ISCAS_NLP at SemEval-2016 Task 1: Sentence Similarity Based on Support Vector Regression using Multiple Features

Cheng Fu, Bo An
Institute of Software, Chinese Academy of Sciences, Beijing, China
{fucheng,anbo}@nfs.iscas.ac.cn

Xianpei Han, Le Sun
State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing, China
{xianpei,sunle}@nfs.iscas.ac.cn

Abstract

This paper describes our system developed for English Monolingual subtask (STS Core) of SemEval-2016 Task 1: "Semantic Textual Similarity: A Unified Framework for Semantic Processing and Evaluation". We measure the similarity between two sentences using three different types of features, including word alignment-based similarity, sentence vector-based similarity and sentence constituent similarity. The best performance of our submitted runs is a mean 0.69996 Pearson correlation which outperforms the median score from all participating systems.

1 Introduction

Semantic Textual Similarity (STS) is the task of measuring the degree of semantic equivalence of a sentence pair (Agirre et al., 2012). STS was first held in SemEval 2012 and has drawn considerable attention in recent years. STS has been widely used in a lot of natural language processing tasks such as information retrieval, machine translation, question answering, text summarization, and so on.

Previous methods for this task could be roughly divided into three categories: alignment approaches, vector space approaches and machine learning approaches (Hänig et al., 2015). Alignment approaches align words or phrases in a sentence pair, and then take the quality or coverage of alignments as similarity measure (Sultan et al., 2014). Vector space approaches represent sentences as bag-of-words vectors and take vector similarity as their similarity measure (Meadow et al., 1992). Machine learning approaches combine different similarity measures and features using supervised machine learning models (Bär et al., 2012).

In our system, we measure semantic text similarity by combining evidence from all above three categories. Specifically, we extract alignment-based similarity features, vector-based similarity features and sentence constituent similarity features from sentence pairs, and produce similarity scores between two sentences by combining these feature through a Support Vector Regression (SVR) model.

2 System Overview

To measure the similarity between two sentences, we first extract a series of features, including alignment-based similarity features, vector-based similarity features and sentence constituent similarity features. Then we combine all these features and get an overall similarity using a Support Vector Regression (SVR). In following we first describe how to extract different types of features. Then we describe how to train the SVR model.

2.1 Alignment-based Similarity Features

Sultan aligner is an open source unsupervised word alignment tool[1] whose core algorithm is described in (Sultan et al., 2014).

This aligner aligns related words in two sentences based on the following two properties of the words: firstly, whether they are semantically similar; secondly, whether they occur in similar semantic contexts in the respective sentences. As to the

[1] *https://github.com/ma-sultan/monolingual-word-aligner/*

Proceedings of SemEval-2016, pages 645–649,
San Diego, California, June 16-17, 2016.

former one, it mainly utilizes information provided by the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013) to make decisions. In the case of the latter, contextual similarity for a word pair is computed as the sum of the word similarities for each pair of words in the context of them. The system based on Sultan aligner achieved the best performance at the SemEval 2014 STS task. And then at the SemEval 2015 English STS task, they constructed a top-performing system combining output of Sultan aligner and another vector-based feature.

In our system, we use the output result of Sultan aligner as follows:

- Given two input sentences, we compute their similarity score based on the produced word alignments, using a similarity function the same as the one described in (Sultan et al., 2015).

- The word aligner is used as the basis of subsequent feature extraction. Specifically, we use the word aligner to determine sentence major constituent similarity, named entity similarity and keyword similarity.

2.2 Vector-based similarity Features

In our system, we extract two vector-based similarity features based on two different types of word vectors. Given two sentences, we first generate two vector representations, and then compute a cosine similarity between these vectors as their similarity score.

The first type of word vectors used in our system comes from Baroni et al. (2014), which were learned by Skip-Gram framework in word2vec toolkit (Mikolov et al., 2013) from a corpus of about 2.8 billion tokens. Details on their approach can be found in (Baroni et al., 2014). We get the vector representation of a sentence by combining all the vectors of words in this sentence via element-wise addition.

