Computing Semantic Text Similarity Using Rich Features

Yang Liu[1], Chengjie Sun[1], Lei Lin[1], Yuming Zhao[2] and Xiaolong Wang[1]
[1] Harbin Institute of Technology, Harbin Heilongjiang 150001, China
[2] Northeast Forestry University, Harbin Heilongjiang 150040, China
{yliu,cjsun,linl,ymzhao,wangxl}@insun.hit.edu.cn

Abstract

Semantic text similarity (STS) is an essential problem in many Natural Language Processing tasks, which has drawn a considerable amount of attention by research community in recent years. In this paper, our work focused on computing semantic similarity between texts of sentence length. We employed a Support Vector Regression model with rich effective features to predict the similarity scores between short English sentence pairs. Our model used WordNet-Based features, Corpus-Based features, Word2Vec-based features, Alignment-Based feature and Literal-Based features to cover various aspects of sentences. And the experiment conducted on SemEval 2015 task 2a shows that our method achieved a Pearson correlation: 80.486% which outperformed the wining system (80.15%) by a small margin, the results indicated a high correlation with human judgments. Specially, among the five test sets which come from different domains used in the estimation, our method got better results than the top team on two of them whose domain-related data is available for training, while comparable results were achieved on the rest three unseen test sets. The experiments results indicated that our solution is more competitive when the domain-specific training data is available and our method still keeps good generalization ability on novel data.

1 Introduction

Semantic text similarity is a fundamental challenge in many Natural Language Processing tasks, such as Machine Reading, Deep Question Answering (Narayanan & Harabagiu, 2004), Automatic Machine Translation Evaluation (Papineni, Roukos et al., 2002), Automatic Text Summarization (Fattah & Ren, 2008) and Query Reformulation (Metzler, Dumais et al., 2007), etc. Previous researches on semantic text similarity have been focused on documents and paragraphs, while comparison objects in many NLP tasks are texts of sentence length, such as Video descriptions, News headlines and beliefs, etc. In this paper, we study semantic similarity between sentences. Given two input text segments, we need to automatically determine a score that indicates their semantic similarity. The difficulties of this task lie on several aspects. First, there were no existing effective measures to represent sentences which could be understood by computers without losing any information. Second, even with good representations, it's very hard to find a metric which can fully compare the equivalence between two sentence representations. Third, similarity itself is a very complex concept, and semantic space is also hard to define and quantize. Given the same pair of sentences, different people may mark different similarity scores; this inconsistency is derived from people's judgments of difference. Although with these difficulties ahead, a lot of methods have been proposed to handle this problem in recent years. And our efforts mainly focused on trying to combine different existing approaches to represent a sentence, and hope to cover as many aspects of sentence as possible on semantic level.

In this paper, we exploited WordNet-Based, Corpus-Based, Word2Vec-based, Alignment-Based and Literal-Based features to measure semantic equivalence between short English sentences. We used a SVR model to combine all of these similarities and predict a final score between 0~5 to denote the magnitude of semantic similarity. And the experiment conducted on SemEval 2015 task 2a shows that our method achieved a Pearson correlation: 80.486% which outperformed the wining system (80.15%) by a small margin. Experimental results demonstrate the effectiveness of our approach.

29th Pacific Asia Conference on Language, Information and Computation pages 44 - 52
Shanghai, China, October 30 - November 1, 2015

Feature Category	Feature Name
WordNet-Based	Path_similarity, Res_similarity, Lin_similarity, Wup_similarity
Corpus-Based	LSA_similarity, IDF_LSA_similarity, Freq_LSA_similarity, Text_LSA_similarity, LDA_similarity, RIC_Difference
Word2Vec-Based	W2V_similarity, IDF_W2V_similarity, S2V_similarity, Text_W2V_similarity
Alignment-Based	Alignment_similarity
Literal-Based	EditDistance_similarity, ShallowSyntatic_similarity, DifferLen_Rate, Digit_similarity, Digit_in_Fea, No_overlap_Fea, Neg_Sentiment_Fea

Table 1 Feature sets of our system configuration

2 Related Work

Previous efforts have focused on computing semantic similarity between documents, concepts or phrases. Recent natural language processing applications show a stronger demand of finding effective methods to measure semantic similarity between texts of variable length, and extensive method have been proposed in these years. Related work could roughly be divided into five major categories: Word co-occurrence methods, Corpus-based and Knowledge-based methods, String similarity methods, Descriptive feature-based methods and Alignment-based methods.

