Evaluating Factors Affecting Sentences Similarity and Paraphrasing Identification Using K-Means Clustering

Marwah ALIAN
Hashemite University, Princess Sumaya University for Technology, Jordan, marwah2001@yahoo.com

Arafat AWAJAN
Princess Sumaya University for Technology, Amman, Jordan, awajan@psut.edu.jo

Abstract

This research considers Arabic Paraphrasing Benchmark for identifying similar and paraphrased sentences using k-means clustering. The benchmark is constructed based on Arabic transformation rules for sentences and provided by labels for similar/not similar and paraphrased/not paraphrased sentences. K-means clustering is applied to partition the dataset into clusters with similar sentence pairs. Three factors that affect the distribution of similar and paraphrased sentences are tested by conducting several experiments with K-means clustering. By analyzing the resulted clusters, paraphrased sentences achieve a recall of 0.81 with the pre-trained embeddings and a recall of 0.78 with introducing words' weight while labeling by majority provides better recall than labeling with a threshold of similarity score.

Keywords

paraphrasing, sentence similarity, K-means, clustering.

1. Introduction

Paraphrasing identification refers to reframing a sentence using different words and structures to produce a new sentence with the same meaning (Fernando, 2008). Semantic similarity has an important role in detecting paraphrases, as it allows the process to compute the similarity score between texts in order to identify whether they are similar in meaning or not (Srivastava and Govilkar, 2017) (Alian and Awajan, 2018). The approaches to measure semantic similarity between sentences can be categorized into four groups, namely, co-occurrence–based approach, statistical corpus-based approach, feature-based approach, and word embedding-based approach. The co-occurrence–based approach represents texts as a bag of words vector while the statistical corpus-based approach uses Latent Semantic Analysis, which represents texts as vectors in a reduced-dimension space. The feature-based approach focuses on the similarity of words and the order between texts where words that have the same part of speech are aligned. The Word-Net based measure is used to compute semantic similarity between words and overlapping word orders in the two texts. Finally, the word embedding approach has the ability to consider the context of the words when representing words in a distributed space (Alian and Awajan, 2018).

For the purpose of analyzing paraphrased sentences in Arabic, we built a paraphrasing benchmark which consists of Arabic sentence pairs; part of these pairs is collected from Arabic books and lexicons while the other is constructed by experts using words form Arabic lexicons and AWSS dataset. Then, six transformation rules are utilized to generate the transformed form of the collected sentences. The dataset is labeled by Arabic specialists with different levels from Art College from Hashemite University (Alian et al., 2019).

The aim of this research is to analyze the constructed paraphrasing benchmark and use the K-means clustering to analyze and study the factors that affect the results of similarity and the distribution of sentences.

The K-means is considered as the simplest method used for partitioning a dataset into clusters of similar objects. Therefore, it is applied to the dataset after representing the sentences as vectors in a continuous vector space using wrod2vec model.

The main factors that will be studied to show their impact are: embedding method, weight of words and labeling dataset. Two ways are considered to implement the word2vec embeddings; the first one is achieved by training word2vec on the dataset and the second one by using pre-trained embeddings. Introducing the weight that represents the importance of words in a sentence is tested using Tf-idf. The way of labeling the dataset for similarity is another factor that may affects the results of similarity and paraphrasing which is tested through experiment.

This paper is organized as follows: section 2 describes the structure of the Arabic paraphrasing benchmark. Section 3 explains K-means clustering algorithm while in Section 4, we experiment k-means clustering on the benchmark. Finally we conclude in Section 5.

2. Arabic Paraphrasing Benchmark

This benchmark is constructed from Arabic sentences that are collected from several books used for teaching Arabic language such as Jordanian curriculums for Arabic language and other books (Jarim and Ali, 2004) (Omar, 1998) (Omar, 1420 - 1999) (Alkholi, 2001). In addition, Experts in Arabic language create a number of sentences based on words from some Arabic lexicons and words from Arabic Word Semantic Similarity (AWSS) dataset (Almarsoomi et al., 2013).

Arabic Paraphrasing can be reached by two methods; the first is based on paraphrasing hypothesis "two sentences are paraphrased if they consist of identical words except one word in the first sentence and its synonym in the second one". The second method is based on the language transformation rules (Al-Kholi, 1999). Therefore, the collected and created sentences are transformed into other sentences based on the transformation rules for Arabic sentences. The applied transformations are reflected in a set of rules referred to as "transformation rules" that were constructed by Chomsky (Chomsky, 1957); the creator of the constructional and transformational school.

