Computing and Informatics, Vol. 39, 2020, 464–480, doi: 10.31577/cai_2020_3_464

MEASURING SENTENCES SIMILARITY BASED ON DISCOURSE REPRESENTATION STRUCTURE

Mamdouh Farouk

Computer Science Department
Assiut University
Assiut, Egypt
e-mail: mamfarouk@aun.edu.eg

Abstract. The problem of measuring similarity between sentences is crucial for many applications in Natural Language Processing (NLP). Most of the proposed approaches depend on similarity of words in sentences. This research considers semantic relations between words in calculating sentence similarity. This paper uses Discourse Representation Structure (DRS) of natural language sentences to measure similarity. DRS captures the structure and semantic information of sentences. Moreover, the estimation of similarity between sentences depends on semantic coverage of relations of the first sentence in the other sentence. Experiments show that exploiting structural information achieves better results than traditional word-to-word approaches. Moreover, the proposed method outperforms similar approaches on a standard benchmark dataset.

Keywords: Sentence similarity, discourse representation structure, structural similarity

1 INTRODUCTION

Natural language processing has gained the focus of research especially after the explosion of data expressed in natural languages. Moreover, the wide use of social media and the need to analyze this data makes natural language tasks crucial. Measuring the similarity between natural language sentences is located at the core of many tasks to process natural language data. For instance, many approaches such as text classification, summarization, question answering, semantic search, and

plagiarism checking depend on sentence similarity [24, 33, 34]. Accuracy of calculation of sentences similarity affects these applications. Consequently, the problem of measuring the sentence similarity has got a lot of focus.

Measuring similarity between natural language sentences means estimating the degree of semantic relatedness between these sentences. The solutions for the problem of measuring sentence similarity still need improvement to accurately assess the similarity. Most of the previously proposed approaches depend on words of sentences. However, a sentence does not contain words only. Semantic relations between words are important components of a sentence.

Deep learning techniques that achieved good results in computer vision are also used in sentence similarity task. Word semantic representation is generated using deep learning techniques [20]. In this representation similar words have close vectors in the representation space. These numerical vector representations, normally of 300 length, for words, are used to get semantic similarity of sentences [21, 4].

Discourse Representation Theory (DRT) is a framework for representing the meaning of natural language sentences in a formal semantic approach [11]. DRT uses mental representation, which is DRS, to handle the meaning across sentence boundaries. DRT is used to implement language understanding systems [5]. DRS, which is used in DRT, consists of two main components: a set of discourse referents and a set of conditions. Consider this sentence "A woman walks. She smokes." This sentence can be represented in DRS as shown in Figure 1. The first line contains the set of referents (x and y). The other part is the set of conditions upon these referents.

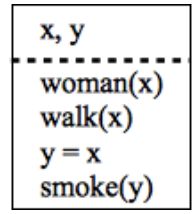

Figure 1. DRS representation for the sentence "A woman walks. She smokes."

This paper proposes a new approach for measuring sentence similarity. The proposed approach extracts semantic relations between words. Based on the similarity of semantic relations in sentences the similarity is calculated.

The main contribution of this work is calculating sentence structural similarity based on the semantic representation DRS that captures semantic and structural information of sentences. Unlike the traditional word-to-word approach, the proposed approach considers semantic relations between words in measuring similarity. Moreover, the proposed approach uses word embeddings to calculate the similarity between words.

The proposed approach is tested using standard datasets. Li2006 dataset [8] which is widely used in the evaluation of sentence similarity approaches is used to

evaluate the proposed approach. Moreover, MSRP dataset [3] which is used for paraphrase detection is also used to evaluate the performance of the proposed method. Experiments show that using DRS in sentence similarity improves measuring similarity.

The rest of this paper is organized as follows. Section 2 mentions the related work. Section 3 explains the proposed approach. A detailed example to get the similarity between two sentences is shown in Section 4. Section 5 describes the experiments and discusses the results. Finally, Section 5 concludes the presented work.

2 RELATED WORK

Different approaches have been proposed to calculate sentence similarity. Some of these approaches are string-based that consider the sentence as a sequence of characters. The similarity between two sequences of characters is assessed using string similarity methods such as q-grams [22] and Levenshtein distance [15].

Moreover, some approaches depend on word similarity to measure sentence similarity. These approaches consider the sentence as a set of words. WordNet [18], which is a lexical database for the English language, is widely used to find similarity between words. However, many approaches depend on analyzing big corpora to capture the semantics of words based on co-occurrences of words [24]. Latent Semantic Analysis (LSA) is one of the approaches that statistically analyzes big corpora to generate word semantic representation in a vector. Cosine similarity between these vectors measures the semantic similarity between words. Some approaches combine both methods (WordNet and corpus analyzing) to find similarity between sentences [25, 1].

