Semantic Similarity Analysis for Paraphrase Identification in Arabic Texts

Adnen Mahmoud
LATICE Laboratory Research Department of Computer Science
University of Monastir, Tunisia
Mahmoud.adnen@gmail.com

Mounir Zrigui
LATICE Laboratory Research Department of Computer Science
University of Monastir, Tunisia
Mounir.zrigui@fsm.rnu.tn

Abstract

Arabic plagiarism detection is a difficult task because of the great richness of Arabic language characteristics of which it is a productive, derivational and inflectional language, on the one hand, and a word can has more than one lexical category in different contexts allows us to have different meanings of the word what changes the meaning of the sentence, on the other hand. In this context, Arabic paraphrase identification allows quantifying how much a suspect Arabic text and source Arabic text are similar based on their contexts. In this paper, we proposed a semantic similarity approach for paraphrase identification in Arabic texts by combining different techniques of Natural Language Processing NLP, such as: Term Frequency-Inverse Document Frequency TF-IDF technique to improve the identification of words that are highly descriptive in each sentence; and distributed word vector representations using word2vec algorithm to reduce computational complexity and to optimize the probability of predicting words in the context given the current center word, which they would be subsequently used to generate a sentence vector representations and after applying a similarity measurement operation based on different metrics of comparison, such as: Cosine Similarity and Euclidean Distance. Finally, our proposed approach was evaluated on the Open Source Arabic Corpus OSAC and obtained a promising rate.

1 Introduction

Plagiarism is defined as the unauthorized use or closer imitation for the language and thought of another author and the representation of them that one's own original work based on a set of rules, such as for example: inadequate referencing, direct copy from one or more sources of the text by displacement of words in a sentence, paraphrase and rewrite texts by presenting other's ideas with different words, and translation by expressing an idea in one language into another one (Abderahman et al, 2016) (Gharavi et al, 2016).

However, the field of paraphrase detection in Arabic texts is a difficult task because of the great variability of morphological and typographical features of Arabic language where a plagiarized text can include more changes: in the vocabulary, or syntactic, and semantic representation of the text compared with other languages such as Latin or English and we also find that a word can have more than one lexical category in different contexts what changes the meaning of the sentence. Nowadays, Arabic paraphrase identification based on semantic similarity analysis between source text and suspicious text is a difficult task in Natural Language Processing NLP of which we examine the similarity degree of a given pair of texts, in varying in different levels such as word, sentence or paragraph (Vo et al., 2015). Thus, many distributional semantic approaches based on the resemblance determination of their signification and their semantic contain (Negre, 2013) have drawn a considerable amount of attention by research community. In this context, our work consists in detecting semantic relatedness between the suspect text and the source text by combining different Natural Language Processing NLP methods to detect paraphrase in Arabic texts by generating word vector representations using word2vec algorithm which they would be combined subsequently to generate sentence vector

representations and thereafter applying a similarity measurement operation. In this paper, we start by present a state of the art in the field of Arabic plagiarism detection in section 2 describing the complexities of Arabic language, on the one hand, and the works that have been proposed in this field in the literature, on the other hand. Thereafter, we detail different phases that make up our proposed method in section 3. Finally, we present the evaluation in section 4 as well as the results obtained and we end by a conclusion and some future works to realize in the field of plagiarism detection especially in Arabic language in section 5.

2 State Of The Art

2.1 Complexities of Arabic Language

Arabic language is very rich of morphological and typographical features (Meddeb et al, 2016) (Zouaghi et al, 2008) which make Arabic semantic analysis a very difficult task for several reasons among which we can mention:

- Arabic language is very rich of morphological features. Thus, Arabic script is cursive whose most letters are tied and written from right to left whose letters change shape depending on whether they appear at the beginning, middle or end of the word, on the one hand, and it consists of: a stem composed by a consonant root and a pattern morpheme; more affixes include time markers, sex and/or number; and enclitics include some propositions, conjunctions, determinants and pronouns. (Meddeb et al, 2016) (Boudhief et al, 2014)
- A word can have more than one lexical category such as: noun, verb, adjective, subject, etc. and can have more than one meaning depending on the context in which it is used (Zrigui et al, 2016) where the identification of some words is very difficult because of the non-capitalization of proper noun, acronyms and abbreviations (Lihioui et al, 2014) as shown in table 1:

