Sentence Semantic Similarity based on Word Embedding and WordNet

Mamdouh Farouk
omputer Science Department
Faculty of Computers and Information
Assiut University, Assiut, Egypt
Email: mamfarouk@aun.edu.eg

***Abstract*—Semantic similarity between sentences is a crucial task for many applications. The emerging of word embedding encourages calculating similarity between words and between sentences based on the new semantic word representation. On the other hand, WordNet is widely used to find semantic distance between sentences. This paper combines the using of pre-trained word vector and WordNet to measure semantic similarity between two sentences. In addition, word order similarity is applied to make the final similarity more accurate. The proposed approach has been implemented and tested using standard datasets. Experiments show that presented methods achieves better results comparing with other approaches previously proposed to measure sentence similarity.**

I. Introduction

Semantic similarity between words and between sentences is very important issue in natural language processing field. The aim of semantic similarity between sentences is to assess the level of relatedness between them based on their meaning[3]. Many tasks in natural language processing depend on semantic similarity such as summarization[6], question answering[8], semantic matching[7], and so on. The accuracy of measuring is important factor in these applications. Moreover, semantic similarity has gained focus in many fields such as biomedical field and other fields. Analyzing biomedical text is very helpful in bioinformatic research. Measuring similarity between sentences supports this research area[11].

Calculating similarity between sentnces in many approaches depends on sentences representation and word representation. Moreover, semantic representation for natural language text is widely used in many appliacations[17]. On the other hand, Using deep learning approach has shown a very promising results in many fields specially computer vision. As in many other fields, researchers start to exploit the deep learning in natural language processing field. Furthermore, Mikolov has used a deep learning model to learn word representation in a semantic space [9]. He used two architectures: Continuous Bag Of Words (CBOW) and Skip-GRAM. The first model, CBOW, is a training model to predict a word from a context. However, the other model, Skip-GRAM, try to predict the context from the word. The generated word representation has achieved a very good results in the semantic representation of words[9]. Word2vec is unsupervised system to represent words in semantic space in such a way that similar words in meaning are close to each others in this space. For example, the word "Paris" and the word "France" are close to each other in the vector space. Usually, the vector that represent a word in that space has size from 200 to 400 depending on the learning parameters. However, word2vec dose not represent semantic relations between words[4]. Other works have tried to extend the word2vec approach.[12] [14]. Moreover, GloVe is another work for representing words in vector space based on leaning from a big corpus[13].

The new semantically rich representation of word has been used in many tasks of NLP. Word vector representation is used in similarity measuring between words. Moreover, short text similarity depends on word vector model to determine the relatedness degree. Some recently proposed approaches using only word embedding to calculate semantic similarity[2]. While others depends on external resources such as WordNet [1].

Many approaches have been proposed to find word similarity and sentence similarity in NLP. A lot of these approaches depend on a lexical resources and/or semantic resources[2]. Recently, using pre-trained word vector generated using deep learning is used to measure similarity between words and sentence. Atish [1] has presented an approach to calculate semantic similarity between words and sentences. His approach depends on WordNet to find the similarity between synsets. As a first step a sentence vector is represented based on word similarity between two sentences. Then he calculates sentences similarity based on the calculated vectors.

WordNet[21] is used as an external resource to measure semantic similarity between words. WordNet organizes words into hierarchy according to word meaning. Relations between words (such as synonymy, hyperonymy, meronymy and so on) are also included in WordNet. Moreover, different methods has been proposed to calculate similarity between words using WordNet hierarchy and relations. Furthermore, many approaches of measuring similarity between sentences depends on WordNet.

On the other hand, Tom in [2] used pre-traind word vector to calculate similarity between short text. In his technique, text is represented by the average vector of words' vectors. Using the averaged vectors, a semantic similarity is calculated. In addition, TF-IDF is used in the similarity equation in order to weight the words of the sentence according to its importance. External resources such as WordNet has not been used in this approach. This is because external resources is not available in all domains and for all natural languages.

Moreover, Using Part Of Speech (POS) is very useful in many NLP application. In similarity measuring, POS is used to find words with similar POS. Similarity between words with same POS are only considered. Another approach of using POS in similarity measure is calculating mean vector for noun words in a sentence and calculating verb vector. So, instead of having single vector for a sentence we have two mean vectors one for noun words and the other for verb words[16]. On the other hand, it can be used as weighting for word according to POS. Different words in a sentence have different weight according to its POS[15].

This paper proposes using word model in measuring semantic similarity between sentences besides external resource such as WordNet. The proposed Method is giving better results then the base line which is the using of cosine similarity between the two sentences. Moreover, experiments show that combining WordNet based similarity and Vector based similarity achieves better Pearson Correlation with human similarity in standard dataset. In addition, word order similarity is used to improve the final measurements.