The approaches for sentence similarity can be classified into three main classes: word-to-word based similarity, vector-based similarity, and structure-based approach [26]. In the word-to-word approach, the sentence similarity is calculated based on the similarity between the words in sentences. The second category depends on converting sentences to vectors that capture the semantic features of these sentences. Sentence similarity is calculated based on the similarity between these vectors. The third class of the approaches that measure sentence similarity is structure-based which exploits the structural information of sentences to calculate similarity.

Kenter and de Rijke in [21] propose an approach for measuring sentence similarity based on word embedding. They used word representation generated from deep learning to measure word similarity. Different pre-trained word vectors are used to measure sentence similarity. In addition to using word embeddings, TF-IDF weighting schema is used to consider word importance in the sentence. This approach is considered a word-to-word based approach. However, this approach ignores structural information of sentences. The structure of a sentence reveals important information that helps in the similarity measure.

Abdalgader and Skabar proposed to use word sense disambiguation and synonym expansion to improve sentence similarity [12]. Firstly, the sentences are processed and the meaning of the words are determined. A union vector for both sentences is constructed by finding the union of both sets of words. Additionally, the original set of words of each sentence is expanded by synonyms of the words belonging to this set. A vector representation for each sentence is constructed by finding the similarity between words in the union vector and words of that sentence vector. Finally, the cosine similarity between the two vectors is calculated as the semantic similarity between the two sentences. Although good results have been achieved using this approach, it needs external resources such as WordNet which is not available for all languages in high accuracy.

Some approaches combine different word similarity methods to calculate sentence similarity. For example, Li et al. in [25] proposed an approach to measure sentence similarity based on semantic net and corpus statistics. They computed semantic similarity between words based on a lexical database, which captures human knowledge, and based on a statistically analyzed corpus. In addition, word order similarity is calculated to measure order similarity for common words. A similar approach has been proposed by Pawar and Mago [1]. They combined WordNet and corpus analysis measures to assess sentence similarity. However, these approaches do not use structure information of sentences. In addition, external resources are needed to compute the similarity. The measured similarity depends on the accuracy of the used resources.

On the other hand, vector-based approaches generate vector representation for sentences and calculate similarity between these vectors. Skip-Thought [13] is a neural network model designed to train sentences and get vector representation that captures features of sentences. This model is similar to the skip-gram model that is used to get word vector representation. The idea is that similar sentences have similar features and close vectors. This model is used for sentence similarity systems. The input to their system is the words' vectors of sentences and the output is sentence vectors. These vectors are used to calculate sentence similarity.

Lee et al. in [14] introduced structure-based method to calculate sentence similarity. They extract grammar links from sentences and construct a grammar matrix in which rows represent links in the smaller (in length) sentence and columns represent links of the other sentence. Moreover, WordNet [18] is used to measure the similarity between words. The final similarity is calculated based on the constructed grammar matrix. Although this approach exploits lexical relations between words, it ignores semantic relations between words. Semantic relations are more helpful to assess semantic similarity between sentences.

Paraphrase detection is one task that is very related to sentence similarity. Recently Ferreira et al. proposed an approach for identifying paraphrases [27]. Their approach depends on extracting features and classifying a pair of sentences based on the extracted features. The extracted features are calculated based on lexical similarity, syntactic similarity, and semantic similarity. Lan and Xu proposed a learning-based approach that used sub-word level representation to detect para-

phrases [35]. However, these approaches can be used in paraphrase detection and do not assign a similarity value for a sentence pair. Moreover, these approaches do need labeled data.

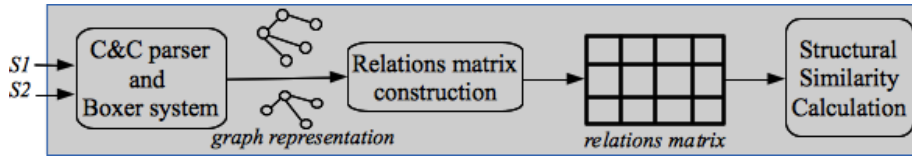

Figure 2. Proposed system architecture

3 SENTENCES SIMILARITY

A lot of NLP applications, such as social media analysis, question answering, and plagiarism, depend on sentence similarity. Consequently, the accuracy of measuring relatedness between sentences is a crucial task for many applications. The proposed approach exploits structure information in DRS representation of sentences to improve measuring sentence similarity. The input to the proposed system is two sentences and the output is a similarity value between 0 and 1.

As shown in Figure 2, calculating structural similarity between two sentences contains three steps. The first is generating DRS graphs for the inputted sentences. The second step is constructing a relation similarity matrix and the final step is calculating the structural similarity based on the relation matrix.

Structural information of a sentence helps to assess the sentence similarity [23]. Moreover combining structural similarity and word-based similarity improves the assessed similarity between sentences. As the first step for calculating structural similarity, each sentence is parsed and the output is passed to a semantic analyzer which outputs DRS graph representation equivalent to the sentence. Based on DRS graph representation of the sentences, a graph matching technique is used to measure the similarity between the two sentences. The following sub-sections explain the details of these steps.

