Measuring Text Similarity based on Structure and Word Embedding

Mamdouh Farouk
Computer Science Dept.
Assiut University, Assiut, Egypt
mamfarouk@aun.edu.eg

January 12, 2020

Abstract

The problem of finding similarity between natural language sentences is crucial for many applications in Natural Language Processing (NLP). Moreover, accurate calculation of similarity between sentences is highly needed. Many approaches depends on word-to-word similarity to measure sentences similarity. This paper proposes a new approach to improve accuracy of sentences similarity calculation. The proposed approach combines different similarity measures in calculation of sentences similarity. In addition to traditional word-to-word similarity measure the proposed approach exploits sentences semantic structure. Discourse representation structure (DRS) which is a semantic representation for natural sentences is generated and used to calculated structure similarity. Furthermore, word order similarity is measured to consider order of words in sentences. Experiments show that exploiting structural information achieves good results. Moreover, the proposed method outperforms the current approaches on Pilot standard benchmark dataset achieving 0.8813 peasron correlation with human similarity.

1 Introduction

Natural language processing has got a lot of attraction specially after the explosion of data expressed in natural language. Moreover, the wide use of social media and the need to analyze this social data makes natural language tasks crucial. Measuring similarity between natural language sentences is the core of many tasks to process natural language text automatically. For instance: many approaches of text classification , summarization , question answering and plagiarism checking depend on sentences similarity. Accurate calculation of similarity between sentences affects many applications of natural language processing. Consequently, the problem of finding similarity between sentences has gained focus.

Measuring similarity between sentences means estimating the degree of closeness in meaning between these sentences. Different approaches have been proposed to calculate similarity between sentences. The string-based approach considers the sentence as a sequence of characters. It finds the similarity using methods such as q-gram and levenshtein distance.

Moreover, some approaches for measuring sentences similarity depend on similarity between words. These approaches consider the sentence as a set of words. WordNet (Miller, 1995), which is a lexical database captures human knowledge, is widely used to find similarity between words (Liu and Wang, 2013). However, many approaches depend on analyzing big corpus to measure similarity between words (Gomaa and Fahmy, 2013). The idea behind corpus analysis depends on the observation that words with similar co-occurrence in the corpus have similar meaning. Latent Semantic Analysis (LSA) is one of the approaches that statistically analyze a big corpus to represent word in numerical vector. Cosine similarity between these vectors represents the semantic similarity between words. Some approaches combine both methods (WordNet and corpus analysis) to find similarity between sentences (Li *et al.*, 2006)(Pawar and Mago, 2018)(Farouk, 2018).

On the other hand, using deep learning to find word vector representation has shown promising results in measuring similarity between words (Mikolov *et al.*, 2013). A word representation is learned through a training model which tries to predict the word from context words. The output of this process is a vector (normally of size 300) which captures the semantics of word. The semantically related words have closed vectors. Many approaches use the generated vector (word embedding) to measure similarity between sentences (Kenter and Maarten, 2015)(Farouk, 2018).

This paper proposes an approach for measuring similarity between sentences based on different similarity methods. Structure similarity and word based similarity are combined in the proposed approach. Semantic relations between words in the sentence are extracted and structure similarity is calculated based on the extracted relations. In addition, word-to-word similarity is measured depending on word embedding. Moreover, the proposed approach exploits word order similarity to improve final calculated similarity. The proposed approach combines these measures to calculate sentences similarity.

This research aims to improve measuring sentences similarity through combining syntactic and semantic similarities. Moreover, this paper proves that considering structural information besides word-to-word approach improves sentences similarity calculation. In addition, applying the proposed approach for different datasets shows the applicability of the approach for general NLP tasks.

The rest of this paper is organized as follow. Section 2 mentions the related work. Section 3 explains the proposed approach and details of its main components. Moreover, a detailed example to calculate similarity between two sentences is shown in section 4. Section 5 describes the experiments and discusses the results of the proposed approach. Finally, Section 6 concludes the presented work.

2 Related work

Many approaches have been proposed to address the problem of measuring similarity between short texts (Farouk, 2019). These approaches can be classified according to working methodology into word-to-word based, vector based and structural based. Word-to-word approach measures the similarity between sentences based on the similarity between words of both sentences. There are many techniques for measuring words similarity (Navigli and Martelli, 2019). On the other hand, vector-based approach represents sentences into vectors and get the similarity between these vectors. Differently, structure based approach takes the structure of sentences into account when calculating sentences similarity.

Li et al proposed an approach to measure sentences similarity based on word-to-word similarity (Li *et al.*, 2006). In addition to WordNet they use (Latent Semantic Analysis) LSA to find similarity between words. Moreover, Atish and Mago proposed a similar approach that combines WordNet and corpus analysis measures to assess similarity between sentences (Pawar and Mago, 2018).

On the other hand, some approaches exploit word embedding to measure similarity between words(Kenter and Maarten, 2015). Word embedding is a vector representation for words. These vectors which capture semantic features of words are obtained using deep learning models. One of these approaches which exploits word embedding to measure similarity between words is proposed in (Kenter and Maarten, 2015) . Different pre-trained word vectors are used to measure sentences similarity. Moreover, TF-IDF weighting schema is used beside word embedding to consider word importance. However, this approach considers the sentence as a set of word and ignores structure of sentences.

