Exploiting Synonymy to Measure Semantic Similarity of Sentences

Youhyun Shin* Yeonchan Ahn* Hyuntak Kim** Sang-goo Lee*

*School of Computer Science and Engineering
Seoul National University
Seoul, Korea
{shinu89, acha21, sglee}@europa.snu.ac.kr

**Software R&D Center
Samsung Electronics
Seoul, Korea
hyuntak7.kim@samsung.com

ABSTRACT

The importance of semantic similarity measures between sentences is increasingly growing in text mining, text clustering, and question answering. Many studies have focused on finding exact term matching to predict sentence similarity. In this paper, we present a method for measuring sematic similarity of sentences based on constructed synonymy graph to avoid considering just exactly matching terms. When we construct graph which has terms as nodes and synonymy relation as edges, we use WordNet and part-of-speech to exploit synonyms. We assume synonym of a synonym is also similar; it takes advantage of the fact friend of a friend is likely to be a friend as well in real-world. With this concept, similarity between words is estimated by exploiting the minimum number of synonym chains between two nodes. The proposed algorithm calculates similarity of two sentences by summing all the similarities between selected words in sentences. Evaluation is conducted on two different data sets, Microsoft Research paraphrase corpus, and Yelp review dataset. Experimental evidences show that 1) the proposed method is more accurate compared to existing sentence similarity measures and 2) using real-world dataset like Yelp reveals that the proposed method has chance to be applied to recommendation.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information Search and Retrieval – *Retrieval model*

General Terms

Algorithms

Keywords

Semantic Similarity, Sentence Similarity, Information Retrieval

1. INTRODUCTION

With the rapid growth of online text, high-quality semantic distance of sentences is needed for the purpose of information retrieval [1]. Semantic similarity measures play an importance role in text mining, question answering, and text clustering.

Most of researchers on sentence similarity have focused on exact term matching such as vector space model to calculate similarity between sentences. As a result, various similarity function using term vector weighting scheme [6] have been developed. However, there exists still have difficulty in distinguishing semantics in sentences such as paraphrased words like *drive* and *operate*, opposite words like *start* and *stop*.

Semantic similarity measures using lexical database such as WordNet would relieve this problem [2, 3, 4, 5]; WordNet has a concept hierarchy that can lead to calculate semantic similarity of words. However, it still has limitations; this only can be applied to nouns and verbs which have hypernyms and it is a way to calculate similarity between words, not sentences. Also, distance between words can be affected by domain such as polysemy cannot take into account; for example, the meaning of "*account"* is different depending on domain. "*account*" is used for meaning both *money deposit at the bank* and *explaining event or process*. This leads to the needs of new method to calculate semantic distance of sentences, use other part-of-speech like adjective and reflect domain specific relationship.

To define how similar between two words, we assume synonym of a synonym is also similar like the concept of friend of a friend. In real-world, friend of a friend is also likely to be a friend. This assumption can reveal hidden similarities, leading to better calculation. For example, in WordNet, synonyms of "*advanced*" = {*progressive, high, sophisticated}*, and synonyms of "*progressive*" = {*modern, advanced, active*}. At this time, there is a chance to be similar between *advanced* and *modern*. Because one of synonyms of *advanced* is *progressive*, and one of synonyms of *progressive* is *modern*. According to our assumption, we calculate distance between two nodes as smallest number of chains between them. For example, the distance between *advance* and *modern* is 2; because they are connected in the graph as *advanced-progressive-modern,* and there are two chains.

In this paper, we propose a novel algorithm using synonymy graph to calculate similarity between sentences. To the best of our knowledge, we believe that there is no existing works that exploiting only synonymy for measuring sentence similarity with considering part-of-speech. From the sentence only nouns, adjectives, and verbs are extracted. And then, nouns, adjectives, verbs and its synonyms become nodes in graph. To exploit synonyms, we use WordNet; nodes in this graph make edge when they have synonymy relationship. At this time, we take advantage of the concept of friend of a friend to estimate semantic similarity between two words. Therefore, the sum of shortest path of all pair words in two sentences is semantic similarity of two sentences. The main contributions of this work can be roughly summarized as follows:

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

IMCOM '15, January 08 - 10 2015, BALI, Indonesia

Copyright is held by the owner/author(s). Publication rights licensed to ACM. ACM 978-1-4503-3377-1/15/01…$15.00

http://dx.doi.org/10.1145/2701126.2701219.

