ECNU: Using Traditional Similarity Measurements and Word Embedding for Semantic Textual Similarity Estimation

Jiang Zhao, Man Lan*, Jun Feng Tian
Shanghai Key Laboratory of Multidimensional Information Processing
Department of Computer Science and Technology,
East China Normal University, Shanghai 200241, P. R. China
51121201042,10112130275@ecnu.cn; mlan@cs.ecnu.edu.cn*

Abstract

This paper reports our submissions to semantic textual similarity task, i.e., task 2 in Semantic Evaluation 2015. We built our systems using various traditional features, such as string-based, corpus-based and syntactic similarity metrics, as well as novel similarity measures based on distributed word representations, which were trained using deep learning paradigms. Since the training and test datasets consist of instances collected from various domains, three different strategies of the usage of training datasets were explored: (1) use all available training datasets and build a unified supervised model for all test datasets; (2) select the most similar training dataset and separately construct a individual model for each test set; (3) adopt multi-task learning framework to make full use of available training sets. Results on the test datasets show that using all datasets as training set achieves the best averaged performance and our best system ranks 15 out of 73.

1 Introduction

Estimating the degree of semantic similarity between two sentences is the building block of many natural language processing (NLP) applications, such as textual entailment (Zhao et al., 2014a), text summarization (Lloret et al., 2008), question answering (Celikyilmaz et al., 2010), etc. Therefore, semantic textual similarity (STS) has been received an increasing amount of attention in recent years, e.g., the Semantic Textual Similarity competitions in Semantic Evaluation Exercises have been held from 2012 to 2014. This year the participants in the STS task in *SemEval* 2015 (Agirre et al., 2015) are required to rate the similar degree of a pair of sentences by a value from 0 (no relation) to 5 (semantic equivalence) with an optional confidence score.

To identify semantic textual similarity of text pairs, most existing works adopt at least one of the following feature types: (1) string based similarity (Bär et al., 2012; Jimenez et al., 2012) which employs common functions to calculate similarities over string sequences extracted from original strings, e.g., lemma, stem, or *n*-gram sequences; (2) corpus based similarity (Šarić et al., 2012; Han et al., 2013) where distributional models such as *Latent Semantic Analysis* (LSA) (Landauer and Dumais, 1997), are used to derive the distributional vectors of words from a large corpus according to their occurrence patterns, afterwards, similarities of sentence pairs are calculated using these vectors; (3) knowledge based method (Shareghi and Bergler, 2013; Mihalcea et al., 2006) which estimates the similarities with the aid of external resources, such as WordNet[1]. Among them, lots of researchers (Sultan et al., 2014; Han et al., 2013) leverage different word alignment strategies to bring word-level similarity to sentence-level similarity.

In this work, we first borrow aforementioned effective types of similarity measurements including string-based, corpus-based, syntactic features and so on, to capture the semantic similarity between two sentences. Beside, we also present a novel feature type based on *word embeddings* that are induced using neural language models over a large raw cor-

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

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 117–122,
Denver, Colorado, June 4-5, 2015. ©2015 Association for Computational Linguistics

pus (Mikolov et al., 2013b). Then these features are served as input of a regression model. Notice that, the organizers provide us seventeen training datasets and five test datasets, which are drawn from different but related domains. Accordingly, we build three different systems in terms of the usage of training datasets: (1) exploit all the training datasets and train a single model for all test datasets; (2) choose one domain-dependent training dataset for each test dataset using *cosine* distance selection criterion and train models individually for each test dataset; (3) to overcome overuse or underuse of training datasets, we adopt multi-task learning (MTL) framework to make full use of available training datasets, that is, for each test set the main task is built upon designated training datasets and the rest training datasets are used in the auxiliary tasks.

The rest of this paper is organized as follows. Section 2 describes various similarity measurements used in our systems. System setups and experimental results on training and test datasets are presented in Section 3. Finally, conclusions and future work are given in Section 4.

