Arabic speech corpus.

The following are some studies that discuss the three above-mentioned acoustic models. Reference [11] indicated that the impression that continuous HMMs are the best choice of acoustic model, however, semi-continuous might have an advantage in small amount of training data due to the need for estimating a large number of parameters. Reference [12] used semi-continuous HMM to reduce the number of parameters and the computational complexity. Reference [13] highlights the problem when using fully continuous with limited training data. In indicated that when large number of basic HMMs has only a few observations in the training data, then the sparse mixture-weight distributions cannot be estimated robustly and are expensive to store. In PTM based model, the number of Gaussians is reduced according to the shared triphone states of the same phoneme. There are some studies employed this method. For instances, Reference [14] showed that PTM systems, if properly trained, can significantly outperform the currently dominant state clustered HMM-based approach. Reference [15] presented that the PTM based acoustic model is easily trained, reliably estimated, and enable the decoder to perform efficient Gaussian pruning.

III. PHONEMES SET

The phoneme is the basic unit of speech that describes the pronunciation of a word. It is also the basic unit that is used for speech recognition in ASR. It is used to represent the language's vowels and consonants thorough straightforward one-to-one rules. The pronunciation dictionary is prepared using the phonemes set by mapping the words pronunciations (e.g. Arabic letters) to their corresponding phonemes based on the given phonemes set. However, in some cases, morphologically driven rules are used for phonetic rich dictionary. In addition, the pronunciation exceptions might manually processed for better acoustic representation. In general, the creation a dictionary requires linguistic experts and deep knowledge of the language sounds.

The selection of a phoneme symbol is an optional and it does not matter what phoneme symbol is used for an individual letter. Each phoneme is modelled using a sequence of HMM states in the acoustic model that is later used to find the most likely spoken words based on the best-matched phonemes. However, the choice of the phoneme in the phonemes set is not straightforward as it has some constraints. For instances, it should represent the relevant information of the language sounds, it should consider the the surrounding context between the letters, and carefully estimating the start and the end of the letters. Table I shows the phonemes set used in this work. It contains 46 phonemes.

TABLE I. THE ARABIC LETTERS AND THE PHONEMES SET

#	Letter	Phoneme	#	Letter	Phoneme
1	ء	E	24	ظ	ZZ
2	آ	AA	25	ع	AE
3	أ	O	26	غ	GH
4	ؤ	EW	27	ف	F
5	إ	I	28	ق	Q
6	ئ	EY	29	ك	K
7	ا	A	30	ل	L
8	ب	B	31	م	M
9	ة	P	32	ن	N
10	ت	T	33	ه	H
11	ث	TH	34	و	W
12	ج	J	35	ى	AY
13	ح	HH	36	ي	Y
14	خ	KH	37	ً	N
15	د	D	38	ٌ	N
16	ذ	DH	39	ٍ	N
17	ر	R	40	َ	AU
18	ز	Z	41	ُ	AW
19	س	S	42	ِ	AI
20	ش	SH	43	ّ	ignored
21	ص	SS	44	ٱ	AUA
22	ض	DD	45	ۥ	AWW
23	ط	TT	46	ۦ	AIY

IV. ACOUSTIC MODELS

The essence of this work is that the growing amount of training speech collections adds more complexities for efficient parameter estimation methods in speech recognition. Intuitively, the goal is, hopefully, to decrease the error rate as well as to minimize the processing time in large vocabulary ASR systems. Acoustic model is a major component of ASR engines that statistically represents the relationship between the speech signals and the phonemes. HMM is one most common type of acoustic models to compile the statistical representation of each phoneme. It has been long observed that the HMM based acoustic models successfully implemented in the state of the art speech recognizers. However, there are other approaches such as artificial neural networks (ANN) [16] and support vector machine (SVM) [17].

CMU Sphinx speech engines support all acoustic modeling types. For instances, the configuration file "Sphinx_train.cfg" of the CMU PocketSphinx has the commands to enable of disable a particular acoustic model type as shown in Fig. 1. The figure shows that the corresponding system implements PTM based acoustic model since the PTM command is enabled. It also shows that the PTM model is supported in PocketSphinx. However, the reader can check the CMU Sphinx website to investigate the latest versions and updates.

```
#$CFG_HMM_TYPE = '.cont.'; # Sphinx 4, PocketSphinx
#$CFG_HMM_TYPE  = '.semi.'; # PocketSphinx
$CFG_HMM_TYPE  = '.ptm.'; # PocketSphinx
```

Fig. 1. CMU configuration settings for acoustic models

In the decoding process, the HMM states of each phoneme is compared with the query acoustic feature vectors to find the probabilities of the best matched phonemes using specials algorithm. The HMMs parameters are estimated using the training speech files and the corresponding text transcription. The size of corpus vocabulary as well as the the size of the speech corpus determines some training parameters such as the number of Senones and the number of Gaussians. Table II shows the approximation number of Senones and the densities according to the vocabulary and the size of some English speech corpora [18].

TABLE II. APPROXIMATION NUMBER OF SENONES AND DENSITIES

Vocabulary	Hours	Senones	Densities	Example
20	5	200	8	Tidigits Digits Recognition
100	20	2000	8	RM1 Command and Control
5000	30	4000	16	WSJ1 5k Small Dictation
20000	80	4000	32	WSJ1 20k Big Dictation
60000	200	6000	16	HUB4 Broadcast News
60000	2000	12000	64	Fisher Rich Telephone Transcription

The major difference between the three acoustic modeling is related to the accuracy and the processing time. The fully continuous model is assigned a separate Gaussian mixture model for each Senone which hugely increase the the Gaussians in the model. On the other hand, the semi-continuous model allows sharing the Gaussians that increases the computations speed. Still, the semi-continuous is more flexible that performs well in the limited amount of training speech hours. PTM model is in between that used relatively low fixed Gaussians that speed up the recognition time. PTM is characterized by fast decoding as well as its ability to handle large amount of speech training collections. Hence, the model selection is based on the speech and accuracy constrains. Hence, PTM might be good option if the decoding time is more important than the accuracy. The PTM model has been farther enhance for better pronunciation variations modeling such the work in [19]. Reference [19] proposes a state-dependent PTM model with variable codebook size to improve the coverage of phonetic variations while maintaining model discriminative ability.

V. PROPOSED METHOD

Preparing a continuous speech corpus is extremely difficult task that demand so much time. It require different phases such as collecting the audio files, segmentation, transcription, and diacritization. Hence, most of the Arabic ASR research studies employed small corpus of isolated words. For this reason, we compile a continuous speech corpus that contains 105,531 words (22,545 unique words) of 1660 speech file. We got the speech files form As-Sabah TV [20] in Kuwait. We performed a preprocessing to divide the file into short files of 30-60 seconds. The speech files cover different news stories and it sums up to 17.6 hours of 29 speakers (19 male speakers and the rest are for female speakers). We split the the corpus into two parts, 13.5 hours for training (1,269 speech files) and 4.1 hours for testing (391 speech files). The speech files were sampled at 16 KHz mono. A silence of 0.1 seconds at the beginning and at the end of each speech file. The proposed method is summarized in the algorithm shown in Fig.2.

- *Prepare the desired speech collection.*
- *Prepare the textual transcription of the collected speech files.*
- *Define the phonemes set.*
- *Based on the training textual transcription and the phonemes set, create the pronunciation dictionary.*
- *Define the senones (n_tied_states).*
- *Train the desired acoustic models.*
- *Prepare the language model using CMUCLMTK [21]*
- *[Optional for speed] How many parts to run Forward-Backward estimation in.*
- *[Optional for speed] Define how many pieces to split decode in.*

Fig. 2. CMU configuration settings for acoustic models

The CMU Sphinxtrain is used for training by performing the following steps: computing feature from audio files, training context independent models, training context dependent models, build trees, prune trees, lattice generation, lattice pruning, and finally decoding using the trained models. The PocketSphinx is used for decoding using the learned acoustic model and the other components such as the pronunciation dictionary and the language model.

VI. EXPERIMENTAL RESULTS

This section presents the experimental results based on the introduced MSA speech corpus. The performance is measured based on different parameters such as the number of Senones and the number of Gaussians. The PocketSphinx configuration file indicates the following constrains regarding the number of Gaussians:

- For fully continuous models, the initial has to be less than the final number of densities.
- For semi-continuous models, the initial and final models have the same density.
- For PTM models, the initial and final models have the same density.
- If you are training semi-continuous or PTM model, use 256 Gaussians [18].

Word Error Rate (WER) was used to evaluate the ASR performance for different acoustic models. Table III shows the WER and the accuracy for the fully continuous model based on different values of Senones and densities. The performance is low for all investigated parameters. The lowest scored WER is 55.9%. This is reasonable since we used a relatively small size corpus. Ideally, ASR requires 200-300 hours speech corpus. One more reason for this low accuracy result is that the used corpus has no filler dictionary. Filler dictionary generally

contains noise and inhalation speech that are not considered as phonemes.

TABLE III. THE CONTINUOUS BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	1➔8	200	44.1	55.9
2	1➔16	1000	42.3	57.7
3	1➔32	1000	42.0	58.0
4	1➔256	1000	43.6	56.4
5	1➔256	5000	89.8	
6	1➔256	200	42.2	57.8

Table IV shows the results of the semi-continuous based models. The results are slightly better than the PTM based model that are shown in Table V. It is clear that the size of the used corpus is suitable for semi-continuous model. The best WER is 64.0% at 256 Gaussians and 1000 Senones.

TABLE IV. THE SEMI-CONTINUOUS BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	256	200	36.8	63.2
2	256	1000	35.8	64.2
3	256	2000	35.9	64.1
4	256	3000	36.1	63.9
5	64	500	36.2	63.8
6	32	1000		

Table V shows the results using PTM based acoustic models. The best WER was found to be 63.5% using 256 Gaussians and 500 and 750 Senones. The experimental results show that the PTM based system is noticeably faster than the semi-continuous based model.

TABLE V. THE PTM BASED PERFORMANCE

Experiment	Densities	Senones	WER (%)	Accuracy (%)
1	256	200	36.6	63.4
2	256	500	36.5	63.5
3	256	750	36.5	63.5
4	256	1000	36.8	63.2
5	256	2000	36.8	63.2
6	256	3000	36.9	63.1
7	256	4000	37.1	62.9
8	256	5000	37.4	62.6
9	256	6000	37.5	62.5
10	256	7000	38.0	62.0
11	256	8000	38.2	61.8
12	256	9000	38.4	61.6
13	256	10000	38.7	61.3

We also considered speeding up the execution time using CMU PocketSphinx configuration (i.e. number of parts to run Forward-Backward estimation➔ $CFG_NPART = 10; and how many pieces to split decode in➔ $DEC_CFG_NPART = 10;). This option is helpful since it clearly reduces the execution time by utilizing a number of processors in multicore machines.

VII. CONCLUSION

We evaluated the performance of three different implementations of acoustic models that include semi-continuous, fully continuous, and PTM acoustic models. We employed CMU PocketSphinx with a speech corpus that contains 17.6 hours of Arabic speech. The results show that the semi-continuous acoustic model slightly outperforms the PTM acoustic models. However, the PTM has less execution time.

REFERENCES

[1] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[2] Huang, Xuedong D., and Mervyn A. Jack. "Semi-continuous hidden Markov models for speech signals." Computer Speech & Language 3.3 (1989): 239-251.

[3] Available: http://cmuSphinx.sourceforge.net/wiki/acousticmodeltypes

[4] Available: http://cmuSphinx.sourceforge.net/wiki/download

[5] Alghamdi, Mansour, Moustafa Elshafei, and Husni Al-Muhtaseb. "Arabic broadcast news transcription system." International Journal of Speech Technology 10.4 (2007): 183-195.

[6] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[7] AbuZeina, Dia, et al. "Within-word pronunciation variation modeling for Arabic ASRs: A direct data-driven approach." International Journal of Speech Technology 15.2 (2012): 65-75.

[8] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[9] Hyassat, Hussein, and Raed Abu Zitar. "Arabic speech recognition using SPHINX engine." International Journal of Speech Technology 9.3-4 (2006): 133-150.

[10] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. "Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus." International Arab Journal of Information Technology (IAJIT) 9.1 (2012): 84-93.

[11] Riedhammer, Korbinian, et al. "Revisiting semi-continuous hidden Markov models." Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012.

[12] Huang, Xuedong, et al. "The SPHINX-II speech recognition system: an overview." Computer Speech & Language 7.2 (1993): 137-148.

[13] Digalakis, Vassilios, and Hy Murveit. "High-accuracy large-vocabulary speech recognition using mixture tying and consistency modeling." Proceedings of the workshop on Human Language Technology. Association for Computational Linguistics, 1994.

[14] Sankar, Ananth. "A new look at HMM parameter tying for large vocabulary speech recognition." ICSLP. 1998.

[15] Lee, Akinobu, et al. "A new phonetic tied-mixture model for efficient decoding." Acoustics, Speech, and Signal Processing, 2000. ICASSP'00. Proceedings. 2000 IEEE International Conference on. Vol. 3. IEEE, 2000.

[16] Wilinski, Piotr, et al. "Toward the border between neural and Markovian paradigms." IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 28.2 (1998): 146-159.

[17] Guo, Guodong, and Stan Z. Li. "Content-based audio classification and retrieval by support vector machines." IEEE transactions on Neural Networks 14.1 (2003): 209-215.

[18] Avialable: http://cmusphinx.sourceforge.net/wiki/tutorialam

[19] Liu, Yi, and Pascale Fung. "State-dependent phonetic tied mixtures with pronunciation modeling for spontaneous speech recognition." IEEE transactions on speech and audio processing 12.4 (2004): 351-364.