Word co-occurrence methods are usually used in Information Retrieval (IR) systems (Manning, Raghavan et al., 2008). This method is based on the hypothesis that more similar documents would have more words in common. This method has some drawbacks when used in sentence. As sentences are relatively short compared to documents, they would share fewer words in common; moreover, IR systems often exclude function words in their method while these words carry structural information in sentences (Li, McLean et al., 2006), which eventually may lead to unsatisfactory results.

Many methods combined both corpus-based and knowledge-based measures to reach a better result. Two well-known corpus-based methods are Latent Semantic Analysis (LSA) (Dumais, 2004) and Hyperspace analogues to Language (HAL) (Burgess, Livesay et al., 1998). Another effective corpus-based measure is Explicit Semantic Analysis (ESA) (Gabrilovich & Markovitch, 2007). ESA is a method that represents the meaning of texts in a high-dimension space of concepts derived from Wikipedia. As this methodology explicitly uses the knowledge collected and organized by humans, common-sense and domain-specific world knowledge are considered in it which leads to substantial improvements in measure semantic similarity between sentences, and it is also easy to interpret by human. Knowledge-based methods are often based on semantic networks such as WordNet. Some well-known knowledge-based measures include: S&P’s Measure, Wu&Palmer Measure, Leakcock&Chodorow’s Measure, Renik’s Measure, Lin’s Measure and Jiang’s Measure.

As to String-based similarity, Islam et al. proposed a normalized and modified version of the Longest Common Subsequence (LCS) string matching algorithm to measure text similarity (Islam & Inkpen, 2008). Combined with a corpus-based measure, their methods achieved a very competing result.

Descriptive feature-based methods uses predefined features to capture information contained in the sentence. Then feed these features into the classifier, this supervised method achieved best in SemEval 2012 (Šarić, Glavaš et al., 2012).

For alignment-based methods, Sultan et al. (Sultan, Bethard et al., 2014a) proposed an effective solution to align words in monolingual sentences which achieved state-of-the-art performance while relying on almost no supervision and a very small number of external. Based on the output of word aligner, they taking the proportion of their aligned content words as the semantic degree of the two sentences. This simple unsupervised method leads to state-of-art results for sentence level semantic similarity in SemEval 2014 STS task.

Specially, SemEval has hold STS for four years in a row, and many wining methods have been published (Bär, Biemann et al., 2012; Han, Kashyap et al., 2013; Sultan, Bethard et al., 2014b).

3 Feature Generation

The core idea of our method is to use the combination of word similarities to estimate sentence similarity, as lots of effective methods have been proposed to measure word-to-word similarity in recent years. Our features could roughly be divided into five categories: WordNet-Based features, Corpus-Based features, Word2Vec-based features, Alignment-Based feature and Literal-Based features. Generally, Word2Vec-Based methods also can be regarded as Corpus-Based methods, to explore the effectiveness of deep learning based methods, in our paper, we separately classified Word2Vec-Based features into a category. Features used in our model are shown in Table 1.

After combination of these features, we got a very competitive result, which indicated that different features capture different aspects of semantics in sentences. We will look into these features in detail in the following sections.

3.1 WordNet-Based Features

WordNet (Miller, 1995) is a widely used semantic net of English, and it is an effective tool to find synonyms of nouns, verbs, adjectives and adverbs. WordNet is particularly well suited for similarity and relatedness measures, since it organizes nouns and verbs into hierarchies of *is-a* relations (Pedersen, Patwardhan et al., 2004). In this paper, these similarity measures were tried in our experiments. After selection, four of them were kept in our final model: Path_similarity, Res_similarity, Lin_similarity, and Wup_similarity. We provided below a short description for each of these metrics first, and then explain how these measures were used in our evaluation of sentence semantic similarity.