Example	Translation	Function
ذهب أحمد إلى الدكان	Ahmed went to the shop	Verb
ذهب هذا الرجل ممتاز	The gold of this man is excellent	Subject

Table 1: Influence of syntactic category on the disambiguation of the word "**ذهب**" (dhhb)

- Inflected language whose lexical units vary in number and in bending such as the number of names or verb tense according to the grammatical relationships which they have with other lexical units. (Boudhief et al, 2014)
- The absence of diacritic marks makes Arabic language more ambiguous (Meddeb et al, 2016) (Zrigui et al, 2016). Therefore, only the diacritics, the occurrence context, and in some cases the grammatical category of the ambiguous word can disambiguate its sense which complicates the automatic processing of Arabic language and especially in its semantic analysis of which there is not a consistent theoretical formalism capable of taking into account all the phenomena encountered in this language (Zouaghi et al, 2012) (Zouaghi et al, 2007) as indicated in table 1 and 2:

Word	Vocalization	Translation
عَمِلَ	amila	Worked
عُمِلَ	omila	was done
عَمَلٌ	amalon	Work
عَمِلَ	amila	Worked

Table 2: Influence of diacritics on the disambiguation of the word '**عمل**' (aml)

- An Arabic word may have several possible divisions such as proclitic, flexive form and enclitic. Thus, clitics stick to nouns, verbs, and adjectives which they relate that makes Arabic language agglutinative, on the one hand, and increase the ambiguity of word segmentation, on the other hand. (Boudhief et al, 2014) (Zouaghi et al, 2012)
- Synonyms are widespread in which there are many words are considered synonyms which require the use of tools of

morphological analysis to find synonyms of a word. (Zrigui et al, 2016)

- The presence of coordination conjunction with a space-free link makes it difficult to distinguish between 'و' as a letter of a word and the word 'و' having the role of conjunction of coordination, on the one hand, and plays an important role in the interpretation of a statement by identifying its proposals, on the other hand. (Lihioui, 2014) (Zouaghi et al, 2007) as illustrated in the following example:

لقد تم إكمال هذا الإنجاز بالحكمة والعمل الدؤوب من أبناء هذه المدينة

"This accomplishment has been completed with the wisdom and hard work of the people of this city."

To conclude, Arabic language is very difficult to treat automatically because of its properties and its morphological, syntactic and semantic specificities that we quoted above and which make also the field of Arabic paraphrase detection difficult because the change of the word order or its meaning in the suspect sentence causes an ambiguity during semantic analysis between the source text and suspect text whose a word can have more than one lexical category in different contexts which allows us to have different meanings of a word what changes the meaning of a sentence.

2.2 Related Work

This section provides an overview on related works that deal with Arabic plagiarism detection especially in paraphrase identification field based on semantic analysis to determine the relatedness between the suspect and source Arabic text documents. Thus, several similarity detection approaches between documents have been proposed in the literature of which there are three types of methods for computing relatedness according to the type of resources that have been used, we distinguish:

- Knowledge-based methods relies on some form of ontology using WordNet which is a well-known knowledge source to compute semantic similarity between words as in (Shenoy et al, 2012) that proposed a semantic plagiarism detection system using ontology mapping where ontologies are a computational model of some domain of the world by describing semantics of terms used in the domain.
- Web-based approach gathers co-occurrence statistics based on the search engine results and used that to compute word relatedness like Point-wise Mutual Information PMI (Niraula et al, 2015). Thus, (Shuka et al, 2016) showed that the use of a web based cross language semantic plagiarism detection approach helps authors and written to secure their files and to make their files sale.
- Corpus-based measurements compute word similarity and relatedness based on word vector representations obtained from a given corpus. Among the most popular methods for inferring word vector representations to select more discriminative features, we can cite:

