Open Access **Baghdad Science Journal** **P-ISSN: 2078-8665**
Published Online First: Suppl. November 2022 **2022, 19(6): 1581-1591** **E-ISSN: 2411-7986**

DOI: https://dx.doi.org/10.21123/bsj.2022.7255

Short Text Semantic Similarity Measurement Approach Based on Semantic Network

Naamah Hussien Hameed*[1] ***Adel M. Alimi*[2]** ***Ahmed T. Sadiq*[1]**

[1]Computer Science Department, University of Technology, Baghdad, Iraq.
[2]REGIM Lab, ENIS, University of Sfax, Tunisia.
*Corresponding author: cs.20.50@grad.uotechnology.edu.iq
E-mail addresses: Adel.alimi@enis.tn, Ahmed.T.Sadiq@uotechnology.edu.iq

Received 30/3/2022, Accepted 8/8/2022, Published Online First 25/11/2022, Published 5/12/2022

This work is licensed under a Creative Commons Attribution 4.0 International License.

Abstract:

Estimating the semantic similarity between short texts plays an increasingly prominent role in many fields related to text mining and natural language processing applications, especially with the large increase in the volume of textual data that is produced daily. Traditional approaches for calculating the degree of similarity between two texts, based on the words they share, do not perform well with short texts because two similar texts may be written in different terms by employing synonyms. As a result, short texts should be semantically compared. In this paper, a semantic similarity measurement method between texts is presented which combines knowledge-based and corpus-based semantic information to build a semantic network that represents the relationship between the compared texts and extracts the degree of similarity between them. Representing a text as a semantic network is the best knowledge representation that comes close to the human mind's understanding of the texts, where the semantic network reflects the sentence's semantic, syntactical, and structural knowledge. The network representation is a visual representation of knowledge objects, their qualities, and their relationships. WordNet lexical database has been used as a knowledge-based source while the GloVe pre-trained word embedding vectors have been used as a corpus-based source. The proposed method was tested using three different datasets, DSCS, SICK, and MOHLER datasets. A good result has been obtained in terms of RMSE and MAE.

Keywords: Natural language processing, Semantic network, Semantic similarity, Text mining, Word embedding

Introduction:

Many NLP and text mining tasks require finding similarity scores between texts, tasks such as information retrieval [1], text classification, text summarization [2], sentiment analysis [3], automatic student short answers assessment, machine translation [4], etc. In the traditional similarity calculation methods, the texts are converted into a vector in vector space [5]. The vector is constructed using the concepts or words of the texts, the similarity between compared texts is then calculated as the cosine similarity of their vectors. This means the similarity score between compared texts is computed based on the number of common words between them. These methods work well with large texts the size of an article or document, based on the assumption that similar texts tend to share similar words, but cannot be relied upon in dealing with short texts with a sentence length or a few sentences, as two short texts can carry almost the same meaning and they do not have a single word in common. For example, the sentence “She is a beautiful woman” and the sentence “She is a nice girl” has similar meaning but do not share any common word except stop words. Here, the need arises to find a method that takes into account the semantic similarity aspects between the words of two texts which means taking into account synonyms, hypernyms, hyponyms, and other relationships between words.

Finding similarities between words is an important element of text similarity, which is subsequently utilized as a starting point for text similarity. Words can be similar in both lexical and semantic aspects. Words are lexically similar if their

character sequences are similar. Words are semantically similar if they have the same meaning, are opposites, are employed in the same way, are employed in the same context, and one word is a type of another. String-Based methods can be used to compute lexical similarity, whereas Corpus-Based or Knowledge-Based algorithms can be used to calculate semantic similarity. String-based algorithms ensure that a string comparison metric is utilized to compare the similarity of distinct character sequences. Corpus-Based algorithms use information acquired in a huge corpus to compute semantic similarity between words, whereas Knowledge-Based algorithms use the knowledge obtained from semantic networks to determine the semantic closeness of words [6, 7].

In this paper, a method has been presented for measuring semantic similarity between texts by building a semantic network that represents the relationship between the words of the texts, where each node represents a word with its part of speech (PoS) tag, and the edge represents the similarity degree between the nodes. To measure the similarity score between words. WordNet [8] as a knowledge-based source and GloVe [9] as a corpus-based source are used. Two sources are used to take the advantage of both and to avoid the problem of the lack of some words in one of the sources.

The next sections are structured as follows: Section 2 briefly reviews some similar works. Section 3 describes the semantic network. Section 4 explains the word-to-word similarity sources. Section 5 describes the proposed method for semantic similarity measurement between short texts. Section 6 details the results obtained. Section 7 presents the conclusion.

Related Work:

In [10] Liu and Wang, 2013 adopted a vector space model to consolidate word-to-word similarity. Initially, both sentences are converted to a bag-of-words form. The approach then creates a combined word set by combining sentence 1 and sentence 2. For every sentence, a semantic vector is created, with the combined word serving as a vector element. The highest similarity degree of a word pair between each word in the combined word set and each word in a sentence is represented by every element of the semantic vector. They develop similarity metrics based on concept vectors to assess the similarity of word pairs. Following the formation of each sentence's semantic vector, the cosine coefficient of these semantic vectors can be used to determine the sentence's similarity. This technique yields a precision of 0.738 and a recall of 0.902 in the paraphrase detection on the Microsoft Research Paraphrase Corpus (MSRP).

