

Tiantianzhu7: System Description of Semantic Textual Similarity (STS) in the SemEval-2012 (Task 6)

Tiantian Zhu

Department of Computer Science and
Technology

East China Normal University

51111201046@student.ecnu.edu.cn

Man Lan

Department of Computer Science and
Technology

East China Normal University

mlan@cs.ecnu.edu.cn

Abstract

This paper briefly reports our submissions to the Semantic Textual Similarity (STS) task in the SemEval 2012 (Task 6). We first use knowledge-based methods to compute word semantic similarity as well as Word Sense Disambiguation (WSD). We also consider word order similarity from the structure of the sentence. Finally we sum up several aspects of similarity with different coefficients and get the sentence similarity score.

1 Introduction

The task of semantic textual similarity (STS) is to measure the degree of semantic equivalence between two sentences. It plays an increasingly important role in several text-related research and applications, such as text mining, Web page retrieval, automatic question-answering, text summarization, and machine translation. The goal of the Semeval 2012 STS task (task 6) is to build a unified framework for the evaluation of semantic textual similarity modules for different systems and to characterize their impact on NLP applications.

Generally, there are two ways to measure similarity of two sentences, i.e., corpus-based methods and knowledge-based methods. The corpus-based method typically computes sentence similarity based on the frequency of word occurrence or the co-occurrence between collocated words. For example, in (Islam and Inkpen, 2008) they proposed a corpus-based sentence similarity measure as a function of string similarity, word similarity and common word order similarity (CWO). The knowledge-

based method computes sentence similarity based on the semantic information collected from knowledge bases. With the aid of a number of successful computational linguistic projects, many semantic knowledge bases are readily available, for example, WordNet, Spatial Date Transfer Standard, Gene Ontology, etc. Among them, the most widely used one is WordNet, which is organized by meanings and developed at Princeton University. Several methods computed word similarity by using WordNet, such as the Lesk method in (Banerjee and Pedersen, 2003), the lch method in (Leacock and Chodorow, 1998) and the wup method in (Wu and Palmer, 1994). Generally, although the knowledge-based methods heavily depend on the knowledge bases, they performed much better than the corpus-based methods in most cases. Therefore, in our STS system, we use a knowledge-based method to compute word similarity.

The rest of this paper is organized as follows. Section 2 describes our system. Section 3 presents the results of our system.

2 System Description

Usually, a sentence is composed of some nouns, verbs, adjectives, adverbs and/or some stop words. We found that these words carry a lot of information, especially the nouns and verbs. Although the adjectives and adverbs also make contribution to the semantic meaning of the sentence, they are much weaker than the nouns and verbs. So we consider to measure the sentence semantic similarities from three aspects. We define the following three types of similarity from two compared sentences to measure

the semantic similarity: (1) Noun Similarity to measure the similarity between the nouns from the two compared sentences, (2) Verb Similarity to measure the similarity between Verbs, (3) ADJ-ADV Similarity to measure the similarity between the adjectives and adverbs from each sentence. Besides the semantic information similarity, we also found that the structure of the sentences carry some information which cannot be ignored. Therefore, we define the last aspect of the sentence similarity as Word Order Similarity. In the following we will introduce the different components of our system.

2.1 POS

As a basic natural language processing technique, part of speech tagging is to identify the part of speech of individual words in the sentence. In order to compute the three above semantic similarities, we first identify the nouns, verbs, adjectives, and adverbs in the sentence. Then we can calculate the Noun Similarity, Verb Similarity and ADJ-ADV Similarity from two sentences.

2.2 Semantic similarity between words

The word similarity measurement have important impact on the performance of sentence similarity. Currently, many lexical resources based approaches perform comparatively well to compute semantic word similarities. However, the exact resources they are based are quite different. For example, some are based on dictionary and/or thesaurus, and others are based on WordNet.