The second type of word vectors we used is the multi-prototype word vectors come from (Liu Y et al., 2015). They assume that a word in different topic express different meanings. In our work, we employ Latent Dirichlet Allocation (LDA) (Ng and Jordan., 2003) framework to infer the topic distribution of text, and collapsed Gibbs sampling algorithm (Griffiths and Steyvers., 2004) to assign a topic to each word in the corpus. In our work, we use Wikipedia English Dump (2012.12 Version) to learn the LDA model and TWE-2 framework, which consider each word-topic pair $<w_i, z_i>$ as pseudo word to learn. To get the representation of a sentence, we first use LDA and collapsed Gibbs sampling algorithm to get the topic assignment for each word in this sentence, and combine these vectors by the same way as the first type of vectors.

2.3 Sentence constituent similarity Features

In our system, we extract the following sentence constituent similarity features:

- **Subject similarity:** If the subjects of two sentences are the same, we assign a subject similarity 1, otherwise 0.

- **Predicate similarity:** If the predicates of two sentences are the same, we assign a predicate similarity 1, otherwise 0.

- **Object similarity:** If the objects of two sentences are the same, we assign an object similarity 1, otherwise 0.

- **Complement similarity:** If the complements of two sentences are the same, we assign a complement similarity 1, otherwise 0.

- **Named entity similarity:** We extract three similarity features based on whether there are aligned time pairs, location pairs or person pairs between compared sentences. For instance, given two sentences, if there exist locations that could be aligned between them, we will assign a 1 to the according named entity similarity, otherwise 0.

- **Keyword similarity:** Given two sentences, we acquire a keyword set for each one from the output of the Stanford CoreNLP tools[2] (Manning, Christopher D. et al., 2014). And then find out the keyword pairs that appear in the result of Sultan aligner. Finally, we get a real number between 0 and 1 as keyword similarity based on the proportion of aligned keywords in the above two sets.

[2] *http://stanfordnlp.github.io/CoreNLP/*

Generally speaking, different words in a sentence belonging to different sentence constituents may have different contributions to the semantic of the whole sentence. Based on this assumption, we extract four features to capture the similarities between the four major constituents of two sentences, including subject, predicate, object and complement. In our implementation, we extract the above constituents from the syntactic parsing result of the Stanford CoreNLP tools, and then we use the alignment result provided by Sultan aligner to judge whether they could be aligned.

In some specific domains, named entities play important roles in sentence semantic similarity. For instance, two headlines usually will be regarded to have little similarity when some named entities such as times and locations are different, although the rest parts may be quite similar. Based on the above observation, we extract three named entity similarity features for our system, according to whether there exist aligned time pairs, location pairs or person pairs between compared sentences. Concretely, if there exist aligned named entities of the above three types between two sentences, we set their corresponding similarity to 1, otherwise set it to 0. When extracting these features, we take outputs of the Stanford CoreNLP tools and the Sultan aligner as input.

Finally, we also extract an additional feature based on the keyword sets outputted by the Stanford CoreNLP tools. It is obtained by measuring the overlap between the two keyword sets of the sentence pair. In detail, given two sentences $S^{(1)}$ and $S^{(2)}$, the keyword similarity feature is a real number between 0 and 1, which is computed as the following formula,

$$Sim_{key} = \frac{2 * N_{\mathrm{Ak}}}{N_k^1 + N_k^2}$$

where N_{Ak} is the number of aligned keywords, N_k^i is the size of keyword set of $S^{(i)}$.

2.4 Support Vector Regression Model

Finally, we combine all the above features using a support vector regression model which is implemented in Scikit-learn (Pedregosa et al., 2011). We use its default SVR parameter settings(C=1.0, cache_size=200, coef0=0.0, degree=3, epsilon=0.1, gamma='auto', kernel='rbf', max_iter=-1, shrinking=True, tol=0.001, verbose=False) without further optimization.

3 Data

There are five test date sets in English Monolingual subtask (STS Core) of SemEval-2016 Task 1, which are Q&A Answer-Answer data set, Headlines data set, Plagiarism Detection data set, Post-Edited Machine Translations data set and Q&A Question-Question data set.

We trained our supervised systems using data from the past four years (2012-2015) of SemEval English STS task. For headlines we used all Headlines (2013), Headlines (2014), Deft-news(2014), SMTnews (2012) and Headlines (2015) sentence pairs. For the other four domains, we used all past annotation data.

4 Evaluation

We submitted three runs named S1, S2 and S3 before the deadline, but S3 is identical to S1 by mistake. So in this paper, we just discuss S1 and S2.

Table 1 lists the settings of run S1 and run S2. For run S1, we used all the features described in this paper. For run S2, we just used word alignment-based and sentence vector-based features.