[20] Available: http://www.alsabahpress.com/

[21] Available: http://cmusphinx.sourceforge.net/wiki/cmuclmtkdevelopment

Acceptance Letter

waset.org

December 26, 2017

Prof. Dr. Fawaz Al-Anzi
Kuwait University
Kuwait

Herewith, the international scientific committee is happy to inform you that the peer-reviewed draft paper code 17TH100043 entitled (The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition by Fawaz S. Al-Anzi, Dia AbuZeina) has been accepted for poster presentation as well as inclusion in the conference proceedings of the ICNMLKD 2017 : 19th International Conference on Network, Machine Learning and Knowledge Discovery to be held in Bangkok, Thailand during October, 26-27, 2017. The high-impact conference papers will also be considered for publication in the special journal issues at http://waset.org/Publications.

Conference Registration and Writing Formatted Paper:

1. Conference registration documents should be submitted to:
http://waset.org/apply/2017/10/bangkok/ICNMLKD?step=2
2. Word Template File should be Downloaded at
http://waset.org/downloads/template.docx
3. Latex Style File should be Downloaded at http://waset.org/downloads/latex.zip
4. Copyright Transfer Statement Document should be Downloaded at
waset.org/publications/copyright?paperCode=17TH100043

Letter of Invitation and Visa Requirements:
If you need an invitation letter to get an entrance Visa, please fill in the online form to get a letter at http://waset.org/apply/2017/10/bangkok/ICNMLKD?step=1.

We look forward to your participation in the ICNMLKD 2017 : 19th International Conference on Network, Machine Learning and Knowledge Discovery.

Sincerely,

International Scientific Committee
ICNMLKD 2017 : Bangkok, Thailand
http://waset.org/conference/2017/10/bangkok/ICNMLKD

World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:11, No:10, 2017

The Capacity of Mel Frequency Cepstral Coefficients for Speech Recognition

Authors : Fawaz S. Al-Anzi, Dia AbuZeina

Abstract : Speech recognition is of an important contribution in promoting new technologies in human computer interaction. Today, there is a growing need to employ speech technology in daily life and business activities. However, speech recognition is a challenging task that requires different stages before obtaining the desired output. Among automatic speech recognition (ASR) components is the feature extraction process, which parameterizes the speech signal to produce the corresponding feature vectors. Feature extraction process aims at approximating the linguistic content that is conveyed by the input speech signal. In speech processing field, there are several methods to extract speech features, however, Mel Frequency Cepstral Coefficients (MFCC) is the popular technique. It has been long observed that the MFCC is dominantly used in the well-known recognizers such as the Carnegie Mellon University (CMU) Sphinx and the Markov Model Toolkit (HTK). Hence, this paper focuses on the MFCC method as the standard choice to identify the different speech segments in order to obtain the language phonemes for further training and decoding steps. Due to MFCC good performance, the previous studies show that the MFCC dominates the Arabic ASR research. In this paper, we demonstrate MFCC as well as the intermediate steps that are performed to get these coefficients using the HTK toolkit.

Keywords : speech recognition, acoustic features, mel frequency, cepstral coefficients
Conference Title : ICNMLKD 2017 : 19th International Conference on Network, Machine Learning and Knowledge Discovery
Conference Location : Bangkok, Thailand
Conference Dates : October 26-27, 2017

International Science Index, Computer and Information Engineering Vol:11, No:10, 2017 waset.org/abstracts/78382

[ICECTA'2017] Your paper #1570395545 ('Exploring the Language ModelingToolkits for Arabic Text')

EM

EDAS Conference Manager <help@edas-help.com>
on behalf of
tpc.icecta@aurak.ac.ae.edas.info

Reply all

Sat 16/09, 15:13
Fawaz Alanzi;
Diaeddin Abuzeina
Inbox

You forwarded this message on 16/09/2017 15:26

Action Items

Dear Prof. Fawaz Al-Anzi:

Congratulations - your paper #1570395545 ('Exploring the Language Modeling Toolkits for Arabic Text') for ICECTA'2017 has been **accepted** and will be presented in the session titled __.

The reviews are below or can be found at https://edas.info/showPaper.php?m=1570395545.

======= Review Model 1 =======

> *** The Research Novelty: Rate the novelty and originality of the ideas or results presented in the paper. It has been said many times before. (1)

> *** Research Methods: Research methods appropriateness and evidence adequacy (the relation between method and problem) Research method cannot be identified (1)

> *** Future Impact: Future impact on related subject of the research/society. Future impact cannot be identified. (1)

> *** The Clarity of Research Question: The clarity of research question/problem and relevance of the literature overview Problem is not reported, and literature is poorly reviewed (1)

> *** Scientific argument and discussion: The strength of argument and discussion in connecting research finding with original research question. - The summary and conclusions (indicating to what degree the problem has been solved) Conclusions are not supported by decent discussion (2)

> *** Detailed comments: Please justify your recommendation and suggest improvements in technical content or presentation.

The authors in this paper demonstrated two well-known LMs toolkits; the CMU-Cambridge and HTK toolkits. They used a small data set to demonstrate the steps to compute N-grams. They also highlight the basic concepts of the grammars language models that is rarely used in NLP systems. Review: 1. The literature section presented the state of the art of this problem. They need to highlight what are the main features in their model which will improve the solution for this problem. 2. No clear definition of the problem 3. The size of the sample is very small. 4. The formula numbers' are missing here.

======= Review Model 2 =======

> *** The Research Novelty: Rate the novelty and originality of the ideas or results presented in the paper. Some interesting ideas and results on a subject well investigated. (3)

> *** Research Methods: Research methods appropriateness and evidence adequacy (the relation between method and problem) Research method is not easily identified, but only partially appropriate to address the problem (4)

> *** Future Impact: Future impact on related subject of the research/society. Future impact is not explicitly defined; the likely impact is limited (3)

> *** The Clarity of Research Question: The clarity of research question/problem and relevance of the literature overview Problem is incompletely reported, and the relevance to literature review is not clear (3)

> *** Scientific argument and discussion: The strength of argument and discussion in connecting research finding with original research question. - The summary and conclusions (indicating to what degree the problem has been solved) Standard arguments that only partially support the conclusion (3)

> *** Detailed comments: Please justify your recommendation and suggest improvements in technical content or presentation.

The paper computes the N-grams for some Arabic test using existing toolkits.If the paper to be accepted, I recommend the following changes to be taken into consideration:

1. The word "that" is overused and in places where it shouldn't be.
2. The paragraph below (from the abstract) needs rewriting. Also, commas are necessary.

The implementing toolkits include the Carnegie Mellon University (CMU)-Cambridge Language Modeling Toolkit and the Cambridge University Hidden Markov Model Toolkit (HTK) language modeling toolkits.

1. There are some typos in the caption of the figures, please revisit.
2. The literature review can be improved. Instead of saying reference x did this and reference y did that.
3. Cited text should be paraphrased, not just copied as is.
4. The whole paragraph that begins with "Perplexity are the most common metrics used to evaluate N-grams LM..." is taken from elsewhere without citation. Also, maybe Perplexities?
5. References of URLs are not in the proper format
6. The entire paper can be improved

Regards,
Prof. Amjad Omar, TPC Chair
ICECTA'2017
AURAK, UAE

Exploring the Language Modeling Toolkits for Arabic Text

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
abuzeina@ku.edu.kw

***Abstract*—Statistical N-grams language models (LMs) have shown to be very effective in natural language processing (NLP), particularly in automatic speech recognition (ASR) and machine translation. In fact, the successful impact of LMs promotes to introduce efficient techniques as well as different types models in various linguistic applications. The LMs mainly include two types that are grammars and statistical language models that is also called N-grams. The main difference between grammars and statistical language models is that the statistical language models are based on the estimation of probabilities for word sequences while the grammars usually do not have probabilities. Despite there are many toolkits that are used to create LMs, however, this work employs two well-known language modeling toolkits with focus on Arabic text. The implementing toolkits include the Carnegie Mellon University (CMU)-Cambridge Language Modeling Toolkit and the Cambridge University Hidden Markov Model Toolkit (HTK) language modeling toolkits. For clarification, we used a small Arabic text corpus to compute the N-grams for 1-gram, 2-gram, and 3-gram. In addition, this paper demonstrates the intermediate steps that are needed to generate the ARPA-format LMs using both toolkits.**

Keywords— Arabic; language model; grammar; N-grams; perplexity

I. INTRODUCTION

Language models (LMs) are of significant contribution to the performance of natural language processing (NLP) systems such as automatic speech recognition (ASR) and machine translation. LMs have been successfully applied in different linguistic applications such as Part-of-Speech (PoS) tagging, parsing, information retrieval, spell correction, summarization, etc. In particular, LM is a critical component in linguistic applications that produce sequences of words as output. In the last decades, extensive research has been devoted to promote new techniques to compile LMs as well as to address some challenges such as missing some of n-grams. In speech recognition, the ASR decoder uses the information that is provided in the LM to find the best possible word sequence of the testing speech for transcription purpose. In general, it is extremely important for the language applications to have the ability to predict the next word given the previous word(s), or the history.

LMs can be either probabilistic or non-probabilistic. The probabilistic LMs are known as statistical LMs, such as N-grams, while the non-probabilistic is known as "any-word" grammar. Any-word grammar does not use probabilities for words. It is unconstraint grammar that leads to very poor accuracy in continuous ASR systems. That is, any-word grammar relies entirely on the acoustic model. On the other hand, statistical language model is based on computing the probabilities of all word combinations (i.e. all possible word sequences) in the training source text. Statistical language models are generally demonstrated using ARPA format textual files that include the statistical estimation of the desired N-grams, typical up to 3-grams. No doubt, compiling a statistical language model requires a large number of words from different textual resources. In addition, the data should not be too specific to a particular domain; otherwise, it will not generalize well to the sentences in question.

During ASR decoding, the recognizer employs the language model to transcribe speech files using the acoustic model and the dictionary (vocabulary). Hence, the most likely hypothesis for each testing speech file is generated as an output. Employing language models reinforces the speech recognition accuracy as the more you can constrain the range of possible utterances, the more accurate the recognizer will be. Based on the information that is provided in the N-grams, the probability of a words sequence is computed using the following formula [1]:

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k | w_1^{k-1})$$

Where n is limited to include the words' history. Hence, the Chain Rule applied to compute joint probability of words in sentence. LMs utilize Markov assumption to simplify data estimation. For instance, for *n*=2, the bigram is calculated for the words sequence as follows:

$$P(w_1\, w_2) = P(w_2\,|w_1)P(w_1\,)$$

This work aims at demonstrating the process of computing the N-grams for small Arabic text corpus that contains five sentences using two well-known toolkits. The toolkits are the Carnegie Mellon University (CMU) -Cambridge language modeling toolkit [2] and the Cambridge University Hidden Markov Model Toolkit (HTK) language modeling tools [3]. Of course, there are other popular tools such as the SRI Language Modeling Toolkit (SRILM) [4]. For clarification, we demonstrate all intermediate steps that are required to produce the language models, which include model estimation using the training data. This work also demonstrates some grammars that are mainly used for isolated words such as digits and control commands. The implementation requires to run the command line under UNIX. In this work we used Cygwin which is a Unix-like environment for Microsoft Windows.

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

In next section, we present the literate review. In section 3, we present the grammars followed by the background of N-gram language models in section 4. In section 5, we present HTK toolkits followed by CMU-Cambridge toolkits in section 6. Finally, we conclude in section 7.

II. Literature Review

The literature shows that LMs have been exploited in many linguistic applications. For instances, Reference [5] used CMU-Cambridge language modeling toolkit LM for continuous Arabic ASR. Reference [6] augmented naive Bayes models with statistical N-gram language models to address short-comings of the standard naive Bayes text classifier. Reference [7] proposed a model, called emoticon smoothed language model for Twitter sentiment analysis. Reference [8] investigated the document categorization task with statistical language models. Reference [9] reported the benefits of largescale statistical language modeling in machine translation. Reference [10] proposed to use a new statistical language model that is based on a continuous representation of the words in the vocabulary for machine translation. Reference [11] employs statistical language models in temporal action detection. Reference [12] explores recent advances in recurrent neural networks for large scale language modeling. Reference [13] employs statistical LMs for large statistical machine translation task. Reference [14] explores the application of neural LMs to machine translation. Reference [15] employs LMs for text summarization.

III. Grammars

In this section, we demonstrate the grammars language models. Such grammars are simple that do not have probabilities and are designed according to the information that is provided in the corresponding application. That is, grammars mainly contain isolated words such as commands, control words, and digits. However, grammars might allow sequences of words. Fig. 1 shows a simple grammar for ten digits that can be used in continuous speech recognition to choose one or more words from the list. The grammar is written using JSGF format as shown in Fig. 1.

```
#JSGF V1.0;
grammar myGrammar;
public <command> = <word>* ;
<word> =( One | Two | Three | … | Ten);
```

Another form of the ten digits grammar

```
$WORD = (One | Two | Three | … | Ten);
($WORD)
```

Fig. 1. A part of any-word grammar for ten digits

Grammars are usually written by hand or it can be generated using a program. Despite most of grammars do not use probabilities, however, some elements might be weighted. In fact, grammars are rarely used in ASR systems since the probabilistic models of a language is more useful than the hard models (i.e. grammars) of the legal sentences in the languages.

IV. N-grams Language Models

Statistical N-grams language models are the most widely used language model in speech recognition. That is, the goal of LM is to compute the probability of a sentence or sequence of words. The most likely sequence of words is estimated, given the speech feature vectors, is given by [1]:

$$\widehat{w} = \underset{W \in L}{\operatorname{argmax}} \frac{P(O|W)P(W)}{P(O)} = \underset{W \in L}{\operatorname{argmax}} P(O|W)P(W)$$

Where $\widehat{w}$ is the most likely recognized words, $P(O|W)$ is the probability of the feature vectors, given a sequence of words that is computed using the acoustic model, $P(W)$ is the probability of the words sequence that is computed using the language model. $P(O)$ is the probability of the acoustic observation sequence and can be ignored. Hence, the statistical language model has to be computed at the first place in order to decode the testing speech files in ASR systems. The statistical N-grams language model is trained by counting N-grams occurrences in a large transcription corpus to be then smoothed and normalized. N-gram models can be trained by counting and normalizing. The following formula is used to estimate the N-grams parameter [1]:

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{Count(w_{n-N+1}^{n-1}w_n)}{Count(w_{n-N+1}^{n-1})}$$

One major problem in LMs is unseen words or n-grams that are found in the testing set while, at the same time, out of vocabulary. Accordingly, a probability of 0.0 is given to the items that are not seen in the training data. That is, not all n-grams will be present (i.e. not observed) in the training data. One solution is smoothing by assigning non-zero or small probabilities to unseen n-grams in which all word sequences can occur with some probability. Hence, smoothing provides a better way of estimating the probability of zero frequency n-grams which never occur in order to produce generalized LMs. Smoothing is also called discounting. When creating a language model, it is more efficient to use log probabilities rather than actual probabilities due to the risk of numerical underflow especially in very long strings. It is also efficient in ASR decoding algorithm such as Viterbi algorithm.

