Textual Similarity for Word Sequences

Fumito Konaka and Takao Miura(✉)

Department of Advanced Sciences, HOSEI University,
3-7-2 KajinoCho, Koganei, Tokyo 184–8584, Japan
fumito.konaka.2t@stu.hosei.ac.jp, miurat@hosei.ac.jp

Abstract. In this work, we introduce new kinds of sentence similarity, called *Euclid similarity* and *Levenshtein similarity*, to capture both word sequences and semantic aspects. This is especially useful for Semantic Textual Similarity (STS) so that we could retrieve SNS texts, short sentences or something including collocations. We show the usefulness of our approach by some experimental results.

Keywords: Euclid similarity · Levenshtein similarity · Semantic textual similarity

1 Introduction

Nowadays there exist a variety of documents spread over internet. One of the typical examples is Social Networking Service (SNS), which is provided through some platform to build community or social relations among people sharing interests, activities, backgrounds or real-life connections in terms of messages, tweets or documents (*Wiki*). A social network service is provided using some mechanisms such as Blog and Twitter, some profiles and social links. These messages are characteristic because they consist of short texts chained many followers (or *retweets*), contain a few duplicate but special buzzwords (i.e., `lol`, `:)`) and ignore grammatical rules very often.

Information retrieval for these kinds of information have been widely discussed. Among others, a model of *Bag-of-Words* is common in text information processing. That is, every document is described as a vector over words with an assumption of Distributed Semantic Model (DSM) [5] which means words in similar contexts carry similar semantics. However, there exist serious deficiencies for SNS texts since sentences in SNS are generally short and sparse so that word sequences may carry characteristic semantics. For example, in two statements `"I hope to marry her"` and `"I hope to divorce her"`, we have same words except one, thus the statements have completely different semantics.

In this work, we introduce new kinds of similarity, *Euclid similarity* and *Levenshtein similarity* between sentences to provide a new approach towards information retrieval of sentences including BLOG/Twitter. The basic idea comes from *semantic* distance. Euclid similarity allows us to obtain better similarity based on multiple word expression (as n-gram and collocation) considered

G. Amato et al. (Eds.): SISAP 2015, LNCS 9371, pp. 244–249, 2015.
DOI: 10.1007/978-3-319-25087-8_23

as a unit. They happen to co-occur often closely positioned. We also introduce semantic similarity into Levenshtein distance and the new difference of two words reflect the similarity. This provides us with independence of sentence length for similarity. In fact, we may have same context but much differnce of size[1] and we see the approach works better than dependent ones.

The rest of the paper is organized as follows. In section 2 we describe semantic similarity for sentences as well as some related works. Section 3 contains a framework of our approach including extended Levenshtein distance. Section 4 contains some experimental results. In section 5 we conclude this work.

2 Semantic Similarity

Semantic Textual Similarity (STS) provides us with new kinds of text retrieval. In fact, it allows us to capture semantic structure directly by means of word/phrase sense and the interrelationship same as grammatical structure instead of word distribution (Bag-Of-Words, BOW) model. For instance, *a small elephant looks at a big ant* is completely different in size from *a big elephant looks at a small ant*, but the two are same from the viewpoint of BOW model.

Here we like to focus attention on sequences as a new feature. The typical issue is *collocation*, which is a sequence of words or terms that co-occur more often than would be expected by chance (*WIKI*). Note that an *idiom* means a phrase carrying semantics different from the constituents[2], and that *collocation* means an expression of several words which likely happens more often[3].

On the contrary, we have different expression to describe identical situation. For example, two sentences `"His lecture came across well."` and `"His lecture resonated well."` talk about identical fact since `come across` means `resonate` though different length. We should examine words and collocation enough for powerful retrieval.

There have been several work of the similarity proposed so far. Tubaki et al. discuss a fundamental model based on word-vector space and sentence structures[5]. In fact, they examine model how to learn word description using sentence structure optimization and decomposition, and propose semantic similarity defined by kernel functions.

Islam et al. has proposed another similarity putting attention on word strings and word similarity[2]. By this approach *miss-spell* aspects could be involved.