The transformation rules can be categorized into a number of patterns; permutation, deletion, addition, reduction, expansion, and replacement. Permutation is the process of reordering words in the first sentence to get the new sentence while deletion is removing one item from the sentence. It is also the inverse of the addition rule which adds an item to the structure of the sentence. The expansion is the process of replacing a word by two different words that provide the same meaning while reduction is to represent two words by one word with the same meaning (Alian et al., 2019).

The transformation rules were applied on the collected sentences by two experts in Arabic language to produce the transformed sentences. Both experts have the degree of philosophy in Arabic where they collect and transform the sentences in three months.

Let A, B, and C are words or phrases in the sentence. Then, the transformation rules can be symbolized as in Table 1 as described by Al-kholi (Al-Kholi, 1999). Furthermore, the transformation is a description of the relationship between the deep structure of the sentence and the surface structure of the sentence.

Table 1: Transformation rules **(Alian et al., 2019)**

Transformation rule	Representation by symbols	Example
Permutation	A+B = B+A	تسلم الفائز الجائزة تسلم الجائزة الفائز
Deletion	A + B = […] + B A + B = A + […]	اسألوا أهل القرية عن اللص اسألوا القرية عن اللص
Addition	A=A+B	السماء صافية إن السماء صافية
Expansion	A = B+C	وددت نزول المطر وددت لو ينزل المطر
Reduction	A + B = C	الجو حار بارد الجو معتدل
Replacement	A = B	شارك الأستاذ في الأمسية الأدبية شارك الأستاذ في الأمسية الشعرية

3. Background

3.1. K-means

K-means is a non-hierarchical clustering algorithm which was introduced by Hartigan and Wong (1979) (Hartigan, 1979). It is considered as the most popular and simple approach used for partitioning a dataset (Steinley, 2007) or a number of observations in order to produce k clusters such that the total variation of within-cluster is minimized.

In this algorithm, the data is partitioned into k clusters where each cluster is represented by the mean of its content points and this is where the name of K-means comes from (Berkhin, 2002). However, there is no special equation for determining the number of clusters (k). It is specified by the problem domain or the application Data (Parsian, 2015).

K-means clustering can be used in many scientific and industrial applications (Berkhin, 2002). For example, it can be used in clustering documents into groups according to their content similarity or grouping customers into clusters according to their behavior (Parsian, 2015). It is a prototype-based partitioning clustering technique that attempts to find a user specified number of clusters (K) presented by their centroid (Tan, 2014).

K-means clustering technique is used in partitioning data into K classes (C1,C2,…,Ck) where K is the number of clusters and it is a predefined parameter while Ck is the group of nk objects in cluster k . The data consists of N objects where each object has P features. If Xnxp denotes NxP matrix, K-means generates k clusters by selecting k centroids then starts an iterative procedure by computing the squared Euclidean distance between any object vector and the centroids vectors. Euclidean distance is computed as in Equation (1)

$$d^2(i,k) = \sum_{j=1}^{P} (x_{ij} - s_j^{(k)})^2 \qquad (1)$$

Where
$d^2(i,k)$ is the squared Euclidean distance
i is the ith object
k is the *kth* centroid vector represented by $(s_1^{(k)}, \dots, s_P^{(k)})$.

Then an object is assigned to the cluster with the closest centroid to that object. The centroid is a point in the P-dimensional space which is selected randomly at the beginning then it is updated by computing the mean of the values on each feature over all objects in the corresponding cluster. The centroid for the jth feature in cluster Ck is found by Equation (2) (Steinley, 2006) (Steinley, 2007)

$$\bar{x}_j^{(k)} = \frac{1}{n} \sum_{i \in C_k} x_{ij} \qquad (2)$$

And the centroid vector for the cluster Ck will be:

$$\bar{x}^{(k)} = (\bar{x}_1^{(k)}, \bar{x}_2^{(k)}, \dots, \bar{x}_P^{(k)})' \qquad (3)$$

K-means attempts to find good partitioning by minimizing the error sum of squares (SSE) in the constructed clusters. It is considered as a loss criterion and calculated as in Equation (4):

$$\text{SEE} = \sum_{j=1}^{P} \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \bar{x}_j^{(k)})^2 \qquad (4)$$

3.2. Aravec

AraVec (Mohammad, Eissa and El-Beltagy, 2017) is an Arabic language pre-trained distributed word embedding model where it was produced using the Word2Vec skip-gram technique trained on World Wide Web pages Arabic content with a vector dimension of 300 and a vocabulary size of 145,428. The Aravec

models have been built based on three domains: Web pages, Arabic tweets and Wikipedia with articles of Arabic content. These domains provide building the models with more than 3.3 billion tokens.

