FBK-TR: Applying SVM with Multiple Linguistic Features for Cross-Level Semantic Similarity

Ngoc Phuoc An Vo
Fondazione Bruno Kessler
University of Trento
Trento, Italy
ngoc@fbk.eu

Tommaso Caselli
TrentoRISE
Trento, Italy
t.caselli@trentorise.eu

Octavian Popescu
Fondazione Bruno Kessler
Trento, Italy
popescu@fbk.eu

Abstract

Recently, the task of measuring semantic similarity between given texts has drawn much attention from the Natural Language Processing community. Especially, the task becomes more interesting when it comes to measuring the semantic similarity between different-sized texts, e.g paragraph-sentence, sentence-phrase, phrase-word, etc. In this paper, we, the FBK-TR team, describe our system participating in Task 3 "Cross-Level Semantic Similarity", at SemEval 2014. We also report the results obtained by our system, compared to the baseline and other participating systems in this task.

1 Introduction

Measuring semantic text similarity has become a hot trend in NLP as it can be applied to other tasks, e.g. Information Retrieval, Paraphrasing, Machine Translation Evaluation, Text Summarization, Question and Answering, and others. Several approaches proposed to measure the semantic similarity between given texts. The first approach is based on vector space models (VSMs) (Meadow, 1992). A VSM transforms given texts into "bag-of-words" and presents them as vectors. Then, it deploys different distance metrics to compute the closeness between vectors, which will return as the distance or similarity between given texts. The next well-known approach is using text-alignment. By assuming that two given texts are semantically similar, they could be aligned on word or phrase levels. The alignment quality can serve as a similarity measure. "It typically pairs words from the two texts by maximizing the summation of the word similarity of the resulting pairs" (Mihalcea et al., 2006). In contrast, the third approach uses machine learning techniques to learn models built from different lexical, semantic and syntactic features and then give predictions on degree of similarity between given texts (Šarić et al., 2012).

At SemEval 2014, the Task 3 "Cross-Level Semantic Similarity" (Jurgens et al., 2014) is to evaluate the semantic similarity across different sizes of texts, in particular, a larger-sized text is compared to a smaller-sized one. The task consists of four types of semantic similarity comparison: paragraph to sentence, sentence to phrase, phrase to word, and word to sense. The degree of similarity ranges from 0 (different meanings) to 4 (similar meanings). For evaluation, systems were evaluated, first, within comparison type and second, across all comparison types. Two methods are used to evaluate between system outputs and gold standard (human annotation), which are Pearson correlation and Spearman's rank correlation (rho).

The FBK-TR team participated in this task with three different runs. In this paper, we present a clear and comprehensive description of our system which obtained competitive results. Our main approach is using machine learning technique to learn models from different lexical and semantic features from train corpora to make prediction on the test corpora. We used support vector machine (SVM) regression model to solve the task.

The remainder of the paper is organized as follows. Section 2 presents the system overview. Sections 3, 4 and 5 describe the Semantic Word Similarity, String Similarity and other features, respectively. Section 6 discusses about SVM approach. Section 7 presents the experiment settings for each subtask. Finally, Sections 8 and 9 present the evaluation and conclusion.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 284–288,
Dublin, Ireland, August 23-24 2014.

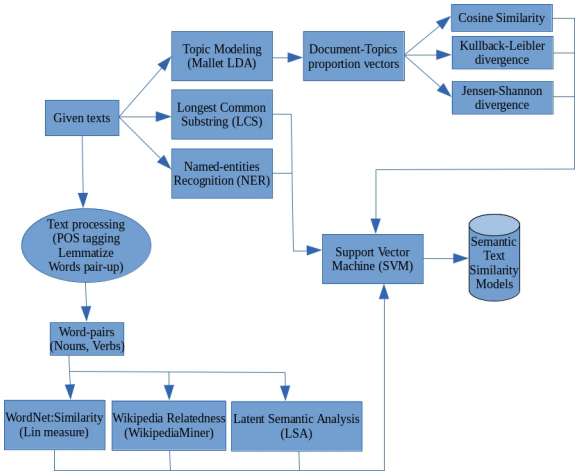

Figure 1: System Overview.

2 System Overview

Our system was built on different linguistic features as shown in Figure 1. By constructing a pipeline system, each linguistic feature can be used independently or together with others to measure the semantic similarity of given texts as well as to evaluate the significance of each feature to the accuracy of system's predictions. On top of this, the system is expandable and scalable for adopting more useful features aiming for improving the accuracy.