In [11] Croft et al, 2013 suggested lightweight semantic similarity (LSS), a short text similarity that integrates the vector space model with path length word-to-word similarity. The first stage in the procedure is to create a combined word set from the two sentences and use it as a vector space dimension. Every sentence is represented as a vector using the procedure. Evert word in the combined word set is evaluated for every word in a sentence. The sum of the word-to-word similarity score for each term in the combined word set is considered as the score of a vector component relating to that phrase. The process is recurring until each vector component (term) has a value. The approach creates a vector representation for the second sentence using a similar process. Cosine similarity on sentence vectors is used to calculate overall sentence similarity. The method's performance is measured on 65-word pairs from Rubenstein and Goodenough, with every word being substituted by its description from the Collin Cobuild lexicon. The LSS algorithm and human judgment are then used to assess the similarity of the noun-sentence pair's definitions. They attained a Pearson correlation of 0.807.

In [12] Kusner et al. 2015, produced word embedding from Google News corpora, using the word2vec approach advanced by Mikolov et al [13]. The term "word embedding" refers to the representation of words as a dense numerical vector. To quantify sentence similarity, the approach constructs the text as normalized bag-of-words vectors. The word mover distance (WMD) function is used to calculate the distance between the two sentences. The function determines the shortest cumulative distance that a word in one sentence must travel to match a word in the second sentence exactly. The Euclidean distance between the word embedding vectors is used to compute the distance between words. As a result of WMD calculation, the greater the space between two sentences, the less similar the two sentences will be.

In [14] Vu et al. 2014, used explicit semantic analysis (ESA) in conjunction with ROUGE (Recall-Oriented Understudy for Gisting Evaluation). ROGUE is a measure of lexical similarity based on n-gram co-occurrence information. They calculated text similarity using each technique, then use a linear combination and a tuning parameter to obtain the final similarity. They put the method to the test by creating their dataset

from Wikipedia articles. They achieved a Person correlation of 0.82.

Concerning the biomedical area, in [15] Soğancıoğlu et al. 2017, presented a technique to compute text similarity. They used the outcome of combining numerous sentence similarity metrics as an input for a supervised machine learning approach. Following text preparation, the technique assesses each sentence's knowledge-based, string-based, and corpus-based similarity. Each measurement's result was fed into the supervised regression model. They synthesized their dataset, which contained biomedical sentence pairings, for testing reasons. They calculated the Pearson correlation by comparing their result to the score of similarity of human judgment. A Pearson correlation of 0.836 was achieved using this strategy.

In [16] Pawar et al. 2018, suggested a technique for determining sentence similarity that takes the semantic and word position information into consideration. They used a method that was both knowledge-based and corpus-based. The approach creates a joint word set by combining two input texts. The sentences are then translated into a semantic vector using WordNet knowledge and a joint vocabulary set. The degree of similarity between compared words was considered using the shortest route between the two words and the deepness of the subsumer in WordNet. The mark of similarity between words was then weighted using information content obtained from a corpus and cosine similarity was applied to the two vectors. Order vectors were likewise produced and order similarity was determined using a similar approach. Lastly, the similarity was calculated by merging semantic and order similarity. Rubenstein and Goodenough word pairings were used to test the approach. The approach had a Pearson correlation of 0. 0.8794, which was rather good. This approach has a disadvantage, even though it produces promising results. Word sense disambiguation is not done, and there is a problem if sentences contain terms that are not in WordNet.

In [17] Yang et al. 2021, suggested a strategy for combining semantic and syntactic information in short text similarities. The semantic information is derived from semantic vectors of short texts, which are dynamically created by comparing short texts and term similarity. A constituency parse tree was used to retrieve syntactic information. The two parts were then linearly integrated. To address the phenomena of polysemy, they employed knowledge and corpora to express the meaning of phrases. They tested their method on semantic textual similarity (STS) tasks which contained 24 datasets. Good results were achieved in terms of the Pearson correlation coefficient, however, using a tree parser was computationally expensive which made the method unsuitable for real-time applications.

In [18] Lubis et al. 2021, proposed semantic similarity based on word embedding for the automatic short answer grading system. They trained the word2vec model on a full Wikipedia dump in Indonesia to obtain a word embedding vector. The student answer and correct answer were converted to sentence vectors by computing the average of their words vector and the semantic similarity was then computed as cosine similarity between their vectors. They tested their method on a dataset consisting of 224 student responses from a computer network engineering class. They achieved a Pearson correlation of 0.7.

In [19] Mijbel et al. 2021, suggested an approach for measuring semantic similarity between texts based on the semantic network and word description. The method began with text processing, parts of speech tagging, and building the semantic network. The semantic similarity was calculated from three aspects, which were the similarity of the nodes, the similarity of the parts of speech, and the similarity of the edge relationship, the final similarity was obtained as a linear combination of the three similarities. They tested the method on the DSCS dataset [20], 1.17 mean absolute error was achieved.