Word-to-word sentences similarity approach depends totally on word similarity approach. Recently, Qu et al propose a novel method to compute semantic similarity based on Wikipedia (Qu *et al.*, 2018). They exploit Wikipedia to calculate Information Content (IC) for concepts and represent concepts semantically. Properties of concept such as neighbors and categories are used to enrich semantic representation of concepts. Although the results of this approach is promising, the high ambiguity of Wikipedia concepts is a problem that needs a solution.

On the other hand, some approaches combine different word similarity methods to calculate sentences similarity (Farouk, 2018). Although these approaches achieve good results they ignore structure of sentence which captures important information that is helpful in calculating similarity.

Differently, vector based approach tries to capture features of a sentence into a numerical vector representation. Similarity between vectors represents sentences similarity. This approach depends on vector representation method. The richer representation the more accurate similarity measure between sentences. Ryan et al proposed a vector representation for sentences based on deep learning (Ryan *et al.*, 2015). Their approach which called Skip-thought trains the model to predict surround sentences from the focused sentence. This vector representation achieves good results in sentences similarity task.

Lee et al in (Lee *et al.*, 2014) introduced structure based method to calculate

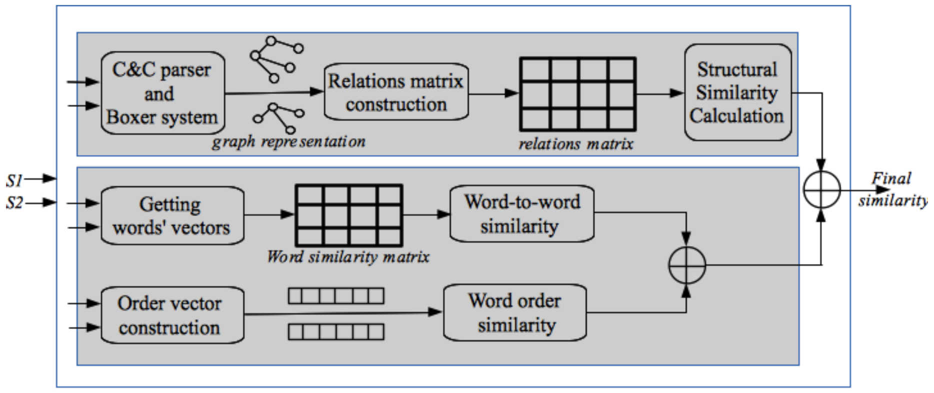

Figure 1: Proposed system architecture: the upper part is structural similarity and the lower part is word embedding based similarity and word order similarity

similarity between sentences. They extract grammar links from sentences and construct grammar matrix in which rows represent links in the small sentence and columns represent links of other sentence. Moreover, WordNet is used to measure similarity between words. The final similarity is calculated based on the constructed grammar matrix. Although they used grammar links which reflect the sentence structure, they ignore semantic relations which reflect structure and semantics of a sentence.

On the other hand, paraphrase detection is one task that is very related to sentences similarity. Recently, Ferreira et al propose an approach for identifying paraphrase (Ferreira *et al.*, 2018). Their approach depends on extracting features and classifying a pair of sentences based on the extracted features. The extracted features calculated based on lexical similarity, syntactic similarity, and semantic similarity. This approach is similar to the proposed approach which combines structure and word-to-word similarity. However, their approach doesn't assign a similarity value for sentences pair. Moreover, the proposed approach does not need labeled data.

Unlike the previous approaches, the proposed approach considers both syntactic and semantic similarities in measuring sentences similarity. Moreover, this paper proposes a new approach that combines structural similarity which is calculated based on Discourse Representation Structure (DRS) and word-to-word similarity which using word embeddings. According to the best of my knowledge, DRS haven't used before to calculate structural similarity between sentences. The proposed system doesn't need labeled data for training.

3 Sentences similarity

A lot of NLP applications, such as question answering, social media analysis, and plagiarism detection depend on sentences similarity. Consequently, performing high accuracy in measuring similarity between sentences is a crucial task. The proposed approach exploits structure information of sentences to improve measuring similarity between sentences. As shown in figure 1 the proposed system consists of three main components. The first component measures structural similarity, the second component calculates similarity based on words of sentences, and the third component calculates word order similarity. The inputs to the proposed system are two sentences and the output is a similarity value between 0 and 1. *Zero* means not related and *one* means completely similar. The following subsections explain these three components in details.

3.1 Structure based similarity

Information of sentence structure helps to assess the similarity between two sentences (Ma and Suel, 2016). As shown in figure 1 calculating structural similarity contains three steps. As a first step each sentence is parsed and the output is passed to semantic analyzer which outputs semantic graph representation equivalent to the sentence. Based on semantic graph representation for the sentences, the second step constructs relations similarity matrix which is used in the third step to calculate structure similarity. The following sub-sections explain details of these steps.

3.1.1 Relation extraction

In order to get the structure of a sentence, parsing process is applied. Moreover, semantic relations between words are extracted and sentence semantic graph is constructed based on extracted relations. In this graph nodes represent words and edges represent semantic relations between words. Structure similarity between sentences is calculated based on the constructed graphs. Moreover, there are two main steps for generating semantic graph for a sentence. The first step is parsing which outputs syntax tree for the sentence. The second step is extraction of semantic relations and constructing sentence graph.