Creating an N-grams language model follows three main steps that are: compute the word unigram counts, convert the word unigram counts into a vocabulary list, and generate bigram and trigram (or more) tables based on this vocabulary. As a preprocessing step, it is must to include special words such as <s> to indicate for the "start of sentence" and the </s> to indicate for the "end of the sentence". CMU-Cambridge toolkits uses <UNK> token to indicate for unknown words whereas HTK toolkits uses !!UNK for the same purpose. In order to demonstrate the process to create a statistical language model for Arabic text, we prepared a small set that contains five sentences to demonstrate the required steps. The training data set is shown in Fig. 2. The sentences include 22 unique words.

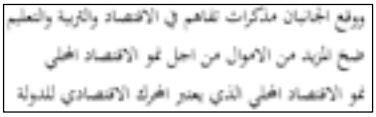


Fig. 2. A small corpus tha contains three sentences

V. THE HTK TOOLKITS

HTK provides two approaches to generate N-grams. The first method employs ***HLStats*** function, which is used exclusively to computer bigram language model (i.e., 2-grams). The words probabilities is computed using the following formula [16]:

$$p(i,j) = \begin{cases} (N(i,j) - D)/N(i) & if\ N(i,j) > t \\ b(i)p(j) & otherwise \end{cases}$$

Where $N(i,j)$ is the number of times word j follows word i and $N(i)$ is the number of times that word i appears. D is the discount constant that has a default value 0.5. t is a threshold that is used to ensure that all bigram probabilities for a given history sum to one. The LMs that is generated using ***HLStats*** use probabilities as base-10 logs.

The second method implements a series of functions to computer N-grams. In this work, we implement the second method that is demonstrated by HTK toolkit in [16] to compute N-grams for general N-grams. That is, it is not restricted for bigram as the case in ***HLStats*** method. However, the second method is not one command as the case in ***HLStats***, but it requires several steps to compute N-grams as the following [16]: ***LNewMap*** to prepare an initial empty word map file. This can help to add future n-gram files without having to rebuild existing ones. The created word map file contains just a header and no words. ***LGPrep*** to process the training text data to produce all newly encountered words and the identifiers that the tool has assigned them. ***LGList*** contains the collected N-grams (3-grams in our case) as shown in Fig. 3.

```
3-Gram File holmes.0/gram.0[30 entries]:
 Text Source: LM
.           <s>         ضع          : 1
.           <s>         نمو         : 1
</s>        .           <s>         : 2
<s>         ضع          المزيد      : 1
<s>         نمو         الاقتصاد!   : 1
<s>         ووقع        الجانبان    : 1
اجل         نمو         الاقتصاد!   : 1
الاقتصاد!   المحلي      </s>        : 1
الاقتصاد!   المحلي      الذي        : 1
الاقتصاد!   والتربية    والتعليم    : 1
الاقتصادي!  للدولة      </s>        : 1
...
من          اجل         نمو         : 1
من          الاموال     من          : 1
نمو         الاقتصاد!   المحلي      : 2
والتربية    والتعليم    </s>        : 1
والتعليم    </s>        .           : 1
ووقع        الجانبان    مذكرات      : 1
يعتبر       المحرك      الاقتصادي   : 1
30 ngram entries printed
```

Fig. 3. A part of the the 3-grams in the triaing data

LGCopy is employed to derive a sequenced set of N-grams files. The word list should be supplied in a spate file that contains the system's vocabulary. The unknown word symbol defaults to !!UNK. ***LGList*** saves the new word map containing the new class symbols (!!UNK in this case) and only words in the vocabulary. ***LFoF*** produces a frequency of frequency (FoF) table for the chosen vocabulary list. Finally, ***LBuild*** builds the actual language model. Fig. 4 shows a part of the generated 3-grams language model using ARPA format.

The 3-grams shows in Fig. 4 shows the contents of the N-grams language model along with the corresponding probabilities of the 1-grams cases (23 unigrams), 2-grams cases (contains 4 bigrams), and 3-grams cases (2 trigrams). The ARPA format shown in Fig. 4 is the language model form that is generally used in in speech recognition systems. In Fig. 4, the probabilities (in term of $\log_{10}$) are stored on the left of the n-grams while the back-off weight (in terms of $\log_{10}$) is stored on the right of the word(s).

```
\data\
ngram 1=23
ngram 2=4
ngram 3=2
\1-grams:
-1.1614 !!UNK    -2.0000
-1.1614 </s>     -1.9690
-99.9900         <s>
-1.4624 اجل
-0.9853 الاقتصاد!  -0.4375
-1.4624 الاقتصادي!
...
-1.1614 نمو      -1.9526
-1.4624 والتربية
-1.4624 والتعليم
-1.4624 ووقع
-1.4624 يعتبر
\2-grams:
-0.0044 !!UNK <s>
-0.0044 </s> !!UNK       +0.0000
-0.1805 المحلي الاقتصاد!
-0.0044 الاقتصاد! نمو    -1.5315
\3-grams:
-0.0044 </s> !!UNK <s>
-0.0044 المحلي الاقتصاد! نمو
\end\
```

Fig. 4. A part of the 3-grams using HTK toolkit

VI. THE CMU-CAMBRIDGE TOOLKITS

We used the CMU-Cambridge toolkit to compute the N-grams for the dataset that is described in section 4. In particular, we computer the 1-grams, 2-grams, and the 3-grams. The CMU-Cambridge toolkit uses the following five commands to produce the LM dump file: ***text2wfreq***, ***wfreq2vocab***, ***text2idngram*** , ***idngram2lm***, ***lm3g2dmp*** [17]. The outputs of the previous commads are shown in Fig. 5.

text2wfreq	wfreq2vocab	text2idngram
والتربية 1 اجل 1 ضع 1 في 1 من 2 الذي 1 نمو 2 ووقع 1 <s> 3 مذكرات 1 الاموال 1 الاقتصادي! 1 الجانبان 1 تفاهم 1 والتعليم 1 الاقتصاد! 3 المحرك 1 المحلي 2 يعتبر 1 المزيد 1 للدولة 1 </s> 3	1) </s> 2) <s> 3) اجل 4) الاقتصاد! 5) الاقتصادي! 6) الاموال 7) الجانبان 8) الذي 9) المحرك 10) المحلي 11) المزيد 12) تفاهم 13) ضع 14) في 15) للدولة 16) مذكرات 17) من 18) نمو 19) والتربية 20) والتعليم 21) ووقع 22) يعتبر	1 2 13 1 1 2 18 1 2 13 11 1 2 18 4 1 2 21 7 1 3 18 4 1 4 10 1 1 4 10 8 1 4 19 20 1 5 15 1 1 ... 11 17 6 1 12 14 4 1 13 11 17 1 14 4 19 1 16 12 14 1 17 3 18 1 17 6 17 1 18 4 10 2 19 20 1 1 20 1 2 1 21 7 16 1 22 9 5 1

Fig. 5. The outpus of three commands to generat LM

The descriptions of the CMU-Cambridge toolkits command is as follows: ***text2wfreq***: list of every word which occurred in the text, along with its number of occurrences. ***wfreq2vocab***: a vocabulary file. ***text2idngram***: list of every id n-gram which

occurred in the text, along with its number of occurrences. ***idngram2lm*** : a language model, in either binary format, or in ARPA format. ***lm3g2dmp***: convert the language model file from ARPA format to DMP format. Fig. 6 shows the outputs of the previous commands. This is a 3-gram language model, based on a vocabulary of 22 words. We highlight that the output of the ***text2idngram*** command is the 3-grams occurred in the source text. For instance, the sequence 18 4 10 reprents the trigram "نمو الاقتصاد المحلي" and it occurs two times as indicated in the Fig. 5, the right colmun.

The ARPA-standard format LM that is shown in Fig. 6 contains 23 items of the 1-grams, 26 items of 2-gram, and 28 item of the 3-grams. For each n-gram, the value on the left represents the actual probability while the value on the right represents the back-off weights. According to the LM in Fig. 6, the actual probability of the word "الاقتصاد" is -0.9853 (in $\log_{10}$ format). The probability in standard from is $10^{-0.9853}$=0.101. The LM also shows that the probability of the 3-gram " نمو الاقتصاد المحلي" is greater than the 3-gram "ووقع الجانبان مذكرات" as the actual probability of the first 3-gram is ($10^{-0.1761}$) 0.6666 while the prbability of the second 3-gram is almost zero ($10^{-99.9990}$).

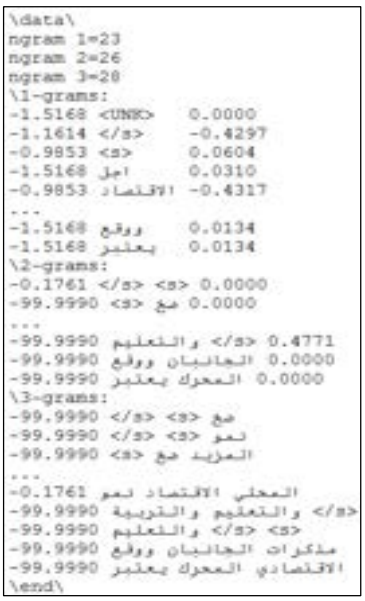


Fig. 6. A part of the the 3-grams using CMU-Cambridge toolkit

Perplexity are the most common metrics used to evaluate N-grams LM. Perplexity is defined in terms of the inverse of the average log likelihood per word. It is an indication of the average number of words that can follow a given word, a measure of the predictive power of the language model. It is a way to measure the quality of a model independent of any NLP system. The lower perplexity system is considered better than one of higher perplexity. The perplexity formula is:

$$PP(W) = N\sqrt{\frac{1}{P(w1, w2, \dots, w_N)}}$$

Where PP is the perplexity, P is the probability of the word set to be tested W=w1, w2, … , w_N, and N is the total number of words in the testing set. Since we used a very small data set, we did not evaluate the perplexity in this work.

VII. CONCLUSION

This paper demonstrates two well-known LMs toolkits; the CMU-Cambridge and HTK toolkits. We used a small data set to demonstrate the steps to compute N-grams. We also highlight the basic concepts of the grammars language models that is rarely used in NLP systems. As a future work, we recommend to investigate new research directions to run NLP systems while skipping the language models. That is, evaluating the performance if we employ an ASR system that entirely use acoustic models.

REFERENCES

[1] Jurafsky, Dan, and James H. Martin. Speech and language processing. Vol. 3. Pearson, 2014.

[2] Available: http://www.speech.cs.cmu.edu/SLM/toolkit.html

[3] Available: http://htk.eng.cam.ac.uk/download.shtml

[4] Available: http://www.speech.sri.com/projects/srilm/

[5] AbuZeina, Dia, et al. "Cross-word Arabic pronunciation variation modeling for speech recognition." International Journal of Speech Technology 14.3 (2011): 227-236.

[6] Liu, Kun-Lin, Wu-Jun Li, and Minyi Guo. "Emoticon smoothed language models for twitter sentiment analysis." AAAI. 2012.

[7] Peng, Fuchun, Dale Schuurmans, and Shaojun Wang. "Augmenting naive bayes classifiers with statistical language models." Information Retrieval 7.3 (2004): 317-345.

[8] Tantug, Ahmet Cüneyd. "Document categorization with modified statistical language models for agglutinative languages." International Journal of Computational Intelligence Systems 3.5 (2010): 632-645.

[9] Brants, Thorsten, et al. "Large language models in machine translation." In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. 2007.

[10] Schwenk, Holger, Daniel Dchelotte, and Jean-Luc Gauvain. "Continuous space language models for statistical machine translation." Proceedings of the COLING/ACL on Main conference poster sessions. Association for Computational Linguistics, 2006.

[11] Richard, Alexander, and Juergen Gall. "Temporal action detection using a statistical language model." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[12] Jozefowicz, Rafal, et al. "Exploring the limits of language modeling." arXiv preprint arXiv:1602.02410 (2016).

[13] Schwenk, Holger, Anthony Rousseau, and Mohammed Attik. "Large, pruned or continuous space language models on a gpu for statistical machine translation." Proceedings of the NAACL-HLT 2012 Workshop: Will We Ever Really Replace the N-gram Model? On the Future of Language Modeling for HLT. Association for Computational Linguistics, 2012.

[14] Vaswani, Ashish, et al. "Decoding with Large-Scale Neural Language Models Improves Translation." EMNLP. 2013.

[15] Rush, Alexander M., Sumit Chopra, and Jason Weston. "A neural attention model for abstractive sentence summarization." arXiv preprint arXiv:1509.00685 (2015).

[16] Young, Steve, et al. "The HTK book (for HTK version 3.4)." Cambridge university engineering department 2.2 (2006): 2-3.

