UWB at SemEval-2016 Task 1: Semantic Textual Similarity using Lexical, Syntactic, and Semantic Information

Tomáš Brychcín
NTIS – New Technologies
for the Information Society,
Faculty of Applied Sciences,
University of West Bohemia,
Technická 8, 306 14 Plzeň
Czech Republic
brychcin@kiv.zcu.cz

Lukáš Svoboda
Department of Computer
Science and Engineering,
Faculty of Applied Sciences,
University of West Bohemia,
Technická 8, 306 14 Plzeň
Czech Republic
svobikl@kiv.zcu.cz

Abstract

We present our UWB system for Semantic Textual Similarity (STS) task at SemEval 2016. Given two sentences, the system estimates the degree of their semantic similarity.

We use state-of-the-art algorithms for the meaning representation and combine them with the best performing approaches to STS from previous years. These methods benefit from various sources of information, such as lexical, syntactic, and semantic.

In the monolingual task, our system achieve mean Pearson correlation 75.7% compared with human annotators. In the cross-lingual task, our system has correlation 86.3% and is ranked first among 26 systems.

1 Introduction

Semantic Textual Similarity (STS) is one of the core disciplines in Natural Language Processing (NLP). Assume we have two textual fragments (word phrases, sentences, paragraphs, or full documents), the goal is to estimate the degree of their semantic similarity.

STS systems are usually compared with the manually annotated data. In the case of SemEval the data consist of pairs of sentences with a score between 0 and 5 (higher number means higher semantic similarity). For example, English pair

```
Two dogs play in the grass.
Two dogs playing in the snow.
```

has a score 2.8, i.e. the sentences are not equivalent, but share some information.

This year, SemEval's STS is extended with the Spanish-English cross-lingual subtask, where e.g. the pair

```
Tuve el mismo problema que tú.
I had the same problem.
```

has a score 4.8, which means nearly equivalent.

Each year STS belongs to one of the most popular tasks at SemEval competition. The best STS system at SemEval 2012 (Bär et al., 2012) used lexical similarity and Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007). In SemEval 2013, the best model (Han et al., 2013) used semantic models such as Latent Semantic Analysis (LSA) (Deerwester et al., 1990), external information sources (WordNet) and n-gram matching techniques. For SemEval 2014 and 2015 the best system comes from (Sultan et al., 2014a; Sultan et al., 2014b; Sultan et al., 2015). They introduced new algorithm, which align the words between two sentences. They showed that this approach can be efficiently used also for STS. Overview of systems participating in previous SemEval competitions can be found in (Agirre et al., 2012; Agirre et al., 2013; Agirre et al., 2014; Agirre et al., 2015).

The best performing systems from previous years are based on various architectures benefiting from lexical, syntactic, and semantic information. In this work we try to use the best techniques presented during last years, enhance them, and combine into a single model.

2 Semantic Textual Similarity

This section describes various techniques for estimating the text similarity.

Proceedings of SemEval-2016, pages 588–594,
San Diego, California, June 16-17, 2016.

2.1 Lexical and Syntactic Similarity

This section presents the techniques exploiting lexical and syntactic information in the text. Some of them have been successfully used by (Bär et al., 2012). Many of the following techniques benefit from the weighing of words in a sentence using *Term Frequency - Inverse Document Frequency* (TF-IDF) (Manning and Schütze, 1999).

- **Lemma n-gram overlaps:** We compare word n-grams in both sentences using *Jaccard Similarity Coefficient* (JSC) (Manning and Schütze, 1999). We do it separately for different orders $n \in \{1, 2, 3, 4\}$. *Containment Coefficient* (Broder, 1997) is used for orders $n \in \{1, 2\}$. We extend original metrics by weighing of n-grams. We define this weight as a sum of *IDF* values of words in n-gram. N-gram match is not counted as 1 but as the weight of this n-gram. According to our experiments, this weighing significantly improves performance.

  We also use information about the length of *Longest Common Subsequence* compared to the length of the sentences.

- **POS n-gram overlaps:** In similar way as for lemmas, we calculate *Jaccard Similarity Coefficient* and *Containment Coefficient* for n-grams of part-of-speech (POS) tags. Again, we use n-gram weighing and $n \in \{1, 2, 3, 4\}$. These features exploit syntactic similarity of the sentences.

- **Character n-gram overlaps:** Similarly to lemma or POS n-grams, we use *Jaccard Similarity Coefficient* and *Containment Coefficient* for comparing common substrings in both sentences. Here the *IDF* weights are computed on character n-gram level. We use n-gram weighing and $n \in \{2, 3, 4, 5\}$.