The main idea of the Path_similarity measure (The Shortest Path based Measure) is that the similarity between two concepts can be derived from the length of the path linking the concepts and the position of the concepts in the WordNet taxonomy (Meng, Huang et al., 2013). Formally, the Path_similarity between concepts c_1 and c_2 is defined as following formula:

$$Sim_{path}(c_1, c_2) = 2 * deep\_\max - len(c_1, c_2) \quad (1)$$

where the deep_max is the maximum depth of the taxonomy and $len(c_1, c_2)$ is the length of the shortest path from synset c_1 to synset c_2 in WordNet.

Res_similarity (Resnik's Measure) is a similarity measure based on information content. It assumes that similarity is dependent on the corpus that generates the information content.

$$Sim_{res}(c_1, c_2) = -logp\big(lso(c_1, c_2)\big) = IC(lso(c_1, c_2)) \quad (2)$$

where $lso(c_1, c_2)$ is the lowest common subsume of c_1 and c_2.

Lin_similarity (Lin's Measure) (Lin, 1998) is a similarity measure based on the Resnik measure, which adds a normalization factor consisting of the information content of the two input concepts:

$$Sim_{lin}(c_1, c_2) = \frac{2*IC(LCS)}{IC(c_1)+IC(c_2)} \quad (3)$$

Wup_similarity (Wu & Palmer's Measure) (Wu & Palmer, 1994) measure is based on the depth of two given concepts in the WordNet taxonomy and that of their Least Common Subsumer (LCS), the similarity score of two concepts is defined as following formula(Resnik, 1999):

$$Sim_{wup}(c_1, c_2) = \frac{2*depth(LCS)}{depth(c_1)+depth(c_2)} \quad (4)$$

In our experiment, we used the NLTK[1] toolkit (Bird, 2006) WordNet APIs to calculate WordNet-based similarities. Based on WordNet and Brown corpus (to obtain IC through statistical analysis of Brown corpus), we generated the four WordNet-based features following the same steps proposed in (Liu, Sun et al.).

Issues that required attention is that the results of Res_similarity measure needs to process normalization to make sure the value lies in the interval [0.0, 1.0].

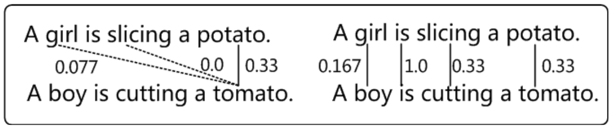
A girl is slicing a potato.
0.077
0.0
0.33
A boy is cutting a tomato.
A girl is slicing a potato.
0.167
1.0
0.33
0.33
A boy is cutting a tomato.

Figure 1 A simple example of word alignment using knowledge-based similarity measures

[1] http://www.nltk.org/

Parameters	**num_topics**	**passes**	**update_every**	**alpha**	**eval_every**
Values	400	10	1	‘auto’	10

Table 2 Parameter setting of LDA model

Figure 1 is an example of how we find the most probable sense in second sentence which has the maximum WordNet similarity with word in first sentence.

3.2 Corpus-Based Features

Latent semantic analysis (LSA) is a technique for comparing texts using a vector-based representation learned from a corpus. A term-document ma trix describes the occurrences of terms in document. The matrix is decomposed by singular value decomposition (SVD). SVD is a factorization of a SVD decompose the term-by-document matrix into three smaller matrixes like follows:

$$X = U\Sigma V^T \qquad (5)$$

real or complex matrix in linear algebra. In LSA, where U and V are column-orthogonal matrixes and Σ is a diagonal matrix containing singular values. Now, columns in U could be preserved as the semantic representations of words. Similarity is then measured by the cosine distance between their corresponding row vectors. To make full use of the semantic information in LSA model, we proposed several methods to compute the sentence similarity based on LSA. These features incude: LSA_similarity,Text_LSA_similarity, IDF_LSA_similarity and Freq_LSA_similarity.

In our experiment, we directly use the LSA model provided by SEMILAR [2] (Ștefănescu, Banjade et al., 2014). The model was decomposed from the whole 2014 Wikipedia articles. One word is represented as a 200-dimension real value vector. We call it “LSA vector” in the rest of the paper. LSA_similarity represent a sentence by summing all LSA vectors of words appeared in the sentence and then averaged it with the length of the sentence. Thus we can get vector representations V_1 and V_2 of the two sentences. The LSA_similarity could be measured with cosine similarity between the two vectors.