3.1 Generation of Discourse Representation Structure

In order to get the structure of a sentence, a parser is used and semantic relations between words are extracted. A sentence semantic graph is constructed based on extracted relations. In this graph nodes represent words and edges represent semantic relations between words. The structural similarity of sentences is calculated based on the constructed graphs. In this paper, C & C parser [28] is used to parse sentences. In addition, the Boxer system [9] is used to get the semantic relations between sentence entities.

C & C parser contains many taggers such as Part Of Speech (POS) tagger and CCG supertagger. These taggers are highly efficient [9]. In addition, C & C contains Name Entity Recognizer which can determine ten different types of entities (organization, location, person, email, URL, first name, surname, title, quotation, and unknown name). C & C parser tags the words in a sentence with POS from the Penn treebank [17]. Then it builds sentence structure based on Combinatorial Categorial Grammar (CCG) paradigm. The output of the parsing is a syntax tree in which each node has POS tag, lemma, and name entity tag.

Based on the output of C & C parser, the Boxer system builds semantic representation for a sentence. The Boxer is a free software for analyzing text semantically. It depends on CCG and C & C parser to generate Discourse Representation Structure (DRS) for sentence text. DRS represents natural language text semantically. DRS captures the semantic of text and models it into related entities. DRS can be converted to other semantic formats such as first-order-logic [2]. The proposed approach uses semantic relations in DRS to calculate sentence similarity.

For example, consider these sentences: S_1 = "The boy who kills the snake is strong." and S_2 = "The boy is injured by a snake." The output of the Boxer system for these sentences is shown in Table 1. The DRS representation contains the words in sentences and the relations between words. For example the relation *theme* in S_1 connects the words *kills* and *snake*. Table 2 shows relations of both sentences.

Based on the output of the Boxer system, a semantic graph representation for the sentence is generated. Figure 3 shows the graph representation for the sentence S_1 = "The boy who kills a snake is strong." This graph captures the structure information of the sentence. Semantic relatedness between sentences is measured based on the generated graphs. Table 2 shows relations of DRS representation in both sentences.

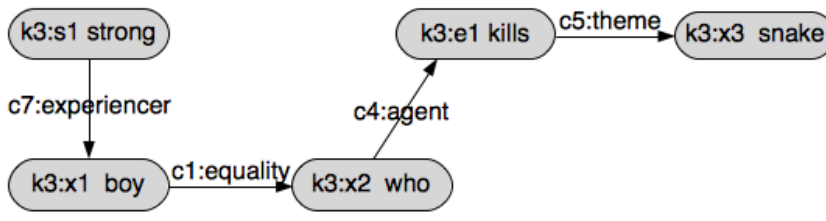

Figure 3. Sentence graph representation

3.2 DRS Graph Based Similarity

There are different techniques for solving graph matching problem. Graph matching is used in many applications in different fields [16]. For example, graph matching is used for measuring the similarity between documents [10]. In this paper structural sentence similarity is measured using sentences graphs. Based on the generated DRS

S_1 = "The boy who kills the snake is strong."	S_2 = "The boy is injured by a snake."
k3 attribute c6:strong:0 0 [ ]	k3 attribute c4:strong:0 0 [ ]
k3 concept c0:boy:0 0 [ ]	k3 concept c0:man:0 0 [ ]
k3 concept c2:snake:0 0 [ ]	k3 concept c5:snake:0 0 [ ]
k3 event c3:kill:0 0 [ ]	k3 event c1:kill:0 0 [ ]
k3 referent k3:e1 0 [ ]	k3 referent k3:e1 0 [ ]
k3 referent k3:s1 0 [ ]	k3 referent k3:s1 0 [ ]
k3 relation c1:equality 0 [ ]	k3 role c2:theme:1 0 [ ]
k3 role c5:theme:1 0 [ ]	k3 role c3:experiencer:-1 0 [ ]
k3 role c7:experiencer:1 0 [ ]	k3 role c6:agent:1 0 [ ]
k3 role c4:agent:-1 0 [ ]	k3 referent k3:x1 1 [The]
k3 referent k3:x1 1 [The]	c0:man:0 instance k3:x1 2 [man]
c0:boy:0 instance k3:x1 2 [boy]	k3 surface k3:e1 2 [is]
k3 referent k3:x2 1 [who]	k3:e1 main k3 1 [ ]
c3:kill:0 instance k3:e1 1 [kills]	c1:kill:0 instance k3:e1 3 [killed]
k3 referent k3:x3 1 [the]	k3 referent k3:x2 2 [a]
c2:snake:0 instance k3:x3 2 [snake]	c4:strong:0 arg k3:s1 1 [strong]
k3 surface k3:s1 2 [is]	c5:snake:0 instance k3:x2 4 [snake]
k3:s1 main k3 1 [ ]	c2:theme:1 int k3:e1 1 [ ]
c6:strong:0 arg k3:s1 3 [strong]	c2:theme:1 ext k3:x1 0 [ ]
c1:equality int k3:x1 3 [ ]	c3:experiencer:-1 int k3:x2 3 [ ]
c1:equality ext k3:x2 0 [ ]	c3:experiencer:-1 ext k3:s1 0 [ ]
c5:theme:1 int k3:e1 2 [ ]	c6:agent:1 int k3:e1 4 [ ]
c5:theme:1 ext k3:x3 0 [ ]	c6:agent:1 ext k3:x2 1 [by]
c7:experiencer:1 int k3:s1 1 [ ]	
c7:experiencer:1 ext k3:x1 0 [ ]	
c4:agent:-1 int k3:x2 2 [ ]	
c4:agent:-1 ext k3:e1 0 [ ]	