3 Semantic Word Similarity Measures

At the lexical level, we built a simple, yet effective Semantic Word Similarity model consisting of three components: WordNet similarity, Wikipedia relatedness and Latent Semantic Analysis (LSA). These components played important and complimentary roles to each other.

3.1 Data Processing

We used the TreeTagger tool (Schmid, 1994) to extract Part-of-Speech (POS) from each given text, then tokenize and lemmatize it. On the basis of the POS tags, we only picked lemmas of content words (Nouns and Verbs) from the given texts and then paired them up regarding to similar POS tags.

3.2 WordNet Similarity and Levenshtein Distance

WordNet (Fellbaum, 1999) is a lexical database for the English language in which words are grouped into sets of synonyms (namely synsets, each expressing a distinct concept) to provide short, general definitions, and record the various semantic relations between synsets. We used Perdersen's package WordNet:Similarity (Pedersen et al., 2004) to obtain similarity scores for the lexical items covered in WordNet. Similarity scores have been computed by means of the Lin measure (Lin, 1998). The Lin measure is built on Resnik's measure of similarity (Resnik, 1995):

$$Sim_{lin} = \frac{2 * IC(LCS)}{IC(concept_1) + IC(concept_2)} \quad (1)$$

where *IC(LCS)* is the information content *(IC)* of the least common subsumer *(LCS)* of two concepts.

To overcome the limit in coverage of WordNet, we applied the Levenshtein distance (Levenshtein, 1966). The distance between two words is defined by the minimum number of operations (insertions, deletions and substitutions) needed to transform one word into the other.

3.3 Wikipedia Relatedness

Wikipedia Miner (Milne and Witten, 2013) is a Java-based package developed for extracting semantic information from Wikipedia. Through our experiments, we observed that Wikipedia relatedness plays an important role for providing extra information to measure the semantic similarity between words. We used the package Wikipedia Miner from University of Waikato (New Zealand) to extract additional relatedness scores between words.

3.4 Latent Semantic Analysis (LSA)

We also took advantage from corpus-based approaches to measure the semantic similarity between words by using Latent Semantic Analysis (LSA) technique (Landauer et al., 1998). LSA assumes that similar and/or related words in terms of meaning will occur in similar text contexts. In general, a LSA matrix is built from a large corpus. Rows in the matrix represent unique words and columns represent paragraphs or documents. The content of the matrix corresponds to the word count per paragraph/document. Matrix size is then reduced by means of Single Value Decomposition (SVD) technique. Once the matrix has been obtained, similarity and/or relatedness between the words is computed by means of cosine values (scaled between 0 and 1) for each word vector in the matrix. Values close to 1 are assumed to

be very similar/related, otherwise dissimilar. We trained our LSA model on the British National Corpus (BNC) [1] and Wikipedia [2] corpora.

4 String Similarity Measures

The Longest Common Substring (LCS) is the longest string in common between two or more strings. Two given texts are considered similar if they are overlapping/covering each other (e.g sentence 1 covers a part of sentence 2, or otherwise). We implemented a simple algorithm to extract the LCS between two given texts. Then we divided the LCS length by the product of normalized lengths of two given texts and used it as a feature.

4.1 Analysis Before and After LCS

After extracting the LCS between two given texts, we also considered the similarity for the parts before and after the LCS. The similarity between the text portions before and after the LSC has been obtained by means of the Lin measure and the Levenshtein distance.

5 Other Features

To take into account other levels of analysis for semantic similarity between texts, we extended our features by means of topic modeling and Named Entities.

5.1 Topic Modeling (Latent Dirichlet Allocation - LDA)

Topic modeling is a generative model of documents which allows to discover topics embedded in a document collection and their balance in each document. If two given texts are expressing the same topic, they should be considered highly similar. We applied topic modeling, particularly, Latent Dirichlet allocation (LDA) (Blei et al., 2003) to predict the topics expressed by given texts.

The MALLET topic model package (McCallum, 2002) is a Java-based tool used for inferring hidden "topics" in new document collections using trained models. We used Mallet topic modeling tool to build different models using BNC and Wikipedia corpora.

