JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 37, 617-633 (2021)
DOI: 10.6688/JISE.202105_37(3).0008

Estimation in Semantic Similarity of Texts

MANH HUNG NGUYEN AND DINH QUE TRAN
Department of Information Technology
Posts and Telecommunications Institute of Technology
Hanoi, 12157 Vietnam
E-mail: mhnguyen@ptit.edu.vn(nmh.nguyenmanhhung@gmail.com); quetd@ptit.edu.vn

The semantic similarity of texts or documents has been widely studied in various areas including natural language processing, document comparison, artificial intelligence, semantic web, *etc*. Several similarity measures have been proposed but they are usually tied to special application domains or to data representation of various types. The purpose of this paper is to present a model for estimation in semantic similarity of texts based on similar sentences in structure of subjects, verbs and objects. And in turn, the semantic similarity of these components in the structure of sentences is estimated by means of the basic semantic similarity of words. The model is evaluated with two experiments: direct similarity and relative similarity among texts. The experimental results indicate that the proposed model is better than some baseline models in some circumstances.

Keywords: semantic computing, text mining, text similarity, sentence similarity, word similarity

1. INTRODUCTION

The semantic similarity between texts or documents is widely studied in various areas including natural language processing, document semantic comparison, artificial intelligence, semantic web, *etc*. These research issues could be formulated with two forms:

- The direct form: Given two texts D_1 and D_2, the problem is how to measure similarity of two such documents and evaluate their similar degree.
- The indirect form: Given a text D, and a set of texts $SD = \{D_1, D_2 ... D_n\}$, the problem is to determine which text in the set SD is the most similar to the text D.

There are several similarity measures proposed in the literature as listed in Table 1. Technically, these models could be considered from a viewpoint with three levels: (i) The matching level; (ii) The level of relation among words in a sentence or in the text; and (iii) The scope level of model.

In the matching level, a model could be only based on the lexical matching, or based on the semantic matching. In the lexical matching based approaches, words are compared only based on their lexical structure, and/or based on the statistic of words in texts. For instances, the models of Buscaldi *et al*. [1], Lintean and Rus [2], Proisl *et al*. [3], Sultan *et al*. [4]. The main advantage of these models is simple processing and the precision of

Received February 22, 2019; revised April 22 & June 15 & July 5 & July 21, 2019; accepted July 30, 2020.
Communicated by Chao-Lin Liu.

the statistic-based models could be increased when the texts are longer. However, their limitation is that these models could not recognize the similarity among the words which have same meaning but different lexical structures.

Table 1. Summary of recent proposed models.

Models	Matching		Relation	Scope		
	WordNet	ontology	syntax/ corpus	word	sentence	text
Arora *et al.* [5]			✓	✓	✓	✓
Buscaldi *et al.* [1]				✓		✓
Han *et al.* [6]	✓		✓	✓		✓
Hanig *et al.* [7]			✓	✓	✓	✓
Lee *et al.* [8]	✓		✓	✓	✓	
Lintean and Rus [2]				✓	✓	
Nguyen and Tran [9]		✓		✓	✓	
Proisl *et al.* [3]				✓		✓
Sultan *et al.* [4]			✓	✓	✓	
Vu *et al.* [10]			✓	✓	✓	✓
Our model		✓	✓	✓	✓	✓

In the semantic matching approaches, words are compared in their semantics. It may be based on WordNet (*e.g.*, the works of Lee *et al.* [8]), or based on an ontology (Nguyen and Tran [9, 11]).

At the level of relation between words in a sentence or a text, the models could be classified in two groups. One group is that it is not concerned with any relationship between words and considers each text (or sentence) as a bag of words. Therefore, when comparing two texts (or two sentences), these models compare each pair of words being collected from the text (or the sentence) such as the model of Lintean and Rus [2]. Such a consideration is suitable for applications requiring the semantic comparison between two sets of keywords or key ideas of two texts (or sentences). However, they might also result in lack of some meaning of texts being formed from the syntax/corpus, or meaning of the word order.

Another group is that it takes into account the relationship among words of texts (or sentences) to compare them. These relations are of various types. For instances, the simplest relationship of words is their order. It is considered in the model given by Nguyen and Tran [9], or in the *words surrounding a word* model of Sultan *et al.* [4]. A more complex relationship among words is the syntax of sentence: subject, verb, adverb, object, *etc.*. and/or the corpus of texts: category, keywords, form, *etc.* It has been investigated, for instance, by Lee *et al.* [8]. Intuitively, the more the number of criteria are taken into account for comparing texts, the more accurate the model is. However, a selection of the syntax/corpus of sentences/texts might result in some increase in the complexity of model. This makes the model less applicable, specially in real-time processing applications.

At the scope level, most of models are able to apply for comparing the similarity among words. Some models may enable to compare sentences, but not texts (Lee *et al.* [8], Lintean and Rus [2], Nguyen and Tran [9, 11], Sultan *et al.* [4]). Meanwhile, some models do the opposite (Buscaldi *et al.* [1], Proisl *et al.* [3]).

This paper presents a model for estimation of similarity among texts based on the semantic similarity of their sentences. We propose a similar measure of two sentences based on the semantic similarity of their structures with three parts: subjects, verbs and objects. In turn, the semantic similarity of the components in structures of sentences is estimated by means of the similarity between two words.

The paper is organised as follows. Section 2 presents the model of semantic similarity of texts. Section 3 describes experiments to validate, evaluate the proposed model as well as compare it with some related works. Section 4 is the conclusion and perspectives.