Table 1. DRS representation generated from Boxer system: (on left) DRS representation for sentence S_1 and (on right) DRS for sentence S_2

graphs for sentences, a relation matrix is constructed. Rows of this matrix represent relations of the first graph and columns represent relations of the second graph. Cell i, j in the matrix is filled with similarity value between relation i belonging to the first sentence and relation j belonging to the second sentence. Structural sentence similarity is calculated from this matrix.

3.2.1 Relation Similarity

As shown in Table 1, each relation has a name and links between the interior word and the exterior word. The similarity value between two relations is calculated in three steps:

Measuring the similarity between names of relations. The proposed approach distinguishes between the case when both relations have the same name

S_1 = "The boy who kills the snake is strong."	S_2 = "The boy is injured by a snake."
boy → **equality** → who	killed → **theme** → man
kills → **theme** → snake	snake → **experiencer** → strong
strong → **experiencer** → boy	killed → **agent** → snake
who → **agent** → kills	

Table 2. Relations of sentence S_1 and relations of sentence S_2 according to DRS representation

and the case when both relations have different names. The similarity value in the first case is higher than in the second case. If both relations have the same name the similarity value is 1. Otherwise the similarity value will be 0.7. This value has been assigned based on a tuning experiment using Li2006 dataset [8]. This value is working for every dataset.

Measuring similarity between interior nodes. Word embeddings [20] are used to calculate the similarity between interiors words of both relations. Word vectors which are trained on part of Google News[1] is used to get the word vector. The cosine similarity between words' vectors is calculated as the similarity between these words. In addition, word expansion is used to improve the word similarity measure. Two lists of words are obtained from the two words using expansion. The max similarity between these two lists is chosen as the similarity between the two words.

Measuring similarity between exterior nodes. Word embeddings are also used to find similarity between exterior words.

The following equation is used to calculate the similarity between two relations R_1 and R_2.

$$RelSim(R_1, R_2) = \frac{Sim(I_{R1}, I_{R2}) + Sim(E_{R1}, E_{R2})}{2} * NameSim(R_1, R_2). \quad (1)$$

$Sim(I_{R1}, I_{R2})$ is the similarity between interior word of R_1 and interior word of R_2. *NameSim* assesses similarity between names of relations.

For example, the similarity between *theme* relation in S_1 and *theme* relation in S_2 is calculated as follows:

- Similarity between names of relations is 1.
- Similarity between interior nodes: the interior word for *theme* relation in S_1 is *kills* and interior word for *theme* relation in S_2 is *killed*. $Sim(kills, Killed)$ is 0.94.
- Similarity between exterior nodes: $Sim(snake, man)$ is 0.08.

The final similarity between these relations is calculated according to Equation (1).

[1] This data set is publicly available at `https://code.google.com/archive/p/word2vec/`

3.2.2 Word Expansion

The proposed approach uses word expansion when measuring the similarity between two words. A word can be considered an expansion to another if there is an *equality* relation between them. For example, in Figure 3 the word *boy* can be used as expansion to the word *who*. When measuring the similarity between a word w_1 and another w_2, the proposed approach gets a list of words equal to w_1 and a list of words equal to w_2. The similarity between all words in the two lists is calculated and the max similarity is selected to represent the similarity between w_1 and w_2.

3.2.3 Calculating Structural Similarity

The proposed approach calculates the structural similarity by guessing to what extent the relations of sentence S_1 are covered by sentence S_2. This can be calculated based on the constructed matrix. In order to measure coverage of a relation belonging to the first sentence in the second sentence, the maximum similarity between this relation and all relations in the second sentence is selected. The structural similarity between S_1 and S_2 is calculated as follows:

$$Sim_{st}(S_1, S_2) = \frac{\sum_i^n maxSim(Ri, S2) * W_{Ri}}{\sum_i^n W_{Ri}} \tag{2}$$

where n is the number of relations in S_1 and W_{Ri} is the weight for the relation R_i. The weight of relations is used to reflect the importance of different relations according to its effect on the sentence meaning. Since the generated relations are limited, a fixed weight is assigned to each relation (Table 3). Common semantic roles have a high weight. Relations such as *agent* and *theme* have higher weights than other relations.