In this research, C&C parser (Clark and Curran, 2007) is used to parser sentences. C&C parser is an advanced statistical parser widely used in NLP tasks (Augenstein*et al.*, 2012)(Clark *et al.*, 2009)(Baroni *et al.*, 2014). C&C parser supports some features which are exactly what is needed in this research. C&C parser contains many taggers such as Part Of Speech (POS) tagger and Combinatorial Categorial Grammar (CCG) supertagger. These taggers are highly efficient (Curran *et al.*, 2007). In addition, C&C contains Name Entity Recognizer (NER) which can determine ten different types of entities (organization, location, person, email, URL, first name, surname, title, quotation, and unknown name). Using C&C parser, the words in a sentence are tagged with POS

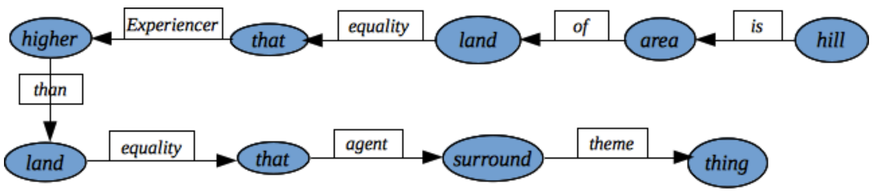

Figure 2: Discourse Representation Structure (DRS) graph representation for the sentence: ”A hill is an area of land that is higher than the land that surrounds it”

from penn tree bank (Marcus *et al.*, 1993). The output of parsing process is a syntax tree in which each node has POS tag, lemma, and name entity tag.

As a second step in generating sentence graph, Boxer system is used to get the semantic relations between words. Boxer system is a semantic construction toolkit (Augenstein*et al.*, 2012). Moreover, Boxer is an open domain software which semantically analyze English text. It is developed by Curran, Clark and Bos (Curran *et al.*, 2007). Based on the output of C&C parser, Boxer system builds semantic representation for the sentence. It depends on CCG and C&C parser to generate Discourse Representation Structure (DRS) for sentence. DRS is a representation corresponds to natural language text. Moreover, DRS captures the semantic of text and models it into related entities. DRS can be converted to other semantic format such as first-order-logic. The performance of Boxer system is very promising (Bos, 2008). The proposed approach uses semantic relations extracted using Boxer system to improve measuring sentences similarity.

Based on the output of Boxer system, a semantic graph representation for the sentence is generated. Figure 2 shows the graph representation generated for the sentence ”A hill is an area of land that is higher than the land that surrounds it”. This graph captures the structure information of the sentence. A relation between two words, as shown in figure 2, has interior node (the source word) and exterior node (the destination word). For example, the interior node for *theme* relation in figure 2 is the node *surround* and the exterior node is *thing*. As mentioned each node in the graph represents a word in the sentence, so the terms *external word* and *external node* are used interchangeably. Semantic similarity between sentences is measured based on the generated graphs.

3.1.2 Graph based similarity

Graph matching is widely used in many fields of computer science (Farouk *et al.*, 2018). In addition, graph matching is used for measuring similarity between documents (Hammouda and Kamel, 2002). In this paper structural similarity between sentences is measured using sentences' graphs. Based on the generated graphs for sentences, relations matrix is constructed. Rows of this matrix represent relations of the first graph and columns represent relations of the second graph. Each cell in the matrix is filled with similarity value between the row relation and the column relation. Cell i, j in the matrix is filled with similarity value between i^{th} relation in the first sentence and j^{th} relation in the second sentence. The similarity value between two relations is calculated in three steps:

1. Measuring similarity between names of relations.
2. Measuring similarity between interiors nodes.
3. Measuring similarity between exteriors nodes.

To calculate similarity between interiors words of both relations, word embedding approach is used. Cosine similarity between words' vectors is calculated as the similarity between these words. The following equation is used to calculate similarity between two relations.

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

Where $Sim(I_{R1}, I_{R2})$ is the similarity between interior word of R_1 and interior word of R_2. $Sim(E_{R1}, E_{R2})$ is the similarity between exterior word of R_1 and exterior word of R_2. $NameSim(R_1, R_2)$ is similarity between names of relations. This similarity depends on the meaning of relations. If the names are same, the similarity will be 1. If the relations are related the value will be higher than if they are not related.

Furthermore, the proposed approach calculates the structural similarity by guessing to what extend the relations of the first sentence are covered by the other sentence. This can be calculated based on the constructed matrix. In order to measure coverage of a relation in the other sentence, the maximum similarity between this relation and all relations in the second sentence is selected. Finally, the structural similarity between two sentences S_1 and S_2 is calculated as follow.

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

Where n is the number of relations in S_1 and W_{Ri} is the weight for the relation R_i. The relations weights are used to reflect the importance of different relations according to their effects on the sentence meaning. Table 1 shows the used wights of relations. There are different classifications for semantic relations(Jaworski and Przepirkowski, 2014)(Gildea and Jurafsky, 2002). However, these classifications agree on some standard semantic relations which include agent, theme, experiencer (Jaworski and Przepirkowski, 2014). Based on this standard and experiments, these values in table 1 are assigned. For example, the *agent* and *theme* relations are the most important in the sentence, so they have the highest weights.

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

Table 1: Relations weights that reflect importance of relations

3.2 Similarity based on word embedding

In addition to measuring structure similarity, the proposed approach calculates word-to-word similarity between sentences. Using word embedding in measuring similarity between words improves similarity measure between sentences(Kenter and Maarten, 2015). In the proposed approach the word-based similarity is calculated using the following equation

$$sim_w(S_1, S_2) = (\sum_{i=1}^{n} sim(w_i, S_2))/n \tag{3}$$

where n is the number of words in S_1 and the $sim(w_i, S_2)$ is the similarity between the word w_i, in sentence S_1, and Sentence S_2. The similarity between a word w and a sentence S is measured by selecting the max similarity between w and every word in the sentence S according to the following equation.

$$sim(w, S) = \max_{j=1}^{m} sim(w, w_j) \tag{4}$$

where $sim(w, w_j)$ is the similarity between the word w and word w_j in the sentence S. This similarity between words is measured using cosine similarity between words' vectors. This research adopts Google's pre-trained word embedding vectors, which is publicly available in https://code.google.com/archive/p/word2vec/, to be used in word-to-word similarity.