2. THE MODEL OF SEMANTIC SIMILARITY OF TEXTS

Our proposed model takes two texts as its input and at the output, it returns the similar degree between two texts. This model is composed of five levels from bottom to top:

- *Word level*: Estimating the semantic similarity of two words based on their relationship in an ontology and their lexicon (presented in Section 2.1).
- *Sequence of words level*: Estimating the semantic similarity of two sequences of words based on the similarity of words (presented in Section 2.2).
- *Sentence structure level*: Estimating the semantic similarity of two subjects, two verbs and two objects of sentences (presented in Section 2.3).
- *Sentence level*: Estimating the semantic similarity of two sentences based on the similarity of their structure parts: subject, verb and object (presented in Section 2.3).
- *Text level*: Estimating the semantic similarity of two texts based on the similarity of each pair of sentences in the two texts (presented in Section 2.4).

These five levels will be presented in detail in the next sections.

2.1 Similarity of Two Words

This section presents the estimation in semantic similarity of two words. We consider the following cases: (i) Two words are in the same ontology; (ii) Both of them are not in any ontology; and (iii) Only one of them is in an ontology.

2.1.1 Semantic similarity of concepts in an ontology

In this section, the similarity of two concepts (words) in an ontology is presented. We consider an *ontology* as a 2-tuple $\mathscr{G} =< \mathscr{C}, \mathscr{V} >$, in which $\mathscr{C}$ is a set of nodes corresponding to concepts defined in the ontology and $\mathscr{V}$ is a set of edges representing relationships between two nodes in $\mathscr{C}$. In this paper, rather than considering the properties of nodes, we focus on relationship between concepts. A relationship in $\mathscr{V}$ is defined as follows: If $x, y \in \mathscr{C}$ and $< x, y > \in \mathscr{V}$, then x is called the parent of y, and y is the child of x. An ontology could be represented in a tree form, in which each node has a unique parent, but may have several child nodes. In this model, we therefore define some concepts on the ontology tree:

- The *nearest common ancestor concept* of two concepts c_i and c_j, denoted $CA(c_i,c_j)$ is the nearest common parent node on the ontology tree of two nodes c_i and c_j.

- The *path length* between concepts c_i and c_j in an ontology, denoted $L(c_i,c_j)$, is the length of the shortest path from node c_i to node c_j on the ontology tree, an edge is counted as an unit of the path length.

Let's c_i and c_j be two concepts defined in an ontology whose root node is *root*. The semantic similar measure between c_i and c_j is defined as follows:

$$s_{ont}(c_i,c_j) = f_{path}(L(c_i,c_j), \frac{L(CA(c_i,c_j), root)}{max(L(c_i,root), L(c_j,root))}) \qquad (1)$$

where *root* is the root node of the ontology tree; $CA(c_i,c_j)$ is the *nearest common ancestor concept* of c_i and c_j in the ontology tree; $L(x,y)$ is the path length between node x and node y in the ontology tree; $f_{path}(x,y)$ is a path-based semantic similarity function.

Suppose that $\Re$ is the set of real numbers and $[0,1]$ is the unit interval. A function $f_{path} : \Re \times [0,1] \to [0,1]$ is called a *path-based semantic similarity function* if it satisfies the following conditions:

(i) $f_{path}(0,1) = 1$. It means that if two concepts are identical, then their semantic similarity is maximal.

(ii) $f_{path}(l_1,r) \geqslant f_{path}(l_2,r)$ if $l_1 \leqslant l_2$. The longer the path from each of them to the other is, the less similar they are.

(iii) $f_{path}(l,r_1) \geqslant f_{path}(l,r_2)$ if $r_1 \geqslant r_2$. The shorter the path from the *nearest common ancestor concept* to the root of the ontology is, the less similar they are.

For example, it is easy to see that the function $f_{path}(x,y) = \frac{y}{(x+1)^{\frac{1}{3}}}$ is the path-based semantic similarity function. We will make use of this function for our experiments in this paper.

2.1.2 Lexical similarity of words with the same core

In reality, there are many of words with the same original core word, but not all of them are always included in an ontology. In order to measure the similarity between these words (called *core similarity*), we use the following concepts:

- The *length* of a word w_i, denoted as $length(w_i)$, is the number of characters formulating the word.

- The *lexical distance* between a word w_i and its original core word w_0, denoted $d(w_i,w_0)$, is the total number of characters that may be added or deleted from the word w_1 to become the original core word w_0.

- The *lexical distance* between two words w_i and w_j, which have the same original core word $w_0 \notin \{w_i,w_j\}$, is the total distance from each of them to the common core word: $d(w_i,w_j) = d(w_i,w_0) + d(w_j,w_0)$.

Suppose that w_0 is the original core word of two words w_i and w_j. We define a *lexical similarity* of w_i and w_j as follows:

$$s_{lex}(w_i,w_j) = f_{lex}(d(w_i,w_j), length(w_i) + length(w_j)) \tag{2}$$

where $d(w_i,w_j)$ is the lexical distance between w_i and w_j; $f_{lex}(x,y)$ is a lexicon-based similarity function.

A function $f_{lex} : \Re \times \Re \rightarrow [0,1]$ is called a *lexicon-based similarity function* if it satisfies the following conditions:

(i) $f_{lex}(0,l) = 1$. If two words are identical, their distance is 0. Then their lexical similarity is maximal.

