MEASURING THE SIMILARITY OF SHORT TEXTS BY WORD SIMILARITY AND TREE KERNELS

Yun Tian, Haisheng Li, Qiang Cai, Shouxiang Zhao
College of Computer and Information Engineering
Beijing Technology and Business University
Beijing, P.R.China, 100048
Corresponding author: lihsh@th.btbu.edu.cn

ABSTRACT

A novel modeling method is presented in this paper to measure the similarity between short texts. We thought that the complete expression of a sentence or a short text, not only depends on the words, but also relies on the syntactic structure, thus the method takes word similarity feature and syntactic feature into account. The proposed method can be used in a variety of applications involving automatic document summarization, text knowledge representation and discovery. Experiment on two different data sets shows that the proposed method performs better than the measure proposed by Li et al.

Index Terms—Sentence similarity; word similarity; tree kernel; semantic similarity;

1. INTRODUCTION

With the information overload problem, presenting user an efficient information retrieval system becomes more and more important. In web page retrieval, sentence similarity has proven to be one of the best techniques for improving retrieval effectiveness. In the area of text mining, text similarity is used to discover knowledge from text clusters. These applications show that computing texts similarity has become a hot topic in information retrieval community.

Most existing text similarity methods are only suitable for long texts because these methods often focused on analyzing shared words. Similar long texts usually have a degree of co-occurring words, but in short texts, there are few or none. In this paper, we directly focus on the similarity of short texts. We thought that the complete expression of short texts, not only depends on the words, but also relies on the structure. Firstly, the proposed method uses WordNet-based similarity measure to calculate the semantic similarity of short texts. Then, through analyzing syntactic structure of the texts by semantic tree kernel, we get the syntactic similarity. After that, we give the two texts similarity results different weight to calculate the overall texts similarity. The experiment shows that our technique is effective in the short texts similarity methods comparing the mainly used measure proposed by Li et al. [1].

The remainder of this paper is organized as follows: Section 2 reviews some related work briefly. Section 3 presents the proposed method in detail. We combine word semantic similarity and syntactic similarity together to calculate the text similarity. The experimental results are given in section 4; two different datasets are used to verify the proposed method. Conclusions are given at the final section.

2. RELATED WORKS

Previous research of text similarity mainly focused on long text. Many effective techniques for long documents can also be used for composing short texts similarity method. They can be classified into four major categories: word co-occurrence-based method, vector-based method, corpus based method and hybrid method [2].

The word co-occurrence method has been improved in variety ways to match the method of calculating short texts. Hatzivassiloglou et al. proposed a method of combining primitive features and composite features [3]. This technique relies on the assumption that more similar texts have more words in common. But it is not always the case that texts with similar meaning necessarily share many words.

The vector-based method is commonly used in information retrieval (IR) systems. Chiang and Yu applied pattern-matching methods, which are widely used in QA and other text mining, to measure short text similarity [4]. While, the sentence representation is not very efficient due to the vector dimension is very large.

The corpus-based method depends on large corpus. Once the method was proposed for an application domain, it can hardly be used in another domain. The Latent Semantic Analysis (LSA) [5] and the Hyperspace Analogues to Language (HAL) model [6] are two well-known methods in corpus-based similarity.

Hybrid methods use both corpus-based measures and knowledge-based measures of word semantic similarity to determine the text similarity. Li et al. proposed a sentence similarity measurement based on lexical database and word ordering [1]. Using word ordering is not a new idea and experiments have shown that it sometimes decreases the

978-1-4244-8886-5/10/$26.00 ©2010 IEEE

accuracy of text-related IR techniques, such as document clustering and question answering.

Existing text similarity methods usually work well for long texts because long texts have adequate information to be expressed by several keywords. For short texts there is little or none, so we must pay more attention to syntactic structure. The proposed method uses a new semantic similarity measure based on WordNet to calculate the similarity between words, which improves the drawbacks of the methods based on corpus. Through calculating the syntactic similarity, we consider that the proposed model will perform more efficient than existing ones.

3. THE PROPOSED METHOD OF TEXT SIMILARITY

In this section, we propose a model which takes syntactic feature and semantic information of words into account to calculate the text similarity. Our approach consists of two steps. First semantic information is obtained from WordNet, and then syntactic feature are given through analyzing the structure of sentences. Figure 1 shows the procedure of calculating similarity between texts in details.