Through this review of previous works, the following points can be noted:

1- The knowledge-based methods can be limited when some words are not present in the lexical database used, especially with the use of informal words in texts.
2- The methods that depend on word embedding vectors are biased by the nature of the corpus that was used to extract the values of word embedding vectors, for example, the use of political corpus shows a great similarity between ‘Iraq’ and ‘Afghanistan’, while the use of cultural or historical corpus shows a similarity between ‘Iraq’ and ‘Mesopotamia’.

Our main contributions are as follows:
- Presenting a new hybrid method for short text semantic similarity measurement that integrates the knowledge-based and corpus-based semantic information.

- The proposed method is based on the semantic network where the semantic, syntactical, and structural knowledge of the sentence are considered in the calculation of the similarity degree.
- Our method is easy to implement, fast, and computationally inexpensive

Semantic Networks:

The semantic network is one of the ways of representing knowledge, where the nodes represent objects or concepts and the edges represent the binary relationship that connects two nodes. The network representation provides a pictorial representation of knowledge objects, their attributes, and the relationship between them [21]. Representing a text as a semantic network is the best representation of knowledge that comes close to the human mind's understanding of texts, where the semantic network reflects the semantic, syntactical, and structural knowledge of the sentence. The relationship between the nodes can be 'is a", 'a kind of', 'a part of' and so on. Fig. 1 is an example of a semantic network.

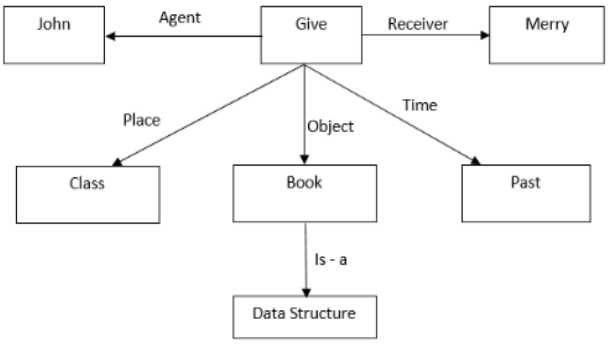

Figure 1. the semantic network of the sentence "John gave Merry a book of data structure in the class"

Word to Word Similarity Sources:

As mentioned, calculating the similarity between words is the cornerstone for measuring similarity between texts. In this section, the sources that have been used to compute the semantic similarity between words are described.

WordNet

WordNet is an example of a semantic network in which words—concepts—are linked by synonymy or meronymy links. WordNet is a lexical database with over 100,000 English words that are commonly used for knowledge-based semantic similarity approaches [22]. The lexicon is divided into nouns, verbs, adjectives, and adverbs by WordNet. These clusters of words are grouped to form synsets or synonym sets. A synset is a concept in which all of the terms have the same meaning. In certain syntax, the words in a synset are interchangeable. The definitions of these words, as well as pointers to other related synsets are included in a synset's knowledge. In WordNet, synsets are organized in a tree-like hierarchical structure, with many specialized terms at the bottom and a few general phrases at the top. Following trails of superordinate terms in "is a" or "is a sort of" (ISA) relations connects the lexical hierarchy. Each word rises the lexical tree until the two climbing paths meet to form a path between them. The subsumer is the synset at the intersection of the two climbing paths; a path connecting the two words is then identified through the subsumer. Counting synset links along the path between the two words yields the path length. Counting the levels from the subsumer to the top of the lexical hierarchy yields the depth of the subsumer. If a word is polysemous (meaning it has numerous meanings), there may be multiple pathways between the two terms [23]. In WordNet, numerous approaches have been developed for identifying semantic similarity between words and concepts. Path-based, information content (IC)-based, gloss-based, feature-based, and hybrid measures are the five categories of measures. The proposed method uses Wu and Palmer [24] measure to compute the similarity between words. Wu and Palmer define the similarity between two concepts as the equation below

$$sim(x1, x2) = \frac{2k}{a1+a2+2k} \quad \ldots\ldots\ldots\ldots\ldots\ldots 1$$

Where a1 and a2 denote the number of links from x1 and x2, to the deepest common subsumer x, and k to the number of links from x to the root of the taxonomy.

GloVe

Word embeddings are representations of words as a vector that maintain the basic linguistic link between words [25]. These vectors are computed by a variety of methods, including neural networks, word co-occurrence matrices, and representations based on the context of the word. [22]. Some of the most often used pre-trained word embeddings are word2vec [26], GloVe [9], fastText [27], BERT [28]. GloVe pre-trained word vector adopted in the proposed method besides WordNet as a word to word similarity resource.

GloVe, developed at Stanford University, employs a global word co-occurrence matrix based on the underlying corpus. It calculates similarity based on the fact that words that are similar to every other commonly transpire together. A single run across the underlying huge corpus is used to populate the co-occurrence matrix with occurrence values. The GloVe model was trained on five corpora, the majority of which were Wikipedia dumps. Words are chosen within a given context

window for constructing vectors because words further away have less significance to the context word in consideration. The GloVe loss function reduces the least square distance between the co-occurrence values in the context window and the global co-occurrence values. To discriminate words based on context, GloVe vectors were enhanced to generate contextualized word vectors [22].