3.3 Word order similarity

Sometimes two sentences have same set of words and have different meaning. For example, consider these sentences: "The young man killed the old woman." and "The young woman killed the old man". The word order similarity is calculated to fix this issue (Li *et al.*, 2004). The idea of word order similarity is to measure order similarity of similar words. In other words, if the similar words have similar order this is better than similar words have different order. In addition to structure similarity and word-to-word similarity, word order similarity is calculated to improve the final similarity measure.

To calculate the word order similarity union set of words for the two sentences is constructed. Then for each sentence a vector is constructed to represent oder of this sentence's words with respect to the union set. the next section explains the process of constructing these vectors by example. The following equation is used to get the word order similarity based on the constructed order vectors.

$$Sim_{or}(S_1, S_2) = 1 - \frac{\|V_1 - V_2\|}{\|V_1 + V_2\|} \tag{5}$$

where V_1 is the order vector for S_1 and V_2 is the order vector for S_2.

Finally, the similarity between two sentences is calculated by combining structural similarity, word-to-word similarity and word order similarity. The following equation is used to measure the similarity.

$$Sim(S_1, S_2) = 0.5 * sim_{st}(S_1, S_2) + 0.3 * sim_w(S_1, S_2) + 0.2 * sim_{or}(S_1, S_2) \tag{6}$$

These wrights in equation 6 are assigned based on the importance of similarity measures. The structure similarity can reveal more information than word-to-word similarity. This is because structure similarity considers relations and words (source and destination words) of these relations. However, word-to-word similarity depends only on words of sentences. Consequently, higher weight is assigned to structure similarity. Moreover, low weight is assigned to word order similarity because it is used only to decrease similarity measure when same set of words have different order.

4 Sentences similarity calculation example

In order to clarify the idea of the proposed approach, this section explains calculating sentences similarity using an example. Consider the following two sentences:
S_1 ="A hill is an area of land that is higher than the land that surrounds it."
S_2 ="A mound of something is a large rounded pile of it."
In this example, the proposed steps to calculate sentences similarity are applied to the given sentences. As explained in the previous section, there are three main steps for calculating sentences similarity: Calculating structural similarity, calculating word-to-word similarity based on word embedding, and calculating word order similarity.

4.1 Calculating Structural Similarity

In order to calculate structural similarity between the given sentences, graph representation for each sentence is generated. The first step to generate this graph is to parse the sentences using C&C parser. The results of parsing process go as input to the Boxer system which generates semantic graph representation for the sentences. The results of this step is shown in table 2.

As shown in table 2, each line represents either a word or a relation. For example, line 9 **c0:hill:0 instance k3:x1 2 [hill]** represents the word *hill*

	First sentence	Second sentence
	A hill is an area of land that is higher than the land that surrounds it.	A mound of something is a large rounded pile of it.
1	c11:theme:1 ext k3:p1:x7 0 [ ]	c10:equality ext k2:p1:x3 0 [ ]
2	c14:equality ext k3:p1:x2 0 [ ]	c3:experiencer:-1 ext k2:p1:s1 0 [ ]
3	c3:equality ext k3:p1:x4 0 [ ]	c5:experiencer:-1 ext k2:p1:s2 0 [ ]
4	c7:equality ext k3:p1:x6 0 [ ]	c0:mound:0 instance k2:x1 2 [mound]
5	k3:p1 event c9:surround:0 0 [ ]	c2:of:0 ext k2:x2 1 [of]
6	c10:agent:-1 ext k3:p1:e1 0 [ ]	k2:p1 relation c10:equality 1 [is]
7	c5:experiencer:-1 ext k3:p1:s1 0 [ ]	k2:p1 referent k2:p1:x3 1 [a]
8	k3 referent k3:x1 1 [A]	c10:equality int k2:x1 4 [ ]
9	c0:hill:0 instance k3:x1 2 [hill]	c4:large:0 arg k2:p1:s1 1 [large]
10	k3:p1 relation c14:equality 1 [is]	c3:experiencer:-1 int k2:p1:x3 2 [ ]
11	k3:p1 referent k3:p1:x2 1 [an]	c6:rounded:0 arg k2:p1:s2 1 [rounded]
12	c14:equality int k3:x1 3 [ ]	c5:experiencer:-1 int k2:p1:x3 3 [ ]
13	c1:area:0 instance k3:p1:x2 2 [area]	c7:pile:0 instance k2:p1:x3 4 [pile]
14	c13:of:0 ext k3:p1:x3 1 [of]	c9:of:0 ext k2:p1:x4 1 [of]
15	c2:land:0 instance k3:p1:x3 2 [land]	c8:thing:12 instance k2:p1:x4 2 [it]
16	c13:of:0 int k3:p1:x2 3 [ ]	k2:p1 punctuation k2:p1:x4 3 [.]
17	k3:p1 referent k3:p1:x4 1 [that]	c9:of:0 int k2:p1:x3 5 [ ]
18	c3:equality int k3:p1:x3 3 [ ]	c2:of:0 int k2:x1 3 [ ]
19	k3:p1 surface k3:p1:s1 1 [is]	c1:thing:12 instance k2:x2 2 [something]
20	c5:experiencer:-1 int k3:p1:x4 2 [ ]	
21	c4:higher:0 arg k3:p1:s1 2 [higher]	
22	c12:than:0 ext k3:p1:x5 1 [than]	
23	k3:p1 referent k3:p1:x5 2 [the]	
24	c12:than:0 int k3:p1:s1 3 [ ]	
25	c6:land:0 instance k3:p1:x5 3 [land]	
26	k3:p1 referent k3:p1:x6 1 [that]	
27	c7:equality int k3:p1:x5 4 [ ]	
28	c9:surround:0 instance **k3:p1:e1 1 [surrounds]**	
29	c10:agent:-1 int k3:p1:x6 2 [ ]	
30	c8:thing:12 instance **k3:p1:x7** 1 [it]	
31	c11:theme:1 int k3:p1:e1 2 [ ]	
32	k3:p1 punctuation k3:p1:x7 2 [.]	

