ISSN 2007-9737

Sentence Similarity Computation based on WordNet and VerbNet

Wafa Wali[1], Bilel Gargouri[1], Abdelmajid Ben Hamadou[2]

[1] MIRACL Laboratory, FESGS-Sfax,
Tunisia

[2] MIRACL Laboratory, ISIMS-Sfax,
Tunisia

wafa.wali@fsegs.rnu.tn, bilel.gargouri@fsegs.rnu.tn, abdelmajid.benhamadou@isimsf.rnu.tn

Abstract. Sentence similarity computing is increasingly growing in several applications, such as question answering, machine-translation, information retrieval and automatic abstracting systems. This paper firstly sums up several methods to calculate similarity between sentences which consider semantic and syntactic knowledge. Second, it presents a new method for the sentence similarity measure that aggregates, in a linear function, three components: the Lexical similarity Lexsim including the common words, the semantic similarity SemSim using the synonymy words and the syntactico-semantic similarity SynSemSim based on common semantic arguments, notably, thematic role and semantic class. Concerning the word-based semantic similarity, a measure is computed to estimate the semantic degree between words by exploiting the WordNet ”is a” taxonomy. Moreover, the semantic argument determination is based on the VerbNet database. The proposed method yielded competitive results compared to previously proposed measures and with regard to the Li’s benchmark, which shown a high correlation with human ratings. Furthermore, experiments performed on the Microsoft Paraphrase Corpus showed the best F-measure values compared to other measures for high similarity thresholds.

Keywords. Sentence similarity, syntactico-semantic similarity, thematic role, semantic class, WordNet, VerbNet.

1 Introduction

Sentence similarity measures have become an important task in several applications, such as information retrieval, text classification, document clustering, question answering, machine translation, text summarization and others.
A number of different metrics for computing the semantic similarity of sentences have been devised. Some of these outlined syntactic methods, to measure the similarity between sentences are based on the co-occurring words between sentences, such as [16] or on the similar syntactic dependencies, like [13]. These methods of assessing the sentence similarity, however, do not take into account the semantic information, such as some words that have different meanings (cancer may be an animal or a disease), synonymous words such as, car/automobile, etc may not be recognized.

In order to fight against the weaknesses of syntactic methods, others investigated approaches to compute sentence similarity based on semantic information using human-constructed lexical resources, such as WordNet like [18], [5] and [11] and/or trained by collecting the statistics of each word from unannotated or highly-annotated text corpora, such as [8] and [14].

However, these sentence similarity methods based on semantic information do not directly induce a real similarity score. For this reason, some approaches estimate the similarity between sentences based on syntactic and semantic information, called hybrid methods, such as [12] and [6] that take account of the semantic information and word order information implied in the sentence, [19] that considers multiple features measuring the word-overlap similarity

doi: 10.13053/CyS-21-4-2853

and syntactic dependency and [21] that takes account of synonymy relations between the word sense and the semantic predicate based on the LMF standardized Arabic dictionary [9]. Indeed, the authors estimate to compute the sentence similarity for Arabic Language. The proposed measure achieved a better correlation of the order of 0.92.

In this paper, we are interested in generalizing the proposal of [21] on the English language using the WordNet and VerbNet databases. Indeed, WordNet is used to having the synonyms of sentence 's words and we benefit also of the VerbNet database, because it is considered as the best lexical resource that gives the adequate properties of semantic arguments in terms of semantic class and thematic role.

This paper is outlined as follow. Firstly, we are concerned about presenting the mostly known hybrid approaches. Secondly, we describe the proposed method to measure the sentence similarity based on semantic arguments, notably, the semantic class and the thematic role. Then, we present the benchmarks used for studying the performance of our method compared to competitive methods. After that, we describe and interpret the obtained results using the Li et al. dataset [12] and the Microsoft Paraphrase Corpus [4]. And finally, we provide a conclusion and perspectives for future research.

2 Hybrid Similarity Measures

Several hybrid methods have already been proposed to measure similarity between sentences. Related work can roughly be classified into three major categories: word order-based similarity, part of speech-based similarity and syntactic dependency-based similarity.

