RICOH at SemEval-2016 Task 1: IR-based Semantic Textual Similarity Estimation

Hideo Itoh
RICOH Company, LTD.
Institute of Information and Communication Technology
16-1, Shinei, Tsuzuki, Yokohama, Japan
hideo.itoh@nts.ricoh.co.jp

Abstract

This paper describes our IR (Information Retrieval) based method for SemEval 2016 task 1, Semantic Textual Similarity (STS). The main feature of our approach is to extend a conventional IR-based scheme by incorporating word alignment information. This enables us to develop a more fine-grained similarity measurement. In the evaluation results, we have seen that the proposed method improves upon a conventional IR-based method on average. In addition, one of our submissions achieved the best performance for the "post-editing" data set.

1 Introduction

Given two sentences, Semantic Textual Similarity (STS) measures their degree of semantic equivalence (Agirre et al., 2015). This fundamental functionality can be used for many applications such as text search, classification and clustering.

This paper describes our monolingual (English) STS system which participated in SemEval 2016 task 1. Our objective is to improve a conventional IR (Information Retrieval) based method for the STS task. In general, an IR-based method estimates semantic similarity between given sentences using similarity between document search results which are obtained using each sentence as a search query. This scheme allows us to utilize a large document database for handling diverse semantic phenomena. However, in our preliminary experiments using past SemEval test data, a conventional IR-based method was not so effective.

In failure analysis of the method, we found the following:

- It is not sufficient only to measure the commonality among sentences.
- It is necessary to measure the importance of the identified commonality in each sentence.

Based on these findings, we propose a new IR-based method for STS. In this method, word alignment techniques are applied to refine the assessment of commonality. The importance of the commonality in each sentence is measured using IR techniques.

2 Related Work

Let us define a conventional IR-based approach more formally and generally. For a given text T, a document retrieval is performed using T as a search query. The search target U is a set of documents. As a result, a document set $R(T, U)$ is obtained. Because U has been fixed in our research, we denote $R(T, U)$ as $R(T)$. The semantic similarity between two texts $T1$ and $T2$ is measured as the similarity between document sets $R(T1)$ and $R(T2)$.

Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007) is a well-known IR-based approach. In their research, Wikipedia articles are used as U. $R(T)$ is regarded as a vector of documents and the cosine similarity of vectors is used to measure the semantic similarity.

In previous STS competitions, we can find IR-based approaches in Buscaldi et al. (2013), more recently Buscaldi et al. (2015). In their research, $R(T)$ is regarded as a list of ranked documents with search scores. They defined an original similarity score using the rank and search score of each document.

Proceedings of SemEval-2016, pages 691–695,
San Diego, California, June 16-17, 2016.

3 Our Approach

Figure 1 both illustrates our approach and contrasts it with conventional IR-based methods.

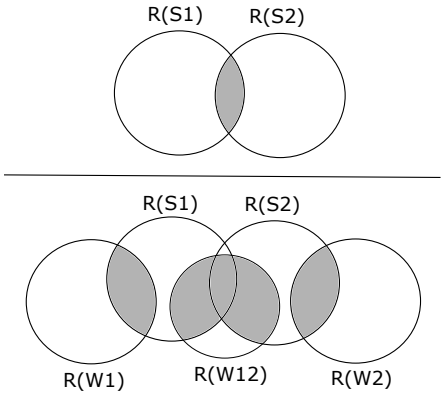

Figure 1: IR-based Semantic Similarity Measures

A conventional IR-based approach is shown at the top of the figure and is followed by our approach. Each of circles $R(T)$ represents documents retrieved for a query text T. Assuming the retrieved documents are ranked by search scores, the top-N documents are used as $R(T)$, where N is a constant. For the STS task, the conventional approach only uses the sentences $S1$ and $S2$ as search queries. The similarity between $S1$ and $S2$ is measured by calculating the ratio of the gray-colored set intersection to the union of $R(S1)$ and $R(S2)$. Specifically, the similarity function Sim is given by the Dice coefficient (Dice, 1945) expressed in Formula 1.