(Hussein, 2015) showed that Arabic document similarity analysis using N-grams and Singular Value Decomposition SVD can generalize the eigen decomposition of a positive semi definite normal matrix[1]. Also, (Hafeez and Patil, 2017) showed that the author analyzed summary of Chinese expression habits using an adaptive weight of word position algorithm based on TF-IDF to dynamically determining the weight of a word position according to the word position. Thus, it introduced the Vector Space Model VSM and designed comparative experiment under the scene of Chinese document clustering of which TF-IDF-AP algorithm improved a promising results. Moreover, Latent Semantic Analysis LSA algorithm allows to measure similarity between texts which represents the meaning not only of individual words but also of whole passages such as sentences, paragraphs, and short essay (Bihi, 2017), as in (Kenter and Rijke, 2015) that proposed a method for inducing polarity to the document-term matrix before applying LSA which was novel and shown to effectively preserve and generalize the synonymous / antonymous information in the projected space. Also, Latent Dirichlet Allocation LDA which is a probabilistic model can capture polysemy where each word had multiple meanings and used to reduce dimensionality to themes that are useful building blocks for representing a gist of

[1] https://en.wikipedia.org/wiki/Singular_value_decomposition

what a collection contains, statically or over time as in (Yih et al, 2012) in order to support navigability between similar documents via dynamically hyperlinks using an LDA based on cosine similarity measurement.

But nowadays, distributed word vector representation is a branch of machine learning where each word is described by the surrounding context and thereafter a vector is generated automatically containing semantic and syntactic information about the word (Gharavi et al, 2016) of which the learned vectors explicitly encode many linguistic regularities and patterns (Towne et al, 2016). Indeed, distributed word representations in a vector refers as word embeddings where each vector can be located and visualized in multi dimensional space and helps learning algorithms to achieve better performance in Natural Language Processing NLP by grouping similar words and has dwarfed older methods for achieving distributed representations, like: Latent Semantic Analysis LSA (Towne et al, 2016) (Mikolov et al, 2013). In this context, several methods have been proposed, despite that there is little works have been proposed in the field of Arabic paraphrase detection, such as:

(Mikolov et al, 2013) showed that the inclusion of Twitter-based word embeddings using word2vec may yield better tagged sentences when it used to train systems designed for downstream NLP tasks. However, a generic and flexible method for semantic matching of short texts as in (Samuel, 2016) which leveraged word embeddings of different dimensionality obtained by different algorithms (GloVe and word2vec) and from different sources where the purpose was to go from word-level to short-text level semantics by combining insights from methods based on external sources of semantic knowledge with word embeddings. On the other hand, (Prazak et al, 2012) attempt to estimate the similarity score between chunks based upon estimating semantic similarity of individual words and compiling them into one number for a given chunk pair. After, it experimented with word2vec and GloVe to estimate similarity of words and compiled all word similarities in one number that reflected semantics of whole chunks via lexical semantic vectors. Moreover, (Niraula et al, 2015) showed that words relatedness and similarity can be measured by combining word representations, like: LSA, LDA, word2vec and GloVe to complement the coverage of semantic aspects of a word and thus better represent the word than individual representations.

3 Proposed Approach

Paraphrase detection between Arabic documents becomes a very important task in the recent years because of the great variability of Arabic language specificities, on the one hand, and the availability of enormous volume of information over the internet. In this context, we propose a method for Arabic paraphrase detection based on the identification of similarity between source document and suspect document using Natural Language Processing NLP techniques. Thus, our proposed approach allows extracting their semantic similarity to detect paraphrase which can be created by direct copy of sentences, replacement of words with similar ones, and changing the order of words or reconstructing the sentences (Sindhu and Idicula, 2015). Indeed, our proposed approach is composed by three phases, as follows:

1- We begin with a preprocessing phase to extract the relevant information from texts.
2- After, we apply a features extraction phase to extract more discriminant features.
3- Thereafter, a paraphrase detection phase is used to identify the rate of similarity between source document and plagiarized document.

Here is the general architecture of our approach for Arabic paraphrase identification as shown in the following figure 1:

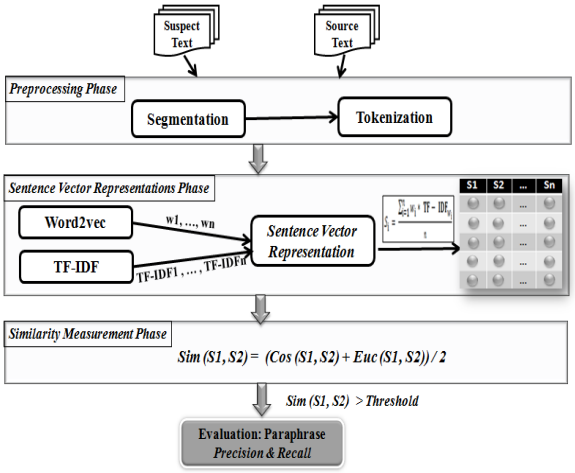

Figure 1. General architecture of Arabic paraphrase detection

3.1 Preprocessing Phase

Our model begins with a preprocessing phase to facilitate further processing by cleaning the Arabic text from noisy information, on the one hand, and to reduce the complexity of Arabic paraphrase identification, on the other hand. So, we proceed by the following steps:

1. We begin by segmenting the source text and suspect text into sentences by identifying their boundaries in order to extract the meaningful information. Among the boundaries used in the literature, we can mention: "," ";" "." ":" "!" "?".

2. After, we try to extract tokens from running text where one Arabic word end and another Arabic word begin by detecting the space between them. (Aliwy, 2012)

3.2 Features Extraction Phase

Term Frequency-Inverse Document Frequency "TF- IDF": TF-IDF is used as a weighting factor in information retrieval and text mining of which it allows the construction of a vector space where each vector represent how a word is important to a document in a collection by the combination between Term Frequency TF (t,d) and Inverse Document Frequency IDF(w). (Shuka et al, 2016)

More formally, given the frequency of the occurrence of term in document d, in order to control the fact that some words are common than others by proceeding as follows (Abderahman, 2016):

- *Term Frequency TF:* is defined as the number of times a term occurs in a document as shown in equation (1). Moreover, a term can appear much more times in long documents than shorter ones since every document is different in length[2].

$$TF(t,d) = \frac{O}{T} \quad (1)$$

Where: O represent the number of times that a term t appears in a document, and T is the number of terms in the document.

- *Inverse Document Frequency IDF***:** is a statistical weight used for measuring the importance of a term in a text document collection. Also, IDF feature is incorporated which reduces the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely as shown in the following equation 2:

$$\text{IDF(t,d)}\log = \frac{|D|}{N} \quad (2)$$

Where: |D| is the total number of documents and N is the number of documents with term t in it.

- *Term Frequency-Inverse Document Frequency TF-IDF*: is calculated for each term in the document by combining TF and IDF as shown in the following equation 3:

$$TF - \text{IDF(t,d,f)} = \text{TF(t,d)} * \text{IDF(t,d)} \quad (3)$$

Word Embeddings "word2vec": Word embeddings are based on a probabilistic feed forward neural network language model to learn a space of continuous word representations in which similar words are expected to be close to each other. Thus, word embeddings allows representing words with low dimensional and dense real-value vectors which capture useful semantic and syntactic features of words (Law and al, 2017), on the one hand, and reducing computational complexity, on the other hand. So, we use word2vec algorithm of which it consists two architectures for learning word embeddings that are less computationally expensive than previous models, such as: Continuous Bag Of Words CBOW and Skip-gram models. Indeed, we use the Skip-Gram model in our work because it showed better performance in Natural Language Processing NLP especially in semantic analysis.

Generally, each input word w is associated with a k-dimensional vector $v_w \in \Re^k$ called the input embedding and each context word w_O is associated with a k-dimensional vector $v'_{w_O} \in \Re^k$ called the output embedding, the probability of observing w_O in the context of w is modeled with a softmax function, as follows in equation 4:

$$P(w_O|w) = \frac{\exp(v'^{T}_{w_O} v_w)}{\sum_{1}^{W} \exp(v'^{T}_{w_i} v_w)} \quad (4)$$

Also, given a sequence of training words $\{w_1, \ldots, w_n\}$ and a larger size of the training context

[2] http://www.tfidf.com/

c in more training examples to lead a higher accuracy at the expense of the training time, the objective of skip gram model is to maximize the average log probability for this sequence, as follows in equation 5 (Mikolov and al, 2013) :

$$\frac{1}{n}\sum_{t=1}^{n}\sum_{-c\le j\le c,\, j\neq 0}\log p(w_{t+j}\,|\,w_t) \qquad (5)$$

Distributed Word Vector Representations "word2vec and TF-IDF": In our work, we try to facilitate the identification of similarity between Arabic texts using word embedding representations and TF-IDF techniques, as follows: On the one hand, we capture semantic properties of words by exploiting vectors as word representation in a multi dimensional space using word2vec algorithm based on skip gram model to optimize the probability of predicting words in the context given the current center word and overcoming the problem of the data sparsity problem. On the other hand, we use TF- IDF technique in order to improve the identification of words that are highly descriptive in each sentence of which TF- IDF value is used as a weighting factor to increase proportionally the number of times a word appears in the document but it is counteracting by the frequency of the word in the corpus.