Proposed Method:

The proposed method for measuring the semantic similarity between two texts is based on semantic network construction that represents the relationship between the elements of the two texts. Fig. 2 below shows the process of finding semantic similarity between two texts. The method starts with text preprocessing, where the input text is converted into a clean format that can be analyzed and processed. In the next step, the semantic network is built, which represents the binary relationships between the words of the two texts. The word-to-word semantic similarity is found through the lexical database (WordNet) and the pre-trained embedding vectors (GloVe). In the last step, the semantic similarity between the two texts is computed using the information provided by the constructed semantic network.

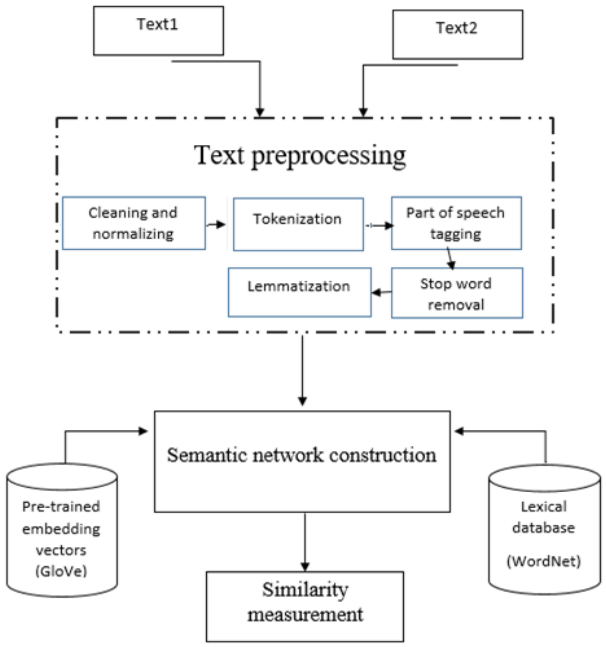

Figure 2. proposed method framework

Text Preprocessing

Text pre-processing is a necessary step to convert the text into a clean format that can be processed and analyzed. The text preprocessing process consists of several steps, which are as follows.

Cleaning and Normalizing

The original text often comes with some unwanted additions that do not affect the semantic meaning of the text, and the process of cleaning the text comes to remove these additions, such as duplicated whitespaces, special characters, HTML tags, punctuation marks, URL links, etc. after cleaning the text is converted to lowercase.

Tokenization

Tokenization is the procedure of dividing the text into smaller components known as tokens [29].

Part of Speech Tagging

Part-of-speech tagging is the procedure of giving a part-of-speech tag to each word in the text. This is done based on its meaning and its context. Tagging is a disambiguation task; ambiguous words have more than one possible part of speech and the goal is to find the correct tag for the situation[30].

Stop Word Removal

Stop words are a list of high-frequency words like pronouns (they, we, you), conjunctions like (for, and, while), etc. They have less impact on the semantic meaning of texts, so deleting these words is a suitable option in many NLP applications.

Lemmatization

It is the procedure of returning a word to its root form. For example, 'run', 'ran', and 'running', are all conjugations of the verb 'run'. In semantic similarity measurement applications, lemmatization is preferred on stemming because it avoids the overgeneralization of the stemmers and takes into account the PoS tag of the word where the word 'running' that has a noun PoS tag remains the same while 'running' with verb PoS tag reduced to 'run'.

Semantic Network Construction

Following the texts are preprocessed, the process of building a semantic network that represents the relationship between the two input texts begins. Each word of the two input texts with its PoS represents a node in this network, while the score of semantic similarity between the words represents the edges. To build this semantic network, the semantic similarity between each node from the first text and the nodes from the second text is found, and the highest value that links a node with the node of the other text represents the weight of this node.

The node to node similarity scores are identified in two case

Case 1: node A and node B have the same word hence the relation assigned to 1

Case 2: node A and node B are not the same, here the external sources are used to compute the similarity between two nodes, calculating Wu & Palmer similarity with WordNet and cosine

similarity of GloVe word embedding vector of two nodes then the relation between node A and node B assigned to the highest score of those two similarities.

Algorithm 1 below shows the process of semantic network construction.

Algorithm 1: semantic network construction

```
Input: two list of words of preprocessed texts
Output: semantic network with nodes weights
Begin:
Node_list_1← get all words of preprocessed text1
Node_list_2← get all words of preprocessed text2
Semantic_net = { }
Nodes_weight_1 = { }
Nodes_weight_2 = { }
For each node1 in Node_list_1:
        Weights1 = [ ]
        For each node2 in Node_list_2:
                If node1 are equal to node2:
                        Add 1 to Weights1
                        Semantic_net [node1, node2] ←1
                Else:
                        Add WordNet similarity (node1, node2) to Weights1
                        Add glove similarity (node1, node2) to Weights1
                        Semantic_net [node1, node2] ← max (WordNet similarity, glove similarity)
        End for
        Nodes_weight_1 [node1] ← max (Weights1)
End for
For node2 in Node_list_2:
        Weights2 = [ ]
        For node1 in Node_list_1:
                If node2 are equal to node1:
                        Add 1 to Weights2
                        Semantic_net [node2, node1] ← 1
                Else:
                        Add WordNet similarity (node2, node1) to Weights2
                        Add glove similarity (node2, node1) to Weights2
                        Semantic_net [node2, node1] ← max (WordNet similarity, glove similarity)
        End for
        Nodes_weight_2 [node2] ← max (Weights2)
End for
Return (Nodes_weight_1, Nodes_weight_2)
End
```

Semantic Similarity Measurement

Following the semantic network is constructed now both texts have a list of nodes with their associated weights that represent their relationship with other text. To compute semantic similarity between the two texts, calculate how the first text is similar to the second text by summing text1 nodes weights that are greater than the threshold (Ɵ) value divided by the number of its nodes. In the same way, the similarity of the second text to the first text is calculated, and the final similarity score is the average of two similarities.