The Cosine similarity is defined as follows:

$$Cos\_Dis(V_1, V_2) = \frac{V_1 \cdot V_2}{\|V_1\|\|V_2\|} \qquad (6)$$

Text_LSA_similarity measures similarity between two sentences S_1 and S_2 using the following scoring function (Mihalcea, Corley et al., 2006):

$$\begin{aligned}\mathrm{Sim}(S_1, S_2) &= \frac{1}{2}\Big(\frac{\sum_{w\in\{S_1\}}\big(maxSim(w, S_2) * idf(w)\big)}{\sum_{w\in\{S_1\}} idf(w)} \\ &+ \frac{\sum_{w\in\{S_2\}}\big(maxSim(w,\ S_1) * idf(w)\big)}{\sum_{w\in\{S_2\}} idf(w)}\Big)\end{aligned} \qquad (7)$$

$$\begin{aligned}&maxSim(w, S) \\ &= \mathrm{MAX}\{Cos\_Dis(LSA(w), LSA(w\_s))\}, w\_s \in S\end{aligned} \qquad (8)$$

This similarity score has a value between 0 and 1, with a score 1 indicating identical text segments, and a score 0 indicating no semantic overlap between two texts.

We also generated two weighted features: IDF_LSA_similarity and Freq_LSA_similarity.

$$IDFV(S) = \sum_{w\in S\ \&\ w\notin StW} IDF(w) * \frac{LSA(w)}{norm(LSA(w))} \qquad (9)$$

where *StW* is the predefined stop words list, *LSA(w)* is LSA vector of w and IDF(w) is the inverse document frequency of w.

$$WFSV(S) \quad = \sum_{w\in S\ \&\ w\notin StW} WF(w) * \frac{LSA(w)}{norm(LSA(w))} \qquad (10)$$

where $WF(w)$ is the word frequency of w. In our experiment, the inverse document frequency and word frequency of each word is computed on Wikipedia corpus dumped in December of 2013.

After got the vector representations of sentences, the cosine distance between two vectors is the value of two features.

Latent Dirichlet Allocation (LDA) (Blei, Ng et al., 2003) is a widely used topic model, typically used to find topics distribution in documents; we tried this technology in our model. The LDA model is trained on the training set of SemEval 2015.

[2] http://www.semanticsimilarity.org/

In our experiments, we use the gensim[3] toolkit (Řehůřek & Sojka, 2010) to find the topic distribution of each sentence, and the cosine distance of the vectors could be regarded as the topic similarity of the sentence pair. The parameter setting in the experiment is shown in Table 2.

RIC_Difference measures difference of information content the sentences bearing. In information theory, the information content of a concept can be quantified as negative the log likelihood -logp(c). In our work, the information content of a word w is defined as:

$$ic(w) = \ln \frac{\sum_{w' \in C} freq(w')}{freq(w)} \quad (11)$$

where C is the set of words in the corpus and $freq(w)$ is the frequency of the word w in the corpus. We use the Wikipedia to obtain word frequency. And the Information Content difference between two sentences S_1 and S_2 could be quantified as:

$$RIC(S_1, S_2) = \frac{\left|\sum_{w \in S_1} ic(w) - \sum_{w \in S_2} ic(w)\right|}{MAX(\sum_{w \in S_1} ic(w)\ ,\ \sum_{w \in S_2} ic(w))} \quad (12)$$

3.3 Word2Vec-Based Features

Word2Vec (Mikolov, Chen et al., 2013) is a language modeling technique that maps words from vocabulary to continuous vectors (usually 200 to 500 dimensions). Recently, word embedding has shown its ability to boost the performance in NLP tasks such as syntactic parsing and sentiment analysis. In our work, we employ this technology to represent a word and use several different methods to combine these word vectors to represent a sentence. These generated features include: W2V_similarity, IDF_W2V_similarity, Text_W2V_similarity and S2V_similarity. Similar to generation of LSA-based features, we generate W2V_similarity Text_W2V_similarity is similar to Text_LSA_similarity, computed using the same formula. Only replace the maxSim with the following formula:

$$maxSim(w, S) = \mathrm{MAX}\{Cos\_Dis(W2V(w), W2V(w_s)), w_s \in S\} \quad (13)$$

