Communications on Applied Electronics (CAE) – ISSN : 2394-4714
Foundation of Computer Science FCS, New York, USA
Volume 6 – No.2, November 2016 – www.caeaccess.org

Automatic Measurement of Semantic Similarity among Arabic Short Texts

Fatma Elghannam
Electronics Research Institute
Giza, Egypt

ABSTRACT
Documents that are dealing with the same topic include normally many identical words. Accordingly, surface words co-occurrence similarity measures has been applied successfully to measure the similarity between these documents. However, the problem is not a trivial task when dealing with short texts that carry the same or close meaning but with different vocabularies. Toward solving this problem, researchers have been investigating methods for word analysis at the semantic level. We introduce a new method to measure the semantic similarity between short texts. In the proposed method, semantic distribution and lexical similarity measures are combined to determine the degree of similarity between two words. The similarity between two words is measured as the lexical similarity between the vectors of similar words extracted from corpus as a second order word vector. The proposed method was applied to measure the semantic similarity between Arabic short texts. The experiments performed showed that the best accuracy achieved by the proposed method was 97% compared to 93% recorded for the second order distribution similarity.

Keywords
Semantic similarity of words, similarity of short texts, corpus based similarity measure, semantic distribution, lexical similarity.

1. INTRODUCTION
Measuring text similarity is a crucial task in many natural language processing applications including information retrieval, document clustering, text mining, machine translation, question answering, word sense disambiguation, paraphrase extraction, summarization and image retrieval. For example, in information retrieval the documents are ranked according to the similarity of a query to each document in the collection. Document clustering is the grouping of pieces of text that carry the same meaning. In text mining, sentence similarity is used as a criterion to discover unseen knowledge from textual databases [1]. In machine translation, systems must choose a translation hypothesis in the target language that is semantically closest, if not identical, to the source language text [26]. Word sense disambiguation is the identification of the sense closest to particular instance of the target word. Paraphrases are pieces of text that carry the same or close meaning but with different vocabularies. Query-based summarization requires, choosing those sentences to be part of the summary that are closest to the query. In image retrieval from the Web, the use of short text surrounding the images can achieve a higher retrieval precision than the use of the whole document in which the image is embedded [7]. The previous mentioned applications show that methods for measuring sentence similarity play an increasingly important role for the research community involved in textual knowledge representation and analysis.

A variety of similarity measures has been defined between documents [24], [10], [14] but there is a less work related to the similarity between short sentences and expressions. In case of measuring the similarity between documents, many researches tend to analyze the surface words co-occurrence between documents [10], [13], [3] as naturally in documents dealing with the same topic there exist many identical words. However, the problem is not a trivial task when dealing with measuring the similarity between short sentences that carry the same or close meaning but with different vocabularies. Toward solving this problem, researchers have been investigating methods for word analysis at the semantic level to evaluate the semantic similarity between words and texts.

Semantic similarity is a measure of identifying the level of relatedness between a set of texts [21]. The most important approaches to implement semantic similarity are knowledge-based and corpus-based measures. Knowledge-based similarity is dependent on semantic network. The similarity between two words can be determined using their relative positions in the knowledge base hierarchy. The two words can have high similarity score if the words are in the same WordNet synset or if one word is a hypernym of another word [27]. Corpus-Based similarity is a semantic similarity measure that determines the similarity between words according to information gained from large corpora. The corpus based similarity relies on the distribution hypothesis "Words that occur in similar contexts tend to have similar Meanings" [9], [12], [17]. The distributional hypothesis suggests that the more semantically similar two words are, the more distributionally similar they will be in turn, and thus the more that they will tend to occur in similar linguistic contexts. Thus, the two words are considered semantically related simply if they have many common co-occurring words.