3.3. Evaluation measures

Precession and recall are commonly used as measures for evaluating the results in Information Retrieval and Machine Learning experiments (Powers, 2007). In information retrieval, they are used for measuring how well the system retrieves relevant items requested by the user (K.M., 2011).
Recall is called sensitivity and it is defined as the percentage of positive items that are predicted correctly positive as in formula (4) (K.M., 2011):

$$Recall = \frac{TP}{(TP+FN)} \qquad (1)$$

Where
TP is the true positives (items that are positive and they are correctly identified positive)
FN is the false negatives (items that are incorrectly identified as negative)
(TP+FN) is the total number of actual positives.

While Precision is called confidence and it is defined as the percentage of predicted positives that are real positives as in formula (2) (K.M., 2011):

$$Precision = \frac{TP}{(TP+FP)} \qquad (2)$$

Where
TP is the true positive
FP is the false positive (items that are negative but falsely identified as positive)
(TP+FP) is the total number of positives predicted

3.4. Term frequency-inverse document frequency

Term frequency-inverse document frequency (TF-IDF) is considered as a numerical and statistical approach that takes into consideration the importance of a term within a group of documents. As the frequency of a word occurrence increased in a document, TF-IDF is proportionally increased but inversely proportional to the word's frequency in all documents or corpus (Mohsen et al., 2018). The TF-IDF is computed as in the following formula:

$$TFIDF = \frac{n_t}{N_d} \times \log\left(\frac{N_j}{n_j}\right) \qquad (3)$$

Where
N_j is the total number of documents in the corpus
n_j is the number of documents that contain term t.
n_t is the number of times where word t appears in the document.
N_d is the total number of terms in the document d.

4. Experiment and Results

Three factors that affect sentences similarity and paraphrasing identification are evaluated using Kmeans clustering. The first factor is the approach used for providing sentence's embeddings; trained and pre-trained embeddings while the second factor is introducing the weight of words. The third factor is the way of labeling the dataset; by majority or by similarity score threshold. The resulted clusters that contain similar sentences are compared with the labeled dataset using recall and precision measures.

4.1 Self-trained Word2Vec word embeddings vs pre-trained Word2Vec

In evaluating the approach used for representing sentences in the distributional space, two experiments are conducted; the first experiment is based on self-trained of word2vec on the dataset that consists of 2020 sentences with 3145 vocabulary while the second experiment is conducted on pre-trained embeddings using

Aravec that utilizes 77,600,000 Arabic tweets, gathered from different locations, in the training of word2vec. The vector that represents a sentence in both experiments has constructed as the mean of its content words' vectors. Then, K-means is applied to partition the dataset into clusters of similar sentences. The number of clusters (K) is chosen as half the number of sentences which is equal to 1010 clusters. Since we compare each pair of sentences if they are similar, paraphrased or not. Table 2 and Table 3 show the confusion matrix for both experiments for the similarity detection.

Table 2: Similarity confusion matrix for self-trained

Training on the dataset	**Labeled**	
	Similar	not similar
Similar	638	95
not similar	226	51

Table 3: Similarity confusion matrix for the Aravec training embeddings

Aravec training	**Labeled**	
	Similar	not similar
Similar	682	102
not similar	182	44

In Table 4 and Table 5, the confusion matrices for paraphrasing test are shown for dataset training embeddings and Aravec embeddings, consequently.

Table 4: The confusion matrix for paraphrasing identification with self-trained embeddings

	paraphrased	**not paraphrased**
paraphrased	576	157
not paraphrased	188	89

Table 5: The confusion matrix for paraphrasing identification with Aravec embeddings

	Paraphrased	**not paraphrased**
paraphrased	611	173
not paraphrased	148	78

Accordingly the recall will be computed as in as in Formula (1) as (in section 3.3) where the true positive is the paraphrased sentences correctly identified while the false negative is the paraphrased incorrectly labeled as not paraphrased.