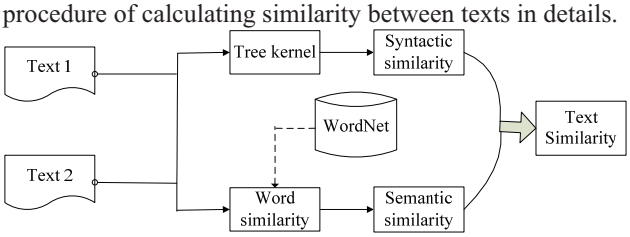

Figure 1. Text similarity model

3.1 Semantic similarity between words

Currently, there are mainly two methods of calculating semantic similarity between words, one is based on information content (IC), and the other is based on path-finding. Traditional methods based on IC usually rely on large corpus. We propose a new method which calculates the IC value only based on the lexical database WordNet [7]. The formula is as follows:

$$IC(c) = 1 - \frac{\log(hypo(c)+1)}{\log(\max_{wn})}$$

where hypo(c) is the synsets(a set of one or more synonyms) of concept c in WordNet, $\max_{wn}$ is a constant that is set to the number of concepts in WordNet.

Based on this IC value formula, we propose a new similarity measure. We deem that this measure is consequently easier to calculate. The formula is as follows:

$$sim(A,B) = \begin{cases} \dfrac{IC(lcs(A,B))}{\alpha * IC(A) + (1-\alpha) IC(B)}, A \neq B \\ 1, A = B \end{cases}$$

$$\alpha = \begin{cases} \dfrac{depth(A)}{depth(A)+depth(B)}, depth(A) \leq depth(B) \\ 1 - \dfrac{depth(A)}{depth(A)+depth(B)}, depth(A) > depth(B) \end{cases}$$

where LCS (least common subsume) refers to the most specific subsume of the two synsets, α is a function to define the relative importance of the non-common characteristics.

A text document is represented by the frequencies of the words it contains, ignoring the order of the words and any punctuation. Most researches are adapted to derive an efficient semantic vector for a short text.

Given two texts T_1 and T_2, a joint unit set is formed:

$$T = T_1 \cup T_2 = \{w_1, q_1 ... w_n\}$$

We use the vector measure of Li et al. [5] to construct the semantic vector. The vector derived from the joint word set is called the lexical semantic vector, denoted by d. Each entry of the semantic vector corresponds to a word in the joint word set, so the dimension equals the number of words in the joint word set. The value of an entry of the lexical semantic vector, d_i(i=1,2…..m), is determined by the semantic similarity of the corresponding word to a word in the text.

- If the word in joint unit appears in texts 1or texts 2, then, d is set to 1.
- If the word does not appear in the text, then we use the word similarity method to calculate the similarity between the word and all of the word in the joint unit to find the most similar ones, then, d is set to the similarity value.

Now, we get the semantic vector d_1 and d_2, then we use the cosine-vector based method to calculate the similarity, the formula is as follows:

$$sim_{sen}(T_1, T_2) = \frac{\sum_{k=1}^{m} (w_{k,d_1} * w_{k,d_2})}{\sqrt{\sum_{k=1}^{m} w_{k,d_1}^{\ 2}} * \sqrt{\sum_{k=1}^{m} w_{k,d_2}^{\ 2}}}$$

where w_{k,d_1} is the weight of W_k in d_1, w_{k,d_2} is the weight of W_k in d_2.

3.2 Syntactic similarity between texts

Tree kernels have been widely used in many applications such as Natural Language Processing (NLP) problems, Support Vector Machines or Principal Component Analysis [8]. In this paper, tree kernels will be used in syntactic structure similarity method of short texts. The most direct form of a sentence is tree structure. Through analyzing the structure of sentences, we find that tree kernel can accurately match the syntax of sentence.

Tree kernels can be used to form representations which are sensitive to large sub-structures of trees or state sequences. It caught as much information as possible from the structure of tree to calculate the syntactic similarity by

matching the same sub trees. Kernels match the syntax tree in a hierarchical way. It means that, from the root of the tree to the bottom, the node must be on the same floor and the path to its root node must be the same.

For example, calculating the similarity of two noun phrases: “the dog” and “the cat”, the two phrases can be described as follows:

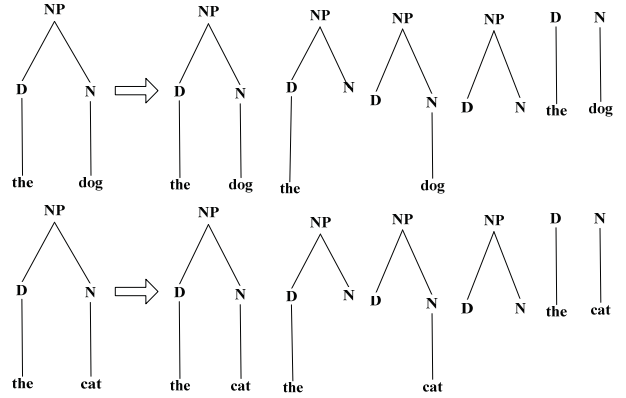

Figure 2. Subtrees of syntactic parse tree(NP =noun phrase; D = definite article; N = noun)

As is shown in Figure 2, in the total five graphs, three of them are the same, so the structure similarity is 3.