In this paper, a new corpus-based method is introduced to measure the semantic similarity between short texts. In the proposed method semantic distribution and lexical similarity measures are combined to determine the degree of similarity between two words. The proposed method is applied to measure the semantic similarity between two types of Arabic short texts, complete sentences and expressions. The experiments performed showed that the proposed method improves the performance of the similarity measure between two short texts instead of relying only on word distribution similarity calculations.

The rest of this paper is organized as follows. Section 2 presents a survey of the most important related works. Section 3 the details of the proposed method. Section 4 the experimental results. Section 5 includes a conclusion of the work.

2. RELATED WORK
Textual similarity refers to the concept of similarity between

texts. There are two trends to measure text similarity: the first is the lexical similarity which is based on surface matching of words, while the second one is the semantic similarity where the similarity is measured on the basis of the actual meaning of words [30]. The following paragraphs review different similarity measure techniques that are used in the both types especially the techniques that have been adopted in the proposed method.

Lexical similarity is a well-known type of similarity that measures the degree of closeness between two given string sequences on the basis of character and term matching [32]. Lexical similarity is categorized into character based similarity and term/token based similarity. Levenshtein, Longest common subsequence, N-gram, Mong, and Jaro are different techniques in the literature described in character based similarity. Levenshtein distance (also called edit distance) is a string metric for measuring the difference between two sequences. The **Levenshtein** distance between two words is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to transform one word into the other [28]. In **N-gram** similarity technique, the similarity is computed on the basis of distance between each character in two strings. This distance is computed by dividing the number of similar grams by maximal number of n-grams. **Jaro** similarity technique is based on the number and order of the common characters between two strings. It takes into account typical spelling deviations and mainly used in the area of record linkage [8]. Cosine, Jaccard, and Pointwise Mutual Information are famous techniques introduced for term based similarity. **Cosine** similarity is a widely used approach to find the similarity between two texts based on the cosine of angle between two vectors [31]. To find the similarity between two texts, each text is represented in the form of vector. Each word in text defines a dimension in the Euclidean space and the frequency of each word corresponds to the value in the dimension. Jaccard coefficient similarity is a count based co-occurrence measure technique. Jaccard coefficient is computed based on number of elements in the intersection set divided by the number of elements in the union set [20]. A survey of these techniques and text similarity approaches exists in [15].

Several successful methods utilize the information gained from large corpora to measure the semantic similarity between words. **Pointwise Mutual Information (PMI)** was first used in the context of word associations by Church and Hanks [6]. PMI is a very simple information-theoretic measure that, when computed between two words x and y, "compares the probability of observing x and y together (*P(x,y)*) with the probabilities of observing x and y independently (*P(x) P(y)*)" [6]. It is defined as:

$$PMI(x;y) = \log \frac{p(x,y)}{p(x)p(y)} \quad (1)$$

PMI has been used for finding collocations and associations between words by counting occurrences and co-occurrences of words in a corpus. Pantel and Lin [29] compute the similarity between two words using the cosine coefficient of their mutual information vectors. **Latent Semantic Analysis (LSA)** proposed by Landauer et al. [23] is another corpus-based measure of semantic similarity. In this technique, context matrix that containing word counts occurs in each paragraph is constructed from large corpus. Rows in such matrix represent the words while columns represent the paragraphs. Then, singular value decomposition (SVD) is used to reduce the number of rows in the matrix. The similarity between two words is measured by computing the similarity between the vectors formed by the rows. **Islam and Inkpen** [19] presented an approach to measure the similarity of two texts based on semantic and syntactic information. The authors considered three different similarity measures to assess the similarity between sentences. First, the longest common subsequence measure is applied. Second, they use a bag-of-words representation to perform a semantic word similarity, which is measured by a corpus-based measure. At the end, they use syntactic information to evaluate the word order similarity. **Mihalcea et al.** [27] combined corpus-based and knowledge-based measures for measuring the semantic similarity of texts. The authors used two corpus-based measures, PMI-IR and LSA and six knowledge-based measures to compute the word-to-word similarity. According to the word-to-word similarity measure they derive the sentence-to sentence similarity. For each word in the first sentence (main sentence), it tries to identify the word in the second sentence that has the highest similarity. Then, the process is repeated using the second sentence as the main sentence. The score of words similarities are then weighted. Finally, the total similarity score is the average of the values found. **Islam and Inkpen** [18] presented corpus-based method for calculating second order co-occurrence pointwise mutual information of two target words. Their method uses PMI to extract the most important neighboring words of the two target words, and adopted the neighboring words to calculate the relative similarity. **Gomaa amd Fahmy** [16] compared several string-based and corpus-based similarity measures and a combination of them for the task of automatic short answer scoring in Arabic language. **Extracting DIStributionally Similar words using CO-occurrences DISCO** word space is a tool for retrieving the distributional similarity between two given words, and for retrieving the distributionally most similar words for a given word [22]. Their method for computing the distributional similarity between words starts by counting words co-occurrences to build the co-occurrence matrix. Every row in the matrix describes a word, and is called a first order word vector. They are used Lin's measure [25] to calculate the first order similarity score between two target words. This score is used as matrix weights to get second order word vectors which are used to compute a second order word similarity measure.