Relation Name	Weight
agent	8
theme	8
experiencer	6
is	4
in	3
other relations	1

Table 3. Weights for relations

4 EXPERIMENTS

The proposed approach has been implemented and tested against standard datasets to prove its effectiveness. The implemented system takes two sentences in natural language as input and measures the similarity between them. The output value

R & G Number	Human Assessment	Li 2006 [25]	Islam [7]	Pawar [1]	Omiotis [6]	Grammar Based [14]	Farouk [4]	Proposed Approach
1	0.01	0.33	0.06	0.023	0.11	0.22	0.104	0.121
5	0.005	0.29	0.11	0.07	0.10	0.06	0.12	0.161
9	0.005	0.21	0.07	0.015	0.10	0.35	0.087	0.067
13	0.108	0.53	0.16	0.292	0.30	0.32	0.204	0.207
17	0.048	0.36	0.26	0.366	0.30	0.41	0.246	0.317
21	0.043	0.51	0.16	0.231	0.24	0.44	0.276	0.178
25	0.065	0.55	0.33	0.279	0.30	0.07	0.30	0.271
29	0.013	0.34	0.12	0.133	0.11	0.20	0.243	0.188
33	0.145	0.59	0.29	0.762	0.49	0.07	0.244	0.423
37	0.13	0.44	0.2	0.10	0.11	0.07	0.218	0.276
41	0.28	0.43	0.09	0.045	0.11	0.02	0.264	0.203
47	0.35	0.72	0.3	0.161	0.22	0.25	0.332	0.283
48	0.355	0.64	0.34	0.54	0.53	0.79	0.386	0.317
49	0.29	0.74	0.15	0.299	0.57	0.38	0.397	0.288
50	0.47	0.69	0.49	0.253	0.55	0.07	0.175	0.378
51	0.14	0.65	0.28	0.302	0.52	0.39	0.133	0.303
52	0.485	0.49	0.32	0.842	0.6	0.84	0.428	0.387
53	0.483	0.39	0.44	0.89	0.5	0.18	0.382	0.433
54	0.36	0.52	0.41	0.783	0.43	0.32	0.286	0.24
55	0.405	0.55	0.19	0.315	0.43	0.38	0.243	0.402
56	0.59	0.76	0.47	0.977	0.93	0.62	0.489	0.521
57	0.63	0.7	0.26	0.477	0.61	0.82	0.318	0.359
58	0.59	0.75	0.51	0.892	0.74	0.94	0.388	0.496
59	0.86	1	0.94	0.856	1	1	0.889	0.80
60	0.58	0.66	0.60	0.898	0.93	0.89	0.549	0.484
61	0.52	0.66	0.29	0.934	0.35	0.08	0.265	0.339
62	0.77	0.73	0.51	1	0.73	0.94	0.594	0.46
63	0.56	0.64	0.52	0.7	0.79	0.95	0.367	0.525
64	0.955	1	0.93	0.873	0.93	1	0.876	0.85
65	0.65	0.83	0.65	0.854	0.82	–	0.578	0.597

Table 4. Results of the proposed approach and other approaches using Li2006 dataset

of the implemented system is between 0 to 1 (0 means no similarity and 1 means completely similar). In order to show the impact of using DRS of sentences, the proposed system is compared to other systems using standard datasets.

4.1 Li2006 Dataset

A short text semantic similarity benchmark dataset [8] is used to evaluate the proposed system. It is one of the widely used datasets in sentence similarity evaluation [25, 14, 7]. Originally, this dataset was created by Rubenstein and Goode-

Method	Pearson Correlation	Spearman Correlation
Li 2006	0.815	0.812
Islam	0.846	0.83
Pawar	0.781	0.823
Omiotis	0.857	0.889
Grammar based	0.714	0.639
word embedding	0.852	0.81
proposed approach	**0.872**	**0.894**

Table 5. Comparison between the proposed method and other methods

nough to measure word similarity [19]. The original dataset contains 65 pairs of words. Li et al. [25] added the definition of each word using the Collins Cobuild dictionary to use this dataset in sentence similarity. These 65 pairs of sentences are manually graded by 32 English native speakers according to the similarity degree.

The proposed system has been fed with pairs of sentences from Li2006 dataset. For each pair of sentences a similarity degree is returned. Table 4 shows the results of the proposed system along with other previously proposed systems. The results are shown in Table 4 for a subset of the selected benchmark. This subset contains 30 pairs of sentences selected carefully to cover different similarity ranges [14]. The proposed system is compared with classic approaches that do not use labeled data such as word-to-word or structure-based approaches. The results of Li approach [25], STS Meth [7] which integrates different word similarity methods, Pawar [1] which combines WordNet and corpus analysis to measure sentences similarity, Omiotis system [6] which is a new measure of semantic relatedness between texts, are included in Table 4. In addition, a similar approach to the proposed system which uses grammar-based similarity technique [14] is also included in the comparison. Moreover, results of Farouk's approach [4] which uses word embeddings in measuring the similarity are also included in Table 4. The Pearson correlation coefficient is calculated between each system results and human rating. Equation (3) is used to calculate the correlation between the human rating and the proposed system.