1) We propose a novel algorithm to calculate semantic similarity between sentences exploiting synonymy with the concept friend of a friend and using part-of-speech. We define similarity as sum of minimum distance between selected terms in each sentence.

2) We introduce graph to estimate word distance which can take into account domain specific similarity. Graph is constructed using training dataset related with domain. Its nodes are words used in sentences and its edges are added when two words are synonymy relation. This domain specific relationship makes difference between the distance from WordNet and our method.

3) We show experimental evidence that our method outperforms traditional approaches to find similar sentences and it can be applied to real-world dataset in recommendation scenarios.

The remainder of this paper is organized as follows: In Section 2, we briefly present the related work in semantic similarity. Section 3 describes the proposed graph model and algorithm. Section 4 presents evaluation experiments and their results. Conclusion and future work are presented in Section 5.

2. RELATED WORK

2.1 Semantic Similarity

Semantic similarity in Natural Language Processing, the idea of computing distance between words or sentences is based on the relatedness or likeness of their meaning unlike exact term matching using similarity measure such as cosine similarity. Semantic similarity can be estimated by the distance between words; for example, many researches with the help of WordNet terms in sentence are represented as nodes of a directed acyclic graph with taxonomic relations. [3, 5, 7, 8, 9] utilize shortest path of words as semantic similarity. Li et al. [2] extend semantic similarity of words to sentences. In this work, WordNet is used to exploit synonymy for only extracted word from sentences to reveal hidden similarities, leading to better calculation for semantic distance in synonymy graph.

2.2 Natural Language Processing

Natural Language Processing (NLP) like part-of-speech tagging, stemming, stop-words removal aims to improve quality. Part-of-speech tagging relieves semantic ambiguity. Also, stop-words which are regarded as words do not have any impact meaning are removed. Julian et al. [4] exploit only nouns, adjectives, and verbs as kind of stop words for better clustering. To avoid missing words with the same meaning appear in various morphological forms, stem of word is used. We exploit part-of-speech, especially nouns, adjective, and verbs as our nodes in synonymy graph.

2.3 Social Network

In real-world, friend of a friend have a big potential to be a friend. Similarly, Small world has a path in a graph which alternate a sequence of nodes with edges that starts with a node and ends with an adjacent node. In addition to this, six degrees of separation theory that everyone and everything is six or fewer steps away, so that a chain of "a friend of a friend" can be lead to connect any two people in a maximum of six steps. According to some social network properties, we define the similarity as number of synonyms between words and adapt six degrees of separation theory.

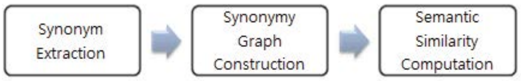

Figure 1. Semantic similarity computation process diagram

3. A NOVEL METHOD FOR MEASURING SEMANTIC SIMILARITY

Figure 1 describes procedure for how semantic similarity is computed. To estimate the sentence similarity, firstly text processing to extract candidate words for finding synonyms based on part-of-speech. Secondly, synonymy graph is constructed to compute similarity between words. Last process is semantic similarity computation; it computes pair-wise word similarities in two sentences, and then, it sums all the maximum similarity of words to compute semantic sentence similarity.

3.1 Synonym Extraction

Text data is pre-processed by Natural Language Processing (NLP) like part-of-speech tagging, stop-word removal, stemming. Only nouns, adjectives, and verbs are extracted from text dataset via part-of-speech tagging. For example, "*Very active and funny space, but pizza was oily. It is little bit far from downtown.*" is transformed into "*active, funny, space, pizza, oily, downtown*".