(ii) $f_{lex}(d_1,l) \geqslant f_{lex}(d_2,l)$ if $d_1 \leqslant d_2$. The longer the lexical distance of two words is, the less the lexical similarity they are.

For example, it is easy to see that the function $f_{lex}(x,y) = 1 - \frac{x}{y}$ is the lexicon-based similarity function. In this paper, we will make use of this function for our experiments.

2.1.3 Transitive semantic similarity of two words

Suppose that c_i, c_j and c_k are concepts, in which only c_j and c_k belong to the same ontology and c_i while c_j have the same core word. The transitive semantic similarity between concepts c_i and c_k via concept c_j is determined by the following formula:

$$s_{tran}(c_i,c_j,c_k) = f_{tran}(s_{lex}(c_i,c_j), s_{ont}(c_j,c_k)) \tag{3}$$

where $s_{lex}(c_i,c_j)$ is the lexical similarity between c_i and c_j; $s_{ont}(c_j,c_k)$ is the semantic similarity between c_j and c_k; $f_{tran}(x,y)$ is a transitive-based similarity function.

A function $f_{tran} : [0,1] \times [0,1] \rightarrow [0,1]$ is a *transitive-based similarity function* if it satisfies the following conditions:

(i) $0 \leqslant f_{tran}(x,y) \leqslant y$. The transitive semantic matching of c_i and c_k must be not bigger than the semantic matching between c_j and c_k because there is no semantic relation between c_i and c_k on the ontology.

(ii) $f_{tran}(x_1,y) \leqslant f_{tran}(x_2,y)$ if $x_1 \leqslant x_2$. The higher the lexical similarity between c_i and c_j is, the higher the transitive semantic similarity between c_i and c_k via c_j is.

(iii) $f_{tran}(x,y_1) \leqslant f_{tran}(x,y_2)$ if $y_1 \leqslant y_2$. The higher the semantic similarity between c_j and c_k is, the higher the transitive semantic similarity between c_i and c_k via c_j is.

For example, it is easy to see that the function $f_{tran}(x,y) = x * y$ is the transitive-based similarity one. In this paper, we will make use of this function for our experiments.

2.1.4 General semantic similarity of two words

Let's c_i and c_j be two words or concepts. In order to measure their semantic similarity in general, we consider the following cases:

- If c_i and c_j are both in the same ontology, then their general semantic similarity is their ontology-based semantic similarity;

- If either c_i or c_j is in an ontology, the other is not, their general semantic similarity is their transitive semantic similarity;
- If neither c_i nor c_j is in an ontology, we consider that they have not any semantic relation but may have some lexical similarity.

Accordingly, the semantic similarity between c_i and c_j is determined by the following formula:

$$s_{word}(c_i,c_j) = \begin{cases} s_{ont}(c_i,c_j) \text{ if } c_i,c_j \in \text{an ontology} \\ s_{tran}(c_i,c_j,c_k) \text{ if } c_i \text{ or } c_j \in \text{an ontology} \\ s_{lex}(c_i,c_j) \text{ if } c_i,c_j \notin \text{any ontology} \end{cases} \tag{4}$$

where $s_{ont}(c_i,c_j)$ is the semantic similarity based on ontology, $s_{tran}(c_i,c_j)$ is the transitive similarity, and $s_{lex}(c_i,c_j)$ is the lexical similarity between c_i and c_j.

2.2 Similarity of Two Sequences of Words

In this section, we consider the similarity of the two sequences of words at two levels: *Semantic similarity* and *Order similarity*.

2.2.1 Semantic similarity of two sequences of words

Let's $S_1 = \{s_1^1, s_1^2, ..., s_1^m\}$ and $S_2 = \{s_2^1, s_2^2, ..., s_2^n\}$ be two sequences of words. We create a *common sequence* of these sequences $S_{12} = S_1 + S_2 = \{s^1, s^2, ..., s^{m+n}\}$ and then construct a *semantic vector* $T = (t^1, t^2, ..., t^{m+n})$ as follows:

$$t^i = min(max(s_{word}(s^i, s_1^k)), max(s_{word}(s^i, s_2^v))), k = 1...n, v = 1...m \tag{5}$$

where $s_{word}(x,y)$ is the semantic similarity between the two words x and y.

The semantic similarity between two sequences of words S_1 and S_2 is defined as follows:

$$s_{sss}(S_1,S_2) = f_{sss}(T) = f_{sss}(t^1, t^2, ..., t^{m+n}) \tag{6}$$

where f_{sss} is a Semantic-Sequence-Similarity (SSS) function.

A function $f_{sss} : [0,1]^k \to [0,1]$ is a semantic similar function of two sequences of words, denoted *Semantic-Sequence-Similarity (SSS)*, if it satisfies the following conditions:

(i) $f_{sss}(0,0,...,0) = 0$

(ii) $f_{sss}(1,1,...,1) = 1$

(iii) $f_{sss}(X_1) \leqslant f_{sss}(X_2)$ if $\|X_1\| \leqslant \|X_2\|$. The bigger the magnitude of the vector T is, the higher the semantic similarity between S_1 and S_2 is.

For instance, the function $f(x_1,x_2,...,x_n) = \frac{\sum_{i=1}^n x_i}{n}$ is a semantic-sequence-similarity one. In this paper, we will use this function for our experiments.