Table 2: Graph representation outputted from Boxer system for example sentences: the left column shows S_1 and the right column shows S_2

First sentence	Second sentence
• surround **theme** thing • hill **is** area • land **equality** that • that **agent** surround • that **experiencer** higher • area **of** land • higher **than** land	• mound **is** pile • pile **experiencer** large • pile **experiencer** rounded • mound **of** thing • pile **of** thing

Table 3: Relations of S_1 and S_2 in the form <interior> <**relation name**> <exterior>

in S_1. In this line **k3:x1** is the reference for the word *hill*. Semantic relations between words are represented as links between words. Each relation in the generated representation has two links. A link is for an interior word and the other link for exterior word, For example, line 1 **c11:theme:1 ext k3:p1:x7**, represents the exterior link for *theme* relation. where $c11$ is the relation id, *theme* is the relation name, *ext* means exterior which means the destination word of the relation, and $k3 : p1 : x7$ is the reference for exterior word. Moreover, line 31 **c11:theme:1 int k3:p1:e1** represents the link for interior word of *theme* relation in which k3:p1:e1 is the reference of the verb *surrounds*. These two lines (1 and 31) represent the theme relation between the words *surround* and *it*. Based on this semantic representation which captures structure of sentence, the sentences similarity is calculated.

The next step is constructing relations similarity matrix based on extracted relations. Firstly, relations belong to each sentence are determined. Table 3 shows the lists of relations in S_1 and S_2. Each relation in table 3 accompanied with its interior and exterior words.

Figure 2 shows the graph representation for the sentence S_1. Each node in this graph represents a word in the sentence and edges represent relations between entities. After determining the list of relations belong to each sentence the relations matrix is constructed by calculating similarity between relations (equation 1). Table 4 shows the constructed matrix for our example sentences.

As shown in the relations similarity matrix rows represent relations of the first sentence S_1 and columns represent relations of the sentence S_2. Each cell in this matrix is filled by the similarity between row relation and column relation. For example, the first cell in this matrix which is filled with the value 0.112836 represents the similarity between *theme* relation in S_1 and *is* relation in S_2. To calculate the similarity between these relations (*theme* and *is*), the similarity between interiors words is calculated. The interior word for *theme* relation is the verb *surround* and interior word for *is* relation is *mound*. These two words are expanded based on the context of each sentence. Relations such as *equality* and *is* are used for word expansion. For example, the word *mound* in S_2 is

relation	is	experiencer	experiencer	of	of
theme	0.112836	0.071832	0.0784489	0.428314	0.428314
is	0.323542	0.248751	0.232218	0.234756	0.234756
equality	0.0845531	0.12681	0.060439	0.148451	0.148451
agent	0.105591	0.100205	0.0913217	0.0805113	0.0805113
experiencer	0.0671812	0.175157	0.0737535	0.0422765	0.0422765
of	0.210981	0.253237	0.186867	0.376546	0.376546
than	0.0671812	0.109438	0.0430672	0.131079	0.131079

Table 4: Relations matrix for our example sentences

matching case	similarity value	notes
both relations have same name	1.0	
is relation and *equality* relation	0.7	these relations are used for word expansion
other cases	0.73	

Table 5: similarity value between names of relations

expanded to *pile*. The similarity between these words (*surround* and *mound*) is measured as the max similarity between the two lists of expanded words. Word embedding is used to calculate word-to-word similarity.

Because of *mound* word can be extended to *pile* word in this example, we measure both $Sim(surround, mound) = 0.0350571$ and $sim(surround, pile) = 0.173464$ and select the max value. Similarity for exterior words of these relations $Sim(thing, pile)$ is 0.135674. The total similarity between these relations (*theme* and *is*) is calculated according to equation 1: $(0.173464 + 0.135674)/2 * 0.73 = 0.112836$. The value 0.73 is the similarity between names of relations. Table 5 shows the proposed similarity values between relations' names. At this stage of research, these values are manually assigned based on experiments.

To get the structure similarity based on the constructed matrix equation 2 is used. The structure similarity between S_1 and S_2 is 0.325996.