2.1 Word Order-based Similarity

A method for measuring the semantic similarity between sentences, based on semantic and word order information was presented in [12], named STATIS. First, semantic similarity is derived from a lexical knowledge base and a corpus. Second, the proposed method considers the impact of the word order on sentence meaning. The derived word order similarity measures the number of different words as well as that of the word pairs in a different order.

The authors of [6] presented a method and called it Semantic Text Similarity (STS). This method determines the similarity of two sentences from a combination between semantic and syntactic information. They considered two mandatory functions (string similarity and semantic word similarity) using corpus-based measures to calculate semantic similarity. Moreover, they took into account an optional function as common-word order similarity to compute the syntactic similarity.

2.2 Part of Speech-based Similarity

The authors of [1] presented an approach that combines corpus-based semantic relatedness measure over the whole sentence along with the knowledge -based semantic similarity scores that were obtained for the words falling under the same syntactic roles in both sentences. All the scores, which are features, were fed to machine learning models, like linear regression, and bagging models to obtain a single score giving the degree of similarity between sentences.

A method named FM3S was introduced by [20] that estimates the sentence similarity between sentences based firstly on the semantic similarity of their words through the separate processing of verbs and nouns and secondly the common word order. Indeed, the method exploits an IC (Information Content)-based semantic similarity measure in the quantification of noun and verb semantic similarity and it is the first which includes the compound nouns and verb tenses in the similarity measure between two sentences.

2.3 Syntactic Sependency-based Similarity

Oliva et al. [17] reported on a method called SyMSS to compute sentence similarity. The method considers that the meaning of a sentence is made up of the meanings of its separate words and the structural way the words are combined. In fact, the semantic information is obtained from a WordNet lexical database and the syntactic information is obtained through a deep parsing process to find the syntactic structure of each sentence. With this syntactic information, SyMSS measures the semantic similarity between terms with the same syntactic role.

The authors of [3] introduced a method to assess the semantic similarity between sentences, which relies on the assumption that the meaning of a sentence is captured by its syntactic constituents and the dependencies between them. They obtain both the constituents and their dependencies from a syntactic parser. The algorithm considers that two sentences have the same meaning if there is a good mapping between their chunks and the chunk dependencies in one text are preserved in the other. Moreover, the algorithm considers that every chunk has a different importance with respect to the overall meaning of a sentence, which is computed based on the information content of the words in the chunk.

3 The Proposed Method

The suggested method aggregates between three modules, namely lexical similarity, semantic similarity and syntactic-semantic similarity. Figure 1 gives an overview of the suggested method.

Indeed, the lexical similarity is computed after removing punctuation signs and determining the lemma of each word using a stemmer, based on common lemmas between sentences using the Jaccard coefficient [7]. Indeed, the choice of the Jaccard coefficient is explained by their simplicity. The lexical similarity computation function between sentences S1 and S2 is defined as follows:

$$LexSim(S1,S2) = \frac{WS1 + WS2 - CW(S1,S2)}{CW(S1,S2)}, \tag{1}$$

where WSi refers to the number words of a sentence Si and CW(S1,S2) is the common word number between a pair of sentences.

The semantic similarity is measured based on semantic vectors. For this module, the authors determined, firstly, the joint word set that contains the distinct words between sentences. Then, each sentence is presented by the use of a joint word set which is called semantic vector as follow. Each element in this vector corresponds to a word in a joint word set. Thus, the value of an element of the semantic vector is determined by the semantic similarity of the corresponding word to word in the sentence. Let us take sentence S1 as example:

— Case 1: if W_i (the word in sentence S1) appears in joint word set, then the cell value of semantic vector of S1 equals 1.

— Case 2: if W_i (the word in the sentence S1) does not appear in the joint word set, then a semantic similarity score is calculated between the word W_i and each word of the joint word set.