3. PROPOSED METHOD

The objective of the proposed method is to improve the performance of measuring semantic similarity between two given short texts. It uses the statistical lexical similarity between the vectors of similar words (second order word vectors) extracted from corpus instead on relying only word distribution similarity calculations. The notion is that two words tend to be semantically close if they have a small distance between their vectors of similar words. To determine the degree of similarity between two words, we measure the lexical similarity between their second order word vectors which are obtained based on the second order distribution similarity. The steps to calculate the semantic similarity between two target texts are: preprocessing, retrieving word profile, normalization, word similarity, and text similarity. The following sections describe these steps in detail: the distributional similarity between words starts by counting words co-occurrences to build the co-occurrence matrix. Every row in the matrix describes a word, and is called a first

order word vector. They are used Lin's measure [25] to calculate the first order similarity score between two target words. This score is used as matrix weights to get second order word vectors which are used to compute a second order word similarity measure

Pre-processing: given two short texts T1, T2 that is required to measure the semantic similarity between them, each of the two texts is partitioned into a list of tokens (words). Then closed words such as articles, pronouns, prepositions, conjunctions, and punctuations are eliminated as they have little semantic discrimination power in our calculations. The Arabic Lemmatizer [11] is adopted in this step. Unlike other methods that including stemming or lemmatization in the preprocessing step, it has been postponed to the normalization step as it will be explained in the next step.

Retrieving word profile: word profile is represented as the list of similar words extracted from corpus. For each word in the two texts, list of similar words is extracted from corpus as a second order word vector. DISCO tool is used for this purpose. Disco builds the second order word vectors by first counting words co-occurrences to build the co-occurrence matrix. Every row in the matrix describes a word, and is called a first order word vector. They are used Lin's measure to calculate the first order similarity score between two target words. This score is used as matrix weights to get second order word vectors. In our preliminary experiments, after investigating the extracted list of similar words for different word derivations that have the same lemma or root, it is found that the word inflection form can serve in improving the results of extracting the most relevant list of similar words. This in turn contributes to clarify the ambiguity of word sense. Fig.1. Shows an example of the extracted list of similar words for the two words (تفاوض, مفاوضات) although they share the same root, their vectors of similar words are dissimilar and have different semantic trends. So, to obtain the best results, the original word form is applied as it is in the process of retrieving the word profile.