  We enrich these features also by *Greedy String Tiling*(Wise, 1996) allowing to deal with reordered text parts and by *Longest Common Substring* (LCS) measuring the ration between LCS and length of the sentences.

- **TF-IDF:** For each word in a sentence we calculate *TF-IDF*. Given the word vocabulary V, the sentence is represented as a vector of dimension $|V|$ with *TF-IDF* values of words present in the sentence. The similarity between two sentences is expressed as cosine similarity between corresponding *TF-IDF* vectors.

2.2 Semantic Similarity

In this section we describe in detail the techniques that are more semantically oriented and are based on *Distributional Hypothesis*. This principle states that we can induce (to some degree) the meaning of words from their distribution in the text. This claim has multiple theoretical roots in psychology, structural linguistics, or lexicography (Firth, 1957; Rubenstein and Goodenough, 1965; Miller and Charles, 1991).

- **Semantic composition:** This approach is based on *Frege's principle of compositionality*, which states that the meaning of a complex expression is determined as a composition of its parts, i.e. words. To represent the meaning of a sentence we use simple linear combination of word vectors, where weights are represented by the *TF-IDF* values of appropriate words. We use state-of-the-art word embedding methods, namely Continuous Bag of Words (CBOW) (Mikolov et al., 2013) and Global Vectors (GloVe) (Pennington et al., 2014). We use cosine similarity to compare vectors.

- **Paragraph2Vec:** Paragraph vectors were proposed in (Le and Mikolov, 2014) as an unsupervised method of learning text representation. Resulting feature vector has fixed dimension while the input text can be of any length. The paragraph vectors and word vectors are concatenated to predict the next word in a context. The paragraph token acts as a memory that remembers what information is missing from the current context. We use cosine similarity for comparing two paragraph vectors.

- **Tree LSTM**: Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) with a complex computational unit. We use tree-structured representation of LSTM presented in (Tai et al., 2015). Tree model

represents the sentence structure. RNN processes input sentences of variable length via recursive application of a transition function on a hidden state vector h_t. For each sentence pair it creates sentence representations h_L and h_R using Tree-LSTM model. Given these representations, model predicts the similarity score using a neural network considering distance and angle between vectors.

- **Word alignment:** Method presented in (Sultan et al., 2014a; Sultan et al., 2014b; Sultan et al., 2015) has been very successful in last years. Given two sentence we want to compare, this method finds and aligns the words that have similar meaning and similar role in these sentences.

  Unlike the original method, we assume that not all word alignments have the same importance for the meaning of the sentences. The weight of a set of words $\boldsymbol{A}$ is a sum of word's *IDF* values $\omega(\boldsymbol{A}) = \sum_{w \in \boldsymbol{A}} \text{IDF}(w)$, where w is a word. Then the sentence similarity is given by

$$\text{sim}(\boldsymbol{S}_1, \boldsymbol{S}_2) = \frac{\omega(\boldsymbol{A}_1) + \omega(\boldsymbol{A}_2)}{\omega(\boldsymbol{S}_1) + \omega(\boldsymbol{S}_2)}, \quad (1)$$

  where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are input sentences (represented as sets of words). $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ denote the sets of aligned words for $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, respectively. The weighing of alignments improves our results significantly.

2.3 Similarity Combination

The combination of STS techniques is in fact a regression problem where the goal is to find the mapping from input space $\boldsymbol{x}_i \in \mathbb{R}^d$ of d-dimensional real-valued vectors (each value $x_{i,a}$, where $1 \leq a \leq d$ represents the single STS technique) to an output space $y_i \in \mathbb{R}$ of real-valued targets (desired semantic similarity). These mapping are learned from the training data $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$ of size N. There exist a lot of regression methods. We experiment with several of them:

- **Linear Regression:** Linear Regression (LR) is probably the simplest regression method. It is defined as $y_i = \boldsymbol{\lambda}\boldsymbol{x}_i$, where $\boldsymbol{\lambda}$ is a vector of weights that can be estimated for example by the *least squares method*.

- **Gaussian processes regression:** Gaussian process regression (GPR) is nonparametric kernel-based probabilistic model for non-linear regression (Rasmussen and Williams, 2005).

- **SVM Regression:** We use Support Vector Machines (SVM) for regression with the radial basis functions (RBF) as a kernel. We use improved Sequential Minimal Optimization (SMO) algorithm for parameter estimation introduce in (Shevade et al., 2000).

- **Decision Trees Regression:** The output of the Decision Trees Regression (DTR) (Breiman et al., 1984) is predicted by the sequence of decisions organized in a tree.

