WSL: Sentence Similarity Using Semantic Distance Between Words

Naoko Miura   **Tomohiro Takagi**
Meiji University, Japan
1-1-1 Higashi-Mita, Tama-ku, Kawasaki-shi,Kanagawa 214–8571
E-mail:{n_miura,takagi}@cs.meiji.ac.jp

Abstract

A typical social networking service contains huge amounts of data, and analyzing this data at the level of the sentence is important. In this paper, we describe our system for a SemEval2015 semantic textual similarity task (task2). We present our approach, which uses edit distance to consider word order, and introduce word appearance in context. We report the results from SemEval2015.

1 Introduction

The Internet, particularly sites related to social networking services (SNS), contains a vast array of information used for a variety of purposes. The vector space model is conventionally used for natural language processing. This model creates vectors on the basis of frequency of word appearance and co-occurring words, without taking word order into account. When it comes to short texts, word co-occurrence is rare (or even non-existent), and the number of words is often less than in a typical newspaper article. Because the average SNS contains data consisting mostly of short sentences, the vector space model is not the best choice.

In this work, we describe a system we developed and submitted to SemEval2015. In the proposed system, we compute sentence similarity using edit distance to consider word order along with the semantic distance between words. We also introduce word appearance in context.

The rest of this paper is organized as follows. Section 2 reviews related work and in Section 3 we present the three systems we submitted for SemEval2015. In Section 4, we discuss the results of our evaluation at SemEval2015.We conclude in Section 5 with a brief summary.

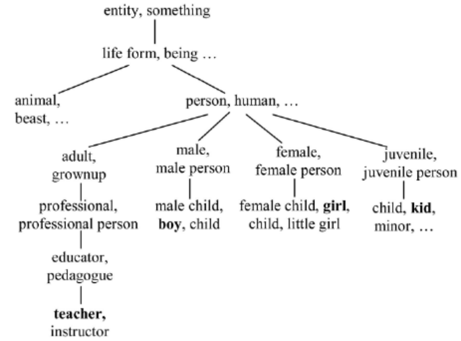

Fig. 1. Hierarchical semantic knowledge base (Li et al., 2006).

2 Related Work

Recent research has introduced the lexical database as a dictionary to analyze short texts(Aziz et al.,2010). Aziz uses a set of similar noun phrases and similar verb phrases and common words to compute sentence similarity. Li combines semantic similarity between words into a hierarchical semantic knowledge base and word order(Li et al.,2006). There are currently a few hierarchical semantic knowledge bases available, one of which is WordNet(Miller,1995). WordNet contains 155,287 words and 117,659 synsets that were stored in 2012 into the lexical categories of nouns, verbs, adjectives, and adverbs(WordNet Statistics, 2014). All synsets have semantic relation to other synsets. An example in the case of using nouns is shown in Fig.1. Li proposed a formula to measure the similarity *s(w₁,w₂)* between words *w₁* and *w₂* as

$$s(w_1, w_2) = e^{-\alpha l} \cdot \frac{e^{\beta h} - e^{-\beta h}}{e^{\beta h} + e^{-\beta h}} \quad , \qquad (1)$$

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 128–131,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

where l is the shortest path length between w_1 and w_2 and h is the depth of the subsumer of w_1 and w_2 in WordNet. For example, we describe the path between "boy" and "girl" in Fig. 1. The shortest path is *boy-male-person-female-girl*, which is 4, so l = 4. The subsumer of "boy" and "girl" is "person, human...", so the depth of this synset is h. In hierarchical semantic nets, words at the upper layers have a general meaning and less similarity than words at the lower layers. Li sets α = 0.2 and β = 0.45.

Not only the similarity between words but also word order is important. For example, the two sentences "a dog bites Mike" and "Mike bites a dog" consist of the same words, but the meanings are very different. In this case, we use vectors such that when each vector completely matches, the sentence similarity is high. Our approach is based on edit distance to take into account word order and combined semantic similarity between words.

3 System Details

The proposed system uses edit distance to take word order into account. It also uses the impact of word appearance in each context.

In this paper, we describe sentence S_1 as $S_1 = \{a_1, a_2, \cdots, a_n\}$ and sentence S_2 as $S_2 = \{b_1, b_2, \cdots, b_m\}$. S_1 consists of n words and S_2 consists of m words. a_i is the i th word of S_1 and b_j is the j th word of S_2. We describe the similarity $Sim(S_1, S_2)$ between S_1 and S_2 within the range of 0 (no relation) to 1 (semantic equivalence).

3.1 Edit Distance

Edit distance is a way of computing the dissimilarity between two strings. Conventionally, the distance is computed for a set of characters with three kinds of operations (substitution, insertion, deletion). However, our approaches are for word sets. Here, we describe the two kinds of edit distance extended in our system.