Feng et al. has discussed similarity between sentences using *similarity of word sequences*[1]. The approach provides us with some improvement caused by short sentences, although they ignore collocation aspects. However, let us note that these approach show the results depending on the length (the number of words) heavily. Also no discussion is found about collocation.

[1] Some texts of different size talk about same content many times: "Please don't let this get you down", "Keep fighting and never give up", "Be strong". Love from Britain, Dec.12, 2014 in FaceBook message for Julia Lipnitskaya.

[2] "*I eat eyeball*" means "*I am scolded*" but not have any food.

[3] We may say "*something like that*" more likely as a custom.

3 Semantic Distance

In this section, we discuss how we should think about semantic distance between sentences putting attention on *sequence* of words and introduce 2 kinds of similarity, *Euclid similarity* and *Levenshtein similarity* to do that. Here we consider similarity as a certain value in $0 \sim 1$ and the smaller value means less similar.

3.1 Euclid Similarity

Assume word similarity $Sim_E(w_i, \bar{w}_i)$ for any two words w_i and $\bar{w}_i$ and we like to extend the definition for capturing sentence similarity between S_1, S_2. If both S_1, S_2 contain the same number (say, k) of words where each i-th word is w_i and $\bar{w}_i$ respectively. Then we define the similarity $Sim_E(S_1, S_2)$, called *Euclid Similarity*, as follows: $Sim_E(S_1, S_2) = \dfrac{\sqrt{\sum_{i=1}^{k} Sim(w_i, \bar{w}_i)^2}}{k}, w_i \in S_1, \bar{w}_i \in S_2$.

Next, let us define similarity of any two sentences using Euclid similarity. Note we assume the same number of words in two sentences in the definition of Euclid similarity. In a sentence S, we call consecutive n words in S by a *shingle*. To introduce the similarity, first we decompose the two sentences into the same number (k) of shingles, and then we give the similarity between the two shingles.

In English, it is well-known that any collocation (n-gram) may carry its own meaning with the length at most $n = 4$. We decompose S into the sequence of shingles of $n = 1, .., 4$ so that we have $s/4 \sim s$ shingles. To decompose two sentences S_1 and S_2 into k shingles, we obtain possible range of the decomposition and select the common possibility suitable for both S_1 and S_2.

We obtain two ranges I_1, I_2 for S_1, S_2 respectively and calculate a new range I_0 as $I_1 \cap I_2$ We say the similarity is 0 if no possibility is found (i.e., $I_0 = \phi$).

For each k in I_0, we decomopose S_1, S_2 into k shingles $(w_1, .., w_k)$ and $(w'_1, .., w'_k)$. Let w, $\bar{w}$ be two shingles and define the similarity $Sim_E(w, \bar{w})$ of the two shingles. Any shingle may or may not contain collocations which we can be see by examining dictionary. If the case, we put the constituent words together into one so that we still consider the shingle as a sequence of words. When we have several collocations in the shingle, we *make* copies of shingle containing different collocations to obtain similarity alternatively. Now the similarity $Sim_E(w, \bar{w})$ is defined as follows: $Sim_E(w, \bar{w}) = \max_{i,j} Path(a_i, b_j)$. Here a_i, b_j mean each word/collocation in w and $\bar{w}$ respectively, none of the two is *stopword* or something like that[4]. $Path(a_i, b_j)$ is calculated using WordNet[5][4].

3.2 Levenshtein Similarity

Here let us introduce another kind of sentence similarity *Levenshtein similarity*, denoted by $Sim_L(S_1, S_2)$. Compared to Euclid similarity, we examine all the pairs of words appeared in S_1 and S_2 while keeping word sequences.

[4] In our approach, we extract only nouns and verbs so that, for instance, pronouns are removed.