While the precision will be calculated as formula (2) as previously mentioned in section 3.3 where the true positive represents the paraphrased sentences correctly identified while the false positive represents the sentences that incorrectly labeled as paraphrased.

The results of recall and precision for detecting similar sentences are shown in Fig.1 while the results for paraphrasing identification are shown in Fig.2.

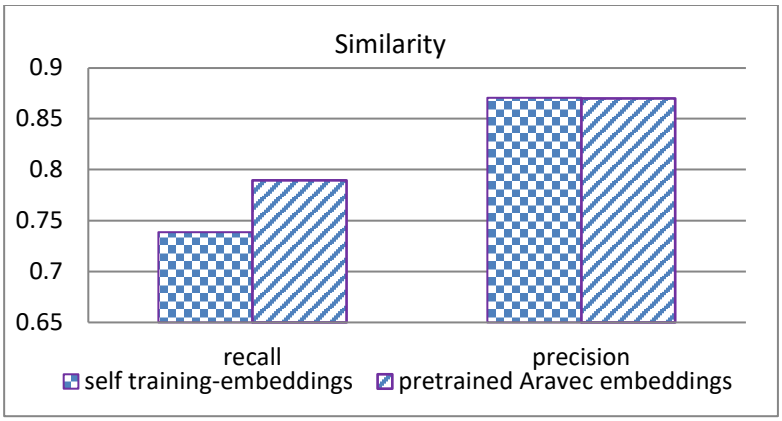

Fig 1. Recall and precision for both dataset and Aravec embeddings for similarity.

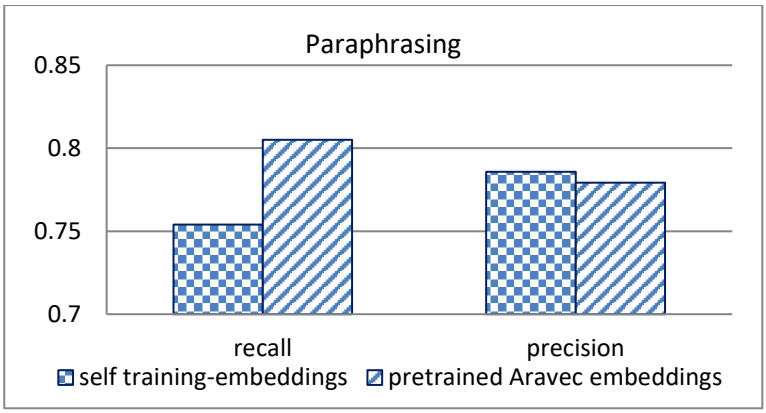

Fig.2. Recall and Precision for both dataset and Aravec for paraphrasing.

The recall achieves better values with the pre-trained Aravec embeddings in similarity detection and paraphrasing identification while the precision with the dataset training embeddings provide better value for paraphrasing identification.

4.2 TF-IDF weight

Each word in the structure of the sentence has its significant importance. In this research, word's importance is expressed as the weight which is computed as the Tf-idf as in formula (3) as in section 3.4.

When we include the weight for each word to produce the embedding of a sentence, the results for paraphrasing identification is little better than the results without taking the weight into consideration. As shown in Table 6.

Table 6: comparing paraphrasing identification results

Experiment	recall	precision	F1 score
K-means TfIDF	0.785166	0.808959	0.796885
K-means	0.779337	0.805007	0.791964

4.3 Labeling Dataset

We used two methods to provide the final label to the dataset. In the first method, we used the label of each sentence pair for similarity by majority of experts' decision about similarity. If more than three experts give a decision of similarity for a sentence pair then this pair is labeled as similar. In the second method, we compute the average of the score of similarity provided by the experts. A sentence pair is labeled as similar if the score average is greater than or equal to 0.50, otherwise the sentence pair is labeled as non-similar.

To evaluate the effect of these two methods on the similarity of sentences, we conduct an experiment by applying the two methods of labeling the dataset with K-means and weighted embeddings. The results of labeling by majority are shown in Table 7 while the results of labeling based on the score of similarity are represented in Table 8. It is shown that choosing the method of labeling a sentence pair has affected the precision and recall measures. From Table 7, it is shown that labeling by majority provides better recall for K-means with Tf-idf weighted embeddings.