Using WordNet, we find synonyms for these extracted words. According to Figure 2, synonyms of "*active*" = {energetic, alive, live, hot}. These 4 words are can be added only when it is used before in that training domain. It means that all the nodes included in semantic graph reflect specific domain context.

3.2 Synonymy Graph Construction

After text data processing, each noun, adjective, and verb now has list of synonyms. In this_step, we construct a graph called synonymy graph. In this graph nodes represent words extracted from sentences and edges represent synonym relation between words. So, the graph would be constructed; its node is each extracted word from sentences, and edge is added when the relationship between words are synonym. For example, if "*active*" has synonym list like "*energetic, alive, live, hot*" and "*energetic*" has "*active, brisk, merry*"; Figure 2 shows process of making synonymy graph which can convey the concept synonym of a synonym would be synonym.

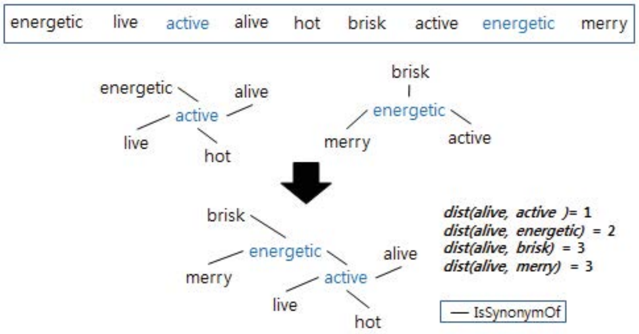

Figure 2. Example of synonymy graph construction

3.3 Semantic Similarity Computation

Based on constructed graph, the similarity between two sentences is calculated as follows:

1. **Generation of candidate pair** We calculate words' pair-wise similarity for all the words of which part-of-speech are matched. For example, if a sentence has words "*great", "dinner*", and the other sentence has words "*delicious", "food*". In this case, only two pairs which have same part-of-speech separately, (*great, delicious*) and (*dinner, food*) are computed to estimate similarity.

2. **Calculation of pair-wise similarity** From pair-wise similarity result, take maximum as similarities between sentences. For example, in Figure 2 there is 3 chains of synonyms between *alive* and *brisk*, and *alive* and *active* have direct synonym relationship, it takes (*alive, active*) to compute similarity, because $dist(alive, active) = 1$ is shorter than $dist(\text{alive}, \text{brisk}) = 3$. To compute the similarity between words, we define similarity between words like $\text{Sim}_w(alive, active) = (2)^{-1} = 0.5$. At this time, we assume if there are no intermediate synonyms between words, we set distance as 6 based on six degrees of separation theory which is in real-world everyone and everything are six or fewer steps away.

3. **Normalization** sum of the maximum similarities and divided by maximum number of words participated in pairwise calculation become semantic similarity between sentences.

w_i, w_j is one of noun, adjective, and verb in sentence S_1, S_2 separately. $dist(w_i, w_j)$ is the number of minimum chains of synonyms between w_i and w_j . Similarity between w_i and w_j , denoted as $Sim_w(w_i, w_j)$ is computed as follows:

$$Sim_w(w_i, w_j) = \begin{cases} 1 & \textit{if } w_i \textit{ and } w_j \textit{ is exactly matched} \\ \left(1 + dist(w_i, w_j)\right)^{-1} & \textit{otherwise} \end{cases}$$

Semantic similarity between sentence S_1 and S_2 is calculated by summing maximum similarities of words and normilizing.

$$Sim_{syn}(S_1,\ S_2) = \frac{\sum_{w_i \in s_1,\, w_j \in s_2}(\max Sim_w(w_i,\ w_j))}{\max(|S_1|, |S_2|)}$$

In case of noun and verb, their hypernym similarity can be computed. To take hypernym similarity into consideration, word similarity becomes α · hypernym similarity plus (1- α) · $Sim_w(w_i, w_j)$. An example of applying to sentences, "*Very active and funny space*" and "*Crowded and energetic space*" is depicted as Figure 3. At step 1, *active, funny, space* and *crowded, energetic, space* are extracted via text processing. At step 2, the word *space* is exactly matched, similarity is 1 based on defined formulation. Then, others calculate their pairwise similarity. Then, set maximum to similarity between two words. Step 4, sum all the maximum semantic similarity of words and normalized it.