WordNet is a machine-readable lexical database. The words in Wordnet are classified into four categories, i.e., nouns, verbs, adjectives and adverbs. WordNet groups these words into sets of synonyms called *synsets*, provides short definitions, and records the various semantic relations between these *synsets*. The *synsets* are interlinked by means of conceptual-semantic and lexical relations. WordNet also provides the most common relationships include *Hyponym/Hypernym* (i.e., is-a relationships) and *Meronym/Holonym* (i.e., part-of relationships). Nouns and verbs are organized into hierarchies based on the hyponymy/hypernym relation between *synsets* while adjectives and adverbs are not.

In this paper, we adopt the wup method in (Wu and Palmer, 1994) to estimate the semantic similar-

ity between two words, which estimates the semantic similarity between two words based on the depth of the two words in WordNet and the depth of their least common subsumer (LCS), where LCS is defined as the common ancestor deepest in the taxonomy.

For example, given two words, w_1 and w_2 , the semantic similarity $s(w_1, w_2)$ is the function of their depth in the taxonomy and the depth of their least common subsumer. If d_1 and d_2 are the depth of w_1 and w_2 in WordNet, and h is the depth of their least common subsumer in WordNet, the semantic similarity can be written as:

$$s(w_1, w_2) = \frac{2.0 * h}{d_1 + d_2} \quad (1)$$

2.3 Word Sense Disambiguation

Word Sense Disambiguation (WSD) is to identify the actual meaning of a word according to the context. In our word similarity method, we take the nearest meaning of two words into consideration rather than their actual meaning. More importantly, the nearest meaning does not always represent the actual meaning. In our system, we used a WSD algorithm proposed by (Ted Pedersen et al., 2005), which computes semantic relatedness of word senses using extended gloss overlaps of their dictionary definitions. We utilize this WSD algorithm for each sentence to get the actual meaning of each word before computing the word semantic similarity.

2.4 Semantic Similarity

We adopt a similar way to compute the three types of semantic similarities. Here we take Noun Similarity as an example.

Suppose sentence s_1 and s_2 are the two sentences to be compared, s_1 has a nouns while s_2 has b nouns. Then we get $a * b$ noun pairs and use the word similarity method mentioned in section 2.2 to compute the Noun Similarity of each noun pair. After that, for each noun, we choose its highest score in noun pairs as its similarity score. Then we use the formula below to compute the Noun Similarity.

$$Sim_{Noun} = \frac{(\sum_{i=1}^c n_i) * (a + b)}{2ab} \quad (2)$$

where c represents the number of noun words in sequence a and sequence b , $c = \min(a, b)$; n_i represents the highest matching similarity score of i -th word in the shorter sequence with respect to one of the words in the longer sequence; and $\sum_{i=1}^c n_i$ represents the sum of the highest matching similarity score between the words in sequence a and sequence b . Similarly, we can get Sim_{Verb} . Since there is no *Hyponym/Hypernym* relation for adjectives and adverbs in WordNet, we just compute ADJ-ADV Similarity based on the frequency of overlap of simple words.

2.5 Word Order Similarity

We believe that word order information also make contributions to sentence similarity. In most cases, the longer common sequence (LCS) the two sentences have, the higher similarity score the sentences get. For example the pair of sentences s_1 and s_2 , we remove all the punctuation from the sentences:

- s_1 : But other sources close to the sale said Vivendi was keeping the door open to further bids and hoped to see bidders interested in individual assets team up
- s_2 : But other sources close to the sale said Vivendi was keeping the door open for further bids in the next day or two

Since the length of the longest common sequence is 14, we use the following formula to compute the word order similarity.

$$Sim_{WordOrder} = \frac{lengthofLCS}{shorterlength} \quad (3)$$

where the *shorter length* means the length of the shorter sentence.

2.6 Overall Similarity

After we have the Noun Similarity, Verb Similarity, ADJ-ADV Similarity and Word Order Similarity, we calculate the Overall Similarity of two compared sentences based on these four scores of similarity. We combine them in the following way:

$$Sim_{sent} = aSim_{Noun} + bSim_{Verb} + cSim_{ADJ-ADV} + dSim_{WordOrder} \quad (4)$$

Where a , b , c and d are the coefficients which denote the contribution of each aspect to the overall sentence similarity, For different data collections, we empirically set different coefficients, for example, for the MSR Paraphrase data, the four coefficients are set as 0.5, 0.3, 0.1, 0.1, because it is hard to get the highest score 5 even when the two sentences are almost the same meaning, We empirically set a threshold, if the score exceeds the threshold we set the score 5.