4.2 Calculating word-to-word similarity

In order to calculate word-to-word similarity for example's sentences, word similarity matrix is constructed by measuring similarity between every pair of words from S_1 and S_2. Table 6 shows the word similarity matrix. The final similarity between S_1 and S_2, which calculated using equation 4, is 0.4605.

	hill	area	land	higher	than	land	surrounds	it
mound	0.462	0.082	0.069	0.0682	0.0146	0.069	0.0339	0.119
something	0.135	0.1168	0.114	0.1466	0.2666	0.114	0.0985	0.616
large	0.004	0.2193	0.1267	0.234	0.0778	0.1267	0.1399	0.131
rounded	0.174	0.084	0.0062	0.032	0.1185	0.006	0.054	0.0073
pile	0.277	0.0337	0.0661	0.0356	0.085	0.066	0.136	0.1786
it	0.139	0.1248	0.1815	0.188	0.283	0.1815	0.062	1

Table 6: words similarity matrix for the example sentences

4.3 Calculating word order similarity

On the other hand, order vector is constructed for each sentence to compute word order similarity between S_1 and S_2. The union vector for both sentences is [hill, area, land, higher, than, surrounds, it, mound, something, large, rounded, pile]. Based on this union vector, order vector for each sentences is constructed by putting the oder of the word in the union vector instead of the word itself. Order vector for S_1 is[1, 2, 3, 4, 5, 7, 8, 0, 0, 0, 0, 0] and order vector for S_2 is [0, 0, 0, 0, 0, 0, 6, 1, 2, 3, 4, 5]. According to equation 5, the value for word order similarity between S_1 and S_2 is 0.32239.

The final similarity between S_1 and S_2 is calculated according to equation 6 which combines the three calculated measures. $Sim(S_1, S_2) = 0.365626$

5 Experiments

The proposed approach has been implemented and tested using standard datasets to prove its effectiveness. In this experiment, the implemented system takes two sentences as input and measures the similarity between them. The output value of the implemented system is ranged between 0 and 1. In order to show the impact of using structural information, the proposed system is tested with structural information and without it against same dataset. Moreover, the system is compared to other systems to show the effectiveness the proposed approach.

5.1 Datasets

In order to prove the applicability and effectiveness of the proposed approach two datasets are selected to apply the implemented approach. Pilot short text semantic similarity benchmark dataset is chosen to evaluate the proposed system. It is one of the most widely used datasets in sentence similarity evaluation (Li *et al.*, 2006)(Lee *et al.*, 2014)(Islam and Inkpen, 2008). Originally this data set was created by Rubenstein and Goodenough to measure word similarity (Rubenstein and John, 1965). It contains 65 pairs of words. Li et al (Li *et al.*, 2006) 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 average value of 32 grades is considered as the similarity value between each pair of sentences. A full list of this dataset and human assessment is in (O'Shea *et al.*, 2008-b). Moreover, a subset of the Poilt benchmark contains 30 pairs of sentences is selected carefully to cover different similarity ranges (Lee *et al.*, 2014). This subset which called Li2006 dataset is widely used for evaluating sentences similarity approaches (Lee *et al.*, 2014)

Moreover, Microsoft Research Paraphrase (MSRP) dataset (Dolan *et al.*, 2004) is also widely used to evaluate sentence similarity techniques. It contains more than 5100 pairs of sentences. It was partitioned into two sets. The first set contains 3400 pair of sentences and is used as a training dataset. The other set contains around 1700 pairs of sentences and used for testing. Each pair is labeled by 1 (paraphrased) or 0 (not paraphrased). In this experiment the testing set is only used since our approach is unsupervised approach and doesn't need training data.

5.2 Results and discussion

In the selected Li2006 dataset similarity of each pair of sentences is assessed by the proposed system. Table 7 shows the results of the proposed system along with other previously proposed approaches. The results of Li approach(Li *et al.*, 2006), O'Shea's approach which depends on LSA (O'Shea *et al.*, 2008-a), STS Meth (Islam and Inkpen, 2008), Omiotis system (Tsatsaronis *et al.*, 2010), and grammar based similarity (Lee *et al.*, 2014) are included in table 7. In addition, the recent approach proposed by (Pawar and Mago, 2018) is also included in the comparison table. Moreover, the pearson correlation coefficient is calculated between each system results and human rating. The proposed approach has achieved the best correlation comparing to other systems, Table 8. The proposed system achieved 0.8813 peasron correlation with human similarity.

In addition, Spearman correlation is calculated to show how well the relationship between results of different system and human measured similarity. As shown in table 8 the proposed system also achieves the best Spearman correlation among all other systems.

On the other hand, the results of using MSRP dataset is also good. The proposed approach calculates similarity between each pair of sentences in MSRP dataset and assigns a value between 0 and 1. A threshold value (0.45) is used to convert the calculated similarity value to 0 or 1. The proposed approach achieves 0.72 accuracy. Table 9 shows the achieved results and other unsupervised approaches results. The results of the proposed approach are comparable to other system. The achieved accuracy is better than accuracy of Omiotis system (Tsatsaronis *et al.*, 2010) and grammar based approach. However, accuracy of the proposed approach is slightly less than the accuracy of Islam's approach (Islam and Inkpen, 2008) which is the best accuracy.