Let S1 be the similarity of text1 with text2

Let S2 be the similarity of text2 with text1

Let S be the final similarity

$$S1 = \frac{A1}{N1} \text{ ……..2}$$

$$S2 = \frac{A2}{N2} \text{ ……..3}$$

Where A1, A2 is the summation of weights of text1 nodes and weights of text2 nodes which is greater than Ɵ value respectively, and N1, N2 is the number of nodes of text1 and text2 respectively

The final similarity is computed as follows.

$$S = \frac{(S1+S2)}{2} \text{ ……...4}$$

Illustrative Example

To illustrate how the proposed method calculates the semantic similarity between two sentences, let's take this example

Text1: "A boy of young age is playing in the park with his mother"

Text2: "A young child and his mum are playing in the field."

In the first step, the input texts are entered into the text preprocessing process. The output of this process is a list of words with their part of speech tags for each text.

List1: "[('boy', 'noun'), ('young', "adjective"), ('age', 'noun'), ('play', 'verb'), ('park', 'noun'), ('mother', 'noun')]"

List2: "[('young', 'adjective'), ('child', 'noun'), ('mum', 'noun'), ('play', 'verb'), ('field', 'noun')]"

These two lists enter the stage of building the semantic network, where the degree of similarity between each word of the first text is calculated with all the words of the second text, and the highest degree of similarity is considered as a weight of this node. The similarity degree obtained from WordNet and GloVe pre-trained embedding vectors and the highest value between them is adopted.

('boy', 'young'): 0.705
('boy', 'child'): 0.909
('boy', 'mum'): 0.545
('boy', 'play'): 0.340
('boy', 'field'): 0.5
('young', 'young'): 1
('young', 'child'): 0.75
('young', 'mum'): 0.571
('young', 'play'): 0.4
('young', 'field'): 0.533
('age', 'young'): 0.541
('age', 'child'): 0.565
('age', 'mum'): 0.5
('age', 'play'): 0.6
('age', 'field'): 0.5
('play', 'young'): 0.4
('play', 'child'): 0.294
('play', 'mum'): 0.461
('play', 'play'): 1
('play', 'field'): 0.8
('park', 'young'): 0.705
('park', 'child'): 0.705
('park', 'mum'): 0.545
('park', 'play'): 0.5
('park', 'field'): 0.875
('mother', 'young'): 0.631
('mother', 'child'): 0.664
('mother', 'mum'): 0.962
('mother', 'play'): 0.705
('mother', 'field'): 0.666

So the weights of text1 nodes are:
'boy': 0.909
'young': 1
'age': 0.6
'play': 1
'park': 0.875
'mother': 0.962

By the same way the text2 nodes weights values calculated

('young', 'boy'): 0.705
('young', 'young'): 1
('young', 'age'): 0.541
('young', 'play'): 0.4
('young', 'park'): 0.705
('young', 'mother'): 0.631
('child', 'boy'): 0.909
('child', 'young'): 0.75
('child', 'age'): 0.565
('child', 'play'): 0.294
('child', 'park'): 0.705
('child', 'mother'): 0.664
('mum', 'boy'): 0.545
('mum', 'young'): 0.571
('mum', 'age'): 0.5
('mum', 'play'): 0.461
('mum', 'park'): 0.545
('mum', 'mother'): 0.962
('play', 'boy'): 0.340
('play', 'young'): 0.4
('play', 'age'): 0.6
('play', 'play'): 1
('play', 'park'): 0.5
('play', 'mother'): 0.705
('field', 'boy'): 0.5
('field', 'young'): 0.533
('field', 'age'): 0.5
('field', 'play'): 0.8
('field', 'park'): 0.875
('field', 'mother'): 0.666

So the weights of text2 nodes are:
{'young': 1
'child': 0.909
'mum': 0.962
'play': 1,
'field': 0.875

Fig. 3 below shows the semantic network that was constructed between the two texts