3.1.1 Levenshtein Distance

The Levenshtein distance between S_1 and S_2 ($|S_1|=n, |S_2|=m$) is $L(n,m)$, where

$0 \leqq i \leqq n,\ 0 \leqq j \leqq m$

$$L(i,j) = \max(i,j) \qquad \text{if } \min(i,j)=0,$$

$$L(i,j) = \min \begin{cases} L(i-1, j-1) + c_1(a_i, b_j) \\ L(i, j-1) + 1 \\ L(i-1, j) + 1 \end{cases} \quad \text{otherwise.} \qquad (2)$$

The indicator function $c_1(a_i, b_j)$ is defined as

$$c_1(a_i, b_j) = \begin{cases} 0 & (a_i = b_j) \\ 1 & (a_i \neq b_j) \end{cases} \qquad (3)$$

3.1.2 Jaro-Winkler Distance

The Jaro distance between S_1 and S_2 ($|S_1|=n, |S_2|=m$) is d_j:

$$d_j = \begin{cases} 0 & (q = 0) \\ \dfrac{1}{3}\left(\dfrac{q}{n} + \dfrac{q}{m} + \dfrac{q-t}{q}\right) & (q \neq 0) \end{cases}, \qquad (4)$$

where q is the number of matching words between S_1 and S_2. We consider two words as matching when they are the same and not father than

$$\left[\frac{\max(n,m)}{2}\right] - 1$$

t is half the number of transpositions.

The Jaro-Winkler distance is d_w:

$$d_w = \begin{cases} d_j & \text{if } \ d_j < 0.7 \\ d_j + (k * p * (1 - d_j)) & \text{otherwise.} \end{cases}, \qquad (5)$$

where k is the length of common words at the start of the sentence. p is constant and usually set to p = 0.1.

3.2 Semantic Distance

We borrow our approach to compute similarity between words from Li (Li et al.,2006)(Eq. (1)). It can be used for both nouns and verbs because both are organized into hierarchies. However, it is not available for adjectives and adverbs, which are not organized into hierarchies. Therefore, in addition to Eq. (1), when $w_1 \in synsetA$, $w_2 \in synsetB$, we define semantic similarity between words if they are adverbs and adjectives as

$$s(w_1, w_2) = \begin{cases} 1 & (synsetA = synsetB) \\ 0 & (synsetA \neq synsetB) \end{cases} \qquad (6)$$

$s(w_1, w_2)$ is 1 if w_1 and w_2 are in the same synset.

Conventionally, we calculated edit distance on the basis of match or mismatch between words and ignored how similar two words are. However, with this approach, if two words have the same meaning although they are different words (e.g., “fall” and “autumn”), edit distance defines them as a mismatch. We address this issue by introducing semantic similarity between words as distance.

(a) Levenshtein distance
We rewrite Eq. (3) as

$$c_1(a_i,b_j) = 1 - s(a_i,b_j) \tag{7}$$

We propose a measure for the sentence similarity of S_1 and S_2 $Sim(S_{1,},S_2)$ as

$$Sim(S_1,S_2) = 1.0 - \frac{L(n,m)}{\max(n,m)} \tag{8}$$

(b) Jaro-Winkler distance
We rewrite Jaro-distance d_j defined by Eq. (4) as

$$d_j = \begin{cases} 0 & (q'=0) \\ \frac{1}{3}\left(\frac{q'}{n} + \frac{q'}{m} + \frac{q'-t}{q'}\right) & (q' \neq 0) \end{cases} \tag{9}$$

We define q' in Eq. (10). q' indicates the sum of all semantic similarity between words in S_1 and S_2 $(1 \leqq i \leqq n, 1 \leqq j \leqq m$, $SUM(c_2(a_i,b_j))$). Further, originally, we calculated t only if two words are matching (a_i=b_j); however, in our proposed methods we change to s(a_i,b_j)>0.5 to take into account of the semantic similarity of words.

$1 \leqq i \leqq n,\ 1 \leqq j \leqq m$

$$q' = q + SUM(c_2(a_i,b_j)) \tag{10}$$

C_2 in Eq. (10) is defined by Eq. (11). It means the semantic similarity of words.

$$c_2(a_i,b_j) = \begin{cases} 0 & (a_i = b_j) \\ s(a_i,b_j) & (a_i \neq b_j) \end{cases} \tag{11}$$

We propose a measure for the sentence similarity of S_1 and S_2 $Sim(S_{1,}, S_2)$ as

$$Sim(S_1,S_2) = d_w \tag{12}$$

3.3 The Impact of Word Appearance in Context

There is one issue when we compute $Sim(S_{1,},S_2)$, as follows. Let us consider two sentences: “I ate an apple” and “I hate an apple”. These sentences indicate opposite meanings. However, except for “ate” and “hate”, both sentences consist of the same words and have the same word order. Therefore, the method we mentioned above (Eq. (8)) computes the $Sim(S_{1,},S_2)$ as high. However, we decide that the similarity between these sentences have opposite meanings because of “ate” and “hate”. For this reason, we introduce conditional probability to estimate word appearance for each context and extract the probabilities from a corpus as training data. Further, we give this word appearance for semantic similarity (Eq. (1)) as a weight.