$$r = \frac{n\sum_i^n x_i y_i - \sum_i^n x_i \sum_i^n y_i}{\sqrt{n\sum_i^n x_i^2 - (\sum_i^n x_i)^2}\sqrt{n\sum_i^n y_i^2 - (\sum_i^n y_i)^2}} \tag{3}$$

where n is the number of sentence pairs, x is the similarity value of the proposed approach and y is human similarity value. The proposed approach has achieved the best results comparing to other systems in Table 4. The proposed system achieved 0.872 Pearson correlation with human similarity.

In addition, Spearman correlation is calculated to show the relationship between the results of different systems and human measured similarity. Equation (4)

explains how to calculate Spearman correlation.

$$r_s = 1 - \frac{6 \sum_i^n D_i}{n^3 - n^2} \tag{4}$$

where n is the number of samples and D is the difference between the human assessment and the system assessment. As shown in Table 5, the proposed system achieves the best results among all other systems.

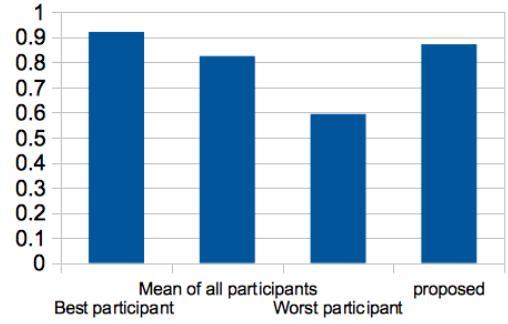

Figure 4. Results of the proposed system comparing to human raters

Figure 4 shows the achieved results comparing to human results in the Li2006 dataset. After calculating the average assessment of all human participants, Pearson correlation is calculated between the average assessment and each individual human assessment. As shown in Figure 4, the worst correlation between all participants is 0.594. The proposed approach achieves better than the mean of all human participants.

4.2 MSRP Dataset

Microsoft Research Paraphrase dataset [3] is widely used to evaluate sentence similarity techniques. It contains more than 5 000 pairs of sentences. It was partitioned into two sets. The first set contains around 4 200 pairs of sentences and is used as a training set. The other set contains around 1 700 pairs of sentences and is used for testing. Each pair is labeled by 1 (paraphrased) or 0 (not paraphrased). In our experiment the testing set is used to test the proposed approach.

In this experiment the proposed approach calculates the similarity between each pair of sentences and assigns a value between 0 and 1. A threshold value should be used to convert the calculated similarity value to 0 or 1. If the calculated similarity value is above the threshold, this pair is considered as the paraphrases. Different threshold values have been used previously in the literature. Omiotis approach used 0.2 as a threshold value [6], and 0.5 is used by Achananuparp in [32]. A tuning

experiment using hill-climbing algorithm [33] on MSRP training dataset determined that 0.45 is the threshold value for the proposed approach.

Metric	Accuracy	Precision	Recall	F-measure
Islam	72.64	74.65	89.13	81.25
Omiotis	69.97	70.78	93.40	80.52
grammar based	71.02	73.90	91.07	81.59
Farouk	71.6	76.2	83.3	79.6
proposed approach	70.46	72.34	89.30	79.93

Table 6. Results of the proposed approach and other approaches using MSRP dataset

Table 6 shows the achieved results and other approaches results in the MSRP dataset. Although the results of the proposed system are not the best, these results are comparable to other systems.

4.3 Results and Discussion

The experiments show that measuring similarity based on structural information of sentences gives better results than the traditional word-to-word approach. The proposed approach which uses C & C parser and the Boxer system to generate DRS representation for sentences outperforms other systems in Pearson correlation and spearman correlation measures.

However, the proposed approach achieves 70 % accuracy on the MSRP dataset. The results of MSRP dataset are not very good such as Li2006 dataset. This is because the proposed approach depends on the structure of sentences. The better structure of sentences the better performance of the proposed system. The first dataset (Li2006) is derived from a dictionary which means its sentences are well-structured. Consequently, the proposed approach achieves good results. However, the second dataset (MSRP) is derived from news sources on the web. This may explain the results of the second experiment.

Although the proposed approach outperforms other classic approaches in Li2006 dataset, it is sensitive to sentence structure. The performance of the proposed system will not be in the same high level with data that loose structure such as twitter messages. Moreover, DRS representation can be generated for many languages such as French [29], Chinese [30]. The proposed approach can be applied to other languages if DRS representation can be generated for that language.

5 CONCLUSION

The problem of finding the similarity between natural language sentences is important for many applications. Moreover, the structure of a sentence can reveal important information that helps in measuring sentence similarity. The proposed approach exploits structural information to calculate sentence similarity.