More formally, given a sentence S composed by n words, our distributed sentence representations of suspect text and source text is composed by the following steps:

1. We begin by determining the Term Frequency - Inverse Document Frequency $TF-IDF \in \Re$ of each word in the sentence. The output of this step is a set of values representing the importance of each word in the text, as follows in equation 6:

$$\text{TF-IDF}_{w1,...,n} = \text{TF-IDF}_{w1},\cdots,\text{TF-IDF}_{wn} \qquad (6)$$

2. Then, we learn word embeddings using word2vec algorithm where the i-th word in the sentence S is mapped into a single vector $w_i \in \Re^k$ represented by a column w_i in a matrix M of size n*k. At the end of this operation, we have a sequence of n word vector representations of the sentence S, as follows in equation 7:

$$w_{1:n} = w_1, w_2, \cdots, w_n \qquad (7)$$

3. The word-level skip-gram model predicts the context of words given the current word vector but our goal is the prediction of the context of sentences given the vector representation of the current sentence vector. (Peng and Gildea, 2016)

So, we proceed as follows: we calculate an average of all word vector representations w_i extracted from W for each sentence S_i composed by n words, as follows in equation 8:

$$S_i = \frac{\sum_{i=1}^{n} w_i * TF-\text{IDF}_{wi}}{n} \qquad (8)$$

At the end of this step of sentence vector representations, each sentence Si will be mapped into a single vector represented by a column in a matrix V, which will be used in subsequent processing, as follows in equation 9:

$$V_{1:m} = S_1, \cdots, S_m \qquad (9)$$

Where: m is the number of sentences in the text.

3.3 Arabic Texts Similarity Measurement

Our goal in this study is how identify the rate of similarity between Arabic texts to conclude that there is a paraphrase between them by combining different metrics of comparison to prove our proposed approach. So, given the sentence vector representations of suspect text S_1 and source text S_2 of dimension k, we compare each sentence of suspect document with all sentences of source document, as follows:

1. We identify the semantic relation between suspect and source sentence using Cosine Similarity based on the calculation of the number of similar words that exist in source sentence S_1 and suspect sentence S_2 to determine the score of similarity between them where the Cosine Similarity is measured using word vectors (Alaa et al, 2016), as follows in equation 10:

$$\text{Cos}(S_1, S_2) = \frac{S_1 \cdot S_2}{\|S_1\| \cdot \|S_2\|} = \frac{\sum_{i=1}^{k} S_{1i}\ S_{2i}}{\sqrt{\sum_{i=1}^{k} S_{1i}^2}\sqrt{\sum_{i=1}^{k} S_{2i}^2}} \qquad (10)$$

2. After, we use the Euclidean distance as another similarity measure which calculates the similarity between two documents as the distance between their vectors representations reduced to a single point (Negre, 2013) as follows in equation 11:

$$Euc(x,y) = \sqrt{\Sigma \left| x_i - y_j \right|^2} \quad (11)$$

So, our proposed similarity method is based on the semantic similarity of sentences in Arabic texts to determine the degree of semantic relatedness between them by combining two methods, such as: Cosine Similarity *Cos* and Euclidean Distance *Euc*, as follows in equation 12:

$$Sim_{Comb}(S_1, S_2) = \frac{Cos(S_1, S_2) + Euc(S_1, S_2)}{2} \quad (12)$$

If the result we found from (12) has also exceeded a threshold α, then, we find that there is actually plagiarism (paraphrase) between the source document and suspect document. Otherwise, it is considered to be not plagiarized (not paraphrase).

At the end of this step, we obtain a vector which contains different scores of similarity according to the suspect text document sizes until reaching the source document size.