Let us show an example. P(I | S_2), P(ate| S_2), P(an| S_2), and P(apple|S_2) are words of S_1 appearance in context S_2. We define S^* as the set of nouns, verbs, adjectives, and adverbs (e.g., when sentence S is “It is a dog”, S^* is {“is”, “dog”}).
We measure each word appearance *weight(w)* in context S as:

$$P(w \mid S) = \frac{doc_{w,S^*}}{doc_{S^*}} \tag{13}$$

$$weight(w) = \frac{1}{(1 + e^{-\gamma * P(w|S)})}, \tag{14}$$

where doc_{w,S^*} is the number of documents that contains both w and S^* and doc_{S^*} is the number of documents that contains S^*. We set $\gamma = 5.0$.

We take into account the impact of words in context and apply it to Levenshtein distance, rewriting Eq. (7) as

$$c_1(a_i,b_j) = (1 - s(a_i,b_j)) * weight(a_i)^{-1} * weight(b_j)^{-1} \tag{15}$$

When a word in one sentence co-occurs with words in the other sentence frequently, the impact is low, and when it co-occurs less frequently, the impact is high. We use Eq.(15) when a_i and b_j are nouns or verbs and $s(a_i,b_j) < 0.7$.

4 Results

STS systems at SemEval 2015 were evaluated on five data sets. Each data set contained a number of sentence pairs that have a gold-standard score in the range of 0–5 as correct answers. The STS systems were evaluated by Pearson correlation between the system output and the gold-standard score. We used the Reuters Corpus as training data.

4.1 Submissions

We submitted the outputs of three of our system runs. In the STS task, the similarity between the $score(S_1,S_2)$ of two sentences needed to be in the range of 0–5. Accordingly, we set $score(S_1,S_2)$ as $score(S_1,S_2) = 5 * Sim(S_1,S_2)$. For pre-processing, we use Stanford-NLP tools for tokenization and POS-tagging. We also remove punctuation marks.
And we use JWNL to measure the similarity between words. (Eq.(1))

-run1
Levenshtein distance approach (Eq. (8))

-run2
Jaro-Winkler distance approach (Eq. (12))

-run3
Using run1 (Eq. (8)) in conjunction with word appearance in context (Eq. (15))

4.2 Evaluation on STS 2015 Data

Table 1 shows the results (Pearson correlation) of each of our three runs evaluated on five data sets. Our best system was **run3**. It was ranked 64 out of 74 systems.

The weighted-mean scores of **run1** and **run2** were almost the same. When we compare the scores of **run1** and **run3**, **run3** performed better on four datasets (the exception was “answers-forums”). Overall, the best performance in terms of weighted-mean score was by **run3**.

Data Set	run1	run2	run3
answers-forums	0.3759	0.4287	0.3709
answers-students	0.5269	0.6028	0.5437
belief	0.6387	0.5231	0.6478
headlines	0.5462	0.6029	0.5752
images	0.5710	0.4879	0.6407
Weighted-Mean	0.5379	0.5424	0.5672

Table 1 . Results of evaluation on SemEval2015 STS task.

5 Conclusion

In this paper, we proposed methods for determining sentence similarity. We adopted the semantic distance of word on edit distance along with word appearance in context. Evaluation results suggest that using word appearance in context is an effective element for determining sentence similarity.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Iñigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, Janyce Wiebe. SemEval-2015 Task 2: Semantic Textual Similarity, English, Spanish and Pilot on Interpretability. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* June,2015, Denver,CO, Association for Computational Linguistics.

Yuhua Li, David McLean, Zuhair A. Bander, James D. O’Shea, and Keeley Crockett (2006). Sentence Similarity Based on Semantic Nets and Corpus Statistics. *IEEE Transactions on Knowledge and Data Engineering Vol. 18 No. 8 August 2006.*

Mehwish Aziz and Muhammad Rafi (2010). Sentence based semantic similarity measure for blog-posts. *Digital Content, Multimedia Technology and its Applications (IDC). 2010 6th International Conference.*

G.A. Miller, “WordNet: A Lexical Database for English”. (1995) *Communications of the ACM, Vol. 38, Issue 11 Nov. 1995.*

WordNet Statistics WordNet.princeton.edu. Retrieved2014-03-11
http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html

Reuters Corpus English Language, 1996/08/20-1997/08/19.
http://about.reuters.com/researchandstandards/corpus/

JWNL (Java WordNet Library)
http://sourceforge.net/projects/jwordnet/