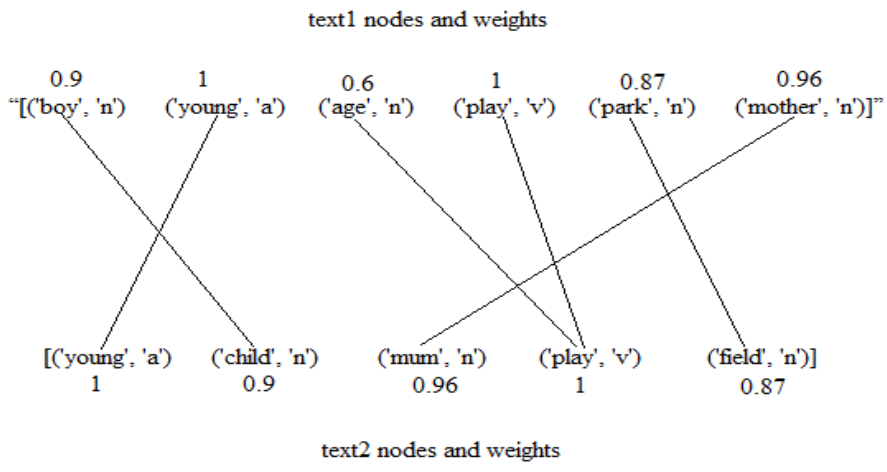

Figure 3. The semantic network of illustrative example

Now, calculating the degree of semantic similarity between the two texts is conducted. This is done by calculating the degree of similarity of the first text with the second text by summing the values of the weights of the first text nodes that are greater than Ɵ and dividing by the number of its nodes. In the same way, we calculate the similarity of the second text with the first text, and the degree of total similarity is the average between the two similarities.

The value of Ɵ is inversely proportional to the semantic similarity score between the texts. The higher the Ɵ value, the less similarity between the two texts and vice versa. To find the semantic properties of words and use them to the maximum extent possible while keeping noise to a minimum, several Ɵ values have been tested and found that the best results have been obtained with Ɵ values between "0.7 to 0.85"

So the sum of text1 nodes weights that are greater than Ɵ (0.7) are: 4.747
The number of nodes: 6
S1=4.747/6
S1= 0.791

The sum of text2 nodes weights that are greater than Ɵ (0.7) are: 4.747
The number of nodes: 5
S2=4.747/5
S2=0.949

The final similarity is
S = (s1+s2)/2
Final similarity= 0.87

Experimental Result:

The proposed method was implemented using Python programming language, and several text processing libraries provided by the language were used. To evaluate the proposed method, it was tested on three different datasets, these datasets are different and varied in their size and domain. The first is a small group of sentencespairs in the field of computer science, the second is a large group that contains English sentences pairs for general purposes, and the third is within the field of automated assessment of students' answers.

Datasets Description

DSCS Dataset

Domain-Specific Complex Sentence (DSCS) Semantic Similarity dataset [20]. It comprises 50 pairings of sentences from the computer science field, with associated similarity scores supplied by 15 human annotators, and the similarity score was calculated by averaging the replies of the 15 annotators. The similarity score between sentence pairs is determined on a scale of 0 to 5, with 0 representing complete dissimilarity and 5 indicating complete similarity.

SICK Dataset

The Sentences Involving Compositional Knowledge (SICK) [31]dataset is made up of around ten thousand English sentence pairs that were constructed from two sources: the 8K ImageFlickr data collection and the SemEval 2012 STS MSR-Video Description dataset. Each pair of sentences has two aspects of annotation: relatedness and entailment. The degree of relatedness sorts from 1 to 5, the entailment relation is categorical, consisting of neutral, contradiction, and entailment.

Mohler Dataset

The Mohler [32] dataset consists of 10 assignments with four to seven questions and two tests with ten questions each. These assignments/exams were given to students in an introductory computer science class at the University of North Texas. There are 87 questions in total, and each question has a standard answer provided by the examiner. Each question was answered by 26 to 31 students. Each answer in the assignment is scored on a scale of 0 (not correct) to 5 (completely correct) by two evaluators who are experts in computer science. The standard score of each answer is the average of the two evaluators' scores.

Results

Mean absolute error (MAE), and root mean squared error (RMSE) has been calculated as evaluation metrics to assess the proposed methods. The best results achieved with the three datasets are shown in Table 1 below. The method was tested with several Ɵ values, and it is clear that the best results are obtained with Ɵ value between (0.7, 0.85) except in the case of the Mohler dataset, where the best results were obtained with no threshold used. This is explained by the fact that the results given by the evaluator tend to be high, which makes each word, even if it has a little degree of weight, influential in the value of total similarity.

Table 1. The obtained results in terms of MAE and RMSE

dataset	threshold (Θ)	MAE	RMSE
DSCS	0.7	0.74	0.92
SICK	0.85	0.68	0.89
MOHLER	Nan	0.81	1.04

For further detailed results, Table 2 below shows the difference between the semantic similarity values of the proposed method and the human similarity value, which shows that the difference between the predicted score and the actual score does not exceed '1' (from range 0-5) in most cases.

Table 2. the difference between actual and predicated similarity score

Difference	DSCS	SICK	MOHLER
< 1	33	7217	1622
< 2	15	2377	501
< 3	2	246	114
< 4	0	0	21
< 5	0	0	6

Out of 50 sentences pair in the DSCS dataset, in 33 pairs the difference between our method score and human score is less than 0.5 (the score scale from 0 to 5), and less than 1.5 in 48 pairs. With the SICK dataset, our method gave very close similarity score to the human judgment score in about 73% of compared text pairs. Our results with the Mohler dataset were also good, where 75% of cases, the degree given by our method close to that given by the professor's assessor. Table 3 provides a comparison between our results and the results of similar previous work.

Table 3. Comparison with other studies.

		tf-idf[32]	ESA[32]	LSA[32]	Mijbel et al [19]	Our method
DSCS	RMSE					0.92
	MAE				1.17	0.74
SICK	RMSE					0.89
	MAE					0.68
Mohler	RMSE	1.085	1.031	1.065		1.04
	MAE				1.8	0.81

Our method got better or competitive results compared to the previous works. Compared with Mijbel et al [19], which is the closest approach to our method, it is based on the semantic network. The results showed a significant improvement in our method in reducing the error rate in terms of MAE on both DSCS and Mohler datasets.