$$Sim(S1, S2) = \frac{2|R(S1) \cap R(S2)|}{|R(S1)| + |R(S2)|} \quad (1)$$

In our approach, additional document sets $R(W1)$, $R(W2)$ and $R(W12)$ are used, where $W12$ is a set of words which are aligned between sentence $S1$ and $S2$, while $W1$ and $W2$ are the set of words that are left unaligned in $S1$ and $S2$, respectively. Our method still computes the conventional $Sim(S1, S2)$ but then also calculate $Sim(S1, W1)$ and $Sim(S1, W12)$. Similarly, $Sim(S2, W2)$ and $Sim(S2, W12)$ are also obtained. All together, five similarity values are used as regression features within a model trained to generate the final similarity score. In other words, our approach extends a conventional IR-based scheme by incorporating word alignment information and this allows us to develop a more fine-grained similarity measurement.

$Sim(S1, W12)$ and $Sim(S2, W12)$ can be regarded as a measurement of the importance of $W12$, the aligned words, in sentence $S1$ and $S2$ respectively. When $Sim(S1, W1) > Sim(S1, W12)$, the aligned material $W12$ may not be central to the meaning of $S1$. Therefore, even if the value of $Sim(S1, S2)$ is large, the overall meaning of $S1$ and $S2$ may not be very similar to each other.

We use the Dice coefficient to keep our system simple. The optimal similarity function is left as an open question for future work.

4 System Description

4.1 System Overview

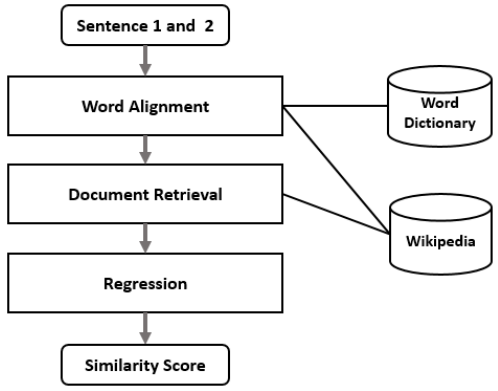

Figure 2: System Overview

We show an overview of our system in Figure 2. As data resources, we used the following:

- The Online Plain Text English Dictionary[1].
- English Wikipedia pages, snapshot on November 2nd, 2015[2].

For each pair of input sentences, we use a fuzzy word alignment procedure described later in this section to produce the following three sets of words:

W12: Aligned words in sentences 1 and 2.
W1: Unaligned words in sentence 1.
W2: Unaligned words in sentence 2.

Each of the word sets is used as a search query. The search target is a document database built from the Wikipedia data set. We used Apache Solr 5.3.1[3] as our IR system. Top-N document IDs returned by the IR system for a given query are used as the search

[1] http://www.mso.anu.edu.au/~ralph/OPTED/
[2] https://dumps.wikimedia.org/enwiki/20151102/
[3] http://lucene.apache.org/solr/

result. The Dice similarity scores obtained from the different query formulations are then used to construct regression features for a model that is trained to generate the final similarity score.

In the following subsections, we describe the details of each step.

4.2 Preprocessing

Each of the input sentences is converted to a word sequence. All non-alphanumeric characters are removed. Words are identified by splitting text using white space. In addition, within-word splitting is performed such as "50mm" to "50" and "mm" whenever the characters transition from numeric to alphabetic and vice versa. Stop words identified using a dictionary[4] are eliminated.

4.3 Spell Correction

Apache Solr 5.3.1 provides a function to suggest similar spellings for a query word. This is performed by fuzzy matching between a query word and words in a database index. The result is a list of words sorted by the degree of the similarity in spelling to the original query term. If a query word is found in a database index, its original spelling is always first ranked. We utilized this function for spell correction. Specifically, we replace each word with its top ranked suggested spelling whenever it is different from the original search term.

4.4 Use of Wikipedia Redirect Relations

The Wikipedia data includes information on redirection between pages. Most of the redirections are based on different names for the same underlying topic. For titles that correspond to a single word, we can use the redirection relations to identify synonymous words. For this purpose, a database of the redirect relations is constructed. A record within this database corresponds to one redirect relation and it includes the titles of source and target pages of the redirection. For each pair of words w1 and w2 from within sentences 1 and 2, we used the database to search for possible redirect relationships. When a redirect relationship is found to exist, the corresponding pair of words are aligned to each other and added to the aligned word set W12.