This paper is organized as follow. Section two explains the proposed system to calculate the semantic similarity between sentences. The phase of preprocessing is dicussed in section three. Measuring semantic similarity between sentences have been discussed in section four. The details of the experiments and the used datasets is stated in section five. Finally, the conclusion of the presented work is reported in section six.

II. SIMILARITY MEASURE

Similarity measure between natural language sentences focuses on finding how much alike two sentences are. Similarity measuring between sentences have gained focus of research since it is the base for many branches of natural language processing such as information retrieval, text summarization, and question answering.

Representing a sentence semantically is an important issue to calculate semantic similarity between sentences. Moreover, there are different ways to represent a sentence semantically. A sentence basically consists of a set words. Each word can be represented by a vector in the semantic space. One of most common representation for sentence is the mean of vectors of that sentence. However, another way is to construct a vector for the sentence depending on sentence information[1]. On the other hand, there are other approaches to represent a sentence in semantic space accurately based on semantic space using deep learning[5].

The input to the proposed system, as shown in figure 1, is two sentences in natural language. The output is a value between 0 and 1 that represent the similarity degree between the inputed sentences. The proposed system consists of two main steps. The first step is a preprocessing step which includes tokenization, stop word removal, and initializing vector for words. The second step is calculating similarity between the two sentences. The similarity calculation depends on two measures. The first is semantic similarity based on pre-train vector representation and WordNet knowledge base. The second is calculating word order similarity, which considers the difference in order of words in the two sentences.

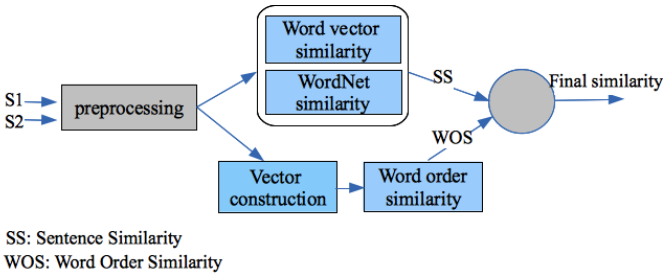

Fig. 1. TOOL architectural design

III. PREPROCESSING PHASE

In this phase the input sentences are prepared to similarity measuring process. The preparation process includes tokenization, stop word removal, and finding word vector in semantic space. After text tokenization and stop word removal we find vector representation for each word in the sentence. A pre-trained word vector, which trained on part of Google News dataset, is used to find the word vector. This trained word vector is public available, https://code.google.com/archive/p/word2vec/. The used pre-trained vector contains 3000000 words and each word is represented by a vector with size 300. However, some words may be not found in the vector space. In this case, there are different ways to apply. One way is to ignore this word form the sentence or assign a zero vector to this word. However, this way is not optimal. Another suggestion is to assign a random vector to this word[2]. The benefits of this way is that the common unfounded words will be matched. For example, a person name that found in both sentences will be matched and improve the final sentence similarity. While using the former way will not contribute to the sentence similarity.

In order to implement this way, an additional vectors is created randomly for unfounded words. If a word is not found in the original vector space we search for this word in the additional vector space and find its vector. If the word is not exist in the additional vector space a new random vector is created to this word and added to the additional random space. After this process the sentence is ready to start semantic similarity calculation based on word vector representation.

In addition, we lower case all words before processing. This is because the same word with capital letter will be considered as a different word in pre-trianing vectors. For example, 'Boy' and 'boy' are considered two words in the vector space. If we measure the similarity between Boy and girl, it will be 0.393372. However, if we measure the similarity between boy and girl, it will be 0.854327. These measures according to the used pre-trained vector space.

IV. CALCULATING SEMANTIC SIMILARITY

Calculating similarity in the proposed approach depends on two different measures: semantic similarity based on word vector and WordNet, and word order similarity. This section explains in details each measure.

A. Calculating semantic similarity

The word vector model has shown a promising results in representing words semantically in a vector space. The

representation of a word in the vector space reflects the word semantics. For example, the word king - man + woman = queen. This means if we subtract the vector representation of the word man from the vector of the word king, then add the vector of the word woman, the result will be the vector of the word queen. This means that the representation of the words in this vector space is very accurate semantically. In the following subsections we show different methods for measuring semantic similarity between sentences based on pre-trained word vector.