2.2.2 Order similarity of two sequences of words

Let's $S_1 = \{s_1^1, s_1^2, ..., s_1^m\}$ and $S_2 = \{s_2^1, s_2^2, ..., s_2^n\}$ be two sequences of words. We also formulate a *common sequence* of these sets $S_{12} = S_1 + S_2 = \{s^1, s^2, ..., s^k\}$ and construct two corresponding *ordered vectors* $T_i = (t_i^1, t_i^2, ..., t_i^k), i = 1,2$ as follows:

$$t_i^j = \begin{cases} l \text{ if } s^j = s_i^l \in S_i \\ 0 \text{ if } s^j \notin S_i \end{cases} \tag{7}$$

The order similarity between two sequences of words S_1 and S_2 is determined by the formula:

$$s_{oss}(S_1, S_2) = f_{oss}(d_1, d_2, ...d_{m+n}) \tag{8}$$

where:

$$d_i = \begin{cases} \dfrac{|t_1^i - t_2^i|}{max(m,n)} \text{ if } min(t_1^i, t_2^i) \neq 0 \\ 1 \text{ if } min(t_1^i, t_2^i) = 0 \end{cases} \tag{9}$$

and $f_{oss}(d_1, d_2, ...d_{m+n})$ is an Order-Sequence-Similarity (OSS) function.

A function $f_{oss} : \Re^n \to [0,1]$ is an order similar function of two sequences of words, denoted *Order-Sequence-Similarity (OSS)*, if it satisfies the following conditions:

(i) $f_{oss}(0, 0...0) = 1$. The order similarity between S_1 and S_2 is the highest when two vectors T_1 and T_2 are identical.

(ii) $f_{oss}(x_1, x_2, ...x_n) \leqslant f_{oss}(y_1, y_2, ...y_n)$ if $x_i \geqslant y_i$ with all $i = 1, ..., n$. The more the vector T_1 is similar to the vector T_2, the higher the order similarity between S_1 and S_2 is.

For example, the function $f(x_1, x_2, ...x_n) = 1 - \frac{\sum_{i=1}^n x_i}{n}$ is a order-sequence-similarity one. In this paper, we will use this function for our experiments.

2.3 Similarity of Two Sentences

A simple sentence is considered to be composed of three parts: subject, verb and object (direct or indirect). Compound sentences could be divided into several simple ones and then, they could be investigated in the same manner. Therefore, we estimate the similarity of two simple sentences by means of the similarity of their subjects, verbs and objects. The extraction of subjects, verbs and objects from a simple sentence is based on the Stack-augmented Parser-Interpreter Neural Network (SPINN) approach proposed by Bowman *et al*. [12].

2.3.1 Semantic similarity of two subjects of sentences

In order to semantically compare the subjects of sentences, we consider two kinds of subjects: (i) Both subjects are a single word; and (ii) At least a subject is composed of many words.

a. Semantic similarity of two single word subjects

If two subjects are noun, we apply the similarity computation of two words from Section 2.1.4:

$$S_{ss}(S_1,S_2) = s_{word}(S_1,S_2) \tag{10}$$

In the case there is at least a subject is not noun (*e.g.*, a pronoun), we define a semantic matching of two subjects as follows:

- *Matching*: Two subjects are *Matching* if they are identical. For instance, *I* and *I*, *We* and *We*, *Mark* and *Mark* are *Matching*.
- *Replace-Matching*: Two subjects are *Replace-Matching* if the one could be replaced by the other. For instance, *A bird* and *It*, *People* and *They*, *A person* and *He* are *Replace-Matching*.
- *Sub-Matching*: Two subjects are *Sub-Matching* if one could be a part of other. For instance, *Some student* and *Many students*, *Somebody* and *Everybody* are *Sub-Matching*.
- *Private-Matching*: Two subjects are *Private-Matching* if they represent the same object but these objects belong to different owners. For instance, *My pet* and *Her pet*, *Its color* and *Their color* are *Private-Matching*.
- *No-Matching*: Two subjects are *No-Matching* if they do not belong to any kinds of matching defined above.

The semantic similarity of two single word subjects S_1 and S_2 of sentences is defined as follows: Given S_1 and S_2 are two single word subjects of sentences, the semantic similarity of two subjects S_1 and S_2 of sentences is determined by the formula:

$$s_{ss}(S_1,S_2) = \begin{cases} s_{word}(S_1,S_2) \text{ if } S_1,S_2 \text{ are nouns} \\ 1 \text{ if } S_1,S_2 \text{ are Matching} \\ a \text{ if } S_1,S_2 \text{ are Replace-Matching} \\ b \text{ if } S_1,S_2 \text{ are Sub-Matching} \\ c \text{ if } S_1,S_2 \text{ are Private-Matching} \\ 0 \text{ if } S_1,S_2 \text{ are No-Matching} \end{cases} \tag{11}$$

where $s_{word}S_1,S_2$ is the semantic similarity of two words S_1 and S_2; (*a*, *b*, *c*) is a 3-tuple such that:

(i) $1 > a > b > c > 0$

(ii) $(a+b+c) \leqslant 1$

(iii) $(b+c) \leqslant a$

In our experiments, we use the 3-tuple *(a,b,c) = (0.5, 0.3, 0.2)*.

b. Semantic similarity of two multi-word subjects

In turn, we consider each subject as a sequence of words. Then we could apply the semantic and the order similarity of two sequences of words. Let $s_{sss}(S_1,S_2)$ and $s_{oss}(S_1,S_2)$ be respectively the semantic and the order similarities between S_1 and S_2. The semantic similarity of two multi-word subjects S_1 and S_2 of sentences is defined as follows:

$$S_{ss}(S_1,S_2) = f_{sos}(s_{sss}(S_1,S_2), s_{oss}(S_1,S_2)) \tag{12}$$

where $f_{sos}(x,y)$ is a Semantic-and-Order-Similarity (SOS) function.