In fact, this semantic similarity score between words is calculated based on the common synonymy between two words with the assistance of the database WordNet [15] and using the Jaccard coefficient [7]. The semantic similarity between two words W_i and W_j is computed as follows:

$$SSim(W_i,W_j) = \frac{SW_i + SW_j - CS(W_i,W_j)}{CS(W_i,W_j)}, \tag{2}$$

where SW_i, SW_j is the synonymy number of each word and $CS(W_i,W_j)$ returns the synonymy common number between W_i and W_j. Thus, the most similar word in S1 to W_i is that with the highest similarity score.

The last step in this module is to calculate the semantic similarity between sentences based on the generated semantic vectors corresponding to sentences S1 and S2 using the cosine similarity. The semantic similarity between two sentences S1 and S2 is computed as follows:

$$SemSim(S1,S2) = \frac{\sum_{i=0}^{n} V1_i \times V2_i}{\sqrt{\sum_{i=0}^{n} V1_i^2}\sqrt{\sum_{i=0}^{n} V2_i^2}}, \tag{3}$$

where $V1_i$ and $V2_i$ are the components of vectors V1 and V2 respectively to S1 and S2. Using the illustrative example below, table 1 shows the computing mode of SemSim(S1,S2):

$$Sim(S,DS) = \max(\forall S_j \in DsSim(S,DS)) \sum . \tag{4}$$

S1=”The **car** was **destroyed** by a **tree**”.

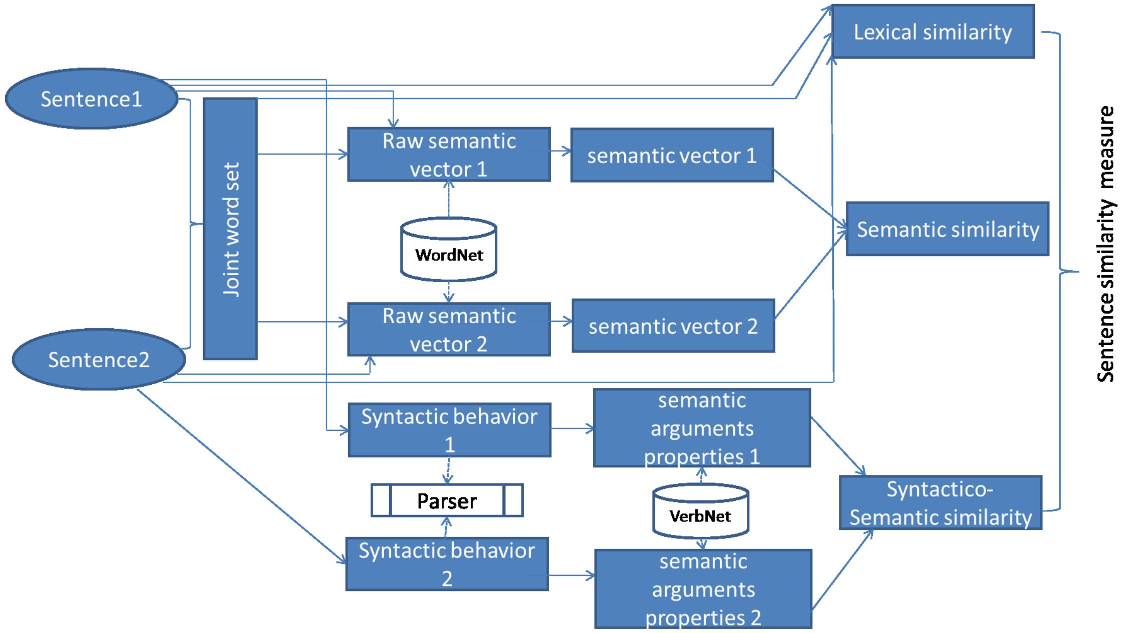

Fig. 1. The proposed sentence similarity measure

S2="The falling **branch crumpled** the **automobile**".

The verbs "was destroyed, crumpled" are reduced to lemma form "destroy, crump". The word "failing" is eliminated because it is considered as stop word.

Moreover, the syntactico-semantic similarity is computed based on the common semantic arguments between sentences in terms of thematic role and semantic class using the Jaccard coefficient [7]. Indeed, this idea is considered original because it has not been employed in former research in the literature. The process of syntactico-semantic computation starts by determining the syntactic structure for each sentence using Stanford parser [2].