[17] http://www.speech.cs.cmu.edu/SLM/toolkit_documentation.html

Notification Mail - CDKP Conference

F

Fifth International Conference on Data Mining & Knowledge Management Process (CDKP 2016) <ckdpconf@icaita.org>

From: Fifth International Conference on Data Mining & Knowledge Management Process (CDKP 2016) <ckdpconf@icaita.org>
To: fawaz alanzi <fawaz.alanzi@ku.edu.kw>
Cc: abuzeina@ku.edu.kw
Sent: Wed, 19 Oct 2016 15:46:50 +0300 (AST)
Subject: Notification Mail - CDKP 2016

DEAR AUTHOR,

First of all, thank you very much for submitting your paper "A Survey of Markov Chain Models In Linguistics Applications" to CDKP 2016 to be held in DUBAI, UAE, NOVEMBER 12 ~ 13, 2016. Based upon the reviewer's reports, we are pleased to inform you that your paper has been ACCEPTED by the conference and will be included in the proceedings published by COMPUTER SCIENCE CONFERENCE PROCEEDINGS in COMPUTER SCIENCE & INFORMATION TECHNOLOGY (CS & IT) series. Congratulations on your excellent work!

In order to achieve the highest quality proceedings, we urge you to carefully consider the reviewer's comments, if any, when preparing the final version of your paper.

1. Please read the following Information carefully to prepare a final manuscript of your paper HTTP://AIRCCSE.ORG/JOURNAL/AIRCC_TEMPLATE.DOC Maximum number of pages without extra payment is 20 (CCSP format). For each extra page you have pay 50 USD additionally.

2. Submit your final camera ready version of paper (.doc version+ .pdf version) and filled CR form CKDPCONF@YAHOO.COM OR CKDPCONF@ICAITA.ORG

3. FINAL MANUSCRIPT SUBMISSION DETAILS:

a) When submitting your final manuscript, please ensure that you send us all source files such as .doc and .pdf

Copy right form: VOLUME EDITORS: DAVID WYLD ET AL.,

b) Please make sure you enter volume editor's name (David Wyld et al.,) in the copy right form.

REGISTRATION DETAILS:

FOR INTERNATIONAL AUTHORS

Regular Registration: 275USD

For International Authors

Payment Methods : WIRE TRANSFER/BANK TRANSFER/NET TRANSFER /PAYPAL

FOR INDIAN AUTHORS:

Payment Methods : WIRE TRANSFER/BANK TRANSFER/NET TRANSFER/DEMAND DRAFT

BANK DETAILS

ACCOUNT NAME : NNN NET SOLUTION PRIVATE LTD.,
ACCOUNT NUMBER : 31071902971
Name of the Bank :State Bank of India, Chennai (7108)
Swift code : SBI NIN BB 458
Address of the Bank : 42-43,East Mada Street, Villivakkam, Chennai, TamilNadu, India
Zip 600049
IFSC Code: SBIN0007108 (for local transaction)
Address of the Beneficiary: 1285 B, Kambar Colony, Anna Nagar, Chennai, Tamil Nadu, India,

Zip 600040

Online Payment Method

**YOU CAN ALSO PAY THROUGH PAYPAL

In paypal you have to choose SEND PAYMENT and enter the mail id PAYAIRCC@GMAIL.COM<MAILTO:PAYAIRCC@GMAIL.COM> and enter the amount.

NOTE: Indian authors can pay local currency (Indian Rupees) to respective bank accounts (for currency conversion use WWW.XE.COM [1] [1 [1]] [1 [1]] [1 [1]] [1 [1]] [1 [2]]).
International authors please use USD/EURO/AUS $.

AFTER WIRE TRANSFER/ BANK TRANSFER, PLEASE SEND SCANNED COPY OF THE RECEIPT TO

CKDPCONF@YAHOO.COM OR CKDPCONF@ICAITA.ORG

NOTE: THE REGISTRATION COVERS CONFERENCE PROCEEDINGS, ADMISSION TO ALL WORKSHOPS, COFFEE BREAKS AND BANQUET (ONLY ONE DAY OF THE CONFERENCE). THE REGISTRATION DOESN'T COVER BREAKFAST, ACCOMMODATION, TRANSPORTATION,
CONFERENCE BAG AND LOCAL TOUR ETC.

JOURNAL VERSION: Extended version of your conference paper will be published in the Journal as mentioned in the web site. There is no publication charge for journal publication.

IMPORTANT DATES

Camera-ready Due : OCTOBER 26, 2016

Registration Deadline : OCTOBER 26, 2016

All detailed information on deadlines, final paper uploading, registrations, and hotel reservation will be available on the web site soon:

Please note that at least one of the authors of accepted papers is required to register and present the paper at the conference;

Thank you again for helping to ensure the success of CDKP 2016. We are looking forward to meeting you at the CDKP 2016 in DUBAI, UAE, NOVEMBER 12 ~ 13, 2016.

--

THANKS!
****
SECRETARY (EDITORIAL),FIFTH INTERNATIONAL CONFERENCE ON DATA MINING & KNOWLEDGE MANAGEMENT PROCESS (CDKP 2016)
HTTP://AIRCCSE.ORG/CONFERENCE.HTML
HTTP://ICAITA.ORG/2016/CDKP/INDEX.HTML

****

Links:
------
[1] http://www.xe.com/
--
THANKS!
****
SECRETARY (EDITORIAL),FIFTH INTERNATIONAL CONFERENCE ON DATA MINING & KNOWLEDGE MANAGEMENT PROCESS (CDKP 2016)
HTTP://AIRCCSE.ORG/CONFERENCE.HTML
HTTP://ICAITA.ORG/2016/CDKP/INDEX.HTML

****

Links:
------
[1] http://WWW.XE.COM
[2] http://www.xe.com/
--
THANKS!
****
SECRETARY (EDITORIAL),FIFTH INTERNATIONAL CONFERENCE ON DATA MINING & KNOWLEDGE MANAGEMENT PROCESS (CDKP 2016)
HTTP://AIRCCSE.ORG/CONFERENCE.HTML
HTTP://ICAITA.ORG/2016/CDKP/INDEX.HTML

****

Links:
------

[1] http://WWW.XE.COM
[2] http://airccse.org/journal/ijdms/index.html
[3] http://airccse.org/journal/ijdkp/ijdkp.html

--
THANKS!
****
SECRETARY (EDITORIAL),FIFTH INTERNATIONAL CONFERENCE ON DATA MINING & KNOWLEDGE MANAGEMENT PROCESS (CDKP 2016)
HTTP://AIRCCSE.ORG/CONFERENCE.HTML
HTTP://ICAITA.ORG/2016/CDKP/INDEX.HTML

****

Links:
------
[1] http://WWW.XE.COM
[2] http://airccse.org/journal/ijdms/index.html
[3] http://airccse.org/journal/ijdkp/ijdkp.html

A SURVEY OF MARKOV CHAIN MODELS IN LINGUISTICS APPLICATIONS

Fawaz S. Al-Anzi and Dia AbuZeina

Department of Computer Engineering, Kuwait University, Kuwait City, Kuwait
fawaz.alanzi@ku.edu.kw , abuzeina@ku.edu.kw

ABSTRACT

Markov chain theory is an important tool in applied probability that is quite useful in modeling real-world computing applications. For a long time, rresearchers have used Markov chains for data modeling in a wide range of applications that belong to different fields such as computational linguists, image processing, communications, bioinformatics, finance systems, etc. This paper explores the Markov chain theory and its extension hidden Markov models (HMM) in natural language processing (NLP) applications. This paper also presents some aspects related to Markov chains and HMM such as creating transition matrices, calculating data sequence probabilities, and extracting the hidden states.

KEYWORDS

Markov chains, Hidden Markov Models, computational linguistics, pattern recognition, statistical

1. INTRODUCTION

Markov chains theory is increasingly being adopted in real-world computing applications since it provides a convenient way for modeling temporal, time-series data. At each clock tick, the system moves into a new state that can be the same as the previous one. A Markov chain model is a mathematical tool that capture the patterns dependencies in pattern recognition systems. For this reason, Markov chain theory is appropriate in natural langue processing (NLP) where it is naturally characterized by dependencies between patterns such as characters or words.

Markov chains are directed graphs (a graphical model) that are generally used with relatively long data sequences for data-mining tasks. Such tasks include prediction, classification, clustering, pattern discovery, software testing, multimedia analysis, networks, etc. Reference [1] indicated that there are two reasons of Markov chains popularity; very rich in mathematical structure and work well in practice for several important applications. Hidden Markov models (HMM) is an extension of Markov chains that used to find the hidden system's states based on the observations.

In order to facilitate the research in this direction, this paper provides a survey of this so popular data modeling technique. However, because of the wide range of the research domains that use this technique. We specifically focus on the linguistics related applications. Reference [2] list some domains that utilize Markov chains theory which include: physics, chemistry, testing, speech recognition, information sciences, queueing theory, internet applications, statistics, economics and finance, social sciences, mathematical biology, genetics, games, music, baseball, Markov text generators, bioinformatics. Reference [3] lists the five greatest applications of Markov chains that include Scherr's application to computer performance evaluation, Brin and Page's application to PageRank and Web Search, Baum's application to HMM, Shannon's application to information theory, and Markov's application to Eugeny Onegin.

This paper is organized as follows. The next section presents a background of Markov chains theory. Section 3 highlights the main concepts of HMM followed by a literature review of Markov chains and HMM in section 4. Finally, we conclude in section 5.

2. MARKOV CHAINS

Markov chains are quite useful in modeling computational linguistics. A Markov chain is a memoryless stochastic model that describes the behaviour of an integer-valued random process. The behaviour is the simple form of dependency in which the next state (or event) depends only on the current state. According to [4], a random process is said to be Markov if the future of the process, given the present, is independent of the past. To describe the transitions between states, a transition diagram is used to describe the model and the probabilities of going from one state to another. For example, Figure 1 shows a Markov chain diagram with three states (Easy, Ok, and Hard) that belong to exam cases (i.e. states). In the figure, each arc represents the probability value for transition from one state to another.

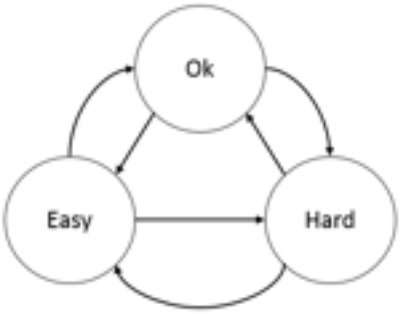


Figure 1. A Simple Markov chain with three states

The Markov chain diagrams are generally represented using state transition matrices that denote the transition probabilities from one state to another. Hence, a state transition matrix is created using the entire states in the system. For example, if a particular textual application has a training data that contains N states (e.g. the size of lexicon), then the state transition matrix is described by a matrix A= {aij} of size N*N. In matrix A, the element aij denote the transition probability from a state i to a state j. Table 1 shows how the state transition matrix used to characterize the Markov diagram shown in Figure 1. That is, the matrix carries the state transitions probabilities between the involved states (Easy, Ok, and Hard). For illustration, the P(E|H) denote to the probability of the next exam to be Easy given that the previous exam was Hard.

Table 1. A state transition matrix of three states

		Next Exam		
	State	**Easy (E)**	**Ok (O)**	**Hard (H)**
Previous Exam	**Easy (E)**	P(E|E)	P(O|E)	P(H|E)
	Ok (O)	P(E|O)	P(O|O)	P(H|O)
	Hard (H)	P(E|H)	P(O|H)	P(H|H)

In Table 1, the sum of the probability values at each row is 1 as the the sum of the probabilities coming out of each node should be 1. Hence, P(E|E)+P(O|E)+P(P(H|E) equal 1. Markov chain is a worthy topic that has many details. For examples, it contains discrete-time, continuous-time, time-reversed, reversible, and irreducible Markov chains. The case shown in Figure 1 is irreducible case, also called ergodic, where it is possible to go from every state to every state.

To illustrate a simple Markov chain data model, a small data set contains two English sentences used to create a transition matrix based on the neighbouring characters sequences. The sentences are inspirational English quotes picked from [5]:

(1) Power perceived is power achieved. (2) If you come to a fork in the road, take it.

Figure 2 shows the transition matrix of these quotes by counting the total number of occurrences of the adjacent two character sequences. It is a 19 × 19 matrix where the value 19 is the total number of unique characters appeared in the sentences (i.e the two quotes). In this example, creating transition matrix is case insensitive where D is same as d, as an example. In addition, a space between two words discarded and not considered in the transition matrix. Figure 2 also shows that the maximum number in the matrix's entries is 3 (a highlighted underlined value) which means that moving from character e to r (e➔r) is the most frequently sequence appeared in this small corpus. The words that contains this sequence are :{ Power (two times) and perceived}.

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	2	0	0	0	1	0	0	0	0	0	**3**	0	0	0	1	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0	1	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	1	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	0	2	0
p	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
r	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 2. A transition matrix of two characters sequences

Based on the information provided in the transition matrix shown in Figure 2. It is possible to answer some questions related to the given data collection. Among inquires, what is the total number of the two characters sequences appeared in the given data set? What are the two characters sequences that did not appear in the data collection? What is the least frequently two characters sequences in the data set? Accordingly, Markov chains are used as prediction systems such as weather forecasting. Therefore, it is possible to predict the tomorrow's weather according to the today's weather. For example, if we have two states (Sunny, Rainy), and the requirement is to find the probability P(Sunny|Rainy), Markov chains make it possible based on the information provided in the probability transition matrix. Another example of the using Markov chains is banking industry. A big portfolio of banks is based on loans. Therefore, Markov chains are used to classify loans to different states such as Good, Risky, and Bad loans.