A function $f_{sos} : [0,1] \times [0,1] \to [0,1]$ is a semantic and order similar function of two objects of sentences, denoted *Semantic-and-Order-Similarity (SOS)*, if it satisfies the following conditions:

(i) $f_{sos}(x_1,y) \leqslant f_{sos}(x_2,y)$ if $x_1 \leqslant x_2$. The higher the semantic similarity $s_{sss}(S_1,S_2)$ is, the higher the semantic similarity between S_1 and S_2 is.

(ii) $f_{sos}(x,y_1) \leqslant f_{sos}(x,y_2)$ if $y_1 \leqslant y_2$. The higher the order similarity $s_{oss}(S_1,S_2)$ is, the higher the semantic similarity between S_1 and S_2 is.

For example, the function $f_{sos}(x,y) = \dfrac{x+y}{2}$ is a semantic and order similar function. We will use it for our experiments.

2.3.2 Semantic similarity of two verbs of sentences

We consider a verb of a sentence as a word. Therefore, the semantic similarity of two verbs is considered as the one between two words. In fact, a verb could be in various forms. For instance, it could be in the past, present, or future tense. It could be in active or passive form. Then, the passive verb will be converted to active form before comparing to other sentences. In this section, we consider some forms of verb: (i) Verb in original form; (ii) Verb in temporal form; and (iii) Verb in direct form with other verb.

a. Verb in the original form

In the reality, there are some verbs which have more or less the same meaning with each other in some contexts. For example: *see* and *meet*, *like* and *love*, *etc*. However, for the simplicity of our model, we make use of these assumptions: If two original verbs are identical, then their similarity is 1; If two original verbs are different, then their similarity is 0. So the similarity of two original verbs is estimated by following formula:

$$s_{ov}(V_1,V_2) = \begin{cases} 1 \text{ if } V_1 = V_2 \\ 0 \text{ if } V_1 \neq V_2 \end{cases} \tag{13}$$

b. Verb in the temporal form

In tense, a verb could be in past, present, future, *etc*. form. We make use of these assumptions: First, if two verbs are in the same temporal form, then their similarity is 1. Second, if two verbs are in different temporal forms, then their similarity is 0.5. So the tense similarity of two verbs is estimated by following formula:

$$s_{tv}(V_1,V_2) = \begin{cases} 1 \text{ if } V_1 \text{ and } V_2 \text{ are in the same temporal form} \\ 0.5 \text{ if } V_1 \text{ and } V_2 \text{ are in different temporal form} \end{cases} \tag{14}$$

c. Verb in direct compose with other verb

In the reality, there are some verbs which could be placed after other special verbs, such as: can, like, want, *etc*. In this model, we call *composed verb* which uses these assumptions: If two original verbs are placed after the same verb, or there are no any verbs before them, then their composed verb similarity is 1. Otherwise, it is 0. So the composed verb similarity of two verbs is estimated by following formula:

$$s_{cv}(V_1,V_2) = \begin{cases} 1 \text{ if } V_1 \text{ and } V_2 \text{ are placed after the same verb} \\ 0 \text{ otherwise} \end{cases} \tag{15}$$

d. Semantic similarity of two verbs in general

The semantic similarity of two verbs, in general, is defined as follows: Given V_1 and V_2 be the two considered verbs of sentences, the semantic similarity of two verbs V_1 and V_2 in sentences is determined by the formula:

$$S_{vs}(V_1,V_2) = s_{ov}(V_1,V_2) * s_{tv}(V_1,V_2) * s_{cv}(V_1,V_2) \tag{16}$$

where $s_{ov}(V_1,V_2)$, $s_{tv}(V_1,V_2)$ and $s_{cv}(V_1,V_2)$ are respectively the original verb similarity, the temporal similarity, and composed similarity of verb V_1 and V_2.

2.3.3 Semantic similarity of two objects of sentences

We use the same principle in the case of two subjects:

- When two objects are single words, their similarity is that of two words: $s_{os}(O_1,O_2) = s_{word}(O_1,O_2)$
- When at least an objects is composed of multi-words, we will consider each object as a sequence of words.

In the second case, the similarity between two objects of sentences is based on: (i) the semantic similarity between two sequences of words and, (ii) the order similarity between two sequences of words.

Let's O_1 and O_2 be two objects of sentences, it means that they are two sequences of words. Suppose that $S_{sss}(O_1,O_2)$ and $S_{oss}(O_1,O_2)$ are respectively the semantic similarity and the order similarity between O_1 and O_2. The semantic similarity between two objects O_1 and O_2 of sentences is determined by the formula:

$$S_{os}(O_1,O_2) = f_{sos}(s_{sss}(O_1,O_2), s_{oss}(O_1,O_2)) \tag{17}$$

where $f_{sos}(x,y)$ is an Semantic-and-Order-Similarity (SOS) function.

2.3.4 Semantic similarity of two sentences based on their structure

Let's $P_1 = \{S_1,V_1,O_1\}$ and $P_2 = \{S_2,V_2,O_2\}$ be two simple sentences. Suppose that $S_{ss}(S_1,S_2)$, $S_{vs}(V_1,V_2)$, and $S_{os}(O_1,O_2)$ are respectively the semantic similarity on their subjects, verbs, and objects. The semantic similarity between two sentences P_1 and P_2 is determined by the formula:

$$S_{sen}(P_1,P_2) = f_{sen}(S_{ss}(S_1,S_2), S_{vs}(V_1,V_2), S_{os}(O_1,O_2)) \tag{18}$$

where $f_{sen}(x,y,z)$ is a sentence similarity (sen) function.