4 Results and Discussion

Open Source Arabic Corpora OSAC[4] includes 22,429 text documents where each text document belongs to 1 of 10 categories such as: Economics, History, Entertainments, Education & Family, Religious and Fatwas, Sports, Heath, Astronomy, Low, Stories, Cooking Recipes). Indeed, the evaluation of our proposed approach is carried out on a collection of historical documents contains 3233 text documents of the Open Source Arabic Corpora (OSAC)[3].

The parameters we used and which made our approach efficient are:

- The word vector representations using word2vec based on Skip-gram model are checked in a matrix of size n*k. In our case, we used more than 350 millions words extracted from Wikipedia and we fixed k at 5 which represent the number of synonyms according to each word context where two words before the word in the middle target and two words after.
- The experiments of this study to identify paraphrase between Arabic texts included the implementation of two combined methods, which are: Cosine Similarity *Cos* and Euclidean Distance *Euc*.
- Two sentences are considered as plagiarism (paraphrase), if they pass the threshold (α) between the result of our proposed method for similarity detection $Sim_{Comb}(S_1, S_2)$ whose the threshold was fine-tuned by several trials on the training corpus and the results achieved when α = 0.3.

Each method was tested individually and the combination method gave us the final result of our proposed method as shown in the following table:

Proposed Approaches	Precision	Recall
TF- IDF + Sim_{Comb}	0.81	0.79
Word2vec + Sim_{Comb}	0.83	0.81
Final Combination: word2vec + TF- IDF + Sim_{Comb}	0.85	0.84

Table 3: Results of paraphrase identification approaches

To conclude, the combination between distributed word vector representations and TF-IDF method have shown good result when we applied each measure of comparison that we cited above (Cosine Similarity and Euclidean Distance), and especially when we have combined them with a set of different measures of similarity (Cosine Similarity and Euclidean Distance) of which a promised plagiarism detection rate was obtained in terms of precision and recall.

5 Conclusion and Future Works

We proposed a semantic textual similarity approach for paraphrase identification in Arabic texts based on the combination of different Natural Language Processing NLP such as: TF-IDF technique to improve the identification of words that are highly descriptive in each sentence, and distributed word vector representations using word2vec algorithm to reduce computational complexity and to optimize the probability of predicting words in the context which they would be subsequently used to generate a sentence vector representations and after applying a similarity measurement operation based on the combination of different metrics of comparison such as: Cosine Similarity and Euclidean Distance. Finally, our proposed approach was evaluated on the Open

[3] https://sites.google.com/site/motazsite/corpora/osac

Source Arabic Corpus OSAC and obtained a promising rate. Despite the promising results that we have obtained using our proposed approach, several improvements will be applied in our method later on, such as : the use of a Convolutional Neural Network CNN to improve the capability to capture statistical regularities in the context of sentences, on the one hand, and we will try to combine word vector representations to improve the similarity measure and to improve the weakness of each method, like: Latent Semantic Analysis LSA, Latent Dirichlet Allocation LDA and distributed representation of words word2vec.

References

Abderahman Y. A., Khalid A. and Osman I. M.. (2016). *A survey of plagiarism detection for Arabic documents, International Journal of Advanced Computer Technology IJACT*, volume 4, n. 6, pp. 34-38.

Alaa Z., Tiun S. and Abdulameer M. H. (2016). *Cross-language plagiarism of Arabic-English documents using linear logistic regressing, Journal of Theoretical and Applied Information Technology, 2005 - 2015 JATIT & LLS*, Volume 83. No.1, , pp. 20-33.

Aliwy A. H. (2012). *Tokenization as Preprocessing for Arabic Tagging System, International Journal of Information and Education Technology*, volume 2, No. 4, pp. 348-353.

Bihi A. (2017). *Analysis of similarity and differences between articles using semantics*, Université Malardalen, Sweden.

Boudhief A., Maraoui M. and Zrigui M. (2014). *Elaboration of a model for an indexed base for teaching Arabic language to disabled people, 6th International Conference on CIST*, pp110-116.

Gharavi E., Bijari K., Zahirnia K. and Veisi H.. (2015). A *deeplearning approach to Persian plagiarism detection*, India.