For simplicity, the information presented in Figure 2 shows the transition matrix based on total number of occurrences. Figure 3 shows the same information but using probabilities instead of

the number of occurrences. That is, it contains the probability of moving from one character to another. As previously indicated, the sum of entries at each row is equal 1. In Figure 3, any matrix entry that has 0 means that there is no transition at that case. Similarly, if the matrix entry is 1, it means that there is only one possible output of that state. For example, the character "o" comes after "y", and this is the only possible arc of the state "y".

	a	c	d	e	f	h	i	k	m	n	o	p	r	s	t	u	v	w	y
a	0	0.33	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0	0	0
c	0	0	0	0.33	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
d	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	0	0	0.29	0	0	0	0.14	0	0	0	0	0	0.43	0	0	0	0.14	0	0
f	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
h	0	0	0	0.5	0	0	0.5	0	0	0	0	0	0	0	0	0	0	0	0
i	0	0	0	0.17	0.17	0	0	0	0	0.17	0	0	0	0.17	0.17	0	0.17	0	0
k	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
n	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
o	0.17	0	0	0	0	0	0	0	0.17	0	0	0	0.17	0	0	0.17	0	0.33	0
p	0	0	0	0.33	0	0	0	0	0	0	0.67	0	0	0	0	0	0	0	0
r	0	0.33	0	0	0	0	0	0.33	0	0	0.33	0	0	0	0	0	0	0	0
s	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
t	0.33	0	0	0	0	0.33	0	0	0	0	0.33	0	0	0	0	0	0	0	0
u	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
v	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
w	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
y	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Figure 3. A probability transition matrix of two characters sequences

3. HIDDEN MARKOV MODELS

Hidden Markov models (HMM) is an extension to Markov chains models as both used for temporal data modeling. However, the difference is that the states in Markov chain models are directly observed while they are hidden in the case of HMM. We explain the concept of HMM based on Figure 1 that shows a three exam's states Markov diagram. As a very simple example, supposed that a student's parents want to know the levels (i.e the difficulty) of their son's exams, naturally, it is possible to recognize the exam as Easy or Ok if the son feels Fine. Similarly, it is possible to recognize the exam as Hard if the son looks Scared. From the parents' point of view, the required states (i.e. Easy, Ok, or Hard) are hidden. However, they directly observe the student's reaction or feeling. Hence, the parents might use the observed reaction as an indication to know the hidden states. HMM is described using three matrices: the initial probability matrix, the observation probability matrix, and the state transition matrix. Figure 4 shows a HMM diagram that shows the states and the observations. In the figure, each arc represents the probability between the states and between the states and the observations.

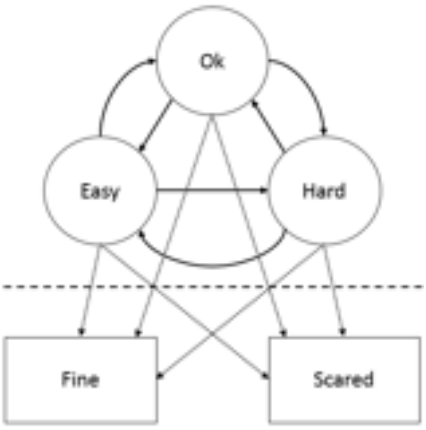


Figure 4. A HMM diagram with the transition and the observation arcs

Based on the information provided in the matrices, either Baum-Welch (also called any path) or Viterbi (also called best path) algorithms used to find the probability scores during recognition phase. Figure 5 shows the trellis diagram for exam states HMM. While Baum-Welch algorithm is used to compute the recognitin probability of a sequence, Viterbi is used to find the best-state sequence associated with the given observtatin, this procoss is also known as back-trakcing. Hence, after computing the observations sequence probability and finding the maximum probability (supposed the star in Figure 5), the Viterbi algorithm leads the process back to identify the states (sources) from which the observations sequence have been emitted. In Figure 5, the maximum probalities supposed to be achieved at the states shown using the dotted lines: Ok, Easy, Hard, respectively.

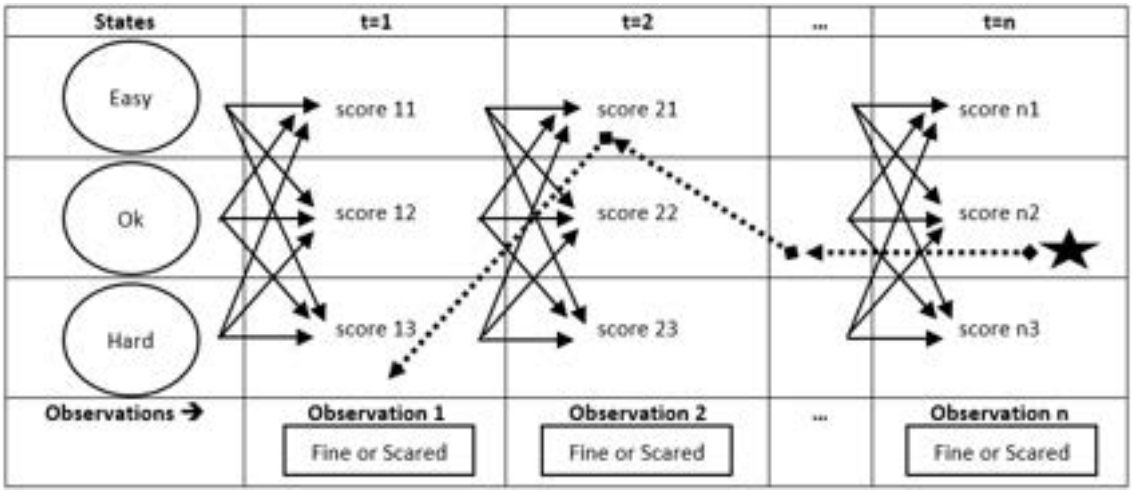


Figure 5. Trellis diagram of three states HMM

4. LINGUISTIC APPLICATIONS

In the literature, there are quite many works on modeling content dependencies for linguistics applications. Markov chain models and HMMs are of great interest to linguistic scholar who primarily work on data sequences. Even though this study focuses on linguistic applications, however, Markov chains used to model a variety of phenomena in different fields. The following are some of studies employed Markov chains. We intentionally ignored the references as the literature has too many studies employed Markov chains:

image processing, text and image compression, video segmentation ,forecasting, networking, signal processing, communications, software testing, genetics, bioinformatics, genome structure recognition, anomaly detection, tumour classification, water quality, epidemic spread, wind power, malicious and cyber-attack detection, traffic management, physics, chemistry, mathematical biology, games, music, multimedia processing, business

The following two subsections include some of the linguistic studies that utilized Markov chain theory. Linguistic applications topics mainly include (but not limited) speech recognition, speech emotion recognition, part-of-speech tagging, machine translation, text classification, text summarization, optical character recognition (OCR), named entity recognition, question answering, authorship attribution, etc. For the reader who interested in NLP, Reference [6] is a good reference as it demonstrates a thorough study of NLP (Almost) from Scratch.

4.1. Markov chains based research

The literature has a large number of studies that employ Markov chains for NLP applications. The following are some linguistic related applications. Reference [7] proposed a word-dividing algorithm based on statistical language models and Markov chain theory for Chinese speech processing. Reference [8] presented a semantic indexing Markov chains algorithm that uses both audio and visual information for event detection in soccer programs. Reference [9] investigated the use of Markov Chains and sequence kernels for the task of authorship attribution. Reference [10] implemented a probabilistic framework for support vector machine (SVM) that allows for automatic tuning of the penalty coefficient parameters and the kernel parameters via Markov chain for web searching via text categorization. Reference [11] demonstrated an automatic video annotation using multimodal Dirichlet process mixture model by collecting samples from the corresponding Markov chain. Reference [12] used a linguistic steganography detection method based on Markov chain models. Reference [13] showed how probabilistic Markov chain models can be used to detect topical structure in large text corpora.

Reference [14] proposed a method of recognizing location names from Chinese texts based on Max-Margin Markov Network. Reference [15] utilized Markov chain and statistical language models in a linguistic steganography detection algorithm. Reference [16] proposed a Markov chain based algorithm for Chinese word segmentation. Reference [17] presented two new textual feature selection methods based on Markov chains rank aggregation techniques. Reference [18] proposed a Markov chain model for radical descriptors in Arabic Text Mining. Reference [19] presented statistical Markov chain models for the distributions of words in text lines. Reference [20] proposed a method for handwritten Chinese/Japanese text (character string) recognition based on semi-Markov conditional random fields (semi-CRFs). Reference [21] presented a Markov chain method to find authorship attribution on relational data between function words. Reference [22] utilized a probabilistic Markov chain model to infer the location of Twitter users. Reference [23] proposed a Markov chain based technique to determine the number of clusters of a corpus of short-text documents. Reference [24] proposed a Markov chain based method for digital document authentication. Reference [25] used Markov chain for authorship attribution in Arabic poetry.

4.2. Hidden Markov models based research

Linguistic HMM based research has been for long an active research area due to the rapid development in NLP applications. The literature has many studies as follows. Reference [26] proposed to extract acronyms and their meaning from unstructured text as a stochastic process using HMM. Reference [27] proposed a morphological segmentation method with HMM method for Mongolian. Reference [28] employed HMM for Arabic handwritten word recognition based on HMM. Reference [29] presented a scheme for off-line recognition of large-set handwritten characters in the framework of the first-order HMMs. Reference [30] proposed the use of hybrid HMM/Artificial Neural Network (ANN) models for recognizing

unconstrained offline handwritten texts. Reference [31] used HMMs for recognizing Farsi handwritten words.

Reference [32] describes recent advances in HMM based OCR for machine-printed Arabic documents. Reference [33] proposed a HMM based method for named entity recognition. Reference [34] combined text classification and HMM techniques for structuring randomized clinical trial abstracts. Reference [35] employed HMM for medical text classification. Reference [36] propose text (sequences of pages) categorization architecture based on HMM. Reference [37] described a model for machine translation based on first-order HMM. Reference [38] introduced speech emotion recognition by use of HMM. Reference [39] presented a HMM based method for speech emotion recognition. Reference [40] discussed the role of HMM in speech recognition. Reference [41] indicated that almost all present day large vocabulary continuous speech recognition (LVCSR) systems based on HMMs. Reference [42] presented a text summarization method based on HMM. Reference [43] presented a method for summarizing speech documents using HMM. Reference [44] used HMM for part-of-speech tagging task. Reference [45] presented a second-order approximation of HMM for part-of-speech tagging task.

5. CONCLUSIONS

This work demonstrates the potential and the size of Markov chains research. The study reveals that the Markov chain and HMM is of high important for linguistic applications. Similarly, Markov chains are also widely used in many other applications. For future work, it worthy to explore the power of Markov chain in new linguistic and scientific directions with more details.

ACKNOWLEDGEMENTS

This work is supported by Kuwait Foundation of Advancement of Science (KFAS), Research Grant Number P11418EO01 and Kuwait University Research Administration Research Project Number EO06/12.

REFERENCES

[1] Rabiner, Lawrence R. "A tutorial on hidden Markov models and selected applications in speech recognition." Proceedings of the IEEE 77.2 (1989): 257-286.

[2] Markov_chain. (2016, August). Retrieved from https://en.wikipedia.org/wiki/Markov_chain

[3] Von Hilgers, Philipp, and Amy N. Langville. "The five greatest applications of Markov Chains." Proceedings of the Markov Anniversary Meeting, Boston Press, Boston, MA. 2006.

[4] Leon-Garcia, Alberto, and Alberto. Leon-Garcia. Probability, statistics, and random processes for electrical engineering. Upper Saddle River, NJ: Pearson/Prentice Hall, 2008.

[5] California Indian Education. (2016, August). Retrieved from http://www.californiaindianeducation.org/inspire/world/

[6] Collobert, Ronan, et al. "Natural language processing (almost) from scratch."Journal of Machine Learning Research 12.Aug (2011): 2493-2537.

[7] Bin, Tian, et al. "A Chinese word dividing algorithm based on statistical language models." Signal Processing, 1996., 3rd International Conference on. Vol. 1. IEEE, 1996.

[8] Leonardi, Riccardo, Pierangelo Migliorati, and Maria Prandini. "Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains." IEEE Transactions on Circuits and Systems for Video Technology 14.5 (2004): 634-643.

[9] Sanderson, Conrad, and Simon Guenter. "On authorship attribution via Markov chains and sequence kernels." 18th International Conference on Pattern Recognition (ICPR'06). Vol. 3. IEEE, 2006.

[10] Lim, Bresley Pin Cheong, et al. "Web search with text categorization using probabilistic framework of SVM." 2006 IEEE International Conference on Systems, Man and Cybernetics. Vol. 4. IEEE, 2006.

[11] Velivelli, Atulya, and Thomas S. Huang. "Automatic video annotation using multimodal Dirichlet process mixture model." Networking, Sensing and Control, 2008. ICNSC 2008. IEEE International Conference on. IEEE, 2008.

[12] Chen, Zhi-li, et al. "Effective linguistic steganography detection." Computer and Information Technology Workshops, 2008. CIT Workshops 2008. IEEE 8th International Conference on. IEEE, 2008.

[13] Dowman, Mike, et al. "A probabilistic model of meetings that combines words and discourse features." IEEE Transactions on Audio, Speech, and Language Processing 16.7 (2008): 1238-1248.

[14] Li, Lishuang, Zhuoye Ding, and Degen Huang. "Recognizing location names from Chinese texts based on max-margin markov network." Natural Language Processing and Knowledge Engineering, 2008. NLP-KE'08. International Conference on. IEEE, 2008.

[15] Meng, Peng, et al. "Linguistic steganography detection algorithm using statistical language model." Information Technology and Computer Science, 2009. ITCS 2009. International Conference on. Vol. 2. IEEE, 2009.

[16] Baomao, Pang, and Shi Haoshan. "Research on improved algorithm for Chinese word segmentation based on Markov chain." Information Assurance and Security, 2009. IAS'09. Fifth International Conference on. Vol. 1. IEEE, 2009.

[17] Wu, Ou, et al. "Rank aggregation based text feature selection." Web Intelligence and Intelligent Agent Technologies, 2009. WI-IAT'09. IEEE/WIC/ACM International Joint Conferences on. Vol. 1. IET, 2009.

[18] El Hassani, Ibtissam, Abdelaziz Kriouile, and Youssef BenGhabrit. "Measure of fuzzy presence of descriptors on Arabic Text Mining." 2012 Colloquium in Information Science and Technology. IEEE, 2012.

[19] Haji, Mehdi, et al. "Statistical Hypothesis Testing for Handwritten Word Segmentation Algorithms." Frontiers in Handwriting Recognition (ICFHR), 2012 International Conference on. IEEE, 2012.