A function $f_{sen} : [0,1]^3 \to [0,1]$ is a semantic similar function of two single sentences, denoted *Sentence similarity (sen)*, if it satisfies the following conditions:

(i) $f_{sen}(x_1,y,z) \leqslant f_{sen}(x_2,y,z)$ if $x_1 \leqslant x_2$. The higher the similarity between their two subjects $S_{ss}(S_1,S_2)$ is, the higher the semantic similarity between P_1 and P_2 is.

(ii) $f_{sen}(x,y_1,z) \leqslant f_{sen}(x,y_2,z)$ if $y_1 \leqslant y_2$. The higher the similarity between their two verbs $S_{vs}(V_1,V_2)$ is, the higher the semantic similarity between P_1 and P_2 is.

(iii) $f_{sen}(x,y,z_1) \leqslant f_{sen}(x,y,z_2)$ if $z_1 \leqslant z_2$. The higher the similarity between their two objects $S_{os}(O_1,O_2)$ is, the higher the semantic similarity between P_1 and P_2 is.

In our experiments, the function $f_{sen}(x,y,z) = \dfrac{x+y+z}{3}$ is used as the semantic similar function.

2.4 Similarity of Two Texts

A text document could be considered as a sequence of ordered sentences. However, in comparing the semantic similarity between two documents, the order similarity of two sequences of sentences is much less important than the semantic similarity of them. So we consider a document as a sequence of sentences without order.

Let's $D_1 = \{P_1^1, P_1^2, ..., P_1^m\}$ and $D_2 = \{P_2^1, P_2^2, ..., P_2^n\}$ be two documents with their sequence of sentences. We formulate a *common sequence* of these two sequences of sentences $D_{12} = (D_1) + (D_2) = \{P^1, P^2, ..., P^{m+n}\}$. And then construct their *semantic similarity vector* $T = (t^1, t^2, ..., t^{m+n})$ as:

$$t^i = min(max(S_{sen}(P^i, P_1^k)), max(S_{sen}(P^i, P_2^v))), k = 1..m, v = 1..n \tag{19}$$

where $S_{sen}(X,Y)$ is the semantic similarity between two sentences X and Y.

The semantic similarity between two documents D_1 and D_2 is determined by the formula:

$$S_{ds}(D_1,D_2) = f_{sss}(T) = f_{sss}(t^1, t^2, ..., t^{m+n}) \tag{20}$$

where f_{sss} is a Semantic-Sequence-Similarity (*SSS*) function.

3. EVALUATION

This section presents two experiments which are designed to evaluate the proposed model. First, an experiment uses the direct similarity of two texts (a pair). It responds to the first question presented in the introduction section: how much are the two texts similar? Second, an experiment uses the relative similarity among two pairs of text. It responds to the second question in the introduction section: Given a set of texts, which text is the most similar to some texts?

3.1 Experiment 1: Direct Similarity Comparison

This experiment uses the direct similarity among two texts to validate the model. We present the used method and then the results of experiment.

3.1.1 Method

This experiment uses the text-similarity-dataset of Dzikovska *et al.* [13], published in SemEval-2015 (http://alt.qcri.org/semeval2015/task2/). This dataset contains totally about 3000 samples of five categories (images, headlines, answer-student, answer-forum, belief). Each sample includes three parts: First, a pair of sentences to compare. Second, an average score of two sentences, this is the average of scores received from a set of student's answers. Each answer gives a score from 0 (two sentences are on different topics) to 5 (two sentences are completely equivalent, as they mean the same thing). Third, an average confidence score self-judged by the answers, noted on 100%. So this score is in the interval [0,100]. The experiment is performed as follows:

- For each sample (pair of texts), the model proposed in this paper is used to estimate the similarity between the two sentences in the pair, called the score of our model.

- Repeat for all samples in each category in the dataset (images, headlines, answer-student, answer-forum, belief), we get a set of scores of our model for each category.

- Compare each set of scores of our model to the set of scores of the corresponding category in the dataset by using the *Pearson correlation coefficient* as follows:

$$r = \frac{\sum_{i=1}^{n}(m_i - \overline{m})(s_i - \overline{s})}{\sqrt{\sum_{i=1}^{n}(m_i - \overline{m})^2}\sqrt{\sum_{i=1}^{n}(s_i - \overline{s})^2} \tag{21}$$

  where n is the number of samples in the sample set; m_i is the score between the two texts in the i^{th} pair calculated from the model; $\overline{m}$ is the mean value of all m_i; s_i is the score of the two texts in the i^{th} pair given in the dataset; $\overline{s}$ is the mean value of all s_i.

- Compare the output parameter from our model to top five best models in the competition of Textual similarity task in Semeval2015 [13].

3.1.2 Results

These results are presented in the Table 2, the first five rows are the results of top five models collected from the Textual similarity task in Semeval2015 [13]. Each model is separated by five categories and the weighted mean value for all categories. These models are sorted by the weighted mean value for all categories. Although our model results are not higher than the best model of Semeval2015 [13], but these results are very competitive: In each category, our model results are always ranked in the top five. Consequently, the weighted mean value from our model are also ranked in the top five regarding the Textual similarity task ranking in Semeval2015 [13].