2 Semantic Similarity Measurements

Following our previous work (Zhao et al., 2014b), we adopted the traditional widely-used features (i.e., string, corpus, syntactic features) for semantic similarity measurements. In this work, we also proposed several novel features using word embeddings.

2.1 Preprocessing

Several text preprocessing operations were performed before we extracted features. We first converted the contractions to their formal writings, for example, *doesn't* is rewritten as *does not*. Then the WordNet-based Lemmatizer implemented in Natural Language Toolkit[2] was used to lemmatize all words to their nearest base forms in WordNet, for example, *was* is lemmatized to *be*. After that, We replaced a word from one sentence with another word from the other sentence if these two words share the same meaning, where WordNet was used to look up synonyms. No word sense disambiguation was performed and all synsets in WordNet for a particular lemma were considered.

2.2 String Based Features

We firstly recorded length information of given sentences pairs using the following eight measure functions: $|A|, |B|, |A-B|, |B-A|, |A \cup B|, |A \cap B|, \frac{(|A|-|B|)}{|B|}, \frac{(|B|-|A|)}{|A|}$, where $|A|$ stands for the number of non-repeated words in sentence A , $|A-B|$ means the number of unmatched words found in A but not in B , $|A \cup B|$ stands for the set size of non-repeated words found in either A or B and $|A \cap B|$ means the set size of shared words found in both A and B .

Motivated by the hypothesis that two texts are considered to be semantic similar if they share more common strings, we adopted the following five types of measurements: (1) longest common sequence similarity on the original and lemmatized sentences; (2) `Jaccard`, `Dice`, `Overlap` coefficient on original word sequences; (3) `Jaccard` similarity using *n*-grams, where *n*-grams were obtained at three different levels, i.e., the original word level (*n*=1,2,3), the lemmatized word level (*n*=1,2,3) and the character level (n=2,3,4); (4) weighted word overlap feature (Šarić et al., 2012) that takes the importance of words into consideration, where Web 1T 5-gram Corpus[3] was used to estimate the importance of words; (5) sentences were represented as vectors in *tf*idf* schema based on their lemmatized forms and then these vectors were used to calculate `cosine`, `Manhattan`, `Euclidean` distance and `Pearson`, `Spearmanr`, `Kendalltau` correlation coefficients.

Totally, we got thirty-one string based features.

2.3 Corpus Based Features

The distributional meanings of words own good semantic properties and *Latent Semantic Analysis* (LSA) (Landauer and Dumais, 1997) is widely used to estimate the distributional vectors of words. Hence, we adopted two distributional word sets released by TakeLab (Šarić et al., 2012), where LSA was performed on the New York Times Annotated Corpus (NYT)[4] and Wikipedia. Then two strategies were used to convert the distributional meanings of words to sentence level: (i) simply summing up the distributional vector of each word w in the sentence, (ii)

[2] http://nltk.org/

[3] https://catalog.ldc.upenn.edu/LDC2006T13

[4] https://catalog.ldc.upenn.edu/LDC2008T19

using the information content (Šarić et al., 2012) to weigh the LSA vector of each word w and then summing them up. After that we used *cosine* similarity to measure the similarity of two sentences based on these vectors. Besides, we used the Co-occurrence Retrieval Model (CRM) (Weeds, 2003) as another type of corpus based feature. The CRM was calculated based on a notion of substitutability, that is, the more appropriate it was to substitute word w_1 in place of word w_2 in a suitable natural language task, the more semantically similar they were.

At last, we obtained six corpus based features.

2.4 Syntactic Features

Besides semantic similarity, we also estimated the similarities of sentence pairs at syntactic level. Stanford CoreNLP toolkit (Manning et al., 2014) was used to obtain the POS tag sequences for each sentence. Afterwards, we performed eight measure functions described above in Section 2.2 over these sequences, resulting in eight syntactic features.