1) cosine similarity: In this method the semantic similarity between the two sentences is calculated based on cosine similarity between the two sentences. Firstly, a vector representing a sentence is calculated. In the word vector model the sentence vector is represented by the average of words vectors. The cosine similarity between means represents the semantic similarity between the two sentences. The cosine similarity between two sentences s1 and s2 can be calculated using the following equation.

$$cos(S1, S2) = \frac{V1.V2}{\|V1\| \, \|V2\|} = \frac{\sum_{i=1}^{n} V1iV2i}{\sqrt{\sum_{i=1}^{n} V1i^2}\sqrt{\sum_{i=1}^{n} V2i^2}} \quad (1)$$

where V1 is the vector representation for S1 and V2 is the vector representation of S1. Cosine similarity is widely used in measuring relatedness between natural text. In this work, cosine similarity is used as a baseline for our experiments.

2) word based similarity: The proposed method to measure similarity between two sentences depends on how much each word in the first sentence is related to the other sentence. Initially, similarity between a word and a sentence is calculated as the max similarity between this word and every word in the sentence. The similarity between a sentence s1 and a sentence s2 is calculated according to the following equation.

$$sim(S1, S2) = (\sum_{i=1}^{n} sim(wi, S2))/n \quad (2)$$

Where n is the number of words in S1 and sim(wi,S2) is the similarity between a word i from s1 and the sentence s2. This similarity is calculated as the max match between the word and each word in the sentence.

Similarity between two words w1 and w2 using pre-trained vector is calculated as the cosine similarity between vector representation of w1 and vector of w2. On the other hand, Using WordNet in measuring word similarity has some advantages. One of the main advantages is considering word POS. In other words, WordNet gives similarity between two words of the same type. For example, it does not measure similarity between noun word and verb. This is because every word type has its own hierarchy in WordNet. The proposed approach combines word vector similarity and WordNet similarity at word level. The word similarity measure is the weighted average between both similarities according to the following equation.

$$sim(w1, w2) = Vsim(w1, w2)*0.75 + WNsim(w1, w2)*0.25 \quad (3)$$

Moreover, in order to make the similarity accurate, similarity between s1 and s2 is calculated and similarity between s2 and s1 is also calculated. The final sentence similarity is the average between both similarities.

words	vector similarity	WordNet similarity	weighted average similarity
boy - girl	0.854327	0.552136	0.778779
boy - saw	0.142304	0.099181	0.131523
boy - flower	0.20061	0.330372	0.233051
boy - garden	0.126913	0.149464	0.132551
cut - girl	0.0191151	0.0812476	0.0346482
cut - saw	0.131987	0.218679	0.15366
cut - flower	0.103757	0.0680001	0.0948176
cut - garden	0.0451568	0.0998537	0.0588311
flower - girl	0.288346	0.272287	0.284331
flower - saw	0.0644533	0.0691874	0.0656368
flower - flower	1	1	1
flower - garden	0.594904	0.0996618	0.471093

TABLE I. CALCULATING SIMILARITY BETWEEN WORDS OF S1 "THE BOY CUT A FLOWER" AND S2 "THE GIRL SAW A FLOWER GARDEN" USING THE PROPOSED METHOD

For example, consider the similarity between these two sentences. S1="the boy cut a flower" and S2="the girl saw a flower garden". To calculate the semantic similarity we get the similarity matrix between every word in S1 and S2 by calculating the average between WordNet based similarity and vector based similarity as shown in table 1. Then we select the max match for each word by getting the best match for each word in the first sentence with the words in the second sentences. Then, get the average of these values. From table 1 sim(s1, s2) = (0.778779 + 0.15366 + 1)/3 = 0.644. and sim(s2, s1) = (0.778779 + 0.15366 + 1 + 0.471093)/4 =0.60. To get the final similarity the two values are averaged.

B. Calculating word order similarity

Order of words in a sentence changes the total meaning of the sentence. For example, consider the sentence "the big mouse clobbers the small cat" and the sentence "the small cat clobbers the big mouse". Although the two sentences have the same set of words, they have difference in meaning. Depending on the vector representation of the words and WordNet will give wrong similarity. Moreover, taking word order into consideration will improve the similarity measuring. In order to calculate word order similarity we adopt the proposed approach in [20].

To measure the word order similarity, a word order vector is constructed for each sentence. The length of this vector equals to the length of the joint set of words in S1 and S2. For each word in S1 we put its index in S1 in the corresponding element in the word order vector for S1. Consider our previous example, words of S1="big mouse clobbers small cat" and S2="small cat clobbers big mouse". The joint set of words between S1and S2 is {big, mouse, clobbers, small, cat}. The corresponding word order vector for S1 is [1 2 3 4 5]. The word order vector for S2 is [4 5 3 1 2]. Another example, where words in sentences are different, consider S1="the boy cut a flower" and s2="The girl saw a flower garden". The joint set will be {boy, cut, flower, girl, saw, garden}. The constructed word order vectors should be V1 = [1 2 3 0 0 0] and V2 = [0 0 3 1 2 4].