In order to show the effect of combining structural similarity and word based similarity, the results of the proposed system is compared with word-to-word similarity measure. In addition, the results of the proposed systems is compared

R&G number	human assessment	Li 2006	LSA	Islam	Atish	Omiotis	grammar based	proposed approach
1	0.01	0.33	0.51	0.06	0.023	0.11	0.22	0.109
5	0.01	0.29	0.53	0.11	0.07	0.1	0.06	0.133
9	0.01	0.21	0.51	0.07	0.015	0.1	0.35	0.074
13	0.1	0.53	0.53	0.16	0.292	0.3	0.32	0.201
17	0.13	0.36	0.58	0.26	0.366	0.3	0.41	0.278
21	0.04	0.51	0.53	0.16	0.231	0.24	0.44	0.212
25	0.07	0.55	0.6	0.33	0.279	0.3	0.07	0.268
29	0.01	0.34	0.51	0.12	0.133	0.11	0.2	0.214
33	0.15	0.59	0.81	0.29	0.762	0.49	0.07	0.334
37	0.13	0.44	0.58	0.2	0.1	0.11	0.07	0.242
41	0.28	0.43	0.58	0.09	0.045	0.11	0.02	0.232
47	0.35	0.72	0.72	0.3	0.161	0.22	0.25	0.298
48	0.36	0.64	0.62	0.34	0.54	0.53	0.79	0.349
49	0.29	0.74	0.54	0.15	0.299	0.57	0.38	0.34
50	0.47	0.69	0.68	0.49	0.253	0.55	0.07	0.282
51	0.14	0.65	0.73	0.28	0.302	0.52	0.39	0.208
52	0.49	0.49	0.7	0.32	0.842	0.6	0.84	0.395
53	0.48	0.39	0.83	0.44	0.89	0.5	0.18	0.401
54	0.36	0.52	0.61	0.41	0.783	0.43	0.32	0.257
55	0.41	0.55	0.7	0.19	0.315	0.43	0.38	0.323
56	0.59	0.76	0.78	0.47	0.977	0.93	0.62	0.49
57	0.63	0.7	0.75	0.26	0.477	0.61	0.82	0.319
58	0.59	0.75	0.83	0.51	0.892	0.74	0.94	0.444
59	0.86	1	1	0.94	0.856	1	1	0.869
60	0.58	0.66	0.83	0.6	0.898	0.93	0.89	0.507
61	0.52	0.66	0.63	0.29	0.934	0.35	0.08	0.301
62	0.77	0.73	0.74	0.51	1	0.73	0.94	0.511
63	0.56	0.64	0.87	0.52	0.7	0.79	0.95	0.444
64	0.96	1	1	0.93	0.873	0.93	1	0.872
65	0.65	0.83	0.86	0.65	0.854	0.82		0.55

Table 7: Results of the proposed approach and other approaches using Li2006 dataset

Method	Pearson correlation	Spearman correlation
Li 2006	0.8104	0.80542
LSA	0.8398	0.86974
Islam	0.8515	0.82871
Atish	0.7848	0.82834
Omiotis	0.8603	0.8907
Grammar based	0.7214	0.6518
proposed approach		
with structural	**0.8813**	**0.90363**
without structural	0.8414	0.80896

Table 8: Comparison between the proposed method and other methods

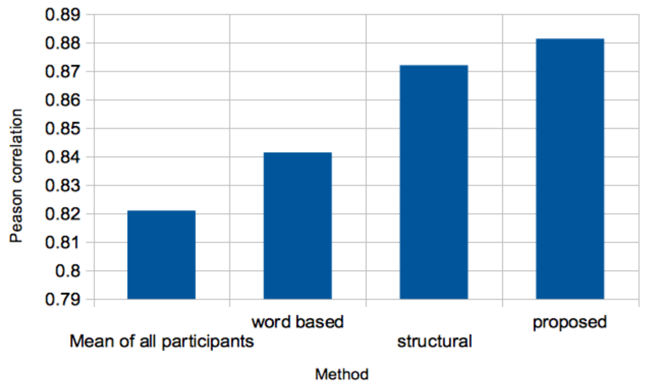

Figure 3: Effect of combining structural similarity and word based similarity

also with results of using structural similarity only. Figure 3 shows the result of the proposed system in the Pilot dataset comparing to other results. As shown in figure 3, the proposed method that combines both word based and structural based similarities gives best results and outperforms each method individually.

The proposed method performs better then others because it considers semantic relations between words. If the two sentences have similar set of words and different relations between words, the similarity value should be less than if they have similar words and similar relations. For example consider the following sentences.
S1 = The lion quickly hunts the big giraffe
S2 = The giraffe was killed by a lion
S3 = The giraffe kills a loin
Although these sentences have similar words, they have different relations. The similarity value between S1 and S2 should be larger than similarity between S1 and S3. Considering semantic relations between words will improve the mea-

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
Mamdouh	71.6	76.2	83.3	79.6
proposed approach	72.12	73.64	90.33	81.14

Table 9: Results of the proposed approach and other approaches using MSRP dataset

sured similarity between such sentences. The experiments show that exploiting structural information of sentences improves the calculation of similarity between sentences.

6 Conclusion

This paper presents an approach for measuring sentences similarity. The proposed approach combines structural similarity, word-to-word similarity and word order similarity approaches. Structural similarity is measured based on graph representation for sentences. C&C parser and Boxer system are used to generate semantic structure representation for sentences. Moreover, word embedding is used to calculate word-to-word similarity. In addition, word order similarity is considered to improve similarity measuring in case of similar words have different positions in the sentences. The proposed approach has been tested against standard benchmark datasets. Experiments demonstrate that the proposed method provides a similarity measure that shows a significant correlation to human intuition. Moreover, combining structure similarity and word based similarity improves the assessed similarity between sentences.

References

Augenstein I., Pad S., Rudolph S. (2012). LODifier: Generating Linked Data from Unstructured Text. In: The Semantic Web: Research and Applications. Vol. 7295 of LNCS. Springer 210-224

Baroni M., Bernardi R., and Zamparelli R. (2014). Frege in space: A program of compositional distributional semantics. Linguistic Issues in Language Technology, 9.