- **Perceptron Regression:** Multilayer Perceptron (MLP) is feed-forward artificial neural network that uses back-propagation to classify instances.

3 System Description

This section describes the settings of our final STS system. For monolingual STS task we submitted two runs. First is based on supervised learning and the second is unsupervised system:

- **UWB sup:** Supervised system based on SVM regression with RBF kernel. We use all techniques described in 2 as features for regression. During the regression we also use the simple trick. We create another features represented as a product of each pair of features $x_{i,a} \times x_{i,b}$ for $a \neq b$. We do so to better model the dependencies between single features. Together, we have 301 STS features. The system is trained on all SemEval datasets from prior years (see Table 1).

- **UWB unsup:** Unsupervised system based only on weighted word alignment (Section 2.2).

We handled with the cross-lingual STS task with Spanish-English bilingual sentence pairs in two steps. Firstly, we translated Spanish sentences to English via *Google translator*. The English sentences

Corpora	Pairs
SemEval 2012 Train	2,234
SemEval 2012 Test	3,108
SemEval 2013 Test	1,500
SemEval 2014 Test	3,750
SemEval 2015 Test	3,000

Table 1: STS gold data from prior years.

	News	Multi-Source	Mean	RR	TR
UWB sup	0.9062	**0.8190**	**0.8631**	1	1
UWB unsup	**0.9124**	0.8082	0.8609	2	1

Table 4: Pearson correlations on cross-lingual STS task of SemEval 2016. *RR* denote the run (system) ranking and *TR* denote our team ranking.

were left untouched. Secondly, we used the same STS systems as for monolingual task.

For preprocessing pipeline we used Standford CoreNLP library (Manning et al., 2014), i.e. for tokenization, lemmatization and POS tagging. Most of our STS techniques (apart from word alignment and POS n-gram overlaps) work with lemmas instead of word forms (this leads to slightly better performance). Some of our STS techniques are based on unsupervised learning and thus they need large unannotated corpora to train. We trained Paragraph2Vec, GloVe and CBOW models on *One billion word benchmark* presented in (Chelba et al., 2014). Dimension of vectors for all these models was set to 300. TF-IDF values were also estimated on this corpus.

All regression methods mentioned in Section 2.3 are implemented in WEKA (Hall et al., 2009).

4 Results and Discussion

This section presents the results of our systems for both English monolingual and Spanish-English cross-lingual STS task of SemEval 2016. In addition we present detailed results on the test data from SemEval 2015. As an evaluation measure we use *Pearson correlation* between system output and human annotations.

In the tables we present the correlation for each individual test set. Column *Mean* represents the weighted sum of all correlations, where the weight are given by the ratio of data set length compared to the full length of all datasets together. The mean value of Pearson correlations is also used as the main evaluation measure for ranking the system submissions.

In the Table 2 we show the results for the test data from 2015. We trained our systems on SemEval STS data from years 2012–2014. We provide comparison of individual STS techniques as well as of different types of regressions. Clearly, the SVM regression and Gaussian processes regression perform best and with our feature set it is 1% better than the winning system of SemEval 2015. The best performing single technique is indisputably the weighed word alignment correlated by 79.6% with gold data. Note that without weighing, we achieved only 74.2% on this data. The original result from authors of this approach was, however, 79.2%. This is probably caused by some inaccuracies in our implementation. Anyway, the weighing improves the correlation even if we compare it to the original results. Note that for estimating regression parameters we use the data from all years apart from 2015 (see Table 1).

The results for monolingual STS task of SemEval 2016 are shown in Table 3. In the time of writing this paper the ranks of submitted systems were not known. Thus we present only our correlations. We can see that our supervised system (SVM regression) performs approximately 3% better than the unsupervised one (weighed word alignment). On the data from SemEval 2015 this difference was not so significant (approximately 1.5%).

Finally, the results for cross-lingual STS task of SemEval 2016 are shown in Table 4. We achieved very high correlations. To be honest we must say that we expected much lower correlation through the fact that we use the machine translation via Google translator causing certainly some inaccuracies (at least in the syntax of the sentence). On the other hand, it proves that our model efficiently generalizes the learned patterns. Here, there is almost no difference in performance between supervised and unsupervised version of submitted systems. Our submitted runs finished first and second among 26 competing systems.