After constructing the vectors, the word order similarity is calculating according to the following equation.

$$Sw = 1 - \frac{\|V1 - V2\|}{\|V1 + V2\|} \quad (4)$$

Method	Pearson Correlation
Atish 2018	0.837
Word Vector	0.717
WordNet based	0.526
proposed	0.852

TABLE II. PEARSON CORRELATION OF THE DISCUSSED METHODS WITH MEAN HUMAN SIMILARITY

The final similarity is the combination between word order similarity and semantic similarity.

$$sim(S1, S2) = \delta Ss + (1 - \delta)Sw \quad (5)$$

where Ss is the semantic similarity and Sw is word order similarity. The value of δ that is used in our experiment is 0.6.

V. EXPERIMENTS

The proposed technique for measuring similarity is implemented and tested with respect to other techniques using the same dataset. Two standard datasets are used in our experiments to show the effectivness of the proposed technique. The first is Pilot Short Text Semantic Similarity Benchmark Data Set [10]. The second dataset is Microsoft paraphrase corpus [18] [19].

Pilot Short Text Semantic Similarity Benchmark Data Set[10] is widely used to evaluate sentence similarity techniques. It contains 65 pair of sentences in English. Each pair is labeled by human assessment similarity. A value between 0 and 1 was assigned to each pair to represent the level of relatedness between these sentences.

Although there are many techniques have been reported previously in this point of research, some selected techniques are used in this experiment to show the effectiveness of the proposed approach. The selected techniques are chosen based on how much they are related to our approach.

In the first experiment the similarity between the 65 pair of sentences are measured using the proposed approach. Moreover, the Pearson correlation coefficient with mean human similarity is calculated to measure the linear correlation between the proposed approach and the human evaluated similarity.

Table II shows a comparison between the proposed approach and other systems using 60 pairs of sentences of Li's Dataset[20]. Five pair of the original dataset was removed to enable real comparison with other approach. The ignored pairs are "17: coast-forest, 24: lad- wizard, 30: coast-hill, 33: hill-woodland and 39: brother-lad"[1]. The similarity measuring approach that proposed by Atish[1] achieves 0.834 Pearson Correlation according to the results stated in his paper. Moreover, the word vector approach that depends on cosine similarity between means of the two sentences achieves 0.717 Pearson Correlation. Li 2006 approached achieves 0.526. The implemented tool in [20] is downloaded and used in our experiment.

On the other hand, MS paraphrase [19] dataset is chosen to test the proposed approach. MS paraphrase corpus contains totally, 5800 pair of sentences. All pairs are manually labeled by 0 or 1. This is depending on whether these two sentences are paraphrasing or not. This dataset contains 4076 pair of

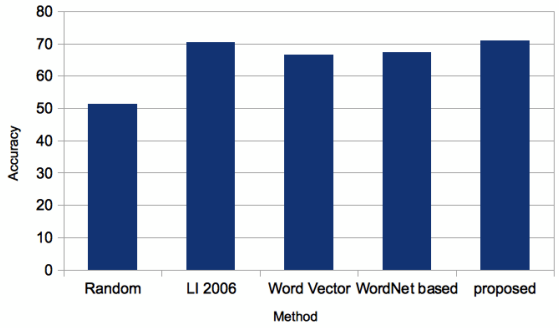

Fig. 2. Accuracy of the proposed approach and others against MS paraphrase dataset

Method	Acc.	Prec.	Rec.	F
Random	51.3	68.3	50.0	57.8
LI 2006	70.3	69.6	97.7	81.3
Word Vector	66.6	66.6	99.8	79.8
proposed	71.6	76.2	83.3	79.6

TABLE III. TEXT SIMILARITY FOR PARAPHRASE IDENTIFICATION

sentences as training data and around 1724 pairs as test data. In our experiment the test data is considered because our approach is unsupervised technique. In this experiment, accuracy, precision, recall, and f-measure are calculated to comapre between the proposed appraoch and other approaches used the same dataset. Table III shows the result of the experiment.

In this experiment, as suggested by Li and others[20] we consider the random function which genrates a random value (0 or 1) for each sentences pair as a baseline. In addition, two other techniques are considers beside the proposed approach. Although the differences are small, the proposed approach achieves the best accuracy and precision with respect to other approaches as shown in figure 2.