تفاوض : (وكويتي, ستشطب, تنفى, عجافها, مشتركان, تقترض, تتفاوض, وترضية, مرشحنا, ينعشون)

مفاوضات : (المفاوضات, محادثات, المحادثات, مباحثات, المباحثات , جولة, عملية, انتخابات, مشاورات, التفاوض)

Fig 1 Vectors of first 10 most similar words for the two words " تفاوض" and "مفاوضات "

Normalization: linguistic processing is used to refine the extracted list of similar words and improve the similarity measurements accuracy between vectors. The root form for each element in the vector of similar words is used instead of their original form. Arabic is a highly inflectional language, and concepts can be represented by varieties of word forms. So, we vote towards deep abstracting for the word vectors by using the root form instead of the lemma or stem. The Arabic Lemmatizer is also adopted in this step.

Word Similarity: At this point, for each word w in both texts T1, T2 we got word profile as a vector of similar words represented in their root forms.

In the current step, for each word w in the first text T1 calculate the similarity measure against the words in T2. The similarity between two words is calculated as the Cosine similarity between their vectors of similar words represented in their root form. The open source library SimMetrics [5] is used in this step.

Text Similarity: in the proposed method, the similarity between texts is calculated by the same way as Mihalcea et al. [27]. So, after calculating the similarity between all the words in the two texts T1 and T2; then to derive the similarity between the two texts we use maximum- row maximum-column method proposed by Mihalcea et al. [27]. For each word w in the text T1 identifies the word in the text T2 that has the highest semantic similarity (maxSim (w, T2)). The same process is applied with words in T2 to determine the most similar word in T1. The highest words similarity for each text are then summed up, and normalized with the length of each text. Finally, the resulting similarity scores are combined using a simple average. The following example illustrates the steps to calculate the similarity between two given short texts.

Example:

Consider the two texts T1, T2

T1: مجلس الشعب يصادق

T2: البرلمان يوافق

The two texts share the same meaning but they completely use different vocabularies. After partitioning the two texts into a list of tokens (words), we have two lists:

T1={*t11("مجلس"), t12("الشعب"), t13("يصادق")*} ,
T2={*t21("البرلمان"), t22("يوافق")*}

Extract the second order word vector (most similar words) for each word in the two texts, Fig. 2 shows a sample of the extracted word vector for the word *t11* ("مجلس ").

{ المجلس, لمجلس , ,البرلمان, مجلسي, اللجنة, لجنة, ومجلس, الوزراء, النواب, الكونغرس, الجمعية, الاتحاد, المؤتمر, بمجلس,الكنيست }

Fig. 2 Sample of the distributionally most similar words for the word t11 ("مجلس"")

Build the similarity matrix by calculating the similarity between the words in T1 and T2. The similarity is calculated as the cosine similarity between the vectors of similar words in their root forms. Fig. 3 shows the results of calculating the similarity between the words in T1 and T2. The figure also shows the highest word similarity for each row and column (Max rows, Max columns) and their averages.

Finally, the overall similarity between the two texts T1 & T2 = (0.485+0.5645) / 2= 0.525

	t_{21} *("البرلمان")*	t_{22} *("يوافق")*	Max rows	Average Max rows
t_{11}*("مجلس")*	0.567	.054	**0.567**	0.485
t_{12}*("الشعب")*	0.326	.051	**.051**	
t_{13}*("يصادق")*	.15	.562	**.562**	
Max columns	**0.567**	**.562**		
Average Max columns	0.5645			

Fig. 3 Similarity between words in T1, T2

4. EXPERIMENTAL RESULTS

To measure the performance of the proposed method it was applied to measure the similarity between pairs of Arabic short texts. It is common in literature of the language the use of expressions or terms. Semantic similarity between expressions is a major task for Natural Language Processing (NLP) applications, especially for those addressing semantic aspects of language such as machine translation. Measuring the semantic similarity between two expressions is mostly more complex than the case of complete sentences. When measuring the similarity between two sentences that are covering the same meaning, it is normally existing identical words, words that have the same stem, or at least words that have high semantic similarity. Furthermore, ambiguous words that can carry various meanings, when included in a sentence the ambiguity can be resolved as other words in the sentence serve to direct the sentence towards the intended meaning of the talk. So, the longer the texts lengths are, the more the opportunity to discover the similarity between texts.