Run	Settings
S1	Use all three different types of features
S2	Use alignment-based similarity features and vector-based similarity features

Table 1: Settings of our submitted runs for SemEval 2016

Data Set	Runs	
	S1	S2
Answer-Answer	0.4937	**0.4965**
Headlines	**0.7976**	0.7904
Plagiarism	**0.8193**	0.8121
Post-Edited	**0.8118**	0.8118
Question-Question	**0.5721**	0.5718
All	**0.6999**	0.6975

Table 2: Performances on STS 2016 test data. Each number in rows 1–5 is the correlation between system output and human annotations for the corresponding data set. The last row shows the values of the comprehensive results of each run.

Performances of our two runs on each of the STS 2016 test sets are shown in Table 2. Each bold number represents the best score on the corresponding test set. The weighted mean of correlations for each run is also shown. The results show that S1 performs equally or better than S2 in all the data sets but Answer-Answer.

Data Set	Align-Sim	Vector-Sim1	Vector-Sim2	S1
Answer-Answer	0.4776	0.1247	0.2339	0.4937
headlines	0.7938	0.6931	0.5785	0.7976
Plagia-rism	0.8185	0.7919	0.6774	0.8193
Post-Edited	0.8179	0.8183	0.7740	0.8118
Question-Question	0.5696	0.6816	0.2287	0.5721

Table 3: Performance of three individual feature and our best run (S1) on STS 2016 test sets.

Table 3 shows the performance of our best run S1 and other three runs which only use one specific kind of features. *Align-Sim* uses only word alignment information, *Vector-Sim1* uses only sentence vector information and word vectors comes from (Baroni et al., 2014), *Vector-Sim2* uses only sentence vector information and word vectors coms from (Liu Y et al., 2015).

Results in table 3 show that our approach can achieve best performance by combining different types of features.

5 Conclusions and Future Work

At SemEval 2016, we present a supervised system for English Monolingual subtask of task1 using multiple features including alignment-based similarity features, vector-based similarity features and sentence constituent similarity features. Experimental results show that our approach can achieve better performance by combining more features.

For future work, we want to better measure the similarity between the major constituents of two sentences, rather than simply use 1 or 0 as their similarity values. We also want to get better sentence vector representation by developing better word composition models.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants no. 61433015, 61572477 and 61272324, and the National High Technology Development 863 Program of China under Grants no. 2015AA015405. Moreover, we sincerely thank the reviewers for their valuable comments.

References

Hänig, Christian, Robert Remus, and Xose De La Puente. "ExB Themis: Extensive Feature Extraction from Word Alignments for Semantic Textual Similarity." SemEval-2015 (2015): 264.

Bär, Daniel, et al. "Ukp: Computing semantic textual similarity by combining multiple content similarity measures." *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 2012.

Han L, Martineau J, Cheng D, et al. Samsung: Align-and-Differentiate Approach to Semantic Textual Similarity[J]. SemEval-2015, 2015: 172.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." The Journal of Machine Learning Research 12 (2011): 2825-2830.

Sultan M A, Bethard S, Sumner T. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence[J]. Transactions of the Association for Computational Linguistics, 2014, 2: 219-230.

Sultan M A, Bethard S, Sumner T. Dls@ cu: Sentence similarity from word alignment[C]//*Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. 2014: 241-246.

Sultan M A, Bethard S, Sumner T. DLS@ CU: Sentence similarity from word alignment and semantic vector composition[C]//*Proceedings of the 9th International Workshop on Semantic Evaluation*. 2015: 148-153.

Manning, Christopher D., Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP Natural Language Processing Toolkit In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pp. 55-60.

Ganitkevitch J, Van Durme B, Callison-Burch C. PPDB: The Paraphrase Database[C]//HLT-NAACL. 2013: 758-764.

Baroni M, Dinu G, Kruszewski G. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors[C]//ACL (1). 2014: 238-247.

Liu Y, Liu Z, Chua T S, et al. Topical Word Embeddings[C]//AAAI. 2015: 2418-2424.

Zellig S. Harris. 1954. Distributional structure. Word, pages 10(23):146–162.

Blei, D. M., Ng, A. Y., & Jordan, M. I. 2003 . Latent dirichlet allocation. Journal of Machine Learning Research, 3: 993–1022

Griffiths, T. L., and Steyvers, M. 2004. Finding scientific topics. PNAS, 101:5228–5235.