Then, the semantic predicate is defined for each sentence by means of a linguistic expert. The determination process of semantic arguments takes the relation between syntactic structure and semantic predicate into consideration using VerbNet. In fact, VerbNet [10] is the largest on-line verb lexicon currently available for English. It is organized into verb classes extending Levin classes through refinement and addition of subclasses to achieve syntactic and semantic coherence among members of a class.

Each verb class in VerbNet is completely described by thematic roles, selectional restrictions on the arguments, and frames consisting of a syntactic description and semantic predicates with a temporal function. The syntactico-semantic similarity computation function between sentences S1 and S2 is determined as follows:

$$SSSim(S1, S2) = \frac{SArgS1 + SArgS2 - CSArg(S1, S2)}{CSArg(S1, S2)}, \tag{5}$$

where CSArg Si refers to the number of semantic arguments of sentence Si and CSArg (S1,S2) is the common semantic argument number between a pair of sentences.

Having applied the procedure on the previous page, the semantic arguments for S1 and S2 are respectively SArg S1 and SArg S2. For the example of the sentence pair, we have:

1. SArg S1= (patient, inanimate), (instrument, inanimate),

Table 1. Example of computing SemSim(S1,S2)

		Car	destroy	tree	branch	crump	Automobile
S1	Car	1					1
	destroy		1				
	tree			1			
	V1	1	1	1	0	0	1
S2	branch				1		
	crump					1	
	automobile	1					1
	V2	1	0	0	1	1	1
SemSim(S1,S2)=0.5							

2. SArg S2 = (Experiencer, inanimate), (theme, inanimate),

3. SynSemSim(S1,S2)=0.

Finally, the similarity between two sentences is based on an aggregation based on three components defined above, namely LexSim, SemSim and SynSemSim. The aggregation combines between them in a linear function using the supervised learning especially the SMO (Sequential Minimal Optimization) algorithm to turn the contribution of each component in the final score:

$$Sim(S1, S2) = \alpha * A + \beta * B + \gamma * C, \qquad (6)$$

where A designed LexSim(S1,S2),B designed SemSim(S1,S2) and C designed SynSemSim(S1,S2).

The component SemSim and SynSemSim have main contribution in the final score because β and $\gamma > \alpha$.

4 Assessment Benchmarks

The sentence similarity measure proposed in the above section is evaluated in two ways. The first way includes the study of correlation between the similarity values to sentence pairs judged by the experts. The second way involves the integration of these sentence similarity measures in a particular application, such as paraphrase determination.

4.1 Benchmarks

In this subsection, we present the benchmarks that are used by the previous measures listed in the second section, such as Li et al. dataset [12] and Microsoft paraphrase corpus (MSPC) [4].

4.1.1 Li et al. Dataset

This dataset, which was created by Li et al., [12] takes a set of 65 noun pairs, replaces the nouns with their dictionary definitions collected from the Collins Cobuild Dictionary, and has 32 human participants that rate the similarity in the meaning of each of the sentence pair on a scale of 0.0 to 4.0. When the similarity scores average the distribution scores, they are heavily skewed toward the low similarity end of the scale, with 46 pairs rated from 0.0 to 0.9 and 19 pairs rated from 1.0 to 4.0 to obtain an uneven distribution across the similarity range. A subset of 30 sentence pairs was selected, consisting of all the sentence pairs rated from 1.0 to 4.0, and 11 taken at equally spaced intervals from the 46 pairs rated from 0.0 to 0.9. Unlike the dataset described above, in which the task is binary classification, this dataset has been used to compare correlation with the human ratings.

4.1.2 MicroSoft Paraphrase Corpus (MSPC)

The Microsoft Research Paraphrase corpus [4] consists of 5801 sentence pairs, 3900 of which were labeled as paraphrases by human annotators. The MSPC corpus is divided into training set (4076 sentences) and a test set (1725 pairs). The number of average words per sentence (sentence length) for this corpus is 17. MSPC is by far the largest publicly available paraphrase annotated corpus, and has been used extensively over the last decade.