In order to use the tree structure to improve our short text similarity, we analyze the syntactic of sentence, and make them all can be transferred into tree structure. For example: the sentence “I got the ball”, through analyzing by tree kernel, can be described as bellows.

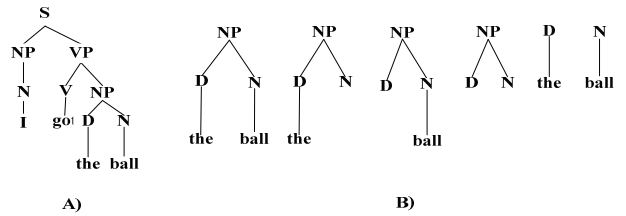

Figure 3. A)An example tree; B) The sub-trees of the NP covering *the ball*.(N = noun, V = verb, VP = verb phrase; D = definite article)

In this section, we use the method proposed by Collins to calculate the syntactic similarity [8]. Conceptually we begin by enumerating all tree fragments that occur in the training data 1, … , n. Note that this is done only implicitly, each tree is expressed by an n dimensional vector where the i^{th} component counts the number of occurrences of the i^{th} tree fragment. We define the function hi(T) to be the number of occurrences of the i^{th} tree fragment in tree T, so that T is now represented as $h(T)=(h_1(T), h_2(T),\ldots, h_n(T))$. Then we can get the method of syntactic similarity, the formula is as follow:

$$sim_{syn}(T_1,T_2) = h(T_1)*h(T_2) = \sum_i h_i(T_1)*h_i(T_2)$$
$$= \sum_{n_1\in N_1}\sum_{n_2\in N_2}\sum_i I_i(n_1)I_i(n_2) = \sum_{n_1\in N_1}\sum_{n_2\in N_2} c(n_1,n_2) \quad ,$$

where n_1 and n_2 are the node set number of T_1, T_2. We define $c(n_1,n_2) = \sum_i I_i(n_1)I_i(n_2)$ and $I_i(n)$ to be 1 if sub-tree I is at node n and 0 otherwise. Next, we note that $c(n_1,n_2)$ can be calculated in polynomial time, due to the following recursive definition:

- If the productions at n_1 and n_2 are different. $c(n_1,n_2)=0$.
- If the productions at n_1 and n_2 are the same, and n_1 and n_2 are pre-terminals, then $c(n_1,n_2)=1$.
- Else if the productions at n_1 and n_2 are the same, and n_1 and n_2 are not pre-terminals,

$$c(n_1,n_2) = \prod_{j=1}^{nc_{n_1}}(1+c(ch(n_1,j),ch(n_2,j))),$$

where $nc(n_1)$ is the number of children of n_1 in the tree, because the productions at n_1/ n_2 are the same, we have $nc(n_1) = nc(n_2)$.

3.3 The overall text similarity

Semantic similarity shows the semantic information between texts, while syntactic similarity conveys the structure information of texts. Both of these two features play an important part in expressing the meaning of texts. Thus, we define the overall similarity method as follow:

$$s(T_1,T_2) = \lambda sim_{sem} + (1-\lambda)sim_{syn}$$

where $0<\lambda<1$ decides the relative contribution of semantic similarity and syntactic similarity to the overall similarity method.

4. EXPERIMENT

In order to evaluate our texts similarity method, we compared the proposed word similarity method with other semantic similarity measure firstly. Then, we compared the proposed texts similarity method with the methods proposed by Li et al.

4.1 Experiment of the word similarity measure

As there is not a standard for evaluation of word similarity, results are mostly judged by human common sense. In 1965 Rubenstein and Goodenough (R&G) organized two pairs of students and 51 experts to estimate the synsets similarity of 63 pairs of concepts. The values range from 0 to 4. Miller and Chades(MC) extracted 30 pairs of nouns from RG dataset, repeated their experiment with 38 subjects[9].

In this paper, we use the word pair set of MC to evaluate our similarity measure. We compare our word similarity measure with the multiple information sources-based semantic similarity measure proposed by Li et al. and Resnik [10].

The correlation coefficient values between the similarity measures (or human ratings) and the replication of Miller and Charles are reported in Table I. The experimental results show that our proposed similarity measure outperforms Li’s measure.