[5] http://wordnet.princeton.edu/

For sentences S_1, S_2, we define *Levenshtein similarity* between S_1 and S_2, denoted by $Sim_L(S_1, S_2)$, as follows:

```
Levenshtein Similarity Calculate $Sim_L(S_1, S_2)$
(1) $M_{i,0} = i(0 \leq i \leq m), M_{0,j} = j(0 \leq j \leq n)$
(2) $M_{i,j} = \min($
        $M_{i-1,j-1} + (1 - Path(w_i, \bar{w}_j)),$
        $M_{i,j-1} + (1 - Path(w_i, \bar{w}_j)),$
        $M_{i-1,j} + (1 - Path(w_i, \bar{w}_j)))$
(3) $Sim_L(S_1, S_2) = 1 - M_{m,n} / \max(m, n)$
```

In the definition we examine WordNet many times for whole sentence. This means our Levenshtein definition captures semantic similarity instead of character matching. Let us note we examine all the word pairs looking at WordNet.

3.3 Using Semantic Distances

Clearly it is hard to decide how well we obtain sentence similarity, because the result depends on contexts, domains and language nature. In this work, we model the situation by a parameter λ as well as the two similarities where $0 \leq \lambda \leq 1.0$: $Sim(S_1, S_2) = \lambda \times Sim_E(S_1, S_2) + (1 - \lambda) \times Sim_L(S_1, S_2)$.

The parameter λ tells us how well sequences give similarity, it is impossible to estimate λ automatically. In the following section, we show some experimental results of our model.

4 Experiments

In this experiment, we examine 30 pairs of nouns among 65 pairs discussed in [3] and referred in [1], [2], [3]. Then we have interpreted these nouns with the *first* interpretation in the Collins Cobuild dictionary [6] and applied TreeTagger[7] for morphological processing in advance.

We examine extended semantic distance. We construct Levenshtein similarity distinguishing nouns from verbs by giving weight 0.5 for Euclid similarity, 0.3 for Levenshtein similarity on nouns and 0.2 for Levenshtein similarity on verbs. These values have been devised through preliminary experiments.

To evaluate the experiment, we examine precision of ranking proposed by [3] with two baseline results [1], [2].

Let us illustrate all the results of Precisions and Ranking in tables 1 and 3 respectively. In table 1, TopRank means the number of pairs ranked highly. A table 1 shows that, compared to our result, Islam approach works 10 percent worse and Feng et al. approach equally.

In table 3, we see that our approach contains a pair of "coast&forest" in top 10 rank which is 21th in Answer, and that our approach doesn't contain any pair in top 10 rank which is 21th or below in Answer.

[6] http://www.collinsdictionary.com/dictionary/english-cobuild-learners

[7] http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/

Table 1. Precision

TopRank	Ours	Islam	Feng
1	1	0	1
5	0.8	0.6	0.8
10	0.6	0.8	0.6
15	0.6	0.8	0.67
20	0.7	0.85	0.75
25	0.88	0.96	0.88
30	1	1	1
Average	0.8	0.72	0.81

Table 2. Unified Explanatory Notes

coast & shore		
Position	”coast”	”shore”
1	the	the, shore, or the
2	coast	shore, of, a, sea
3	is	lake, or, wide, river
4	an, area	is, the, land, along
5	of, land	the, edge, of, it
6	that, is	some one, who, is_on, shore
7	next, to	is_on, the, land, rather
8	the, sea	than, on, a ship
coast & forest		
Position	”coast”	”forest”
1	the	a, forest, is
2	coast, is, an, area	a, large, area
3	of, land, that, is	where, trees
4	next, to, the, sea	grow, close, together