In our experiments, two types of data are used, a set of pairs of complete sentences, and another set of short expressions. Data are collected from different resources; most of them are news web pages. To collect pairs of texts that carry the same or close meaning, we explore different web sites that present the same story but with a variety of vocabularies to express the intended meaning. The data includes positive and negative text pairs with equal ratio. Two texts that share the same meaning are called positive text pair, while the negative pair is not. The first dataset is a set of 100 pairs of complete sentences. The sentence length is between 4 to 9 words. The second is the expression data set which contains of a set of 240 expression pairs with maximum expression length 3 words. In the expression dataset, the existence of similar vocabularies in an equivalent pair is not allowed. Tables 1,2 show samples of the collected pairs of sentences and expressions that are used in our experiments.

Table1. Sample of sentence pairs dataset.

دول العالم تتوصل الى اتفاق تاريخي لمواجهة تغيرالمناخ	اقراراتفاق عالمي لمكافحة التغيرالمناخي
تهاوى اسعارالنفط فى البورصات العربية	اسعارالنفط تواصل هبوطها لأدنى مستوياتها
قمة أوروبية تركية للتوصل لاتفاق حول أزمة المهاجرين	الاتحاد الاوروبي وتركيا يتفقان على مبادئ خطة لمواجهة أزمة اللاجئين
الشبكات الذكية حل سحري لمواجهة الطلب المتنامي على الطاقة	الشبكة الذكية لمواجهة ازدياد الحاجة للطاقة الكهربائية
اختراع جديد يساعد العميان على الحركة	تطويرجهازجديد يتيح للمكفوفين الحركة بسهولة

Table 2. Sample of expression pairs dataset.

مجلس الشعب	البرلمان
زعزعة الأمن والاستقرار	تهديد السلم
خرق معاهدة	انتهاك الاتفاقية
موجات ارتدادية	زلزال
وباء	انتشار فيروس

To measure the validity of the proposed method, we measure its accuracy to classify pairs of texts as similar or not. The classification accuracy of the proposed method is also compared against the second order distribution similarity measure. A supervised learning algorithm is applied to classify pairs of texts as similar or not. First, each of the proposed method and the second order distribution similarity (implemented by DISCO toolkit) is applied to measure the similarity between pairs of texts. Each method produces a list of similarity scores ranges from 1 to 0 according to the degree of similarity between text pairs. The text pairs are also classified manually as a positive or negative class, provided with their contexts. A text pair that shares the same meaning is a positive class, otherwise it is negative. Then, for each of both methods a supervised learning algorithm is used to build a model. The input to the learning algorithm is a set of examples that include the text pairs similarity scores and their corresponding manual classification. Then, based on the input examples, the system builds the classifier which is used to classify other set of pairs either positive or negative class. Cross-validation technique is used in the classification process to estimate the model prediction performance.

Two datasets are used in the experiment: complete sentences and expressions as described before. WEKA platform [1], [2], [3] is used for the classification process. We have examined two different classification algorithms: NaiveBayes, and DecisionTable. Tables 3&4 illustrate the results of the two classifying algorithms for both of the proposed method and the second order distribution similarity method. The performance results of the proposed method compared to the second order distribution similarity show a significant improvement in the accuracy in terms of correctly classified instances obtained by the proposed method. The best result (97%) in sentence test is obtained by the proposed method compared to (93%) in the second order similarity method. In the expression test the best result (85%) is obtained by the proposed method compared to (83%) in the second order similarity method.

Tables 5, 6 show the confusion matrix results of NaiveBayes algorithm for both methods using the two datasets. Rows correspond to the two classes A (similar) , B (not similar). Columns correspond to classes in the classification result. The diagonal elements in the matrix represent the number of correctly classified texts of each class. The off-diagonal elements represent misclassified texts or the classification errors. For example in the table 5, number of instants in classes A, B that are correctly classified are (50, 48) for the proposed method, compared to (47, 47) in the second order similarity. The number of instants in classes A, B that are incorrectly classified are (1,2) in the proposed method, compared to (4, 3) in the second order similarity.