Furthermore, to improve our performance, we also used the recently proposed-Sent2Vec (also known as paragraph vector) (Le & Mikolov, 2014) to represent a sentence. Paragraph Vector is an unsupervised learning algorithm that learns vector representations for variable length pieces of texts such as sentences and documents. In our experiment, we use the open source code Sentence2vec[4] to train paragraph vectors on Wikipedia. And the cosine distance between two paragraph vectors denote the sentence semantic similarity. This feature is called S2V_similarity. In our development stage, we observed that if more corpora were given to train Sent2Vec, this feature could be more effective.

3.4 Alignment-Based Features

Alignment_similarity is a similarity measure based on monolingual alignment. We first align related words across the two input sentences. And the proportion of aligned content words is regarded as their semantic similarity. In our model, we directly used the monolingual word aligner provided by (Sultan et al., 2014). The aligner is based on the hypothesis that words with similar meanings represent potential pairs for alignment if located in similar contexts. More details about the aligner may refer to the paper, we didn't discuss here. Based on the alignment results, we can compute the similarity using the following formula:

$$sts(S_1, S_2) = \frac{n_c^a(S_1) + n_c^a(S_2)}{n_c(S_1) + n_c(S_2)} \quad (14)$$

where $n_c^a(S_\mathrm{i})$ and $n_c(S_\mathrm{i})$ are the number of content words and the number of aligned content words in S_i. We didn't achieve as good results as in the paper, the reason may because that we didn't consider some stopwords in that filed.

In our experiments, we also used plenty of style-related features, we call it "literal-based" features. Here, we give a short description to each of them.

3.5 Literal-Based Features

EditDistance_similarity is based on the hypothesis: two sentences that look more similar are closer in semantics. So we use the Levenshtein Distance

[3] http://radimrehurek.com/gensim/

[4] https://github.com/klb3713/sentence2vec

over characters to measure the similarity between two sentences.

DifferLen_Rate measures the difference of length of two sentences which can be regarded as evidence of comparing the similarity between sentences.

Shallow Syntactic Similarity considers the similarity in terms of English voices. After Part-Of-Speech tagging to each sentence, we use the Jaccard Distance to compute the syntactic constituent overlap.

Neg_Sentiment_Fea is feature measures shallow sentiment of sentences, we manually chose a list NEG_SENTIMENT = {'no', 'not', 'never', 'little', 'few', 'nobody', 'neither', 'seldom' 'hardly', 'rarely', 'scarcely'} to judge the sentiment, the appearance of word in this list indicating an opposed meaning, if only one word in the list appeared only once in this pair of sentences, we think that this pair of sentences expressing opposite meaning.

Digit_in_Fea is a binary feature which cares about whether there is digit numbers appeared in only sentence in the pair. To our intuitive, if only one sentence obtain numbers in it and another contains only text, then human annotators tend to give a lower score to this pair. So, if Digit_in_Fea of a pair of sentences was set to '1', this can be interpreted to give classifier a signal to give a lower similarity score.

Digit_similarity could be regarded as complement to feature Digit_in_Fea. We implemented a simple algorithm to extract numbers from text.and then compares the difference of numbers appeared in two sentences.

No_overlap_Fea measures whether two sentences are totally different in terms of words appeared in the sentences. Although this hypothesis is not always true, but we observed that this assumption is correct under most cases and this feature still contributes to our overall performance.

4 Experiments

We conduct our experiments on the SemEval 2015 STS English subtask. Given two sentences of English text, S_1andS_2, we need to compute how similar S_1 and S_2 are, returning a similarity score between 0.0 (no relation) to 5.0 (semantic equivalence), indicating the semantic similarity between two sentences.

4.1 Datasets

In SemEval 2015 2a, the trial dataset comprises the 2012, 2013 and 2014 datasets, which can be used to develop and train models. The details of the dataset refer to (Agirre & Banea, 2015).