Model \Corpora	Answers-forums	Answers-students	Belief	Headlines	Images	Mean
Winner of SemEval 2015	0.7390	0.7725	0.7491	0.8250	0.8644	0.8015
Linear regression – all lexical	0.7053	0.7656	0.7190	0.7887	0.8246	0.7728
Linear regression – all syntactic	0.3089	0.3165	0.4570	0.2900	0.1862	0.2939
Tf-idf	0.5629	0.6043	0.6762	0.6603	0.7530	0.6593
Tree LSTM	0.4181	0.5490	0.5863	0.7324	0.8168	0.6501
Paragraph2Vec	0.5228	0.7017	0.6643	0.6562	0.7385	0.6725
CBOW composition	0.6216	0.6846	0.7258	0.6927	0.7831	0.7085
GloVe composition	0.5820	0.6311	0.7164	0.6969	0.7972	0.6936
Weighted word alignment	0.7171	**0.7752**	0.7632	0.8179	0.8525	0.7964
Linear regression	**0.7411**	0.7589	0.7739	0.8193	0.8568	0.7982
Gaussian processes regression	0.7363	0.7701	**0.7846**	0.8393	0.8749	0.8112
Decision trees regression	0.6700	0.6991	0.7281	0.7792	0.8206	0.7495
Perceptron regression	0.7060	0.7481	0.7467	0.8093	0.8594	0.7858
SVM regression	0.7375	0.7678	**0.7846**	**0.8398**	**0.8776**	**0.8116**

Table 2: Pearson correlations on SemEval 2015 evaluation data and comparison with the best performing system in this year.

Model \Corpora	Answer-answer	Headlines	Plagiarism	Postediting	Question-question	Mean
UWB sup	0.6215	**0.8189**	0.8236	**0.8209**	**0.7020**	**0.7573**
UWB unsup	**0.6444**	0.7935	**0.8274**	0.8121	0.5338	0.7262

Table 3: Pearson correlations on monolingual STS task of SemEval 2016.

5 Conclusion

In this paper we described our UWB system participating in SemEval 2016 competition in the task of Semantic Textual Similarity. We participated on both monolingual and cross-lingual parts of competition.

Our best results have been achieved by SVM regression of various STS techniques based on lexical, syntactic, and semantic information. This approach has been shown to work well for both subtasks.

Acknowledgments

This publication was supported by the project LO1506 of the Czech Ministry of Education, Youth and Sports and by Grant No. SGS-2016-018 Data and Software Engineering for Advanced Applications. Computational resources were provided by the CESNET LM2015042 and the CERIT Scientific Cloud LM2015085, provided under the programme "Projects of Large Research, Development, and Innovations Infrastructures".

References

Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. 2012. Semeval-2012 task 6: A pilot on semantic textual similarity. In *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*, SemEval '12, pages 385–393, Stroudsburg, PA, USA. Association for Computational Linguistics.

Eneko Agirre, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, and Weiwei Guo. 2013. *sem 2013 shared task: Semantic textual similarity. 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 32–43, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Eneko Agirre, Carmen Banea, Claire Cardie, Daniel

Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2014. Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 81–91, Dublin, Ireland, August. Association for Computational Linguistics and Dublin City University.

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.

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 6th International Workshop on Semantic Evaluation, held in conjunction with the 1st Joint Conference on Lexical and Computational Semantics*, pages 435–440, Montreal, Canada, June.

Leo Breiman, Jerome Friedman, Charles J. Stone, and R.A. Olshen. 1984. *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Andrei Z. Broder. 1997. On the resemblance and containment of documents. In *SEQUENCES '97 Proceedings of the Compression and Complexity of Sequences*, pages 21–29, Jun.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. 2014. One billion word benchmark for measuring progress in statistical language modeling. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association*, pages 2635–2639, Singapore, September.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Science 41*, pages 391–407.

John R. Firth. 1957. A Synopsis of Linguistic Theory, 1930-1955. *Studies in Linguistic Analysis*, pages 1–32.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, November.

Lushan Han, Abhay L. Kashyap, Tim Finin, James Mayfield, and Jonathan Weese. 2013. Umbc_ebiquity-core: Semantic textual similarity systems. 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 44–52, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Christopher D. Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA, USA.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space.

George A. Miller and Walter G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Carl Edward Rasmussen and Christopher K. I. Williams. 2005. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, October.

Shirish K. Shevade, Sathiya S. Keerthi, Chiru Bhattacharyya, and K.R.K. Murthy. 2000. Improvements to the smo algorithm for svm regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, Sep.

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

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2014b. 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.

Md Arafat Sultan, Steven Bethard, and Tamara Sumner. 2015. Dls@cu: Sentence similarity from word alignment and semantic vector composition. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 148–153, Denver, Colorado, June. Association for Computational Linguistics.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China, July. Association for Computational Linguistics.

Michael J. Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *Proceedings of the Twenty-seventh SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '96, pages 130–134, New York, NY, USA. ACM.