Discussion

The proposed method has achieved good and encouraging results, as shown in Table 2, the difference between the estimated similarity score of the proposed method and the actual value is less than 1 (from range 0-5) in most cases. After examining the few cases in which the difference was large, it was found that the reasons were due to the following:

1. One of the two texts has its meaning expressed in a mathematical form, which causes the system to fail to measure the similarity between the mathematical and textual expressions.
2. One of the two texts expresses a meaning very briefly in one or two words, while the comparative text is much longer.
3. The presence of spelling errors in the student's answer (in the Mohler dataset), which the assessor overlooked and considered that the answer was correct.

In general, our method showed encouraging results, as our method gives a degree of similarity between two short texts compared close to that given by a human evaluator, but some limitations can be observed which are:

1- Automatic correction of spelling errors was not adopted, adopting the correction of spelling errors in future work may give better results.
2- The word order is not taken into account, for example, the sentence "Ahmad bought Ali's car" and the sentence "Ali bought Ahmed's car" is considered to be completely similar. However, even for human judgment, the two sentences remain related.

Conclusion:

A method for the semantic similarity measurement between texts has been presented in this paper, based on the semantic network.

Knowledge-based and corpus-based semantic information were combined to build the semantic network. WordNet lexical database and GloVe pre-trained vectors have been combined to

calculate word-to-word similarity. The method is simple, effective, and fast to implement. The results that have been achieved are good and can be improved in the future. The threshold (Θ) value has an important effect on the results. Choosing a higher Θ value leads to lower similarity values between the compared texts and vice versa. The experimental results showed that the best Θ value ranges from 0.7 to 0.85 with some exceptions. Utilizing word embedding vectors that are trained on domain-specific corpora and domain-specific lexical databases may give better results. For future work, incorporating machine learning techniques that take the information provided by the semantic network as features to predict the value of semantic similarity seems a suitable option that can add more accuracy to the process of determining the semantic similarity of texts.

Authors' declaration:

- Conflicts of Interest: None.
- We hereby confirm that all the Figures and Tables in the manuscript are ours. Besides, the Figures and images, which are not ours, have been given the permission for re-publication attached with the manuscript.
- Ethical Clearance: The project was approved by the local ethical committee in University of Technology.

Authors' contributions statement:

N.H.H., A.M.A. and A.T.S. contributed to the design and implementation of the research, to the analysis of the results and to the writing of the manuscript.

References:

1. Liu J, Kong X, Zhou X, Wang L, Zhang D, Lee I, et al. Data mining and information retrieval in the 21st century: A bibliographic review. Comput Sci Rev. 2019; 34: 100193.
2. El-Kassas WS, Salama CR, Rafea AA, Mohamed HK. Automatic text summarization: A comprehensive survey. Expert Syst Appl. 2021; 165: 113679.
3. AL-Jumaili AS. A hybrid method of linguistic and statistical features for Arabic sentiment analysis. Baghdad Sci J. 2020; 17(1 (Suppl.)): 0385-0385.
4. Moussallem D, Wauer M, Ngomo A-CN. Machine translation using semantic web technologies: A survey. J Web Semant. 2018; 51: 1-19.
5. Singh R, Singh S. Text similarity measures in news articles by vector space model using NLP. J Inst Eng (India): B. 2021; 102(2): 329-338.
6. Sánchez Rodríguez I. Text similarity by using GloVe word vector representations [Master thesis]: Polytechnic University of Valencia; 2017.
7. Hassani H, Beneki C, Unger S, Mazinani MT, Yeganegi MR. Text mining in big data analytics. Big Data Cogn Comput. 2020; 4(1): 1.
8. Lee YY, Ke H, Yen TY, Huang HH, Chen HH. Combining and learning word embedding with WordNet for semantic relatedness and similarity measurement. J Assoc Inf Sci. 2020; 71(6): 657-670.
9. Qiu X, Sun T, Xu Y, Shao Y, Dai N, Huang X. Pre-trained models for natural language processing: A survey. Sci China Technol Sci. 2020; 63(10): 1872-1897.
10. Liu H, Wang P. Assessing Sentence Similarity Using WordNet based Word Similarity. J Softw. 2013; 8(6): 1451-1458.
11. Croft D, Coupland S, Shell J, Brown S, editors. A fast and efficient semantic short text similarity metric. 13th UK workshop on computational intelligence (UKCI); 2013.
12. Kusner M, Sun Y, Kolkin N, Weinberger K, editors. From word embeddings to document distances. International conference on machine learning. Proc Int Conf Mach Learn. 2015; 37: 957-966. Available from: https://proceedings.mlr.press/v37/kusnerb15.html.
13. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J. Distributed representations of words and phrases and their compositionality. Adv Neural Inf Process Syst. 2013; 26.
14. Vu HH, Villaneau J, Saïd F, Marteau P-F, editors. Sentence similarity by combining explicit semantic analysis and overlapping n-grams. Proc Int Conf TSD . 2014: Springer. Available from: https://doi.org/10.1007/978-3-319-10816-2_25.
15. Soğancıoğlu G, Öztürk H, Özgür A. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. Bioinformatics. 2017; 33(14): i49-i58.
16. Pawar A, Mago V. Calculating the similarity between words and sentences using a lexical database and corpus statistics. arXiv preprint arXiv:180205667. 2018.
17. Yang J, Li Y, Gao C, Zhang Y. Measuring the short text similarity based on semantic and syntactic information. Future Gener Comput Syst. 2021; 114: 169-180.
18. Fetty Fitriyanti Lubis MDWTSYRAPAAA. Automated Short-Answer Grading using Semantic Similarity based on Word Embedding. Int J Technol 2021; 12(3): 291-319.
19. Mijbel SH, Liatsis P, Sadiq AT. Text Similarity Approach based on Semantic Networks and Words Description. Des Eng. 2021: 15217-15228.
20. Chandrasekaran D, Mago V. Domain Specific Complex Sentence (DCSC) Semantic Similarity Dataset. arXiv preprint arXiv:201012637. 2020.
21. Chowdhary K. Fundamentals of artificial intelligence. New Delhi: Springer; 2020. Available from: https://doi.org/10.1007/978-81-322-3972-7.
22. Chandrasekaran D, Mago V. Evolution of semantic similarity—A survey. ACM Comput Surv. 2021; 54(2): 1-37.