3.2 Experiment 2: Relative Similarity Comparison

This experiment uses the relative similarity among two pairs of texts to validate the model. We present the dataset construction, the method and then the results of experiment.

Table 2. Comparison to top 5 models from Textual similarity task in [13].

rank	Models	images	headlines	students	forums	belief	weighted mean
1	Arora *et al*. [5]	.8434	.8417	.7879	.7390	.7717	.8071
2	Vu *et al*. [10]	.8640	.8250	.7730	.7390	.7490	.8015
3	Sultan *et al*. [4]	.8644	.8250	.7725	.7390	.7491	.8015
4	Hanig *et al*. [7]	.8527	.8245	.7784	.6946	.7482	.7943
5	Han *et al*. [6]	.8701	.8342	.7827	.6589	.7029	.7920
-	Our model	.8590	.8109	.7789	.7187	.7458	.7953

3.2.1 Construction of sample set

We construct a sample set based on the text-similarity-dataset of Dzikovska *et al*. [13] as being used in Experiment 1. Based on this dataset, we construct our sample sets of 1500 samples for experiment. In which, each sample includes (Table 3):

- Two pairs of sentences from the dataset of [13]. These two pairs need to satisfy the condition: Each pair has the confidence score greater than 50%.
- The score of the first pair and the second pair.
- The *value* of the sample. The value of attribute is assigned as follows: If the score of the first pair is greater than the second pair, the *value* is assigned to be 1. If the score of the first pair is smaller than the second pair, the *value* is assigned to be 2.
- The *result* of the sample. The value of this attribute is assigned once the sample is tested.

Table 3. An example of a sample.

Attribute	**Content**	
pair 1	Drug lord captured by marines in Mexico	Suspected drug lord known as El Taliban held in Mexico
pair 2	Explosion hits oil pipeline in Syria's Homs	Explosion hits pipeline as Assad attacks cities
score 1	1.98	
score 2	4.23	
value	2	
result	to be determined!	

3.2.2 Method

We choose two models proposed by Buscaldi *et al*. [1] and Lee *et al*. [8] to compare since they are closed to our model regarding main technical features (summarised in Table 4). The experiment is performed as follows for each considered model:

- Our model makes use of the ontology in OntNotes ([14]), which has more than 300 thousands concepts and their semantic relationships. Meanwhile, the model given by [8] uses WordNet ([15]) to match the words.

Table 4. Technical features of three models.

Model	Model of [1]	Model of [8]	Our model
Split text	n-grams (1-gram)	structure of sentence	structure of sentence
Matching method	lexicon-based	semantic-based (WordNet)	semantic-based (ontology)
Scope	word, sentence, text	word, sentence	word, sentence, text

- For each sample, the model proposed in this paper is used to estimate the similarity of two sentences of the first pair. This returns the similarity of the first pair.
- The model is also used to estimate the similarity between two sentences of the second pair. This returns the similarity of the second pair.
- If the similarity of the first pair is greater than one of the second pair, then the *result* of this sample is 1. Contrarily, if the similarity of the first pair is smaller than one of the second pair, then the *result* of this sample is 2. If the similarity of the first pair is equal to that value of the second pair, then the *result* of this sample is 0.
- The value of attribute *result* is then compared to the attribute *value* of the same sample. If they are identical, we increase the variable *number of correct sample* by 1.
- Repeat these steps for 1500 samples in the set.

At the output, the *accuracy* [16] of the model over the given sample set is calculated as follows:

$$accuracy = \frac{\text{number of correct sample}}{\text{number of sample}} * 100\%. \tag{22}$$

3.2.3 Results

The results are presented in the Table 5. The model given by Buscaldi *et al*. [1] has the lowest *accuracy* value (65.53%). This is reasonable since this model is lexicon-based and statistic-based. Due to basing on lexicon to match the words, it could not match the semantic similarity of words which have a semantic relation in the sample set.

Table 5. *Accuracy* (%) in comparing with other models.

Model	Accuracy (%)
Buscaldi *et al*. [1]	65.53
Lee *et al*. [8]	79.33
Our model	83.00

With the *accuracy* value of 79.33%, the model given by Lee *et al*. [8] is much better than the model given by Buscaldi *et al*. [1], and a bit worse than our model. This could be explained by two aspects. First, although both models are based on semantic matching, the semantic matching in our model is based on ontology, otherwise the model by Lee *et al*. is based on WordNet. In the scope of a domain, an ontology is richer than WordNet because it deeply represents a hierarchy of many related concepts in a specific domain. Meanwhile WordNet represents only a set of words which have the same meaning with a given term. Consequently, with a given concept, many concepts represented in an ontology are related to it, but less terms represented in WordNet are related to the given concept. Therefore, in the scope of an ontology, using ontology could classify better the similarity among concepts than using WordNet.

Second, WordNet is larger than an ontology in scope, so in the case that the sentences in a sample are outside of the used ontology, the model of Lee *et al*. [8] could distinguish the concepts in different domains better than ours.

Furthermore, two models are technically based on structure of sentences. However, the model given by Lee *et al*. uses the sentence structure for detecting the corpus of each word in the sentence only. The word-in-sentence matching is taken without consider the order and the position of words in the sentence as in our model. For instance, in the model given by Lee *et al*., two sentences *A bird eats a fish* and *A fish eats a bird* are considered as complete matching. Since they match the word *bird* in the subject of the first sentence to the word *bird* in the object of the second sentence, and the word *fish* in the object of the first sentence to the same word in the subject of the second sentence, the word *eat* in the verb of two sentences are identical. Consequently, two sentences are considered as the same. Meanwhile, in our model, thanks to sentence structure-based matching, we match separately each component of the sentence: the two subjects are different, the two objects are also different, only two verbs are identical. So two sentences are not the same.