[4] http://www.ranks.nl/stopwords

4.5 String Matching Based Alignment

Additional alignment pairs between words and phrases are extracted using string matching heuristics. The following alignments are handled.

- Word unigram to bigram alignment (e.g., "backstroke to back stroke"). The global alignment algorithm (Needleman and Wunsch, 1970) is used for this approximate matching.
- Acronym alignment (e.g., "GE" to "General Electric"). This is performed by matching consecutive capitalized letters with a phrase that can be used to construct the given sequence of capitalized letters as an acronym. We do not use a dictionary of acronym for this.

Aligned words or phrases are added to W12.

4.6 Dictionary Based Alignment

Using the English word dictionary mentioned in subsection 4.1, the system detects alignments between a word and its derivative forms such as "wear" versus "worn" or "America" versus "American". Semantic information such as synonyms in the dictionary are not used in order to avoid spurious alignments which would be only appropriate in specific contexts. All of the dictionary aligned words are added to W12.

4.7 Spell Expansion

Apache Solr has an API for word stemming. In addition, we can use the suggester mentioned in subsection 4.3, which provides a list of similar spellings for a given word. Both functions are used to compute spelling variations for the purposes of word alignment. Specifically, the built in Porter stemmer[5] (Porter, 1997) is used for English word stemming. Solr's FuzzySuggester mechanism[6] is called for each term and we retain the top-five suggested alternative spellings as candidates for matching by the aligner. If word w1 and w2 from sentence 1 and 2 share the same spelling in any of the expanded alignment candidates, they are aligned and added to W12.

4.8 IR-based Similarity Estimation

After the word alignment processes, the word set W1, W2, W12 are fixed. Using W1, W2, W12 and

[5] http://wiki.apache.org/solr/LanguageAnalysis
[6] http://wiki.apache.org/solr/Suggester

sentence 1 and 2 as a natural language query (disjunctive word combination), a document search is performed. The BM25 relevance function provided by Apache Solr is used. The search target is the Wikipedia page abstracts[7]. The number of the pages in the database is around 12 million.

The search result is a list of ranked document IDs in descent order of the BM25 score. Top-N are used as $R(T)$ explained in section 3. The value of N is empirically set to 100 based on preliminary experiments on the STS 2015 data. This produces the document ID sets $R(S1)$, $R(S2)$, $R(W1)$, $R(W2)$ and $R(W12)$.

4.9 Scoring by Regression

The values of the features F1 – F5 are calculated using the expressions given below. The metric Sim is given by Formula 1 in section 3. The notation $|X|$ means the number of elements in a set X.

F1: $\{Sim(S1, W1) + Sim(S2, W2)\}$ / 2
F2: $\{Sim(S1, W12) + Sim(S2, W12)\}$ / 2
F3: $|W12|$ / $(|W12| + |W1| + |W2|)$
F4: $Sim(S1, S2)$
F5: F3 value calculated including stop words.

The features F1 and F2 are newly introduced by our proposal explained in section 3. F3 roughly corresponds to a similar feature used in Sultan et al. (2014)'s well-known word alignment based STS method. F4 is the similarity measure used in conventional IR-based methods explained in section 3. F5 is introduced to handle the case that input sentences contain stop words only.

We used LIBSVM (Chang and Lin, 2011) for support vector regression to generate similarity scores using the training environment provided by the TakeLab tool (Šarić et al., 2012). We used the Radial Basis Function (RBF) kernel. Training on the STS 2015 test data was performed to identify the optimal regression parameters. Similarity scores generated by the regression were trimmed to a range 0.0 - 5.0 by setting similarity scores that are less than 0 or greater than 5 to be 0 and 5, respectively.

If the word set W12 is empty, we use the feature F4 only and directly convert it to a similarity score by $\mathtt{F4} \star \mathtt{c}$, where $\mathtt{c}$ is a constant to adjust scale between the metric Sim and a similarity score with range 0.0 - 5.0. The $\mathtt{c}$ value was empirically set to 70 using the STS 2015 test data.