TABLE I. EXPERIMENT DATA

Similarity measure	Correlation to MC data
Resnik	0.9583
Li's measure	0.8271
Our measure	0.8729

4.2 Experiment of the texts similarity measure

The CMU newspaper dataset [11] is used in this section to simulate some short text clustering scenarios. This is a well-known collection of messages from 20 newsgroups, with 1000 messages selected from each newsgroup to give a total of 20,000 documents. Most of the messages are relatively short, and consist of just a few sentences.

We choose 6000 sentences from the CMU dataset as our raw dataset. 5000 sentences are un-related; the other 1000 sentences are selected to make up the standard dataset. In the standard dataset, we divided the sentences into 300 groups by their similarity. In each group, three or four sentences are un-related, which means one or two sentences are considered to be similar.

In the experiment, we define λ=0.5, which means that semantic similarity and syntactic similarity are equally important to text similarity.

For the 300 groups standard dataset, we choose one according to order, then we calculate the similarity between this sentence and the candidate sentences in test dataset. Then we pick up the sentence which has the max similarity value .If the sentence belongs to the 1000 standard dataset, we consider this sentence similarity is successful.

We experiment on the datasets with the method proposed by Li et al. and our own measure, and do some research on the results. Analysis on results can be computed as follow:

$$P = \frac{T}{C} * 100\%,$$

where P is defined as the accuracy of results, T is the correct number of sentences, C is the total number of test sentences.

TABLE II. PRECISION OF TWO METHODS

Similarity method	C	T	P
Li's measure	300	268	89.33%
Our measure	300	283	94.33%

From table II, we can see that, comparing with Li's measure, the proposed method got a more accuracy result. The result is in our expectation, because the word semantic similarity method is improved, in addition, the syntactic features are taken into consideration to calculate the text similarity.

5. CONCLUSIONS

In this paper, we propose a novel modeling method which combines semantic similarity obtained from WordNet and syntactic similarity through analyzing the structure by tree kernel.

Several word similarity experiments show that the proposed word similarity measure is more consistent with human judgment than other measures. We also do some experiments on the proposed text similarity. The results show that, comparing Li's similarity method, our method improves the accuracy of the texts similarity.

6. ACKNOWLEGEMENT

This research is partially supported by Science and Technology Development Program of Beijing Municipal Education Commission KM200910011007 and Funding Project for Academic Human Resources Development in Institutions of Higher Learning under the Jurisdiction of Beijing Municipality PHR20110875.

7. REFERENCES

[1] Y.H. Li, D. McLean, Z.A. Bandar, J.D. O' Shea, K. Crockett, *Sentence similarity based on semantic nets and corpus statistics*, IEEE Transactions on Knowledge and Data Engineering , 18 .1138–1150, 2006.

[2] Islam, A., & Inkpen, D, *Semantic text similarity using corpus-based word similarity and string similarity*, ACM Transactions on Knowledge Discovery from Data, 2(2) 1–25, 2008.

[3] V. Hatzivassiloglou, J. Klavans, and E. Eskin, *Detecting text similarity over short passages: exploring linguistic feature combinations via machine learning*, Proceedings of the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora, 1999.

[4] J.H. Chiang, H.C. Yu, *Literature extraction of protein function using sentence pattern mining*, IEEE Transactions on Knowledge and Data Engineering, 17.1088–1098, 2008.

[5] P.W. Foltz, W. Kintsch, and T.K. Landauer, "The Measurement of Textual Coherence with Latent Semantic Analysis", *Discourse Processes,* vol. 25, nos. 2-3, pp. 285-307, 1998.

[6] C. Burgess, K. Livesay, and K. Lund, "Explorations in Context Space: Words, Sentences, Discourse," *Discourse Processes*, vol. 25, nos. 2-3, pp. 211-257, 1998.

[7] Giuseppe Pirró et al, *A semantic similarity metric combining features and intrinsic information content*, Data & Knowledge Engineering 68 1289–1308, 2009.

[8] Collins M, Duffy N, *Convolution Kernels for Natural Language*, Proceedings of the 42nd Annual Meeting on Association for ComPutational Linguisties Table of Contents, Spain, 2004:119-126.

[9] G. A. Miller and W. G. Charles, "Contextual correlates of semantic similarity", *Language and Cognitive Processes*, Vol. 6' No.1. pp. I-28, 1991.

[10] Resnik P, *Using information content to evaluate semantic similarity*, In Proceedings of the 14th International Joint Conference on Artificial Intelligence, Pages 8-453. Montreal, 1995

[11] http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/news20.html/.