2.5 Word Embedding Features

Recently, deep learning has archived a great success in the fields of computer vision, automatic speech recognition and natural language processing. One result of its application in NLP, i.e., word embeddings, has been successfully explored in named entity recognition, chunking (Turian et al., 2010) and semantic word similarities(Mikolov et al., 2013a), etc. The distributed representations of words (i.e., word embeddings) learned using neural networks over a large raw corpus have been shown that they performed significantly better than LSA for preserving linear regularities among words (Mikolov et al., 2013a). Due to its superior performance, we adopted word embeddings to estimate the similarities of sentence pairs. In our experiments, we used two different word embeddings: *word2vec* (Mikolov et al., 2013b) and *Collobert and Weston* embeddings (Turian et al., 2010). The word embeddings from Word2vec are distributed within the word2vec toolkit[5] and they are 300-dimensional vectors learned from Google News Corpus which consists of over a 100 billion words. The Collobert and Weston embeddings are learned over a part of RCV1 corpus which consists of 63 millions words, resulting in 100-dimensional continuous vectors. To obtain the sentence representations from word representations, we used *idf* to weigh the embedding vectors of words and simply summed them up. Although the word embedding is obtained from large corpus in consideration of its context, using this bag of words (BOW) representation of sentences, the current word sequence in sentence is neglected. After that, we used `cosine`, `Manhattan`, `Euclidean` functions and `Pearson`, `Spearmanr`, `Kendalltau` correlation coefficients to calculate the similarities based on these synthetic sentence representations.

2.6 Other Features

Besides the shallow semantic similarities between words and strings, we also calculated the similarities of named entities in two sentences using longest common sequence function. Seven types of named entities, i.e., *location*, *organization*, *date*, *money*, *person*, *time* and *percent*, recognized by Stanford CoreNLP toolkit (Manning et al., 2014) were considered. We designed a binary feature to indicate whether two sentences in a given pair have the same polarity (i.e., *affirmative* or *negative*) by looking up a manually-collected negation list with 29 negation words (e.g., *scarcely*, *no*, *little*). Finally, we obtained in eight features.

3 Experiments and Results

3.1 Datasets

Participants built their systems on seventeen datasets in development period and evaluated their systems on five test datasets in test period. Each dataset consists of a number of sentence pairs and each pair has a human-assigned similarity score in the range [0, 5] which increases with similarity. The datasets were collected from different but related domains. Due to limitation of page length, we only provide a brief description of test sets in Table 1. Refer (Agirre et al., 2014) for more details. As we can see from this table, datasets from different domains have distinct average lengths of sentence `A` and `B`.

[5] https://code.google.com/p/word2vec

Dataset	# of pairs	average length
answers-forums	2000	(17.56,17.37)
answers-students	1500	(10.49,11.17)
belief	2000	(15.16,14.56)
headlines	1500	(7.86,7.91)
images	1500	(10.59,10.58)

Table 1: The statistics of test datasets for STS task in *SemEval* 2015.

3.2 Experimental Setups

We built three different systems according to the usage of training datasets as follows.

allData: We used all the training datasets and built a single global regression model regardless of domain information of different test datasets.

DesignatedData: For each test dataset, we calculated the cosine distance with every candidate training dataset. Then the training dataset with the lowest distance score was chose as the training dataset to fit a regression model for specific test dataset.

$$\texttt{Dist}(X_{tst}, X_c) = 1 - \sum_{x_i \in X_{tst}} \sum_{x_j \in X_c} \frac{\text{cosine}(x_i, x_j)}{|X_{tst}||X_c|}$$

MTL: On one hand, taking all the training datasets into consideration may hurt the performance since training and test datasets are from different domains. On the other hand, using the most related datasets leads to insufficient usage of available datasets. Therefore, we considered to adopt multi-task learning framework to take full advantage of available training sets. Under multi-task learning framework, a main task learns together with other related auxiliary tasks at the same time, using a shared representation. This often leads to a better model for the main task, because it allows the learner to use the commonality among the tasks. Hence, for each test dataset we selected the datasets whose *cosine* distances are less than 0.1 (at least one training set) as training set to construct the main task, and then used the remaining training sets to construct auxiliary tasks. In this work, we adopted the robust multitask feature learning (rMTFL) (Gong et al., 2012), which assumes that the model W can be decomposed into two components: a shared feature structure P that captures task relatedness and a group-sparse structure Q that detects outlier tasks. Specifically, it solves following formulation:

$$\min_{W} \sum_{i=1}^{t} \|W_i^F X_i - Y_i\|_F^2 + \rho_1 \|P\|_{2,1} + \rho_2 \|Q^T\|_{2,1}$$

$$\text{subject to} : W = P + Q$$

where X_i denotes the input matrix of the i-th task, Y_i denotes its corresponding label, W_i is the model for task i, the regularization parameter ρ_1 controls the joint feature learning, and the regularization parameter ρ_2 controls the columnwise group sparsity on Q that detects outliers.

In our preliminary experiments, several regression algorithms were examined, including Support Vector Regression (SVR, *linear*), Random Forest (RF) and Gradient Boosting (GB) implemented in the scikit-learn toolkit (Pedregosa et al., 2011). The system performance is evaluated using Pearson correlation (r).

3.3 Results on Training Data

To configure the parameters in the three systems, i.e., the trade-off parameter c in SVR, the number of trees n in RF, the number of boosting stages n in GB in **allData** and **DesignatedData**, $\rho_{1,2}$ in **MTL**, we conducted a series of experiments on STS 2014 datasets (eleven datasets for training, six datasets for development). Table 2 shows the Pearson performance of our systems on development datasets. We explored a large scale of parameter values and only the best result for each algorithm was listed due to the limitation of page length. The numbers in the brackets in algorithms column indicate the parameter values and those in bold font represent the best performance for each dataset and system. From the table we find that (1) GB and SVR obtain the best averaged results in system **allData** and **DesignatedData** respectively; (2) although **DesignatedData** uses only one most-closely dataset for training for each test set, it achieves comparable or even better performance on some datasets when compared with **allData**; (3) our multi-task learning framework can indeed boost the performance.

3.4 Results on Test Data

According to the results on training datasets, we configured three submitted runs as following:

Algorithms	deft-forum	deft-news	headlines	images	OnWN	tweet-news	Mean
SVR (0.01)	0.458	**0.761**	**0.728**	**0.813**	0.836	0.727	0.721
RF (65)	0.491	0.751	0.718	0.789	**0.873**	**0.741**	0.727
GB (50)	**0.499**	0.760	0.725	0.805	0.863	0.739	**0.732**
SVR (0.1)	**0.549**	**0.725**	**0.765**	**0.790**	0.810	0.740	**0.730**
RF (75)	0.513	0.709	0.741	0.768	**0.814**	**0.767**	0.719
GB (50)	0.504	0.694	0.738	0.790	0.809	0.751	0.714
MTL (0.1, 0.1)	0.556	0.772	0.738	0.808	0.819	0.745	0.740

Table 2: Pearson of **allData,DesignatedData** using different algorithms and **MTL** on STS 2014 datasets.

RUN	answers-forums	answers-students	belief	headlines	images	Mean	Rank
ECNU-1stSVMALL	**0.715**	0.712	**0.728**	0.798	0.847	**0.755**	15
ECNU-2ndSVMONE	0.687	0.733	0.698	**0.820**	0.836	0.747	19
ECNU-3rdMTL	0.692	**0.752**	0.695	0.805	**0.858**	0.752	18
DLSCU-S1	0.739	0.773	0.749	0.825	0.864	0.785	1
ExBThemis-themisexp	0.695	0.778	0.748	0.825	0.853	0.773	2

Table 3: Results of our three runs on STS 2015 test datasets, as well as top rank runs.

ECNU-1stSVMALL which builds a global model on all datasets using SVR with parameter *c*=0.1; **ECNU-2ndSVMONE** which fits individual model for each test set on a designated training set using GB with parameter *n*=50; **ECNU-3rdMTL** which employs robust multi-task feature learning with parameter $\rho_1 = \rho_2 = 0.1$.