[20] Zhou, Xiang-Dong, et al. "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields." IEEE transactions on pattern analysis and machine intelligence 35.10 (2013): 2413-2426.

[21] Segarra, Santiago, Mark Eisen, and Alejandro Ribeiro. "Authorship attribution using function words adjacency networks." 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2013.

[22] Rodrigues, Erica, et al. "Uncovering the location of Twitter users." Intelligent Systems (BRACIS), 2013 Brazilian Conference on. IEEE, 2013.

[23] Goyal, Anil, Mukesh K. Jadon, and Arun K. Pujari. "Spectral approach to find number of clusters of short-text documents." Computer Vision, Pattern Recognition, Image Processing and Graphics (NCVPRIPG), 2013 Fourth National Conference on. IEEE, 2013.

[24] Shen, Jau Ji, and Ken Tzu Liu. "A Novel Approach by Applying Image Authentication Technique on a Digital Document." Computer, Consumer and Control (IS3C), 2014 International Symposium on. IEEE, 2014.

[25] Ahmed, Al-Falahi, et al. "Authorship attribution in Arabic poetry." 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA). IEEE, 2015.

[26] Osiek, Bruno Adam, Geraldo Xexéo, and Luis Alfredo Vidal de Carvalho. "A language-independent acronym extraction from biomedical texts with hidden Markov models." IEEE Transactions on Biomedical Engineering 57.11 (2010): 2677-2688.

[27] He, Miantao, Miao Li, and Lei Chen. "Mongolian Morphological Segmentation with Hidden Markov Model." Asian Language Processing (IALP), 2012 International Conference on. IEEE, 2012.

[28] Alma'adeed, Somaya, Colin Higgens, and Dave Elliman. "Recognition of off-line handwritten Arabic words using hidden Markov model approach." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 3. IEEE, 2002.

[29] Park, Hee-Seon, and Seong-Whan Lee. "Off-line recognition of large-set handwritten characters with multiple hidden Markov models." Pattern Recognition 29.2 (1996): 231-244.

[30] Espana-Boquera, Salvador, et al. "Improving offline handwritten text recognition with hybrid HMM/ANN models." IEEE transactions on pattern analysis and machine intelligence 33.4 (2011): 767-779.

[31] Imani, Zahra, et al. "offline Handwritten Farsi cursive text recognition using Hidden Markov Models." Machine Vision and Image Processing (MVIP), 2013 8th Iranian Conference on. IEEE, 2013.

[32] Prasad, Rohit, et al. "Improvements in hidden Markov model based Arabic OCR." Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. IEEE, 2008.

[33] Zhou, GuoDong, and Jian Su. "Named entity recognition using an HMM-based chunk tagger." proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2002.

[34] Xu, Rong, et al. "Combining Text Classification and Hidden Markov Modeling Techniques for Structuring Randomized Clinical Trial Abstracts." AMIA. 2006.

[35] Yi, Kwan, and Jamshid Beheshti. "A hidden Markov model-based text classification of medical documents." Journal of Information Science (2008).

[36] Frasconi, Paolo, Giovanni Soda, and Alessandro Vullo. "Hidden markov models for text categorization in multi-page documents." Journal of Intelligent Information Systems 18.2-3 (2002): 195-217.

[37] Vogel, Stephan, Hermann Ney, and Christoph Tillmann. "HMM-based word alignment in statistical translation." Proceedings of the 16th conference on Computational linguistics-Volume 2. Association for Computational Linguistics, 1996.

[38] Schuller, Björn, Gerhard Rigoll, and Manfred Lang. "Hidden Markov model-based speech emotion recognition." Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on. Vol. 2. IEEE, 2003.

[39] Nwe, Tin Lay, Say Wei Foo, and Liyanage C. De Silva. "Speech emotion recognition using hidden Markov models." Speech communication 41.4 (2003): 603-623.

[40] Juang, Biing Hwang, and Laurence R. Rabiner. "Hidden Markov models for speech recognition." Technometrics 33.3 (1991): 251-272.

[41] Gales, Mark, and Steve Young. "The application of hidden Markov models in speech recognition." Foundations and trends in signal processing 1.3 (2008): 195-304.

[42] Conroy, John M., and Dianne P. O'leary. "Text summarization via hidden markov models." Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001.

[43] Maskey, Sameer, and Julia Hirschberg. "Summarizing speech without text using hidden markov models." Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers. Association for Computational Linguistics, 2006.

[44] Kupiec, Julian. "Robust part-of-speech tagging using a hidden Markov model." Computer Speech & Language 6.3 (1992): 225-242.

[45] Thede, Scott M., and Mary P. Harper. "A second-order hidden Markov model for part-of-speech tagging." Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics. Association for Computational Linguistics, 1999.

ICCSE2018 submission

E

EasyChair

Mon 04/12/2017, 20:22

Fawaz Alanzi

Inbox

You forwarded this message on 04/12/2017 20:26

Dear Fawaz Al-Anzi,

Fawaz Al-Anzi <Fawaz.alanzi@ku.edu.kw> submitted the following paper to ICCSE2018:

A Literature Survey of Arabic Speech Recognition

You are listed as one of the authors of this paper. To enter the ICCSE2018 Web pages you should visit

https://easychair.org/conferences/?conf=iccse2018

and enter your EasyChair user name and password.

If you forgot your user name or password, please visit

https://easychair.org/account/forgot.cgi

and specify Fawaz.alanzi@ku.edu.kw as your email address.

Best regards,
EasyChair Messenger.

A Literature Survey of Arabic Speech Recognition

Fawaz S. Al-Anzi
Department of Computer Engineering
Kuwait University
Kuwait
fawaz.alanzi@ku.edu.kw

Dia AbuZeina
Department of Computer Engineering
Kuwait University
Kuwait
abuzeina@ku.edu.kw

ABSTRACT
Speech recognition poses some interesting challenges such as varying acoustic conditions, dialects, and articulation at word's boundaries. Large vocabulary speaker-independent continuous speech recognition systems have recently received significant attention. In this paper, we present a survey of Arabic speech recognition that even has more challenges such as the optional diacritization of the Arabic script. Even though Arabic is a live language that is spreading widely throughout a large area, the research devoted to this technology still in the early stages compared to other languages such as English language. In this study, we highlight the progress made so far in Arabic speech recognition field that include corpora, phonemes, language models, acoustic models, and some promising research directions. This survey reveals that the shortage of freely available continuous speech corpora deserves more research attention in this domain. It also shows a need to compile large corpora or a benchmark as it will be a key factor to promote the Arabic language research for effective human-computer interaction.

Keywords
Arabic language; speech recognition; corpus; phoneme; language model

1. INTRODUCTION
Having human speech interpreted by a computer is called Automatic Speech Recognition (ASR). It is defined as the process of converting spoken language (sound waves) into a machine-readable text. With the fast growth of powerful communication devices, it is making man-machine interfaces even more valuable and pervasive. Developing commercial speech recognizers have been shown to be successful business interactive solutions in various industry sectors such as healthcare, telecommunication, banking and finance, retail and mall management, education, hospitality, governmental institutions, and travel,[1]. In the last decade, there has been great enthusiasm by developers to have this attractive property that can be of great advantage in the new technologies such as search engines, voice maps, communications, etc. Recently, speech was used in security and protection fields for authentication purposes (also called voiceprints). Findbiometrics list five unique applications of voiceprints; targeting the developers, hands free interface, call center authentication, proof of life, and multi-factor logical access control [2]. Nuance lists some of benefits of using voiceprints that include simpler authentication, wipe out fraud, and almost-instant return on investment [3]. Even though the utilizing of voiceprint in banking industry is not new, this technology has been transferred to Arab countries. For examples, Kuwait Finance House (KFH) uses a speech recognition platform for users' authentications. The service that was initially only available to VIP customers is now in the process of being rolled out to KFH's entire customer base, [4]. Abu Dhabi Commercial Bank (ADCB) has also turned to voiceprints, [5]. Fortunately, the communications infrastructure that already exist help to spread the voiceprint biometric rather than installing a new machine such as image scanner to read the fingerprint before sending the figure print for authentication.

In fact, globally utilizing speech recognition in natural language processing (NLP) and linguistic applications has pushed to utilize this technology for the Arabic language that has more than 380 million speakers [6]. Most importantly, the holy Quran that was revealed in Arabic has to be read in Arabic by the entire Muslim world. This constraint might reinforce Arabic speech research, and therefore the technology to server the holy Quran readers and learners. Reference [7] presented a speech recognition technique for verification of Quranic recitation of sound files and media. Reference [8] provided a structural overview of speech recognition system for developing Quranic verse recitation recognition with Tajweed checking rules function.

However, speech recognition is not an easy task and there is a long way for efficiently utilizing speech recognition to fulfil people requirements. Despite the successive research attempts, the high accurate transcription of human natural spoken words (speech-to-text) is still a difficult task problem. In fact, speech processing is much complicated than other pattern recognition problems such as text or images classification. For illustration, while locating a particular text or an image has achieved great success, locating a particular speech segment of a particular word in a speech collection audio file is still an active research problem.

Speech recognition is classified as a multidiscipline field that includes machine learning, phonetics, linguistics, and signal processing. Accordingly, significant integration is required for satisfied performance. Unfortunately, while the remarkable evolution in the research toward enhanced ASR; Arabic research is still behind when compared to other languages such as English. In general, varying acoustic conditions, pronunciation variations, dialects, accent, age and others factors are all degrade the performance. Reference [9] presented various factors that affect the way in which words are pronounced such as assimilation, co-articulation, reduction, deletion, and insertion. The Arabic language has even more challenges such as morphological complexity and diacritization wherein short vowels are usually missed in formal writings.

In addition to the Arabic intrinsic challenges, there are some other logistic researching challenges such as resources availability. The absence of unified large continuous speech corpora is an obstacle that might restrain the research in this flourishing domain. It has been noticed that almost all-Arabic speech recognition studies have been investigated using in-house small corpora. This is unlike English language that has many common large corpora such as North America business (NAB) and Broadcast News switchboard, [10]. Working on common corpora saves time as well as gradually enhance the research since the outputs can be compared and improved. It is known in speech recognition

community that creating a large speech corpus is time demand and extremely expensive task. Consequently, it is hard and might be inconvenient for individuals to perform such task. Reference [11] indicated that preparing large training corpora for dialectal Arabic acoustic modeling is too difficult compared to Modern Standard Arabic (MSA).

Even though-isolated words speech recognition is an important task for some digits and commands applications, the continuous or conversational speech recognition has had even more interest. For several years, there have been two well-known ASR engines that are used for speech-to-text task; Carnegie Mellon University (CMU) Sphinx and the Cambridge University Hidden Markov Model Toolkit (HTK). Both are statistical based engines that are based on Hidden Markov models (HMMs). Developing a speech recognition engine is a complex task and requires highly expert staff; therefore, most researchers used the free (black box) Sphinx and/or HTK engines. However, some other techniques have been used in the literature such as artificial neural networks (ANNs) and support vector machines (SVM). Sphinx engine does support Arabic, but HTK does not support Arabic and conducting speech recognition research requires a transliteration process on the training text that can be performed, as an example, using the transliteration system available at [12]. Figure 1 shows the architecture of an ASR that include three knowledge databases: the acoustic models contains the trained HMMs, the language mode represents the statistical words co-occurrences, and the dictionary (also called pronunciation dictionary or vocabulary) has the pronunciation of each words in terms of phonemes, the basic unit of sounds.

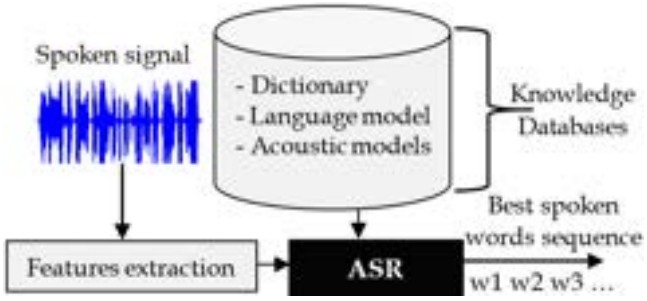


Figure 1. Automatic Speech Recognition (ASR) Architecture

The goal of this paper is to present the recent advances in Arabic ASR with a particular highlight of the essential components of a typical ASR. The topics include the corpora used for both isolated-words and continuous speech. The corpora information include vocabulary size, nature of data, topics, speakers, number of male and female, age, etc. features extraction methods, classification approaches, phonemes sets, pronunciation dictionary, language models (LMs). The paper also includes some new directions that could be investigated for Arabic.

This paper is organized as follows. The next section presents the Arabic speech corpora of both isolated-words and continuous speech. Section 3 presents the phonemes set and pronunciation dictionary, followed by language models in section 4. The performance evaluation is demonstrated in section 5, and the conclusion and future works are presented in section 6.

2. ARABIC SPEECH CORPORA

The preliminary work in speech recognition requires in the first place to specify the type of speech; either isolated-words or continuous speech. Therefore, the speech corpora will be the first topic to be discussed in this survey. In isolated-words speech (also called discrete words), a pause are existed between digits or words while such constraint is not existed in continuous or conversational speech. Isolated-words speech recognition is characterized by easy to implement when compared to the continuous speech recognition that it suffers from co-articulation phenomenon, the critical factor for recognition results and performance.

2.1 Isolated-words speech recognition

In this subsection, we present the isolated-words corpora, the speech features, and the classifiers used. In general, isolated-words size is represented using the number of recorded speech files while continuous speech file is represented using the number of hours (all recorded speech files length). In the following, MFCC is the shorthand for Mel-Frequency Cepstral Coefficients, the widely used features for speech recognition; LPCC is the shorthand of Linear Predictive Cepstral Coefficients. The research contributions presented in Table 1 are date ordered, the earliest first. Otherwise indicated, all the researches listed in Table 1 are categorized as isolated-words speech recognition, MFCC features, and HMM-based classifiers.