Table 7: labeling by majority results

Experiment	Recall	Precision	F1 score
K-means TfIDF	0.87468	0.790751	0.830601
K-means	0.789352	0.869898	0.82767

Table 8: the results of labeling by score >=50%

Experiment	Recall	Precision	F1 score
K-means TfIDF	0.748082	0.80137	0.77381
K-means	0.79863	0.743622	0.770145

5. Conclusion

In this research, k-means clustering is applied for Arabic paraphrasing benchmark that consists of pairs of Arabic sentences with two labels: one for similarity and the other for paraphrasing. K-means is used to explore the dataset as clusters of similar sentences and experiment which pairs are paraphrased and allocated in the same cluster. The experiment includes two ways for producing sentences' embeddings: one way is constructing embeddings by training word2vec on the dataset and the other utilizes the embeddings of Aravec model which provides a pre-trained embeddings. The results for pre-trained embeddings produce a better recall in detecting similar and paraphrased sentences. Applying K-means with introducing Tf-idf weighting for words provides a recall of 0.78 while labeling the dataset for similarity by majority of experts provides a recall of 0.87 which is better than the recall achieved with labeling by similarity score threshold with Tf-idf weighting.

References

Alian, M. and Awajan, A. (2018) 'Semantic similarity approaches- Review', 2018 International Arab Conference on Information Technology (ACIT2018), Werdanye, Lebanon, 1-6.

Alian, M., Awajan, A., Al-Hasan, A. and Akuzhia, R. (2019) 'Towards building Arabic paraphrasing benchmark', the Second International conference on Data Science, E-learning and Information Systems (DATA' 2019), Dubai.

Al-Kholi, M.A. (1999) *Transformation rules for Arabic language*, dar Al-Falah.

Alkholi, M.A. (2001) *Semantics*, Amman: dar Al-falah.

Almarsoomi, F.A., O'shea, J.D., Bandar, Z. and & Crockett, K. (2013) 'AWSS: An Algorithm for Measuring Arabic Word Semantic Similarity', 2013 IEEE International Conference on Systems, Man, and Cybernetics, 504-509.

Berkhin, P. (2002) 'Survey of Clustering Data Mining Techniques'.

Chomsky, N. (1957) *Syntactic Structure*, Paris: Mouton publishers.

Fernando, S.a.S.M. (2008) 'A semantic similarity approach to paraphrase detection', the 11th Annual Research Colloquium of the UK Special Interest Group for Computational Linguistics.

Hartigan, J.A.a.W.M.A.. (1979) 'Algorithm AS 136: A K-Means Clustering Algorithm', *Applied Statistics*, vol. 28, pp. 100-108, Available: http://dx.doi.org/10.2307/2346830.

Jarim, A. and Ali, M. (2004) *Clear grammar in the rules of Arabic language*, 2nd edition, Egyptian and Saudi dar for Publishing.

K.M., T. (2011) 'Precision and Recall.', in Sammut C., W.G.I. (ed.) *Encyclopedia of Machine Learning*, Boston, MA: Springer.

Mohammad, A.B.S., Eissa, K. and El-Beltagy, S.R. (2017) 'AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP', *Procedia Computer Science*, vol. 117, pp. 256-265.

Mohsen, G., Al-Ayyoub, M., Hmeidi, I. and Al-Aiad, A. (2018) 'On the Automatic Construction of an Arabic Thesaurus', 9th International Conference on Information and Communication Systems (ICICS).

Omar, A.M. (1420 - 1999) *Language exercises and grammar*, Kuwait University.

Omar, A.M. (1998) *semantics*, 5th edition, Cairo: Book World.

Parsian, M. (2015) *Data Algorithms,* O'Reilly Media, Inc.

Powers, D.M.W. (2007) 'Evaluation: From Precision,Recall and F-Factor to ROC, Informedness, Markedness & Correlation'.

Srivastava, S. and Govilkar, S. (2017) 'A Survey on Paraphrase Detection Techniques for Indian Regional Languages', *International Journal of Computer Applications*, vol. 163, no. 9, pp. 0975 – 8887.

Steinley, D..B.M.J. (2007) 'Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques', *Journal of Classification*, vol. 24, pp. 99-121.

Tan, P.N.S.M..K. (2014) *Introduction to Data Mining*, 1st edition, Pearson education limited.