4.2 Evaluation Metrics

The official estimation is based on the average of Pearson correlation. This metric is determined as:

$$\rho_{X,Y} = \frac{\sum XY - \frac{\sum X \sum Y}{N}}{\sqrt{\left(\sum X^2 - \frac{(\sum X)^2}{N}\right)\left(\sum Y^2 - \frac{(\sum Y)^2}{N}\right)}} \qquad (15)$$

where X is the golden-standard scores vector, and Y is the output of SVRs.

4.3 Results and Discussion

To achieve a better result, we trained three Support Vector Regression models to predict similarity scores on different test sets, we used all the datasets (except SMT of 2013, we got worse performance after added it, so we exclude SMT in our final model) before 2015 as training set for the first three test sets which are unseen data for the classifier. This classifier was denoted as Clf-1. For headlines and images, all headlines / images data sets appeared before were used as training sets. The trained classifier was denoted as Clf-2 and Clf-3 respectively.
In terms of implementation, we used Scikit-learn[5] toolkit(Pedregosa, Varoquaux et al., 2011) to do the classification and the parameter settings for three SVR models are shown in the following table, we chose these parameters by experiences, Clf-2 and Clf-3 used the same setting, and a better result may be achieved through fine tuning:

Classifier	**kernel**	**Gamma**	**C**	**epsilon**
Clf-1	'rbf'	0.1	1.8	0.1
Clf-2	'rbf'	0.16	100	0.1
Clf-3	'rbf'	0.16	100	0.1

Table 3 Parameter settings of our three classifiers

After the prediction of the similarity scores of sentences, we conducted a post-processing step to boost and correct results, we truncate at the extre-

[5] http://scikit-learn.org/stable/

Data Set	Ans-for	Ans-stu	Belief	Hdlines	Images	Mean
All features	0.7381	0.7644	0.7377	**0.8521**	**0.8650**	**0.8049**
w/o WordNet-based	0.7356	0.7516	0.7260	0.8450	0.8560	0.7959
w/o Corpus-based	0.7150	**0.7850**	0.7389	0.8387	0.8620	0.8032
w/o Word2Vec-based	**0.7460**	0.7498	0.7366	0.8510	0.8536	0.7989
w/o Alignment-based	0.7278	0.7551	0.7168	0.8355	0.8614	0.7926
w/o Literal-based	0.7175	0.7320	**0.7501**	0.8240	0.8618	0.7879

Table 4 Performance of different feature combinations (exclude one kind each time)

Feature Set	Ans-for	Ans-stu	Belief	Hdlines	Images	Mean
All features	**0.7381**	**0.7644**	**0.7377**	**0.8521**	**0.8650**	**0.8049**
WordNet-based	0.6813	0.7252	0.7289	0.7509	0.8352	0.7541
Corpus-based	0.6182	0.6245	0.6652	0.7257	0.8254	0.7043
Word2Vec-based	0.6065	0.7305	0.6904	0.7365	0.8369	0.7381
Alignment-based	0.6675	0.7789	0.6699	0.7891	0.7872	0.7560
Literal-based	0.6666	0.5725	0.5235	0.5493	0.3326	0.5123

Table 5 Results of comparing the importance of different kinds of features on SemEval 2015

mes to keep the score in [0.0, 5.0], and an additional step similar to the details in (Bär et al., 2012). The post-processing step contributed a 0.1% improvement in our overall performance.

Test Set	Winning team	Our System
answers-forums	0.7390	0.7381
answers-students	0.7725	0.7644
belief	0.7491	0.7377
headlines	0.8250	**0.8521**
images	0.8644	**0.8650**
Weight Mean	0.8015	**0.8049**

Table 6 Performances of our model and winning system on SemEval 2015 STS test sets

Table 4 reported the results of our method on SemEval 2015 Task 2a, from which we can know that our method outperformed the winning system by a big margin on the headlines, but only slightly better on the images. The reason may because that in the winning system, images was already achieved a very high accuracy, but due to the incomplete use of the semantic information, didn't perform as well as in headlines. As our method used more sufficient features, our approach achieved both state-of-the-art results on headlines and images. The winning system mainly based on word alignment, which guaranteed very good generalization ability, but much of the semantic information contained in the training set was not used, while these information can also contribute to the system performance, especially for domain-specific test set, in other word, our method can be used to verify this idea. For the first three datasets, our method may achieve much better performance if more domain-specific data was given for learning. Overall, our system performed slightly better than the wining system in terms of average Pearson correlation.