Hafeez S. and Patil B. (2017). *Using Explicit Semantic Similarity for an Improved Web Explorer with ontology and TF-IDF, International Journal of Advance Scientific Research and Engeneering Trends*, Volume 2, Issue 7, pp. 171-173.

Hussein A. S. 2017. *Arabic document similarity analysis using N-gram and Singular Value Decomposition, 9th International Conference on Research Challenges in Information Science RC IS.*

Kenter T. and Rijke M. D. (2015). *Short text similarity with word embeddings, International Conference on Information and Knowledge Management CIKM'15*, Australia.

Law J., Zhuo H. H., He J. and Rong E. (2017). *LTSG: Latent Topical Skip-Gram for Mutually Learning Topic Model and Vector Representations*, Cornell University Library, Coputer Science, Computation and language, United States.

Lihioui C., Zouaghi A. and Zrigui M. (2014). *Towards a hybrid approach to semantic analysis of spontaneous Arabic speech, International Journal of Computational Linguistics and Applications*, volume 5, n. 2, pp. 165-193.

Meddeb O., Maraoui M. and AlJawerneh S. (2016). *Hybrid modeling of an offline arabic handwriting recognition system AHRS, International Conference on Engineering & MIS ICEMIS*, Maroc

Mikolov T., Sustskever I., Chen K., Corrado G., and Dean J. (2013). *Distributed representations of words and phrases and their compositionality, Neural Information Processing Systems NIPS*, United States

Vo N. P. A., Magnolini S. and Popescu O. (2015). *Paraphrase identification and semantic similarity in Twitter with simple features, Proceedings of Social NLP* ,pp. 10-19, Colorado.

Negre E. (2013). *Comparaison de textes: quelques approche , Cahier du LAMSADE 338, Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision UMR 7243*, Paris.

Niraula N. B., Gautam D., Banjadae R., N. Maharjan, and Rus V. (2015). *Combining word representations for measuring word relatedness and similarity, Twenty Eight International Florida Artificial Intelligence Research Society Conference*, Florida

Prazak O., Steinberger D., Konopik M., and Brychain T. (2012). *Interpretable semantic textual similarity with distributional semantics for chunks, Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp1212-1222.

Peng X. and Gildea D. (2016*). Exploring phrase-compositionality in skip-gram models, Cornell University Library, Computer Science, Computation and language*, United States.

Samuel D. S. (2016). *On the use of vector representation for improved accuracy and currency of Twitter POS Tagging*, Dalhousie University, Halifax, Nova Scotia.

Shenoy M. K., Set D. C. and Achrya U. D. (2012). *Semantic plagiarism detection system using ontology mapping, Advanced Computing: An International Journal ACIJ*, volume 3, n. 3, pp. 59-62.

Shuka V., Khan F. and Mody K. (2016). *Plagiarism detection for document, International Journal on Recent and Innovation Trends in Computing and Communication*, volume 4, issue 2, pp. 175-178.

Sindhu L., and Idicula S. M. (November 2015). *SRL based plagiarism detection system for Malayalam documents, International Journal of Computer Science Issues IJCSI*, volume 12, issue 6, pp. 91-97.

Towne W. B., Rosé C. P. and Herbsleb J. D. (2016). *Measuring similarity: LDA and Human Perception, ACM Transaction on Intelligent Systems and Technology*, pp. 1-29, volume 7, n. 2.

Yih W. T., Zweig G. and Platt J. C. (2012). *Polarity inducing Latent Semantic Analysis, Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 1212–1222, Korea.

Zouaghi A., Zrigui M., Ben Ahmed M. and Antoniadis G. (2007). *L'influence du contexte sur la compréhension de la parole spontanée, Proceedings de la Conference Traitement Automatique de La Langue Naturelle TALN'07*

Zouaghi A., Zrigui M. and Antoniadis G. (2008). *Compréhension automatique de la parole arabe spontanée , Traitement Automatique des Langues*, Beljique.

Zouaghi A., Marhbène L. and Zrigui M. (2012). *A hybrid approach for Arabic word sense disanbiguation, Internatonal Journal of Computer Processing of Languages*, volume 24, n.2, pp. 133-151.

Zrigui S., Zouaghi A., Ayadi R., Zrigui M. and Zrigui S. (2016). ISAO*: An intellgent system of opinion analysis, Research in Computing 110* , pp. 21-31.