4.2 Evaluation Metrics

In this subsection, we present the metrics used to evaluate the performance of the hybrid approaches, such as Pearson's coefficient and Spearman's coefficient that are used in Li et al. dataset [12] and recall, precision and f-measure that are used in Microsoft Research Paraphrase corpus [4].

4.2.1 Pearson's Coefficient

The Pearson's coefficient measure indicates how well the results of such a measure are similar to human judgements. Pearson's coefficient, called r, is computed as follows:

$$r = \frac{n(\sum x_i y_i) - (\sum x_i) * (\sum y_i)}{\sqrt{n(\sum x_i^2) - (\sum x_i)^2}\sqrt{n(\sum y_i^2) - (\sum y_i)^2}}, \quad (7)$$

where X_i corresponds to the i-th element in the list of human judgement, Y_i corresponds to the i-th element in the list of sentence similarity computed values and n corresponds to the pair sentence number.

4.2.2 Spearman's Coefficient

The classification is produced based on the sentence similarity measure compared to the one produced on the basis of human judgments. Spearman's is computed as follows:

$$p = 1 - \frac{6\sum di^2}{n(n^2-1)}, \quad (8)$$

where d_i corresponds the difference of the ranks of x_i and y_i.

4.2.3 Recall

The recall measure is calculated as follows: the number of the determined relevant paraphrases divided by the existing number of paraphrases:

$$Recall = \frac{D}{E}, \quad (9)$$

where D corresponds to number of pairs correctly annotated as paraphrases by the measure and E designed Number of parapharses in the dataset.

4.2.4 Precision

The precision measure is based on the number of the determined relevant paraphrases divided by the number of returned paraphrases:

$$Precision = \frac{D}{F}, \quad (10)$$

where F corresponds to number of pairs annotated as paraphrases by the measure.

4.2.5 F-measure

F-measure combines the precision and the recall and expresses a trade-off between those two measures:

$$F - measure = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (11)$$

5 Experiments and Results

This section presents the obtained results for the employed benchmarks, Li et al. dataset and MSPC. All the experiments are performed using the α=0.2, β=0.45 and γ=0.35 parameters, which are empirically determined with respect to equation (5).