Table 3. Ranking

Rank	Answer	Ours	Islam	Feng
1	midday&noon	midday&noon	cock&rooster	midday&noon
2	cock&rooster	cock&rooster	midday&noon	cock&rooster
3	cemetery&graveyard	serf&slave	gem&jewel	cemetery&graveyard
4	gem&jewel	forest&woodland	boy&lad	gem&jewel
5	forest&woodland	gem&jewel	automobile&car	boy&lad
6	coast&shore	cemetery&graveyard	implement&tool	cord&string
7	implement&tool	coast&forest	cemetery&graveyard	serf&slave
8	boy&lad	boy&rooster	cord&string	automobile&car
9	automobile&car	journey&voyage	coast&shore	grin&smile
10	cushion&pillow	automobile&car	serf&slave	boy&rooster
11	grin&smile	hill&woodland	journey&voyage	boy&sage
12	serf&slave	boy&lad	magician&wizard	magician&wizard
13	cord&string	magician&oracle	forest&graveyard	journey&voyage
14	autograph&signature	grin&smile	grin&smile	asylum&fruit
15	journey&voyage	magician&wizard	furnace&stove	magician&oracle
16	magician&wizard	boy&sage	cushion&pillow	coast&shore
17	furnace&stove	autograph&signature	hill&woodland	autograph&signature
18	hill&mound	forest&graveyard	glass&tumbler	cushion&pillow
19	oracle&sage	coast&shore	coast&forest	furnace&stove
20	hill&woodland	cord&string	forest&woodland	autograph&shore
21	glass&tumbler	implement&tool	magician&oracle	glass&tumbler
22	coast&forest	autograph&shore	autograph&signature	forest&woodland
23	magician&oracle	glass&tumbler	boy&rooster	implement&tool
24	boy&rooster	asylum&fruit	boy&sage	forest&graveyard
25	forest&graveyard	furnace&stove	hill&mound	hill&woodland
26	boy&sage	cushion&pillow	bird&woodland	oracle&sage
27	cord&smile	oracle&sage	autograph&shore	hill&mound
28	asylum&fruit	bird&woodland	oracle&sage	bird&woodland
29	bird&woodland	hill&mound	asylum&fruit	cord&smile
30	autograph&shore	cord&smile	cord&smile	coast&forest

To examine our approach and the baselines, let us discuss the differences. In table 3, our approach says ”coast & shore” and ”coast & forest” are ranked as 19th and 7th respectively which are 6th and 22th in Answer.

Explanatory notes (in the Collins) of the words ”coast”, ”shore” and ”forest” are *The coast is an area of land that is next to the sea.”*, *”The shores* *or the shore of a sea, lake, or wide river is the land along the edge of it. Someone who is on shore is on the land rather than on a ship."* and *"A forest is a large area where trees grow close together."* respectively.

First we see big difference of the notes length of "coast & shore". Table 2 contains the unified result of the notes. We examine "shore" and obtain collocation, in fact, we have Paths values in similarities. $Path(coast, shore) = 0.5$, $Path(area, land) = 0.08$, $Path(sea, ship) = 0.08$. Note that 1st, 3rd, 5th, 6th and 7th have similarity 0.

As for "coast & forest", we get almost same lengh of the explanatory notes (of "coast" and "forest"). Again a table 2 contains the unified result of the notes. Also Path values show $Path(area, area) = 1$ and $Path(land, tree) = 0.17$. The 1st and 4th words contain similarity 0.

Similarly Path values see that "land & tree" are more similar rather than "area &land" and "sea & ship". That's why "coast & shore" is ranked lower and "coast & forest" higher. Though the result heavily depends on the notes (in the Collins), we might expect our approach generally collects possible notation fluctuations about focused terms in a sentence.

5 Conclusion

In this work, we introduced two kinds of similarity, Euclid similarity and Levenshtein similarity to model sequence and semantics of words for the purpose of STS and short text retrieval. Then we introduced semantic similarity between sentences.

Our experimental result shows that the precision results are generally nice results, say 10 percent better than Islam[2] in top 10 pairs, for example.

References

1. Feng, J., Zhou, Y.M., Martin, T.: Sentence similarity based on relevance. In: Proceedings of IPMU, vol. 8, p. 833 (2008)
2. Islam, A., Inkpen, D.: Semantic text similarity using corpus-based word similarity and string similarity. ACM Transactions on Knowledge Discovery from Data (TKDD) **2**(2), 10 (2008)
3. Li, Y., McLean, D., Bandar, Z.A., O'shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering **18**(8), 1138–1150 (2006)
4. Pedersen, T., Patwardhan, S., and Michelizzi, J.: WordNet: similarity: measuring the relatedness of concepts. In: Demonstration Papers at HLT-NAACL 2004, pp. 38–41. Association for Computational Linguistics, May 2004
5. Tsubaki, M., Duh, K., Shimbo, M. and Matsumoto, Y.: Modeling and learning semantic co-compositionality through prototype projections and neural networks. In: Conf. on Empirical Methods in Natural Language Processing (EMNLP) (2013)