Table 3 summarizes the results of our submitted runs on test datasets officially released by the organizers, as well as the top rank runs. In terms of mean Pearson measurement, system **ECNU-1stSVMALL** performs the best, which is comparable to **ECNU-3rdMTL**. However, the **ECNU-2ndSVMONE** performs the worst. This is inconsistent with the results on training datasets wherein **ECNU-3rdMTL** yields the best performance. On test dataset, we find that **ECNU-3rdMTL** has much worse performances than **ECNU-1stSVMALL** on answers-forums and belief while it achieves much better results on answers-students, headlines and images datasets. The possible reason may be that the training dataset selected from the candidate datasets in main task are ill-suited for answers-forums and belief test datasets, which is also verified by the results of system **ECNU-2ndSVMONE**. It is noteworthy that on answers-students and headlines **ECNU-2ndSVMONE** achieves much better results than **ECNU-1stSVMALL** although the former system only uses much less training instances (750,750 vs. 10592). In addition, the difference between top system **DLSCU-S1** and our systems is about 3%, which means our systems are promising.

4 Conclusion

We used traditional NLP features including string-based, corpus-based and syntacitc features, for textual semantic similarity estimation, as well as novel word embedding features. We also presented three different systems to compare the strategies of different usage of training data, i.e., single supervised learning with all training datasets and individual training dataset for each test dataset, and multitask learning framework. Our best system achieves 15th place out of 73 systems on test datasets. Noticeably each system achieves the best performance on different test datasets, which indicates the usage of training datasets is important, we will explore more sophisticated way to utilize these training datasets in future work.

Acknowledgements

This research is supported by grants from Science and Technology Commission of Shanghai Municipality under research grant no. (14DZ2260800 and 15ZR1410700) and Shanghai Collaborative Innovation Center of Trustworthy Software for Internet of

Things (ZF1213).

References

Eneko Agirre, Carmen Banea, and et al. 2014. SemEval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91.

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

Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. 2012. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics*, pages 435–440.

Asli Celikyilmaz, Dilek Hakkani-Tur, and Gokhan Tur. 2010. LDA based similarity modeling for question answering. In *Proceedings of the NAACL HLT 2010 Workshop on Semantic Search*, pages 1–9.

Pinghua Gong, Jieping Ye, and Changshui Zhang. 2012. Robust multi-task feature learning. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 895–903.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic textual similarity systems. In *Second *SEM*, pages 44–52.

Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. 2012. Soft cardinality: A parameterized similarity function for text comparison. In **SEM 2012 and (SemEval 2012)*, pages 449–453.

Thomas K Landauer and Susan T Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*.

Elena Lloret, Oscar Ferrández, Rafael Munoz, and Manuel Palomar. 2008. A text summarization approach under the influence of textual entailment. In *Proceedings of NLPCS 2008*, pages 22–31.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd ACL: System Demonstrations*, pages 55–60.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, et al. 2011. Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, 12:2825–2830.

Ehsan Shareghi and Sabine Bergler. 2013. CLaC-CORE: Exhaustive feature combination for measuring textual similarity. In *Second Joint Conference on Lexical and Computational Semantics (*SEM)*.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014. DLS@CU: sentence similarity from word alignment. In *SemEval 2014*, pages 241–246.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 384–394.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. TakeLab: Systems for measuring semantic text similarity. In **SEM 2012 and (SemEval 2012)*, pages 441–448.

Julie Elizabeth Weeds. 2003. *Measures and applications of lexical distributional similarity*. Ph.D. thesis, University of Sussex.

Jiang Zhao, Man Lan, Zheng-Yu Niu, and Dong-Hong Ji. 2014a. Recognizing cross-lingual textual entailment with co-training using similarity and difference views. In *IJCNN 2014, Beijing, China, 2014*, pages 3705–3712.

Jiang Zhao, Tiantian Zhu, and Man Lan. 2014b. Ecnu: One stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. In *Proceedings of the SemEval 2014*, pages 271–277, Dublin, Ireland, August.