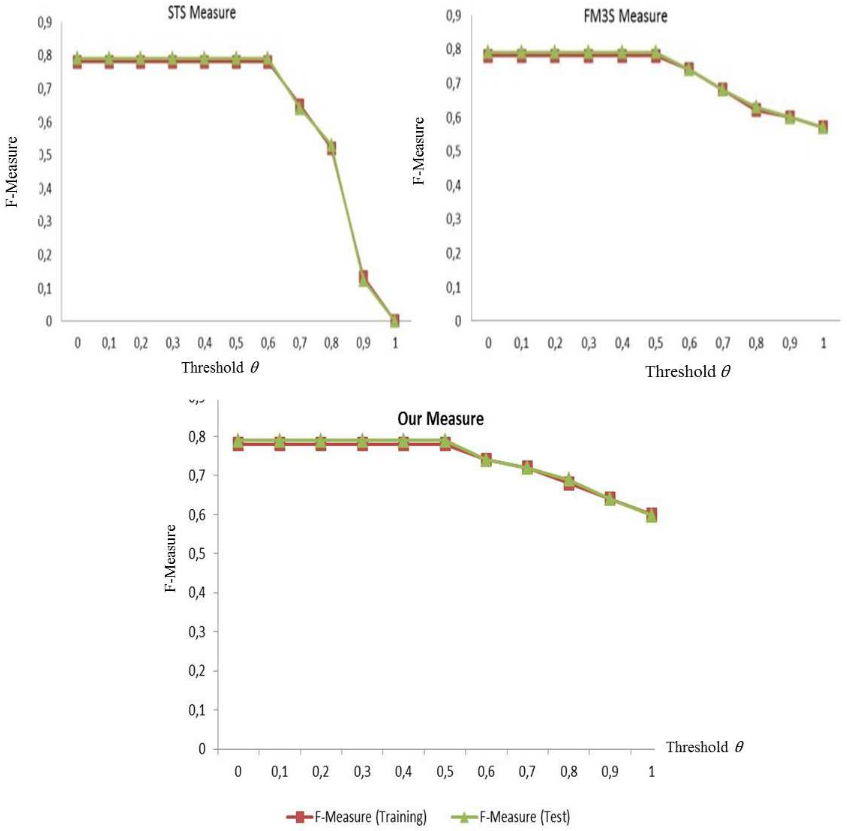

Fig. 2. Comparison between STS, F3MS and our proposal measures using F-measure values and varying the threshold θ on the MSPC dataset

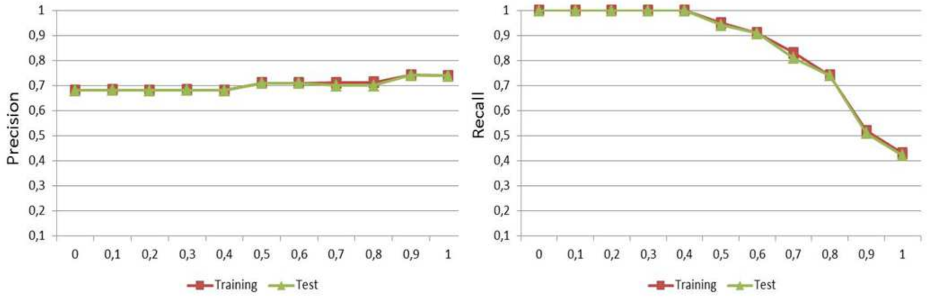

Fig. 3. Precision and recall of our measure applied on the MPSC dataset

5.1 Experiment with Li et al. Dataset

Table 2 shows the correlation coefficients of Pearson (r) and Spearman (p) obtained from different measures on the Li et al. dataset. Our method achieved the best results compared to other computational methods.

Our proposal can reach results that approximates to 100%, but unfortunately 28 sentence pairs in the Li et al. dataset contain the verb

”to be”, which negatively affects the contribution of SynSemSim(S1,S2) component to the final similarity score.

5.2 Experiment with MSPC

Due to the deficiency of published research presenting results on the MPSC dataset, our method is compared only to STS [6] and FM3S [20] measures. Accordingly, this method provides results at the threshold $\theta \in$[0.7, 1.0] as shown figure 2.

Our proposal yielded the competitive results compared to ”FM3S” method for the Training data and for the Test data.

Moreover, the F-measure values obtained with the ”STS” approach for the same interval tend towards 0. This provides further support for the advanced efficiency of our method.

The results illustrated in figure 3 show the precision and recall of the proposed measure, respectively. In fact, precision reached a peak with θ=0.9, showing a value of 0.742 for the Training and Test datasets. This demonstrates that the sentence pairs judged as highly similar by our measure are qualified as paraphrases in the MSPC dataset.

Table 2. Results obtained using the Li et al. dataset

	r	p
STATIS [12]	0.81	0.81
STS [6]	0.85	0.85
SyMSS [17]	0.76	0.71
FM3S [20]	0.76	0.79
Our proposal	**0.87**	**0.87**

6 Conclusion and Future Works

The proposed measure determines the similarity between two sentences regarding the semantic and syntactico-semantic information they contain. The aggregate function Sim(S1,S2) presented in equation (5), contains the common words, Synonymy words and the properties of semantic arguments in a linear way. Our proposal is based on the word semantic similarity. It exploits the WordNet in order to determine the synonymy of the words using the Jaccard measure. The word semantic similarity measure is arranged in semantic vectors so that it has for the sentence semantic similarity using the Cosine similarity.

The proposed method also, takes the common semantic argument properties, notably, the semantic class and the thematic role using VerbNet dataset. Our method yielded competitive results compared to other computational methods, such as of Li et al. dataset.

For the paraphrase recognition task, our proposal outperforms other measures, mainly at a high threshold $\theta \in$ [0.7,1]. These results provide a strong support for the utility of a number of sentence features, such as semantic arguments and properties in the process of computing sentence similarity.