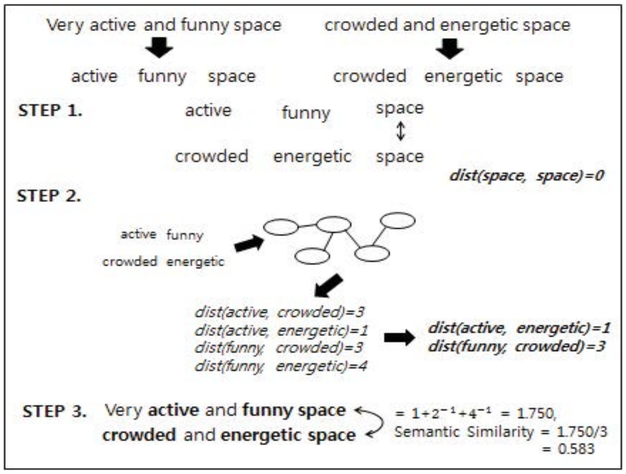

Figure 3. Example of semantic similarity computation

4. EXPERIMENT

We conduct experiments on two data sets to evaluate the proposed semantic similarity measure. First, we compare our similarity measure with other methods for Microsoft Research Paraphrase Corpus dataset. Second, we show that our similarity measure can be adapted to application of recommendation using Yelp dataset.

4.1 Experimental Setup

In this section, two different datasets are used to conduct evaluation in various aspects. MSRP is Microsoft Research Paraphrase Corpus, which contains 4,076 sentence pairs for training and 1,725 sentence pairs for test and its semantic equivalence is judged by two human assessors. As a base line, there are 67% of positive values. Precision, Recall, and F-Measure are computed as:

$$Precision = \frac{|TP|}{|TP| + |FP|} \qquad Recall = \frac{|TP|}{|TP| + |FN|}$$

$$F - Measure = 2 \cdot \frac{pecision \cdot recall}{precision + recall}$$

Also, we evaluate the proposed semantic graph for data set extracted from Yelp. 60,000 reviews for constructing semantic graph are randomly sampled train dataset. To verify the possibility to use semantic sentence similarity for recommendation, we use Yelp data set described as Table 1, 1,573 venues and 1,896 users from Yelp review dataset. This experiment is designed to show indirect impact on guessing review rating, when we assume that similar reviews have similar ratings.

Table 1. Yelp data set statistics

Venues	User	Density
1573	1896	0.00656

Test dataset for evaluation is 500 reviews. Each has 1 to 5 stars. $R_{i,j}$ is rating venue i from user j, $\hat{R}_{i,j}$ is estimated rating for venue i,user j. $|T|$ is number of ratings in test set. Evaluation measure to show the performance is Root Mean Square Error(RMSE) :

$$\text{RMS}E = \sqrt{\frac{1}{|T|} \sum_{R_{i,j} \in T} (R_{i,j} - \hat{R}_{i,j})^2}$$

4.2 Results

In order to evaluate the proposed method, denoted as sim_{syn}, we compare the performance of sim_{syn} with different sentence similarity method [6]; other semantic similarity measuring which exploit WordNet is sim_{sem} , vector model having weighting scheme as TFIDF is sim_{tfidf}, and vector model using only nouns, adjectives, and verbs is sim_{pos}. These results imply our method finds paraphrased sentence pair more accurately than other existing approaches in Table 2.