The proposed approach uses C & C parser and the Boxer system to generate DRS representation for sentences. This semantic representation captures the relations between words. Sentence similarity calculated based on the similarity between relations in both sentences. Moreover, word embedding is used to measure the similarity between words of relations. Experiments using standard datasets show the effectiveness of the proposed approach. The proposed approach performs well especially in case of well-structured sentences. Moreover, the proposed system achieves 0.872 Pearson correlation with human similarity. The proposed system outperforms other classical systems that depend on word-to-word and structure-based similarity.

REFERENCES

[1] Pawar, A.—Mago, V.: Calculating the Similarity Between Words and Sentences Using a Lexical Database and Corpus Statistics. 2018, arXiv preprint arXiv:1802.05667.

[2] Bos, J.: Wide-Coverage Semantic Analysis with Boxer. Proceedings of the 2008 Conference on Semantics in Text Processing (STEP '08), 2008, pp. 277–286, doi: 10.3115/1626481.1626503.

[3] Dolan, W. B.—Quirk, C.—Brockett, C.: Unsupervised Construction of Large Paraphrase Corpora: Exploiting Massively Parallel News Sources. Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004), Geneva, Switzerland, 2004, pp. 350–356.

[4] Farouk, M.: Sentence Semantic Similarity Based on Word Embedding and WordNet. 2018 13th International Conference on Computer Engineering and Systems (ICCES), 2018, pp. 33–37, doi: 10.1109/ICCES.2018.8639211.

[5] Guzman, F.—Joty, S.—Marquez, L.—Nakov, P.: Using Discourse Structure Improves Machine Translation Evaluation. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2014, pp. 687–698, doi: 10.3115/v1/P14-1065.

[6] Tsatsaronis, G.—Varlamis, I.—Vazirgiannis, M.: Text Relatedness Based on a Word Thesaurus. Journal of Artificial Intelligence Research, Vol. 37, 2010, No. 1, pp. 1–39, doi: 10.1613/jair.2880.

[7] Islam, A.—Inkpen, D.: Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity. ACM Transactions on Knowledge Discovery from Data, Vol. 2, 2008, No. 2, Art. No. 10, doi: 10.1145/1376815.1376819.

[8] O'Shea, J.—Bandar, Z.—Crockett, K.—McLean, D.: Pilot Short Text Semantic Similarity Benchmark Data Set: Full Listing and Description. Computing, 2008, available at: `https://semanticsimilarity.files.wordpress.com/2011/09/trmmucca20081_5.pdf`.

[9] Curran, J. R.—Clark, S.—Bos, J.: Linguistically Motivated Large-Scale NLP with C & C and Boxer. Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, 2007, pp. 33–36.

[10] Hammouda, K. M.—Kamel, M. S.: Phrase-Based Document Similarity Based on an Index Graph Model. 2002 IEEE International Conference on Data Mining (ICDM '02), 2002, pp. 203–210, doi: 10.1109/ICDM.2002.1183904.

[11] Kamp, H.—Reyle, U.: From Discourse to Logic: An Introduction to Modeltheoretic Semantics, Formal Logic and Discourse Representation Theory. Springer Netherlands, Studies in Linguistics and Philosophy, Vol. 42, 1993, doi: 10.1007/978-94-017-1616-1.

[12] Abdalgader, K.—Skabar, A.: Short-Text Similarity Measurement Using Word Sense Disambiguation and Synonym Expansion. In: Li, J. (Ed.): AI 2010: Advances in Artificial Intelligence. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6464, 2010, pp. 435–444, doi: 10.1007/978-3-642-17432-2_44.

[13] Kiros, R.—Zhu, Y.—Salakhutdinov, R.—Zemel, R. S.—Torralba, A.—Urtasun, R.—Fidler, S.: Skip-Thought Vectors. In: Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., Garnett, R. (Eds.): Advances in Neural Information Processing Systems 28 (NIPS 2015), 2015, pp. 3294–3302.

[14] Lee, M. C.—Chang, J. W.—Hsieh, T. C.: A Grammar-Based Semantic Similarity Algorithm for Natural Language Sentences. The Scientific World Journal, Vol. 2014, 2014, Art. No. 437162, 17 pp., doi: 10.1155/2014/437162.

[15] Levenshtein, V. I.: Binary Codes Capable of Correcting Deletions, Insertions, and Reversals. Soviet Physics – Doklady, Cybernetics and Control Theory, Vol. 10, 1966, No. 8, pp. 707–710.

[16] Farouk, M.—Ishizuka, M.—Bollegala, D.: Graph Matching Based Semantic Search Engine. In: Garoufallou, E., Sartori, F., Siatri, R., Zervas, M. (Eds.): Metadata and Semantics Research (MTSR 2018). Springer, Cham, Communications in Computer and Information Science, Vol. 846, 2018, pp. 89–100, doi: 10.1007/978-3-030-14401-2_8.