Experiments show that using the proposed approach for calculating semantic similarity between sentences outperforms other systems. The proposed method that combines both approaches outperforms the approach that is using Word vector only and outperforms the approach that is using WordNet only. Moreover, combining word vector model and WordNet based approaches takes the advantages of both approaches. While the word vector model represent the words accurately in the vector space depending on statisticals of huge corpus, WordNet has relation between words and considers type of word during word matching.

VI. CONCLUSION

The task of finding similarity between sentences has gained focus recently because it is an important issue in different fields. Utilizing deep learning gives a good semantic representation of natural language words. Moreover, semantic representation of words in a vector space facilitates the task of similarity measure and achieves good results. On the other hand, WordNet has the advantage of human designed hierarchy and matches words with same POS. This work proposes an approach that combines between using vector representation

of words and WordNet. In addition, the proposed approach uses word order similarity to fix the problem of typical words in two sentences with totally different meaning between the two sentences. Experiments using standard datasets show that using the proposed method gives good results. In addition, considering word order similarity in the calculated similarity between sentences improves the final measures.

REFERENCES

[1] Atish Pawar, Vijay Mago, Calculating the similarity between words and sentences using a lexical database and corpus statistics, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 2018

[2] Tom Kenter , Maarten de Rijke, Short Text Similarity with Word Embeddings, Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, October 18-23, 2015, Melbourne, Australia

[3] Harispe, S., Ranwez, S., Janaqi, S., Montmain, J.: Semantic Similarity from Natural Language and Ontology Analysis. Morgan and Claypool Publishers, San Rafael (2015)

[4] Handler, A.: An empirical study of semantic similarity in WordNet and Word2Vec. Ph.D. thesis, University of New Orleans (2014)

[5] J. H. Lau and T. Baldwin. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. In Proc. of the RepL4NLP 2016, pages 78–86

[6] R.M. Aliguyev, A New Sentence Similarity Measure and Sentence Based Extractive Technique for Automatic Text Summarization, Expert Systems with Applications, vol. 36, pp. 7764-7772, 2009.

[7] Mamdouh Farouk, Mitsuru Ishizuka and Danushka Bollegala, Graph Matching based Semantic Search Engine, 12th International Conference on Metadata and Semantics Research, Cyprus, 2018

[8] De Boni, M. and Manandhar, S. The Use of Sentence Similarity as a Semantic Relevance Metric for Question Answering. Proceedings of the AAAI Symposium on New Directions in Question Answering, Stanford. 2003.

[9] Mikolov, T., Chen, K., Corrado, G., and Dean, J. Efficient estimation of word representations in vector space. 2013, arXiv preprint arXiv:1301.3781.

[10] J.OShea,Z.Bandar,K.Crockett,and D.McLean,Pilotshorttext semantic similarity benchmark data set: Full listing and descrip- tion, Computing, 2008.

[11] Zhu Y, Yan E, Wang F. Semantic relatedness and similarity of biomedical terms: examining the effects of recency, size, and section of biomedical publications on the performance of word2vec. Vol. 17, BMC Medical Informatics and Decision Making. 2017

[12] Wolf, L., Hanani, Y., Bar, K., Dershowitz, N.: Joint word2vec networks for bilingual semantic representations. International Journal of Computational Linguistics and Applications 5(1) (2014) 2742

[13] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global vectors for word representation. In EMNLP, 2014

[14] Sascha Rothe and Hinrich Schutze. Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In Proceedings of the 53rd ACL, volume 1, 2015. pages 17931803, Beijing, China.

[15] Vuk Batanovic, Dragan Bojic, Using Part-of-Speech Tags as Deep-Syntax Indicators in Determining Short-Text Semantic Similarity, Computer Science and Information Systems, vol. 12, no. 1, pp. 1-31, 2015.

[16] Ming Che Lee, Jia Wei Zhang, Wen Xiang Lee, Sentence Similarity Computation Based on POS and Semantic Nets, Fijlh International Joint Conference on INC, IMS and IDC ,2009.

[17] Mamdouh Farouk, Ontology-based Semantic Representation for Arabic Text: A Survey, Journal of Informatics and Mathematical Sciences, Volume 9 Number 4, 2017.

[18] Quirk, C., C. Brockett, and W. B. Dolan. 2004. Monolingual Machine Translation for Paraphrase Generation, In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, Barcelona Spain.

[19] Dolan W. B., C. Quirk, and C. Brockett. Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. COLING 2004, Geneva, Switzerland. 2004.

[20] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, Sentence similarity based on semantic nets and corpus statistics, IEEE Transactions on Knowledge and Data Engineering, vol. 18, no. 8, pp. 11381150, 2006.

[21] George A. Miller. WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41., 1995.