Table 1. Isolated-words corpus information

Ref.	Year	Corpus Information, Features, Classifier/s
[13]	2001	The training set is composed by 50 speakers each of them uttered three times the ten digits. The test set comprises two groups, 30 speakers and 10 speakers. LPCC features were used.
[14]	2002	They used fuzzy NNs for recognition of isolated words. Cannot have access to the paper to present corpus information.
[15]	2003	Speech corpus consists of the 10 isolated digits, with 20 repetitions for each digit, using single male speaker. LPCC and MFCC features were used.
[16]	2004	The corpus contains of a total 1800 digits pronounced by 60 speakers (30 males and 30 females). For testing, they used 1000 digits pronounced by 50 others speakers (25 males and 25 females) .NNs classifier was used.
[17]	2006	The corpus consists of 92 speakers. (46 male and 46 female) pronounce each word two times where 20/92 of the corpus used for learning. (HMM,SVM) classifiers were used.
[18]	2007	Training set: 300 token (10 digits * 5 repetitions * 6 Moroccan speakers). Testing set: 30 token of different individuals.
[19]	2007	128 words for training and other 7 words for testing. Dynamic time warping (DTW) was used as similarity measure for classification.
[20]	2008	The corpus consists of 600 utterances (10 speaker, 10 words, 6 repetitions) split into 300 utterances for training and 300 utterances for testing. NNs was used for recognition.
[21]	2008	The corpus contains about 1.5 hours of commands and less than 1 hour of digits.
[22]	2008	The corpus contains Egyptian 59 men, (33 speakers for and 26 for testing). Speakers asked to utter 16 sentences of proverbs.
[23]	2008 NNs	The training set contains 340 tokens (17 speakers × 2 repetitions × 10 digits). For testing 1,700 tokens (17 speakers × 10 repetitions × 10 digits) were used. NNs was used for classification.

[24]	2009	The corpus was created from all 10 Arabic digits. A number of 60 Moroccan speakers (35 males and 25 females) were asked to utter all digits 5 times.
[25]	2010	The corpus contains 3650 speech files recorded by 13 speakers. Training set contains 3000 speech files and 650 for testing.

2.2 Continuous speech recognition

The research contribution toward continuous Arabic speech recognition is less than what we have seen in isolated-words that make sense as previously indicated regarding the difficulty of preparing a continuous speech corpus. However, there is considerable work initiated by the Linguistic Data Consortium (LDC). The LDC is an open consortium of universities, libraries, corporations and government research laboratories, [26]. More information about LDC Arabic speech corpora can be found at LDC catalog [27] that contains hundreds of (not free) holdings. One of important Arabic speech contribution is the work fielded by IBM in the Gale project that used LDC corpora. Gale project has many phases that gradually improve the performance. The Gale acoustic training set composed of approximately 1800 hours of transcribed Arabic broadcasts provided by LDC. The published work the described the phases include: The IBM 2006 Gale Arabic ASR System [28], The IBM 2009 GALE Arabic speech transcription system [29], and the IBM 2011 GALE Arabic speech transcription system [30]. LDC produced CallHome (CH) corpus of Egyptian Colloquial Arabic (ECA) [31]. This corpus is a collection of informal phone conversations between close friends or family members. This corpus has many sets as shown in Table 2, [32].

Table 2. LDC ECA CallHome datasets

collection	Number of conversations	Number of words	Number of hours
train	80	146,298	14
dev	20	32,148	3.5
eval96	20	15,584	1.5
eval97	20	17,670	1.8
h5_new	20	16,752	1.8
eval03	10	11,015	1.9

In speech recognition systems, it is highly recommended and even more accurate to use large speech collections. Reference [33] explains the meanings of large vocabulary speaker-independent continuous speech recognition as follows. Large vocabulary means that the corpus has vocabulary (unique words) of about 20,000 to 60,000 words. Speaker-independent is the classifier's ability to recognize the speech of people whose speech has never been exposed before. The continuous mean that the speech is recorded according to the human natural language.

One of the earliest attempts to develop a speech corpus was made by reference [34] who describes the OrienTel speech dataset. They indicated that the OrienTel is the first time makes an effort to create speech data on a large scale. The participants of OrienTel collected standard and colloquial varieties of Arabic in Saudi Arabia, the UAE, Egypt, Palestine, Tunisia and Morocco. GlobalPhone project produced a read speech corpus that was designed for the development and evaluation of large continuous speech recognition systems, [35]. Reference [36] presented a Saudi accented Arabic telephone speech database. It contains 96 hours that was collected on a telephone network during 2002 and 2003 using 1033 native speakers (51% males, 49% females). Reference [32] used CallHome corpus for morphology-based LMs at different stages of Arabic ASR. New versions were used by authors as Reference [37] used a new LDC test set called Dev07 that was distributed by LDC in March 2007 and consisting of 2.5 hours of speech (18186 words). The Reference [37] research work consists of using neural network LMs for Arabic ASR. Reference [38] developed an MSA broadcast news speech recognition system. The system was trained on 7.0 hours of a 7.5 hours and tested on the remaining half an hour. The corpus contains a total of 235 news items, 41 news items cover sport news and the rest of the items cover mainly economic news. Among the speakers, 88 of the news items were by female speakers. Reference [21] presented a Holly Qura'an corpus that contains about 18.5 hours. However, the actual challenge is developing a broadcast news corpus since the holy Quran recordings are already available. However, Reference [21] indicated that it takes about 732 working hours to build their holy Quran corpus. Reference [11] used MSA acoustic models as multilingual models to decode Egyptian dialect. They chose the Nemlar broadcast news speech corpus to build the acoustic models. The corpus consists of 40 hours of MSA news broadcast. The total number of speakers is 259 with a lexicon of 62,000 words. Reference [39] presented a MSA continuous speech corpus composed of 200 sentences pronounced by 300 Algerian native speakers selected from eleven regions of Algeria. Reference [40] developed an Arabic ASR system based on phonetically rich and balanced speech corpus. That work was based on 8,043 utterances gathered from 8 (5 male and 3 female) speakers resulting about 8 hours of speech. The round robin testing approach was applied.

The speech dataset is used to train the acoustic models that are the statistical representations of the MFCC speech features vectors. Hence, the acoustic models represent the statistical co-occurrences between phonemes. HMM is one of the most common type to represent acoustic models as the example represented Figure 2. In the figure, each number represents a phoneme and the all three phonemes represent what is called triphone that represents a phoneme surrounded by specific left and right phonemes. There are even more details as each phoneme is internally represented using three states (beginning, middle, end) using mixture density Gaussian distributions. Hence, Figure 2 shows how HMMs used to represent acoustic models.

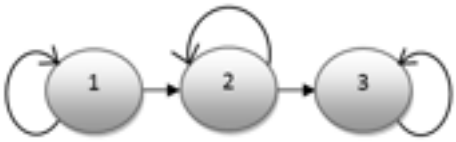


Figure 2. HMM-based Triphone

The procedures that can be applied to produce a continuous speech corpus are summarized as follows. A number of audio files are recorded (under equal conditions) from radio and/or TV broadcast news. It is highly recommended to cooperate with local or international stations of radio and TVs to get some prerecorded speech collections (e.g. a part of the archive). This cooperation avoids the labor and cost intensive manual generation to produce speech corpora. If the lengths of audio files are too long, these audio files should be split into small audio files of 10-30 seconds. In fact, there is no problem to have more than 30 seconds, but the speech recognizer might fail during the training process to align the recordings with their phonetic transcription. Hence, it will be

productive and efficient if the recordings are short. However, if the recordings are long say 10 minutes, initial alignment might be fail and therefore causing a problem during training process. During recording to create a corpus, the following parameters are set as indicated by Sphinx [41]: Sampling Rate = 16 kHz (or 8 kHz, depending on the training data), No. of Bits = 16 bits, No. of Channels = Mono (= single channel), and File Format = ".wav". Once have the recordings completed, the spoken words in the audio files are transcribed and then diacritized. Based on the diacritized text, the pronunciation dictionary is produced. After this step two models can be created, the acoustic model and the language model. The three knowledge bases (acoustic models, pronunciation dictionary, and language model) will be ready for setting up and testing a continuous speech recognition system.

3. ARABIC PHONEMES SET

Speech recognition task is generally performed using one of three approaches based on the basic units of classification. The units include words, syllables, and phonemes. Word-based recognition has a drawback that it needs large number of data for training. A syllable that is a single unit of written or spoken word has relatively smaller number of used units and runs faster than word-based recognition. However, the recognizer which depends on the distinctive unit of sound (i.e., phoneme) is a wide spread approach since it is easy to train, [22]. Either making a research using isolated words or continuous words speech, phonemes set has to be defined before training stage, of course if the research is based on phoneme-based method. Language linguistics define the phoneme set of a particular language after studying and careful classification of speech sounds. In this section, we demonstrate the phonemes-based research that found in the literature. Many studies indicated that Arabic language has 34 phonemes (28 consonants and 6 vowels) such as References [20], [25], and [38]. However, the number of Arabic phonemes is a debated by researchers. For example, Reference [42] used a phoneme set that contains 46 phonemes for developing a tool that is used to create a pronunciation dictionary. The tool generated by Reference [42] had later used in some researches such as Reference [38] and Reference [43]. Reference [19] indicated that Arabic has at least one hundred twelve phonemes as they considered that every letter has four diacritics, therefore, four phonemes. Reference [23] used 37 MSA phoneme as given by Language Data Consortium (LDC). Reference [11] indicated that MSA consists of 38 phonemes, 28 are original consonants, 4 are foreign and rare consonants and 6 are vowels. An example of pronunciation dictionary's entries as produced by Reference [42] is shown in Figure 3. It shows some words and the phonetic transcription (the phonemes) of each word.

```
…
يُوَائِم Y UH W AE: E IH M
يُوَاجِه Y UH W AE: JH IH H
يُوَازِي Y UH W AE: Z IY
يُوَاصِلُ Y UH W AE: SS IH L UH
يُوَاكِب Y UH W AE: K IH B
يُوَقِّعُ Y UH W AE Q IX AI UH
يُوَقِّعُهُ Y UH W AE Q IX AI UH H UH
…
```

Figure 3. Some entries of a pronunciation dictionary

4. LANGUAGE MODELS

Language models is a statistical component of automatic speech recognition systems that is used to estimates the most likely co-occurring sequence of words (possible words) in the language. In combination with acoustic models, language models are used to recognize the spoken words given the sequence of speech signal features vectors. Reference [15] indicated that the language model can be incorporated to constrain the recognizer to recognize only valid word sequences. Reference [25] demonstrated that one benefit of the language model is to reduce the search recognition probability by forcing the recognition to follow certain rules to ensure a better accuracy in the recognition output. Reference [23] indicated that the absence of diacritics in Arabic text decreases predictability in the language model. In general, much larger amounts of text leads to develop more powerful and enhance the goodness of the language models. In speech recognition, N-grams language model is used to indicate for a contiguous sequence of n items that are also called unigram, bigrams, and trigrams of the language text. In case of isolated words speech recognition, context free grammar is used. While language models are mainly used in large continuous speech recognition systems, Context-Free Grammar is also used in speech recognition to to predict subsequent words in small corpora or isolated words speech recognition. As language models Context-Free Grammar is used to reduce the possible words to be considered as a next word. While statistical language models generally describe complex language, Reference [44] indicated that grammars describe very simple type of the language for command and control, and they are usually written by hand. Therefore, grammars usually do not have probabilities for word sequences, but some elements might be weighed. The perplexity is the common way to evaluate N-gram language model, it is an indication of the average number of words that can follow a given word. Reference [39] used a bigram language model for continuous speech recognition of the Arabic language. Reference [38] used both bi-grams and trigrams for the language model. Reference [40] used different language models (bigram and context free grammar) for continuous Arabic speech recognition system. Reference [21] used the tool available in SPHINX-IV to generate the N-grams language model. Reference [32] used n-gram models up to an order of n = 6 for improving the perplexity of the language models. Reference [37] indicated that using of neural network language models for Arabic broadcast news and broadcast conversations outperforms the 4-Grams based language model. Reference [25] used an Arabic grammar file that contains some words and commands to be represented in their isolated words speech system. Reference [45] investigated the use of morphology-based language model at different stages in a speech recognition system for conversational Arabic. Reference [18] used the tool available in CMUSphinx to specify the grammars of spoken Arabic digits recognition system. Figure 4 shows an example of the entries in a language model generated using CMU statistical language tool [44] of the corpus produced by Reference [38]. The number besides the n-grams in Figure 4 is related to the probabilities of occurring for each case.

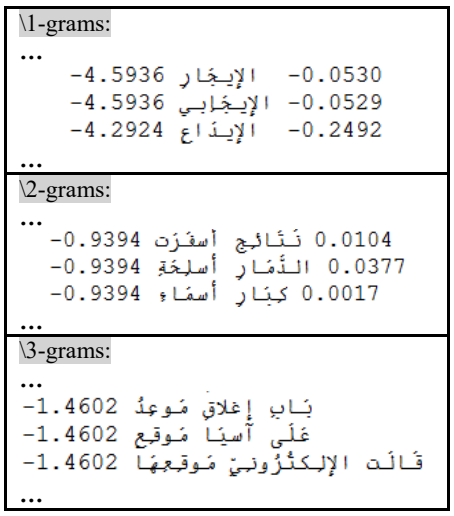


Figure 4. Some entries of a language model

5. PERFORMANCE EVALUATION

The isolated-words speech recognition is generally measured using recognition accuracy rate that is the percentage of correctly recognized patterns such as words or digits. However, in continuous speech, word error rate (WER) is the common metric to measure performance of ASRs. WER is computed using the following formula: WER=(S+D+I)/N, Where: S is the number of substitutions words errors, D is the number of the deletions words errors, I is the number of the insertions words errors, N is the number of words in the testing set. The word accuracy can be measured using WER formula: Word Accuracy = 1 – WER.