Bos J. (2008). Wide-Coverage Semantic Analysis with Boxer. In: Proceedings of Semantics in Text Processing, pp. 277-286. College Publications.

Clark S. and Curran J. (2007). Wide-Coverage Efficient Statistical Parsing with CCG and Log-Linear Models. In Computational Linguistics Vol. 33, No. 4

Clark S., Copestake A., Curran J, Zhang Y., Herbelot A., Haggerty J., Ahn B., Wyk C., Roesner J., Kummerfeld J., and Dawborn T. (2009). Large-Scale Syntactic Processing: Parsing the Web. Final Report of the JHU CLSP Workshop, Johns Hopkins University.

Curran, J.R., Clark, S., Bos, J. (2007). Linguistically Motivated Large-Scale NLP with C&C and Boxer. In: Proceedings of the ACL 2007 Demo Session. pp. 33-36.

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

Farouk M. (2018). Sentence Semantic Similarity based on Word Embedding and WordNet, In IEEE 13th International Conference on Computer Engineering and Systems (ICCES 2018), pp. 33-37.

Farouk M., (2019). Measuring Sentences Similarity: A Survey, Indian Journal of Science and Technology, Vol. 12, No. 25.

Farouk M., Ishizuka M. and Bollegala D., (2018). Graph Matching based Semantic Search Engine, 12th International Conference on Metadata and Semantics Research, Cyprus

Ferreira R., George D.C. Cavalcanti, Freitas F., Lins RD., Steven J. Simske, and Riss M. (2018). Combining sentence similarities measures to identify paraphrases, Computer Speech and Language journal, Vol. 47 No. C, pp 59-73.

Jaworski W., Przepirkowski A. (2014). Semantic roles in grammar engineering. In: Proceedings of the 3rd Join Conference on Lexical and Computational Semantics (*SEM 2014), Dublin, Ireland.

Gildea D. and Jurafsky D. (2002). Automatic labeling of semantic roles. Computational Linguistics, Vol. 28, No. 3, pp.245-288

Gomaa W. H., Fahmy A. A., (2013). A survey of text similarity approaches, International Journal of Computer Applications, Vol. 68, No. 13, pp. 13-18.

Hammouda K., Kamel M., (2002). Phrase-Based Document Similarity Based on an Index Graph Model, Proc. 2002 IEEE Int'l Conf. Data Mining (ICDM '02), pp. 203-210.

Islam A. and Inkpen D. (2008). Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data, Vol. 2, No. 2, article 10.

Kenter T., Maarten de Rijke, (2015). Short Text Similarity with Word Embeddings, Proceedings of the 24th ACM International Conference on Information and Knowledge Management, Melbourne, Australia, October 18-23

Lee M.C., Chang J.W., Hsieh T.C.: A grammar-based semantic similarity algorithm for natural language sentences, Sci. World J., 2014

Li Y., Bandar Z., McLean D., O'Shea J. (2004). A Method for Measuring Sentence Similarity and its Application to Conversational Agents, Proc.17th Int. Florida AI Research Society Conf. (FLAIRS 2004), pp. 820-825.

Li Y., McLean D., Bandar Z., O'Shea J., Crockett K. (2006). Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering, Vol. 18, No. 8, pp.1138-1149.

Liu H., Wang P. (2013). Assessing Sentence Similarity Using WordNet based Word Similarity. Journal of Software, Vol. 8, No. 6, pp.1451-1458.

Ma W., Suel T. (2016). Structural sentence similarity estimation for short texts. Proceedings of the Twenty-Ninth International Florida Artificial Intelligence Research Society Conference (FLAIRS), pp. 232-237.

Marcus M., Marcinkiewicz M.A., Santorini B. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, Vol. 19, No. 2, pp.313-330.

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

Miller GA. (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11, pp.39-41.

Navigli R., and Martelli F. (2019). An overview of word and sense similarity. Natural Language Engineering, Vol. 25, No. 6 pp. 693-714.

O'Shea J., Bandar Z., Crockett K., and McLean D. (2008). A comparative study of two short text semantic similarity measures. In Agent and Multi-Agent Systems: Technologies and Applications, Vol. 4953 of Lecture Notes in Artificial Intelligence, pp. 172-181.

O'Shea J., Bandar Z., Crockett K., and McLean D. (2008). Pilot short text semantic similarity benchmark data set: Full listing and description, Computing.

Pawar A., Mago V. (2018). Calculating the similarity between words and sentences using a lexical database and corpus statistics". arXiv preprint arXiv:1802.05667.

Qu R., Fang Y., Bai W., and Jiang Y. (2018). Computing semantic similarity based on novel models of semantic representation using Wikipedia", Information Processing & Management, Vol. 54, No. 6, pp.1002-1021.

Rubenstein H., and John B. G.(1965). Contextual correlates of synonymy. Communications of the ACM, Vol 8, No 10, pp.627-633.

Ryan K., Zhu Y., Salakhutdinov R., Richard S. Zemel, Torralba A., Urtasun R., and Fidler S. (2015). Skip-thought vectors. In the 28th International Conference on Neural Information Processing Systems, pp. 3294-3302.

Tsatsaronis G., Varlamis I., and Vazirgiannis M. (2010). Text relatedness based on a word thesaurus. Journal of Artificial Intelligence Research, Vol. 37, pp. 1-39.