3 Experiment and Results on STS

Firstly, Stanford parser¹ is used to parse each sentence and to tag each word with a part of speech (POS). Secondly, WordNet SenseRelate AllWords², a WSD tool from CPAN is used to disambiguate and to assign a sense for each word based on the assigned POS.

We submitted three runs: run 1 with WSD, run 2 without WSD, run 3 removing stop words and without WSD. The stoplist is available online³. Table 1 lists the performance of these three systems as well as the baseline and the rank 1 results on STS task in SemEval 2012.

We can see that run1 gets the best result, which means WSD has improved the accuracy of sentence similarity. Run3 gets better result than run2, which proves that stop words do disturb the computation of sentence similarity, removing them is a better choice in our system.

4 Conclusion

In our work, we adopt a knowledge-based word similarity method with WSD to measure the semantic similarity between two sentences from four aspects: Noun Similarity, Verb Similarity, ADJ-ADV Similarity and Word Order Similarity. The results show that WSD improves the pearson coefficient at some degree. However, our system did not get a good rank. It indicates there still exists many problems such as wrong POS tag and wrong WSD which might lead to wrong meaning of one word in a sentence.

¹<http://nlp.stanford.edu/software/lex-parser.shtml>

²<http://search.cpan.org/Tedpederse/WordNet-SenseRelate-AllWords-0.19>

³<http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop>

Table 1: STS system configuration and results on STS task.

Run	ALL	ALLnrm	Mean	MSRpar	MSRvid	SMTeur	OnWN	SMTnews
rank 1	.7790	.8579	.6773	.6830	.8739	.5280	.6641	.4937
baseline	.3110	.6732	.4356	.4334	.2996	.4542	.5864	.3908
1	.4533	.7134	.4192	.4184	.5630	.2083	.4822	.2745
2	.4157	.7099	.3960	.4260	.5628	.1546	.4552	.1923
3	.4446	.7097	.3740	.3411	.5946	.1868	.4029	.1823

Acknowledgments

The authors would like to thank the organizers for their invaluable support making STS a first-rank and interesting international event.

References

- Chukfong Ho, Masrah Azrifah Azmi Murad, Rabiah Abdul Kadir, Shyamala C. Doraisamy. 2010. *Word Sense Disambiguation-based Sentence Similarity*. In Proc. COLING-ACL, Beijing.
- Jin Feng, Yiming Zhou, Trevor Martin. 2008. *Sentence Similarity based on Relevance*. Proceedings of IPMU'08, Torremolinos.
- Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett. 2009. *Sentence Similarity Based on Semantic Nets and Corpus Statistics*.
- LIN LI, XIA HU, BI-YUN HU, JUN WANG, YI-MING ZHOU. 2009. *MEASURING SENTENCE SIMILARITY FROM DIFFERENT ASPECTS*.
- Islam Aminul and Diana Inkpen. 2008. *Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity*. ACM Transactions on Knowledge Discovery from Data.
- Banerjee and Pedersen. 2003. *Extended gloss overlaps as a measure of semantic relatedness*. In Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), pages805C810, Acapulco, Mexico.
- Leacock and Chodorow. 1998. *Combining local context and WordNet similarity for word sense identification*. In Christiane Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press, Cambridge,MA.
- Z.Wu and M.Palmer. 1994. *Verbs semantics and lexical selection*. In Proceedings of the 32nd annual meeting on Association for Computational Linguistics, Morristown, NJ, USA.
- Ted Pedersen, Satanjeev Banerjee, Siddharth Patwardhan. 2005. *Maximizing Semantic Relatedness to Perform Word Sense Disambiguation*.