Reference [21] said that the WER of MSA is in the range: 15–20%. However, WER depends on some other parameters such as the size of training corpus, the number of words in the dictionary, and the perplexity of the language model. We emphasize that measuring performance is required to show if the achieved WER is statistically significant. It has been noticed that the majority of Arabic speech recognition authors do not use the appropriate statistical significant test to prove their performance. Therefore, a baseline system should be developed and tested, and then the output of the proposed method is compared with the baseline results to find the statistically significance enhancement. If the result does not have statistically significant enhancement, then the enhancement is considered accidental and the proposed method is not as strong method to enhance the classification performance. Reference [46] has a description how to perform a statistical significant test.

In this section, we present some of performance evaluations of continuous speech since it has more interest and challenge than isolated-words speech recognition. Reference [32] performed performance evaluation for morphology based LMs. The WER improved by 1.8% and 1.5% for two different test sets. Reference [37] achieved WER improvements by 0.8% and 3.8% for 2 different configurations of neural probabilistic models. Reference [38] achieved a WER of 13.66% of broadcast news corpus. Reference [21] achieved a WER of 46.182% using the holy Quran corpus. Reference [11] reported a recognition accuracy of 99.34% for Egyptian Colloquial Arabic. Reference [39] presented an accuracy rate of 91.65 % for MSA continuous speech corpus. Reference [40] demonstrated a WER of 11.27% and 10.07% with and without diacritical marks respectively for MSA continuous speech corpus. Table 3 shows some of WER for some systems on different English speech corpora, [33].

Table 3. WERs for a number of ASRs (English corpora)

Pronunciation Corpus	Vocabulary	WER %
TI Digits	11 zero-nine, oh	0.5
Wall Street Journal read speech	5,000	3
Wall Street Journal read speech	20,000	3
Broadcast News	64000+	10
Conversational Telephone Speech (CST)	64000+	20

6. CONCLUSION AND FUTURE WORKS

The review of Arabic speech recognition shows that the research is still in the raw stage specially the continuous speech type. Most of the research attempts belong to isolated-words speech recognition. However, there are some research activities towards continuous speech recognition. The major obstacle is the corpora availability. Hence, reinforcing Arabic speech recognition needs affording the professionality producing of large Arabic continuous speech collections (corpora).

The following are some research topics that can be investigated for Arabic. Even there are some studies regarding pronunciation variations such as Reference [43], more studies are needed to tackle this phenomenon. The researches could also investigate using long distance words relationships in the language models. The traditional N-grams language models assume that a word is only influenced by a few proceeding words, typically one or two. However, it is much better to account for longer-distance constraints. Data mining association rules algorithms such as Apriori could help to find some words association rules. Then, N-best ASR hypotheses can be used in combination with words association rules for rescoring the ASR outputs for better accuracy. The semantic relationships can also be used to enhance ASR performance. The semantic and syntactic relationships can be obtained using knowledge bases such as WordNet [47]. The syntactic relationships can be obtained in combination of part of speech tagging and data mining algorithms. The deep neural network hidden Markov model (DNN-HMM) hybrid architecture is another direction that can be employed for Arabic [48]. Further research is needed to clarify the realistic Arabic phoneme set. Reference [49] had some preliminary work in this direction. However, an approach is needed for automatic extracting the Arabic phonemes using data-driven approaches and clustering methods. There has been current research to train a recognizer using synthesized data, because if it is possible then we could get as much data as we want in a preferred domain or with a new and large vocabulary,[38].

ACKNOWLEDGMENT

This work is supported by Kuwait University Research Administration Research Project Number EO06/12.

REFERENCES

[1] Emerging Technologies: 2017. *http://www.em-t.com/*. Accessed: 2017- 12- 4.

[2] Voice Month: 5 Unique Applications of Voice Biometrics - FindBiometrics: 2017. *http://findbiometrics.com/voice-month-5-unique-applications-of-voice-biometrics-22186/*. Accessed: 2017- 12- 4.

[3] Nuance | Nuance - PDF, Customer Service, HIM, and Speech Recognition Solutions: 2017. *http://www.nuance.com/*. Accessed: 2017- 12- 4.

[4] Speech Recognition Case Study Kuwait Finance House (KFH): 2017. *http://*www.*em-t.com/content/speech-recognition-case-study-kuwait-finance-house-kfh*. Accessed: 2017- 12- 4.

[5] ADCB: 2017. *http://www.adcb.com/*. Accessed: 2017- 12- 4.

[6] Mubarak, Hamdy, and Kareem Darwish. 2014.Using Twitter to collect a multi-dialectal corpus of Arabic. *ANLP* (2014): 1.

[7] Mohammed, A., Sunar, M. and Hj Salam, M. 2015. Quranic Verses Verification using Speech Recognition Techniques. *Jurnal Teknologi*. 73, 2 (2015).

[8] Jamaliah Ibrahim, N., Yamani Idna Idris, M., Razak, Z. and Naemah Abdul Rahman, N. 2013. Automated tajweed checking rules engine for Quranic learning. *Multicultural Education & Technology Journal*. 7, 4 (2013), 275-287.

[9] Strik, H. and Cucchiarini, C. 1999. Modeling pronunciation variation for ASR: A survey of the literature. *Speech Communication*. 29, 2-4 (1999), 225-246.

[10] Rabiner, L. and Schafer, R. 2007. Introduction to Digital Speech Processing. *FNT in Signal Processing*. 1, 1â€“2 (2007), 1-194.

[11] Elmahdy, Mohamed, et al. 2009. Modern standard Arabic based multilingual approach for dialectal Arabic speech recognition. *Natural Language Processing*. *SNLP'09. Eighth International Symposium* on. IEEE, 2009.

[12] Arabic Transliteration/Encoding Chart: 2017. *http://languagelog.ldc.upenn.edu/myl/ldc/morph/buckwalter.html*. Accessed: 2017- 12- 4.

[13] Bahi H, Sellami M. 2001. Combination of vector quantization and hidden Markov models for Arabic speech recognition. *ACS/IEEE international conference on computer systems and applications*, 2001

[14] Alimi AM, Ben Jemaa M. 2002. Beta fuzzy neural network application in recognition of spoken isolated Arabic words. *Int J Contr Intell Syst* 30(2), Special issue on speech processing techniques and applications

[15] Elmisery FA, Khalil AH et al.2003. A FPGA-based HMM for a discrete Arabic speech recognition system. In: Proceedings of the *15th international conference on microelectronics, 2003. ICM 2003*

[16] Amrouche, Abderrahmane, and Jean Michel Rouvaen. 2003. Arabic isolated word recognition using general regression neural network. *Circuits and Systems, 2003 IEEE 46th Midwest Symposium* on. Vol. 2. IEEE, 2003

[17] Bourouba H, Djemili R et al. 2006. New hybrid system (supervised classifier/HMM) for isolated Arabic speech recognition. *2nd Information and Communication Technologies*, 2006. ICTTA'06

[18] Satori H, Harti M, Chenfour N.2007. Introduction to Arabic speech recognition using CMU Sphinx system. *Information and communication technologies international symposium proceeding ICTIS07*, 2007

[19] Haraty, R. and El Ariss, O. 2007. CASRA+: A Colloquial Arabic Speech Recognition Application. *American Journal of Applied Sciences*. 4, 1 (2007), 23-32.

[20] Essa EM, Tolba AS et al. 2008. A comparison of combined classifier architectures for Arabic speech recognition. *International conference on computer engineering and systems*, 2008. ICCES 2008

[21] Hyassat, H. and Abu Zitar, R. 2006. Arabic speech recognition using SPHINX engine. *Int J Speech Technol*. 9, 3-4 (2006), 133-150.

[22] Azmi M, Tolba H,Mahdy S, FashalM.2008. Syllable-based automatic Arabic speech recognition in noisy-telephone channel. In: *WSEAS transactions on signal processing proceedings, World Scientific and Engineering Academy and Society (WSEAS),* vol 4, issue 4, pp 211–220

[23] Alotaibi, Y. 2008. Comparative Study of ANN and HMM to Arabic Digits Recognition Systems. *eng*. 19, 1 (2008), 43-60.

[24] Satori, Hassan, et al. 2009. Investigation Arabic speech recognition using CMU sphinx system. *Int. Arab J. Inf. Technol*. 6.2 (2009): 186-190.

[25] Al-Qatab, Bassam AQ, and Raja N. Ainon. 2010. Arabic speech recognition using hidden Markov model toolkit (HTK). *Information Technology (ITSim), 2010 International Symposium* in. Vol. 2. IEEE, 2010.

[26] About LDC | Linguistic Data Consortium: 2017. *https://www.ldc.upenn.edu/about*. Accessed: 2017- 12- 4.

[27] Linguistic Data Consortium - Linguistic Data Consortium: 2017. *https://catalog.ldc.upenn.edu/*. Accessed: 2017- 12- 4.

[28] Soltau, Hagen, et al. 2007. The IBM 2006 Gale Arabic ASR system. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007*. *IEEE International Conference* on. Vol. 4. IEEE, 2007.

[29] Kingsbury, Brian, et al.2011. The IBM 2009 GALE Arabic speech transcription system. *Acoustics, Speech and Signal Processing (ICASSP)*, 2011 IEEE International Conference on. IEEE, 2011.

[30] Mangu, Lidia, et al. 2011. The IBM 2011 GALE Arabic speech transcription system. *Automatic Speech Recognition and Understanding (ASRU)*, 2011 IEEE Workshop on. IEEE, 2011.

[31] CALLHOME Egyptian Arabic Speech - Linguistic Data Consortium: 2017. *https://catalog.ldc.upenn.edu/LDC97S45*. Accessed: 2017- 12- 4.

[32] Kirchhoff, K., Vergyri, D., Bilmes, J., Duh, K. and Stolcke, A. 2006. Morphology-based language modeling for conversational Arabic speech recognition. *Computer Speech & Language*. 20, 4 (2006), 589-608.

[33] Jurafsky, D. and Martin, J. 2000. Speech and language processing. Prentice Hall.

[34] Siemund, Rainer, et al. 2002. OrienTel—Arabic speech resources for the IT market. *LREC 2002 Arabic Workshop*. 2002.

[35] Schultz, Tanja. 2002. Globalphone: a multilingual speech and text database developed at karlsruhe university. *INTERSPEECH*. 2002.

[36] Alghamdi, M., Alhargan, F., Alkanhal, M., Alkhairy, A., Eldesouki, M. and Alenazi, A. 2008. Saudi Accented Arabic Voice Bank. *Journal of King Saud University - Computer and Information Sciences*. 20, (2008), 45-64.

[37] Emami, Ahmad, and Lidia Mangu. 2007. Empirical study of neural network language models for Arabic speech recognition. *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop* on. IEEE, 2007.

[38] Alghamdi, M., Elshafei, M. and Al-Muhtaseb, H. 2007. Arabic broadcast news transcription system. *Int J Speech Technol*. 10, 4 (2007), 183-195.

[39] Selouani, Sid Ahmed, and Malika Boudraa. 2010. Algerian Arabic speech database (ALGASD): corpus design and automatic speech recognition application. *Arabian Journal for Science and Engineering* 35.2C (2010): 158.

[40] Abushariah, Mohammad Abd-Alrahman Mahmoud, et al. 2012. Arabic speaker-independent continuous automatic speech recognition based on a phonetically rich and balanced speech corpus. *International Arab Journal of Information Technology (IAJIT)* 9.1 (2012): 84-93.

[41] Frequenty Asked Questions (FAQ) [CMUSphinx Wiki]: 2017. *http://cmusphinx.sourceforge.net/wiki/faq*. Accessed: 2017- 12- 4.

[42] Ali, M., Elshafei, M., Al-Ghamdi, M. and Al-Muhtaseb, H. 2009. Arabic Phonetic Dictionaries for Speech Recognition. *Journal of Information Technology Research*. 2, 4 (2009), 67-80.

[43] AbuZeina, D., Al-Khatib, W., Elshafei, M. and Al-Muhtaseb, H. 2011. Cross-word Arabic pronunciation variation modeling for speech recognition. *Int J Speech Technol*. 14, 3 (2011), 227-236.

[44] Building Language Model [CMUSphinx Wiki]: 2017. *http://cmusphinx.sourceforge.net/wiki/tutoriallm*. Accessed: 2017- 12- 4.

[45] Vergyri D., Kirchhoff K., Duh K., and Stolcke A. 2004. Morphology Based Language Modeling for Arabic Speech Recognition. *in Proceedings of Interspeech, Germany*, pp. 2245-2248, 2004.

[46] Plötz T. 2005. Advanced stochastic protein sequence analysis, Ph.D. thesis, *Bielefeld University*

[47] Ruiz-Casado, M., Alfonseca, E. and Castells, P. 2007. Automatising the learning of lexical patterns: An application to the enrichment of WordNet by extracting semantic relationships from Wikipedia. *Data & Knowledge Engineering*. 61, 3 (2007), 484-499.

[48] Dahl, G., Dong Yu, Li Deng, and Acero, A. 2012. Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition. *IEEE Transactions on Audio, Speech, and Language Processing*. 20, 1 (2012), 30-42.

[49] Nahar, Khalid MO, et al. "Data-driven Arabic phoneme recognition using varying number of HMM states." Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on. IEEE, 2013.

[50] Prof Allan Ramsay (BSc, MSc, PhD) research profile - research | The University of Manchester: 2017. *http://www.manchester.ac.uk/research/Allan.ramsay/research*. Accessed: 2017- 12- 4.

ACKNOWLEDGEMENT

I hereby acknowledge the support of Kuwait University Research Sector in granting the Project and facilitating the research implementation.

And I also agree to the best of my knowledge that the information herein are true and complete.

Principal Investigator (Applicant)

Name: FAWAZ S. AL-ANZI

Project Code: EO06/12

Faculty: FACULTY OF ENGINEERING & PETROLEUM

Department: COMPUTER ENGINEERING

Date: 16/01/2018

KUWAIT UNIVERSITY
RESEARCH SECTOR

Name: Fawaz S. Al-Anzi

Date: 16/01/2018 10:03:46 a.m.