The results verified that combining semantic distribution and lexical similarity measures to calculate the text improves the performance of measuring the semantic similarity, instead of relying only on distribution similarity calculations. As expected, the proposed method has on the average better classification accuracy for sentences than that of expressions data due to barriers in ambiguity and text length, and insuring completely different vocabularies in case of the expressions dataset as mentioned above.

Table 3. Performance results for the expressions test.

	Second Order Similarity	**Proposed Method**
NaïveBayes	83	**85**
Decision Table	82	84

Table 4. Performance results for the sentences test.

	Second Order Similarity	**Proposed Method**
NaïveBayes	93	**97**
Decision Table	92	95

Table5. Confusion matrix for the second order similarity and proposed method- sentences test.

Second Order Similarity		**Proposed Method**		
A	B	A	B	
47	4	50	1	A=similar
3	47	2	48	B=not similar

Table6. Confusion matrix for the second order similarity and proposed method- expressions test.

Second Order Similarity		**Proposed Method**		
A	B	A	B	
80	23	81	22	A=similar
12	92	10	95	B=not similar

5. CONCLUSION

The paper presents a new method to further improve the performance of automatic measuring the semantic similarity between short texts. In the proposed method, the similarity is measured as the lexical similarity between the vectors of similar words extracted from corpus as a second order word vector. The proposed method was evaluated using two types of Arabic data including a set of pairs of complete sentences, and another set of expressions. The experiments performed showed a significant improvement in the accuracy obtained by the proposed method compared to the second order distribution similarity method. The best result 97% in sentence test was obtained by the proposed method compared to 93% in the second order similarity method. The results verified that combining semantic distribution and lexical similarity to determine the degree of similarity between two words improves the performance of measuring the text semantic similarity, instead of relying only on word distribution similarity calculations.

6. REFERENCES

[1] Atkinson-Abutridy, J., Mellish, C. and Aitken, S., 2004. Combining information extraction with genetic algorithms for text mining. IEEE Intelligent Systems, 19(3), pp.22-30.

[2] Bouckaert, R.R., Frank, E., Hall, M., Kirkby, R., Reutemann, P., Seewald, A. and Scuse, D., 2013. WEKA Manual for Version 3-7-8. Hamilton, New Zealand.

[3] Carenini, G., Cheung, J.C.K. and Pauls, A., 2013. MULTI?DOCUMENT SUMMARIZATION OF EVALUATIVE TEXT. Computational Intelligence, 29(4), pp.545-576.

[4] CBA. Data mining tool Downloading URL : http://www.comp.nus.edu.sg/~dm/p_download.html.

[5] Chapman, S., 2005. SimMetrics-open source similarity measure library. URL: http://nazou. fiit. stuba. sk/home/documentation/concom/concom. doc, Visited:(April 2016).

[6] Church, K.W. and Hanks, P., 1990. Word association norms, mutual information, and lexicography. Computational linguistics, 16(1), pp.22-29.

[7] Coelho, T.A., Calado, P.P., Souza, L.V., Ribeiro-Neto, B. and Muntz, R., 2004. Image retrieval using multiple evidence ranking. IEEE Transactions on Knowledge and Data Engineering, 16(4), pp.408-417.

[8] Cohen, W., Ravikumar, P. and Fienberg, S., 2003, August. A comparison of string metrics for matching names and records. In Kdd workshop on data cleaning and object consolidation (Vol. 3, pp. 73-78).

[9] Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K. and Harshman, R., 1990. Indexing by latent semantic analysis. Journal of the American society for information science, 41(6), p.391.