23. Li Y, McLean D, Bandar ZA, O'shea JD, Crockett K. Sentence similarity based on semantic nets and corpus statistics. IEEE Trans Knowl Data Eng. 2006; 18(8): 1138-1150.
24. Araque O, Zhu G, Iglesias CA. A semantic similarity-based perspective of affect lexicons for sentiment analysis. Knowl Based Syst. 2019; 165: 346-359.
25. Rodriguez PL, Spirling A. Word embeddings: What works, what doesn't, and how to tell the difference for applied research. J Polit. 2022; 84(1): 101-115.
26. Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:13013781. 2013.
27. Bojanowski P, Grave E, Joulin A, Mikolov T. Enriching word vectors with subword information. Trans Assoc Comput Linguist. 2017; 5: 135-146.
28. Devlin J, Chang M-W, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.
29. Al Maksur I, Muhajir M. MyBotS Prototype on Social Media Discord with NLP. Baghdad Sci J. 2021; 18(1 (Suppl.)): 0753-0753.
30. Chiche A, Yitagesu B. Part of speech tagging: a systematic review of deep learning and machine learning approaches. J Big Data. 2022; 9(1): 1-25.
31. Pawar A, Mago V. Challenging the boundaries of unsupervised learning for semantic similarity. IEEE Access. 2019; 7: 16291-16308.
32. Mohler M, Bunescu R, Mihalcea R, editors. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies. Portland, Oregon, USA. 2011: 752–762 Available from: https://aclanthology.org/P11-1076.

نهج قياس التشابه الدلالي للنص القصير على أساس الشبكة الدلالية

نعمة حسين حميد[1] **عادل عليمي[2]** **احمد طارق صادق[1]**

[1] قسم علوم الحاسوب، الجامعة التكنولوجية، بغداد، العراق
[2] المدرسة الوطنية للمهندسين بصفاقس، جامعة صفاقس، صفاقس، تونس

الخلاصة:

يلعب تقدير التشابه الدلالي بين النصوص القصيرة دورًا بارزًا بشكل متزايد في العديد من المجالات المتعلقة بتعدين النص وتطبيقات معالجة اللغة الطبيعية ، خاصة مع الزيادة الكبيرة في حجم البيانات النصية التي يتم إنتاجها يوميًا. الأساليب التقليدية لحساب درجة التشابه بين نصين بناءً على الكلمات التي يتشاركانها لا تعمل بشكل جيد مع النصوص القصيرة. لأن نصين متشابهين يمكن كتابتهما بعبارات مختلفة من خلال استخدام المرادفات. نتيجة لذلك ، يجب مقارنة الجمل الموجزة من حيث المعنى الدلالي. في هذا البحث ، يتم تقديم طريقة قياس التشابه الدلالي بين النصوص والتي تجمع بين المعلومات الدلالية القائمة على المعرفة والنصوص لبناء شبكة دلالية تمثل العلاقة بين النصوص المقارنة وتستخلص درجة التشابه بينها. يمثل تمثيل النص كشبكة دلالية أفضل تمثيل معرفي يقترب من فهم العقل البشري للنصوص ، حيث تعكس الشبكة الدلالية المعرفة الدلالية والنحوية والهيكلية للجملة. تمثيل الشبكة هو تمثيل مرئي لأشياء المعرفة وصفاتها وعلاقتها. تم استخدام قاعدة بيانات WordNet المعجمية كمصدر قائم على المعرفة بينما تم استخدام متجهات تضمين الكلمات المدربة مسبقًا من GloVe كمصدر مستند إلى النصوص. تم اختبار الطريقة المقترحة باستخدام ثلاث مجموعات بيانات مختلفة ، مجموعات بيانات SICK و DSCS و MOHLER. تم الحصول على نتائج جيدة بصيغة RMSE و MAE.

الكلمات المفتاحية: معالجة اللغة الطبيعية، الشبكة الدلالية، التشابه الدلالي، تعدين النص، تضمين الكلمات .