Table 2. Performance comparison on MSRP data set Result in () indicate the number of correct pairs.

	sim_{syn}	sim_{sem}	sim_{tfidf}	sim_{pos}
Precision	0.724	0.690	0.716	0.722
Recall	0.912	0.964	0.795	0.803
F-Measure	0.807	0.804	0.753	0.760

4.2.1 Performance of Similarity Measure

Both sim_{syn} and sim_{sem} measures semantic similarity to compute pairwise sentence similarity. Table 2 shows that sim_{syn} measures pairwise semantic similarity better than sim_{sem} to find paraphrased pair accurately. This can be explain as WordNet-based similarity measures which only uses words having hypernyms have limit in word coverage. Also, sim_{syn} and sim_{sem} which measure semantic similarity perform better in f-measure compared with sim_{tfidf} which uses exact match to calculate vector similarity. sim_{pos} uses vector space model with TFIDF weighting scheme, and term vectors are only nouns, adjective, and verbs. It performs better than sim_{tfidf} which uses all part-of-speech. Also, in Table 3, RMSE of sim_{pos} is better than sim_{tfidf}; it shows that influence of part-of-speech clearly.

Table 3. Performance comparison on Yelp data set.

	sim_{syn}	CF	sim_{sem}	sim_{tfidf}	sim_{pos}
RMSE	**1.121**	1.133	1.284	1.309	1.268

4.2.2 Possibility of high-quality recommendation

We consider applying semantic similarity to find neighbor. sim_{syn} makes slight improvement of the performance compared with well-known approach for recommender system CF in Table 3. This observation support that fine tuning proposed synonymy graphs can contribute to high-quality recommendation.

5. CONCLUSION AND FUTURE WORK

In this paper we propose a novel sentence similarity measuring method based on synonymy graph which trains synonymy relation between words used in domain dataset. We extract nouns, adjectives, and verbs by natural language processing to use these as nodes and synonymy relation. We define similarity, which is based on the concept friend of a friend; synonym of a synonym is similar as well. This leads to compute how similar between two words. We sum all the minimum distance between nodes which match with words in sentences to compute sentence similarity. Evaluation results show that the proposed method performs better, and further verify chance to be applied to application like recommender system.

As future work we will conduct experiment on other dataset like Q&A dialogue which has many paraphrased sentences. We also consider edge weighting to further improve the performance of measuring semantic similarity.

6. ACKNOWLEDGMENTS

This paper was supported by Samsung Electronics.

7. REFERENCES

[1] Harispe, S. Ranwez, S. Janaqi, S. Montmain, J. 2013. Semantic Measures for the Comparison of Units of Language, Concepts or Entities from Text and Knowledge Base Analysis. ArXiv Corr.

[2] Li, Y. McLean, D. Bandar, Z. A. O'Shea, D. J. Crockett, K. 2006. Sentence Similarity Based on Semantic Nets and Corpus Statistics, IEEE Transactions on Knowledge and Data Engineering (TKDE'06), 1138-1150.

[3] Ahsaee, M.G. Naghibzadeh, M. Yasrebi, S.E. 2010. Using WordNet to determine semantic similarity of words, Telecommunications, 5th International Symposium on (IST'10), 1019-1027.

[4] Sedding, J. Kazakov, D. 2004. WordNet-based Text Document Clustering, Proceedings of the 3rd Workshop on Robust Methods in Analysis of Natural Language Data (ROMAND'04), 104-113.

[5] Liu, G. Wnakg, R. 2011. A WordNet-based Semantic Similarity Measure Enhaced by Internet-based Knowledge, Proceedings of the 23rd International Conference on Software Engineering & Knowledge Engineering (SEKE'11), 175-178.

[6] Achananuparp, P. Hu, X. Shen, X. 2008. The Evaluation of Sentences Similarity Measures. Proceedings of the 10th international conference on Data Warehousing and Knowledge Discovery (DaWak'08), 305-316.

[7] Lin, D. 1998. An information-theoretic definition of similarity. Proceedings of the 14th International Conference on Machine Learning (ICML'98), 296-304.

[8] Resnik P. 1995. Using information content to evaluate semantic similarity in a taxonomy. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'95), 448-453

[9] Jiang J. Conrath D. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. Proceedings of Research in Computational Linguistics, 19–33.