4. CONCLUSIONS

This paper presented a model to estimate the semantic similarity among texts. The similarity among texts is based on the semantic similarity among sentences of the texts. In turn, the semantic similarity between two sentences is based on the similarity between their structure, including the semantic similarity between their subjects, verbs, and objects. These semantic measures are based on the semantic similarity between words. Our experiment results indicate that the proposed model is better than some statistic-based models, or models based on WordNet-based semantic matching in the case of relative similarity comparison. In the case of direct similarity comparison, our proposed model results are also very competitive regarding the best models in the Textual similarity task in Semeval2015. However, the used dataset contains the texts having only single sentence. So the comparison among texts which have several sentences is not still validated. This is considered as one of our perspectives in the near future.

REFERENCES

1. D. Buscaldi, P. Rosso, J. M. Gomez-Soriano, and E. Sanchis, “Answering questions with an n-gram based passage retrieval engine,” *Journal of Intelligent Information Systems*, Vol. 34, 2010, pp. 113-134.
2. M. C. Lintean and V. Rus, “Measuring semantic similarity in short texts through greedy pairing and word semantics,” in *Proceedings of the 25th AAAI International Florida Artificial Intelligence Research Society Conference*, 2012, pp. 244-249.

3. T. Proisl, S. Evert, P. Greiner, and B. Kabashi, “Robust semantic similarity at multiple levels using maximum weight matching,” in *Proceedings of the 8th International Workshop on Semantic Evaluation*, 2014, pp. 532-540.
4. M. A. Sultan, S. Bethard, and T. Sumner, “DLS@CU: Sentence similarity from word alignment and semantic vector composition,” in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 148-153.
5. P. Arora, C. Hokamp, J. Foster, and G. Jones, “DCU: Using distributional semantics and domain adaptation for the semantic textual similarity semeval-2015 task 2,” in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 143-147.
6. L. Han, J. Martineau, D. Cheng, and C. Thomas, “Samsung: Align-and-differentiate approach to semantic textual similarity,” in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 172-177.
7. C. Hänig, R. Remus, and X. de la Puente, “Exb themis: Extensive feature extraction from word alignments for semantic textual similarity,” in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 264-268.
8. M. C. Lee, J. W. Chang, and T. C. Hsieh, “A grammar-based semantic similarity algorithm for natural language sentences,” *The Scientific World Journal*, Vol. 2014, 2014, Article No. 437162.
9. M. H. Nguyen and D. Q. Tran, “A semantic similarity measure between sentences,” *South-East Asian Journal of Sciences*, Vol. 3, 2014, pp. 63-75.
10. T. T. Vu, Q. H. Tran, and S. B. Pham, “TATO: Leveraging on multiple strategies for semantic textual similarity,” in *Proceedings of the 9th International Workshop on Semantic Evaluation*, 2015, pp. 190-195.
11. D. Q. Tran and M. H. Nguyen, “A mathematical model for semantic similarity measures,” *South-East Asian Journal of Sciences*, Vol. 1, 2012, pp. 32-45.
12. S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning, and C. Potts, “A fast unified model for parsing and sentence understanding,” in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 1, 2016, pp. 1466-1477.
13. M. Dzikovska, D. Bental, J. Moore, N. Steinhauser, G. Campbell, E. Farrow, and C. Callaway, “Intelligent tutoring with natural language support in the beetle ii system,” in M. Wolpers, P. Kirschner, M. Scheffel, S. Lindstaedt, and V. Dimitrova, (eds.), *Sustaining TEL: From Innovation to Learning and Practice*, Springer Berlin Heidelberg, 2010, LNCS Vol. 6383, pp. 620-625.
14. E. Hovy, M. Marcus, M. Palmer, L. Ramshaw, and R. Weischedel, “Ontonotes: The 90% solution,” in *Proceedings of the Human Language Technology Conference of the NAACL*, 2006, pp. 57-60.
15. G. A. Miller, “Wordnet: A lexical database for english,” *Communications of the ACM*, Vol. 38, 1995, pp. 39-41.
16. G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, Inc., NY, 1986.

Manh Hung Nguyen (Born in 1982) received his Bachelor degree of Computer Science at The Posts and Telecommunication Institute of Technology (PTIT) in 2004, his M.Sc. in Computer Science at the L'Institut de la Francophonie pour l'Informatique (old IFI) in 2007, and his Ph.D. in Computer Science at the University of Toulouse III Paul Sabatier, France, in 2010. He is currently a Lecturer at the Faculty of Information Technology, Posts and Telecommunications Institute of Technologies (PTIT), Hanoi, Vietnam. His research interests include artificial intelligence, multi-agent system, machine learning, distributed intelligent computing.

Dinh Que Tran (Born in 1956) received the M.Sc. in Mathematics (1981) from Hanoi University of Education, Vietnam; M.Sc. in Computer Science (1998) from Melbourne University, Australia and Ph.D in Computer Science (2000) from The National Institute of Research in Computer Science, Vietnam. Currently, he is an Associate Professor at the Faculty of Information Technology, Posts and Telecommunications Institute of Technologies (PTIT), Hanoi, Vietnam. His research interests include artificial intelligence, multi-agent systems, social computing