To compare the importance of each kind of feature, we separately exclude one kind of them in our model and compare new model's performance.

The results are shown in Table 5. And the performance of using only one kind of feature showed in Table 6.

The experiment results demonstrated the effectiveness of our generated features, except Liter-based features, each kind of other features alone could lead to a relatively good performance. Although Literal-based features didn't perform well on its own, exclude it from our model leads to the biggest decrease in Mean correlation, which indicated it is an important complement to other features. We also observed that corpus-based features seem less effective compared to other features as they didn't perform as well as other semantic related features and the absence of it has little impact on the overall performance. The different combinations of them boosted the results to achieve a higher correlation. Also, SVR model played an important role in our approach, it provide a good out-of-sample generalization as the loss function typically leads to a sparse representation of the decision rule which makes our model more robust on novel data. And we think that the appropriate choice of kernel function in SVR may also help a lot in the model.

5 Conclusion

In this paper, we presented our approach to evaluate semantic similarity between short English sentences. We employed a Support Vector Regression model combined with WordNet-Based features, Corpus-Based features, Word2Vec-based features, Binary Features and some other features to predict the semantic similarity score between sentence pairs. Our experiment results showed a high correlation with human annotations which outperformed the top system in SemEval 2015 task 2a. We also observed that our method performed much better compared to winning system on two test sets whose domain-specific data is available for training, results also indicated that our solution still maintains good generation ability on novel datasets which means this technique could be well generalized to other data domains. While the context of the sentence is unavailable and the information about the tone of sentence was eliminated by us (most modal particles and punctuations appeared in sentences were treated as stop words in our process), our model could not distinguish the tone of sentence, for example we may give a high similarity score to a sentence pair consists of a declarative and an imperative if they shared many words. This situation was not considered in feature generation stage, but will be researched latter. Our future work will include the refinements of training effective representations for words and sentences on corpus (LSA, Word2Vec and Sent2Vec), the expansion of stop word list through adding proper selected domain-specific stop words and the re-implementation of a well-designed feature selection process to simplify our model. We hope that these measures could be helpful for improvement, make our model more robust and improve our method’s generalization ability as well.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful suggestions and comments. This work is supported by the National Natural Science Foundation of China (61300114 & 61572151) and Natural Science Foundation of Heilongjiang Province (F201132).

References

Agirre Eneko, & Banea Carmen. (2015). *SemEval-2015 task 2: Semantic textual similarity, English, S-panish and pilot on interpretability.* Paper presented at the Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), June.

Bär Daniel, Biemann Chris, Gurevych Iryna, & Zesch Torsten. (2012). *Ukp: Computing semantic textual similarity by combining multiple content similarity measures.* Paper presented at the 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.

Bird Steven. (2006). *NLTK: the natural language toolkit.* Paper presented at the Proceedings of the COLING/ACL on Interactive presentation sessions.

Blei David M, Ng Andrew Y, & Jordan Michael I. (2003). Latent dirichlet allocation. *the Journal of machine Learning research, 3,* 993-1022.

Burgess Curt, Livesay Kay, & Lund Kevin. (1998). Explorations in context space: Words, sentences, discourse. *Discourse Processes, 25*(2-3), 211-257.

Dumais Susan T. (2004). Latent semantic analysis. *Annual review of information science and technology, 38*(1), 188-230.

Fattah Mohamed Abdel, & Ren Fuji. (2008). Automatic text summarization. *World Academy of Science, Engineering and Technology, 37*, 2008.

Gabrilovich Evgeniy, & Markovitch Shaul. (2007). *Computing Semantic Relatedness Using Wikipedia-based Explicit Semantic Analysis.* Paper presented at the IJCAI.

Han Lushan, Kashyap Abhay, Finin Tim, Mayfield James, & Weese Jonathan. (2013). UMBC EBIQUITY-CORE: Semantic textual similarity systems. *Atlanta, Georgia, USA, 44*.