[17] Marcus, M. P.—Santorini, B.—Marcinkiewicz, M. A.: Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, Vol. 19, 1993, No. 2, pp. 313–330.

[18] Miller, G. A.: WordNet: A Lexical Database for English. Communications of the ACM, Vol. 38, 1995, No. 11, pp. 39–41, doi: 10.1145/219717.219748.

[19] Rubenstein, H.—Goodenough, J. B.: Contextual Correlates of Synonymy. Communications of the ACM, Vol. 8, 1965, No. 10, pp. 627–633, doi: 10.1145/365628.365657.

[20] Mikolov, T.—Chen, K.—Corrado, G.—Dean, J.: Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 2013, arXiv preprint arXiv:1301.3781.

[21] Kenter, T.—de Rijke, M.: Short Text Similarity with Word Embeddings. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (CIKM '15), 2015, pp. 1411–1420, doi: 10.1145/2806416.2806475.

[22] Ukkonen, E.: Approximate String-Matching with Q-Grams and Maximal Matches. Theoretical Computer Science, Vol. 92, 1992, No. 1, pp. 191–211, doi: 10.1016/0304-3975(92)90143-4.

[23] Ma, W.—Suel, T.: Structural Sentence Similarity Estimation for Short Texts. Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference (FLAIRS), 2016, pp. 232–237.

[24] Gomaa, W. H.—Fahmy, A. A.: A Survey of Text Similarity Approaches. International Journal of Computer Applications, Vol. 68, 2013, No. 13, pp. 13–18, doi: 10.5120/11638-7118.

[25] Li, Y.—McLean, D.—Bandar, Z. A.—O'Shea, J. D.—Crockett, K.: Sentence Similarity Based on Semantic Nets and Corpus Statistics. IEEE Transactions on Knowledge and Data Engineering, Vol. 18, 2006, No. 8, pp. 1138–1150, doi: 10.1109/TKDE.2006.130.

[26] Farouk, M.: Measuring Sentences Similarity: A Survey. Indian Journal of Science and Technology, Vol. 12, 2019, No. 25, pp. 1–11, doi: 10.17485/ijst/2019/v12i25/143977.

[27] Ferreira, R.—Cavalcanti, G. D. C.—Freitas, F.—Lins, R. D.—Simske, S. J.—Riss, M.: Combining Sentence Similarities Measures to Identify Paraphrases. Computer Speech and Language, Vol. 47, 2018, pp. 59–73, doi: 10.1016/j.csl.2017.07.002.

[28] Clark, S.—Curran, J. R.: Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. Computational Linguistics, Vol. 33, 2007, No. 4, pp. 493–552, doi: 10.1162/coli.2007.33.4.493.

[29] Le, N. L.—Haralambous, Y.—Lenca, P.: Towards a DRS Parsing Framework for French. Advances in Natural Language Processing, Granada, Spain, 2019.

[30] Wang, Q.—Zhang, L.: Formal Semantics of Chinese Discourse Based on Compositional Discourse Representation Theory. In: Deng, H., Miao, D., Wang, F. L., Lei, J. (Eds.): Emerging Research in Artificial Intelligence and Computational Intelligence (AICI 2011). Springer, Berlin, Heidelberg, Communications in Computer and Information Science, Vol. 237, 2011, pp. 470–476, doi: 10.1007/978-3-642-24282-3_65.

[31] Achananuparp, P.—Hu, X.—Shen, X.: The Evaluation of Sentence Similarity Measures. In: Song, I. Y., Eder, J., Nguyen, T. M. (Eds.): Data Warehousing and Knowledge Discovery (DaWaK 2008). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 5182, 2008, pp. 305–316, doi: 10.1007/978-3-540-85836-2_29.

[32] Russell, S. J.—Norvig, P.: Artificial Intelligence: A Modern Approach. 2nd Edition. Prentice, Upper Saddle River, New Jersey, 2003.

[33] Ramírez-Noriega, A.—Juárez-Ramírez, R.—Jiménez, S.—Inzunza, S.—Martínez-Ramírez, Y.: ASHuR: Evaluation of the Relation Summary-Content without Human Reference Using ROUGE. Computing and Informatics, Vol. 37, 2018, No. 2, pp. 509–532, doi: 10.4149/cai_2018_2_509.

[34] Zhang, L.—Hu, X.: Word Combination Kernel for Text Classification with Support Vector Machines. Computing and Informatics, Vol. 32, 2013, No. 4, pp. 877–896.

[35] Lan, W.—Xu, W.: Character-Based Neural Networks for Sentence Pair Modeling. Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018), Volume 2 (Short Papers), 2018, pp. 157–163, doi: 10.18653/v1/N18-2025.

Mamdouh Farouk is Assistant Professor in Assiut University, Egypt. He received his Ph.D. degree in artificial intelligence from the University of Tokyo, Japan in 2012. He received his B.Sc. and M.Sc. degrees in computer science from Cairo University in 2001. His research interests include data representation and Arabic language processing.