[7]The data file "enwiki-20151102-abstract.xml" was used.

5 Results

5.1 Evaluation

We submitted three variants of our system to the shared task evaluation. The three systems differed in how they used Wikipedia redirect relationships (explained in section 4.4) to aid in word alignment for each word w in input sentences.

RUN-b : Attempts to match w with the targets of the redirect relationships. If no matches are found, tries to match w with the titles of the redirection sources.

RUN-s : Attempts to match w with only the titles of the redirection sources.

RUN-n : Does not use Wikipedia redirection relationships.

The evaluation results (Pearson correlation with the gold standard data) of our three submitted runs are shown in Table 1. As a baseline, we also include the performance of our system when configured to operate as a conventional IR-based method. Specifically, the baseline system uses only the features F3, F4 and F5 (see section 4.9). As with Run-n, the baseline does not use the Wikipedia redirect relationships.

DATA SET	RUN-b	RUN-s	RUN-n	Baseline
answer-answer	0.5087	0.5129	0.5075	0.5287
headlines	0.7869	0.7800	0.7741	0.7701
plagiarism	0.8266	0.8299	0.8225	0.8212
postediting	0.8655	0.8625	0.8669	0.8480
question-question	0.5625	0.5232	0.5426	0.4566
MEAN	0.7116	0.7042	0.7047	0.6891

Table 1: Evaluation Results on SemEval 2016 Task 1

5.2 Discussion

Among the three submitted runs, Run-b has the best performance on average. However, in detailed analysis, we found that the contribution of the Wikipedia redirect relations is small and the differences mainly come from differences in the optimal parameter settings used for LIBSVM training with the parameters arrived at for Run-b resulting in better generalization to the test data. When the hyperparameters that were originally used to train Run-b (cost:10, gamma:1) are also used for Run-n, the mean performance of Run-n increases to 0.7104.

Run-n outperforms the baseline system on all of the data sets except answer-answer. In failure analysis, we found that words in answer-answer typically have larger document frequencies as compare to the other data sets. In other words, answer-answer consists of more general and less topical words. This harms both the performance of our approach and the conventional IR-based baseline.

On the postediting data set, Run-n achieved the best performance of participating systems in the 2016 STS shared task. However, our performance on the question-question data set is relatively poor, lowering the mean performance of our system. Within the question-question data, many of the sentences share the words from common question formulations (e.g., "What is the best way to .."). The alignment of such words puts too much focus on generic material that is less central to the core meaning of the question.

6 Conclusion

We proposed a new IR-based method for STS. The main feature is to extend a conventional IR-based scheme by incorporating word alignment information. The evaluation results show that the proposed method improves upon a conventional IR-based method on average. While we used the Dice coefficient for our IR-based similarity measure for simplicity, future work may see improved performance from alternative mechanisms for contrasting two different collections of ranked documents, such as Webber et al. (2010)'s Rank-Biased Overlap.

References

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, German Rigau, Larraitz Uria, and Janyce Wiebe. 2015. 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)*, pages 252–263, Denver, Colorado, June. Association for Computational Linguistics.

Davide Buscaldi, Joseph Le Roux, Jorge J. Garcia Flores, and Adrian Popescu. 2013. Lipn-core: Semantic text similarity using n-grams, wordnet, syntactic analysis, esa and information retrieval based features. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 1: Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 162–168, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Davide Buscaldi, Jorge Garcia Flores, Ivan V. Meza, and Isaac Rodriguez. 2015. Sopa: Random forests regression for the semantic textual similarity task. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 132–137, Denver, Colorado, June. Association for Computational Linguistics.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Lee R Dice. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of The Twentieth International Joint Conference for Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453.

M. F. Porter. 1997. Readings in information retrieval. chapter An Algorithm for Suffix Stripping, pages 313–316. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. Dls@cu: Sentence similarity from word alignment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 241–246, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for measuring semantic text similarity. In *Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 441–448, Montréal, Canada, 7-8 June. Association for Computational Linguistics.

William Webber, Alistair Moffat, and Justin Zobel. 2010. A similarity measure for indefinite rankings. *ACM Trans. Inf. Syst.*, 28(4):20:1–20:38, November.