[10] El-Ghannam, F. & El-Shishtawy, T. 2013. Multi-Topic Multi-Document Summarizer, International Journal of Computer Science & Information Technology (IJCSIT) Vol. 5, No 6, December 2013

[11] El-Shishtawy, T. & El-Ghannam, F. 2012. An Accurate Arabic Root-Based Lemmatizer for Information Retrieval Purposes, International Journal of Computer Science Issues, Volume 9, Issue 1, pp. 58-66.

[12] Firth, J.R., 1957. {A synopsis of linguistic theory, 1930-1955}. In Studies in Linguistic Analysis, pp. 1-32. Blackwell, Oxford.

[13] Fung, B.C., Wang, K. and Ester, M., 2003, May. Hierarchical document clustering using frequent itemsets. In SDM (Vol. 3, pp. 59-70).

[14] Glass, J. and Derr, E., Miavia, Inc., 2004. Document similarity detection and classification system. U.S. Patent Application 10/710,918.

[15] Gomaa, W.H. and Fahmy, A.A., 2012. Short answer grading using string similarity and corpus-based similarity. International Journal of Advanced Computer Science and Applications (IJACSA), 3(11).

[16] Gomaa, W.H. and Fahmy, A.A., 2013. A survey of text similarity approaches. International Journal of Computer Applications, 68(13).

[17] Harris, Z.S., 1954. Distributional structure. Word, 10(2-3), pp.146-162.

[18] Islam, A. and Inkpen, D., 2006, May. Second order co-occurrence PMI for determining the semantic similarity

of words. In Proceedings of the International Conference on Language Resources and Evaluation, Genoa, Italy (pp. 1033-1038).

[19] Islam, A. and Inkpen, D., 2008. Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD), 2(2), p.10.

[20] Jaccard, P., 1901. Etude comparative de la distribution florale dans une portion des Alpes et du Jura. Impr. Corbaz.

[21] Jiang, J.J. and Conrath, D.W., 1997. Semantic similarity based on corpus statistics and lexical taxonomy. arXiv preprint cmp-lg/9709008.

[22] Kolb, P., 2008. Disco: A multilingual database of distributionally similar words. Proceedings of KONVENS-2008, Berlin.

[23] Landauer, T.K. and Dumais, S.T., 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological review, 104(2), p.211.

[24] Lee, M., Pincombe, B. and Welsh, M., 2005. An empirical evaluation of models of text document similarity. Cognitive Science Society.

[25] Lin, D., 1998, August. Automatic retrieval and clustering of similar words. In Proceedings of the 17th international conference on Computational linguistics-Volume 2 (pp. 768-774). Association for Computational Linguistics.

[26] Marton, Y., Callison-Burch, C. and Resnik, P., 2009, August. Improved statistical machine translation using monolingually-derived paraphrases. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1 (pp. 381-390). Association for Computational Linguistics.

[27] Mihalcea, R., Corley, C. and Strapparava, C., 2006, July. Corpus-based and knowledge-based measures of text semantic similarity. In AAAI (Vol. 6, pp. 775-780).

[28] Navarro, G., 2001. A guided tour to approximate string matching. ACM computing surveys (CSUR), 33(1), pp.31-88.

[29] Pantel, P. and Lin, D., 2002, July. Discovering word senses from text. In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 613-619). ACM.

[30] Pradhan, N., Gyanchandani, M. and Wadhvani, R., 2015. A Review on Text Similarity Technique used in IR and its Application. International Journal of Computer Applications, 120(9).

[31] Qian, G., Sural, S., Gu, Y. and Pramanik, S., 2004, March. Similarity between Euclidean and cosine angle distance for nearest neighbor queries. In Proceedings of the 2004 ACM symposium on Applied computing (pp. 1232-1237). ACM.

[32] Rensch, C.R., 1992. Calculating lexical similarity. Windows on bilingualism, pp.13-15.

[33] Witten, I.H. and Frank, E., 2005. Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.