Due to the promising performance of this measure, it can be applied in other applications, such as plagiarism detection.

References

1. **Aggarwal, N., Asooja, K., & Buitelaar, P. (2012).** Deri&upm: Pushing corpus based relatedness to similarity: Shared task system description. *Proceedings of the First Joint Conference on Lexical and Computational Semantics, Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, Association for Computational Linguistics, pp. 643–647.
2. **Chen, D. & Manning, C. D. (2014).** A fast and accurate dependency parser using neural networks. *EMNLP*, pp. 740–750.
3. **Ştefănescu, D., Banjade, R., & Rus, V. (2014).** A sentence similarity method based on chunking and information content. *LNCS, Proceedings of the 15th International Conference on Computational Linguistics and Intelligent Text Processing*, Vol. 8403, Springer-Verlag New York, Inc., pp. 442–453.
4. **Dolan, B., Quirk, C., & Brockett, C. (2004).** Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. *Proceedings of the 20th international conference on Computational Linguistics*, Association for Computational Linguistics, p. 350.

5. **Hirst, G. & St-Onge, D. (1997).** *Lexical chains as representations of context for the detection and correction of malapropisms*.

6. **Islam, A. & Inkpen, D. (2008).** Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10.

7. **Jaccard, P. (1901).** *Etude comparative de la distribution florale dans une portion des Alpes et du Jura*. Impr. Corbaz.

8. **Jiang, J. J.& Conrath, D. W. (1997).** Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*.

9. **Khemakhem, A., Gargouril, B., Hamadou, A. B., & Francopoulou, G. (2016).** Iso standard modeling of a large Arabic dictionary. *Natural Language Engineering*, 22(6):849–879.

10. **Kipper, K., Korhonen, A., Ryant, N., & Palmer, M. (2006).** Extending VerbNet with novel verb classes. In *Proceedings of LREC*, volume 2006, page 1. Citeseer.

11. **Leacock, C. & Chodorow, M. (1998).** Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database*, 49(2):265–283.

12. **Li, Y., McLean, D., Bandar, Z. A., O'shea, J. D., & Crockett, K. (2006).** Sentence similarity based on semantic nets and corpus statistics. *IEEE transactions on knowledge and data engineering*, 18(8):1138–1150.

13. **Mandreoli, F., Martoglia, R., & Tiberio, P. (2002).** A syntactic approach for searching similarities within sentences. *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, CIKM '02, New York, NY, USA, ACM, pp. 635–637.

14. **Mihalcea, R., Corley, C., & Strapparava, C. (2006).** Corpus-based and knowledge-based measures of text semantic similarity. *AAAI*, volume 6, pp. 775–780.

15. **Miller, G. A. (1995).** WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

16. **Nirenburg, S., Domashnev, C., & Grannes, D. J. (1993).** Two approaches to matching in example-based machine translation. *Proc. of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-93)*, pp. 47–57.

17. **Oliva, J., Serrano, J. I., del Castillo, M. D., & Iglesias, Á. (2011).** Symss: A syntax-based measure for short-text semantic similarity. *Data & Knowledge Engineering*, 70(4):390–405.

18. **Resnik, P. (1995).** Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007.*

19. **Šarić, F., Glavaš, G., Karan, M., Šnajder, J., & Bašić, B. D. (2012).** Takelab: Systems for measuring semantic text similarity. *Proceedings of the First Joint Conference on Lexical and Computational Semantics, Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, Association for Computational Linguistics, pp. 441–448.

20. **Taieb, M. A. H., Aouicha, M. B., & Bourouis, Y. (2015).** Fm3s: Features-based measure of sentences semantic similarity. *International Conference on Hybrid Artificial Intelligence Systems*, pp. 515–529.

21. **Wali, W., Gargouri, B., & Hamadou, A. B. (2016).** Using sentence semantic similarity to improve LMF standardized Arabic dictionary quality. In *Computational Linguistics and Natural Language Processing*.

Article received on 26/12/2016; accepted on 23/02/2017.
Corresponding author is Wali Wafa.