Islam Aminul, & Inkpen Diana. (2008). Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD), 2*(2), 10.

Le Quoc V, & Mikolov Tomas. (2014). *Distributed representations of sentences and documents.* Paper presented at the Proceedings of NIPS 2013

Li Yuhua, McLean David, Bandar Zuhair, O'shea James D, & Crockett Keeley. (2006). Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on, 18*(8), 1138-1150.

Lin Dekang. (1998). *An information-theoretic definition of similarity*. Paper presented at the ICML.

Liu Yang, Sun Chengjie, Lin Lei, & Wang Xiaolong. yiGou: A Semantic Text Similarity Computing System Based on SVM. Paper presented at the Proceedings of the 9th International Workshop on Semantic Evaluation, pages 80-84, Denver, Colorado, USA.

Manning Christopher D, Raghavan Prabhakar, & Schütze Hinrich. (2008). *Introduction to information retrieval* (Vol. 1): Cambridge university press Cambridge.

Meng Lingling, Huang Runqing, & Gu Junzhong. (2013). A review of semantic similarity measures in wordnet. *International Journal of Hybrid Information Technology, 6*(1), 1-12.

Metzler Donald, Dumais Susan, & Meek Christopher. (2007). *Similarity measures for short segments of text*: Springer.

Mihalcea Rada, Corley Courtney, & Strapparava Carlo. (2006). *Corpus-based and knowledge-based measures of text semantic similarity*. Paper presented at the AAAI.

Mikolov Tomas, Chen Kai, Corrado Greg, & Dean Jeffrey. (2013). *Efficient estimation of word representations in vector space*. Paper presented at the ICLR, 2013.

Miller George A. (1995). WordNet: a lexical database for English. *Communications of the ACM, 38*(11), 39-41.

Narayanan Srini, & Harabagiu Sanda. (2004). *Answering questions using advanced semantics and probabilistic inference*. Paper presented at the Proceedings of the Workshop on Pragmatics of Question Answering, HLT-NAACL, Boston, USA.

Papineni Kishore, Roukos Salim, Ward Todd, & Zhu Wei-Jing. (2002). *BLEU: a method for automatic evaluation of machine translation*. Paper presented at the Proceedings of the 40th annual meeting on association for computational linguistics.

Pedersen Ted, Patwardhan Siddharth, & Michelizzi Jason. (2004). *WordNet:: Similarity: measuring the relatedness of concepts*. Paper presented at the Demonstration papers at hlt-naacl 2004.

Pedregosa Fabian, Varoquaux Gaël, Gramfort Alexandre, Michel Vincent, Thirion Bertrand, Grisel Olivier, . . . Dubourg Vincent. (2011). Scikit-learn: Machine learning in Python. *the Journal of machine Learning research, 12*, 2825-2830.

Řehůřek Radim, & Sojka Petr. (2010). Software framework for topic modelling with large corpora. Paper presented at the Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, pages 45–50, Valletta, Malta, May 2010. ELRA

Resnik Philip. (1999). Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.(JAIR), 11*, 95-130.

Šarić Frane, Glavaš Goran, Karan Mladen, Šnajder Jan, & Bašić Bojana Dalbelo. (2012). *Takelab: Systems for measuring semantic text similarity*. Paper presented at the 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.

Ştefănescu Dan, Banjade Rajendra, & Rus Vasile. (2014). Latent semantic analysis models on wikipedia and tasa. The 9th Language Resources and Evaluation Conference (LREC 2014).

Sultan Md Arafat, Bethard Steven, & Sumner Tamara. (2014a). Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics, 2*, 219-230.

Sultan Md Arafat, Bethard Steven, & Sumner Tamara. (2014b). DLS@CU: Sentence Similarity from Word Alignment. Paper presented at the Proceedings of the 8th International Workshop on Semantic Evaluation, pages 241-246, Dublin, Ireland

Wu Zhibiao, & Palmer Martha. (1994). *Verbs semantics and lexical selection*. Paper presented at the Proceedings of the 32nd annual meeting on Association for Computational Linguistics.