We noticed that, in LDA, the number of topics plays an important role to fine grained predictions. Hence, we built different models for different numbers of topics, from minimum 20 topics to maximum 500 topics (20, 50, 100, 150, 200, 250, 300, 350, 400, 450 and 500). From the proportion vectors (distribution of documents over topics) of given texts, we applied three different measures to compute the distance between each pair of texts, which are Cosine similarity, Kullback-Leibler and Jensen-Shannon divergences (Gella et al., 2013).

5.2 Named-Entity Recognition (NER)

NER aims at identifying and classifying entities in a text with respect to a predefined set of categories such as person names, organizations, locations, time expressions, quantities, monetary values, percentages, etc. By exploring the training set, we observed that there are lot of texts in this task containing named entities. We deployed the Stanford Named Entity Recognizer tool (Finkel et al., 2005) to extract the similar and overlapping named entities between two given texts. Then we divided the number of similar/overlapping named entities by the sum length of two given texts.

6 Support Vector Machines (SVMs)

Support vector machine (SVM) (Cortes and Vapnik, 1995) is a type of supervised learning approaches. We used the LibSVM package (Chang and Lin, 2011) to learn models from the different linguistic features described above. However, in SVM the problem of finding optimal kernel parameters is critical and important for the learning process. Hence, we used practical advice (Hsu et al., 2003) for data scaling and a grid-search process for finding the optimal parameters (C and gamma) for building models. We trained the SVM models in a regression framework.

7 Experiment Settings

For subtasks paragraph-to-sentence and sentence-to-phrase, since the length between two units is completely different, we decided, first to apply topic model to identify if two given texts are expressing a same topic. Furthermore, named entities play an important role in these subtasks. However, as there are many named entities which are not English words and cannot be identified by the NER tool, we developed a program to detect and identify common words occurring in both given texts. Then we continued to extract other lexical and semantic features to measure the similarity between the two texts.

[1] http://www.natcorp.ox.ac.uk

[2] http://en.wikipedia.org/wiki/Wikipedia:Database_download

Team	Para2Sent (Pearson)	Para2Sent (Spearman)
UNAL-NLP, run2 (ranked 1st)	**0.837**	0.820
ECNU, run1(ranked 1st)	0.834	**0.821**
FBK-TR, run2	0.77	0.775
FBK-TR, run3	0.759	0.770
FBK-TR, run1	0.751	0.759
Baseline (LCS)	0.527	0.613

Table 1: Results for paragraph-to-sentence.

Team	Sent2Phr (Pearson)	Sent2Phr (Spearman)
Meerkat_Mafia, SuperSaiyan (ranked 1st)	0.777	0.760
FBK-TR, run3	0.702	0.695
FBK-TR, run1	0.685	0.681
FBK-TR, run2	0.648	0.642
Baseline (LCS)	0.562	0.626

Table 2: Results for sentence-to-phrase.

For the subtask word-to-sense, we used the Semantic Word Similarity model which consists of three components: WordNet similarity, Wikipedia relatedness and LSA similarity (described in section 3). For phrase-to-word, we extracted all glosses of the given word, then computed the similarity between the given phrase and each extracted gloss. Finally, we selected the highest similarity score for result.

8 Evaluations

As a result, we report our performance in the four subtasks as follows.

8.1 Subtasks: Paragraph-to-Sentence and Sentence-to-Phrase

The evaluation results using Pearson and Spearman correlations show the difference between our system and best system in these two subtasks in the Tables 1 and 2.

Team	Para2Sent	Sent2Phr	Phr2Word	Word2Sens	Sum
SimCompass (ranked 1st)	0.811	0.742	0.415	0.356	2.324
FBK-TR	0.759	0.702	0.305	0.155	1.95
Baseline	0.527	0.562	0.165	0.109	1.363

Table 3: Overall result using Pearson.

Team	Para2Sent	Sent2Phr	Phr2Word	Word2Sens	Sum
SimCompass (ranked 1st)	0.801	0.728	0.424	0.344	2.297
FBK-TR	0.770	0.695	0.298	0.150	1.913
Baseline	0.613	0.626	0.162	0.130	1.528

Table 4: Overall result using Spearman.

8.2 Subtasks: Phrase-to-Word and Word-to-Sense

Even though we did not submit the results as they looked very low, we report the scores for the phrase-to-word and word-to-sense subtasks. In the phrase-to-word subtask, we obtained a Pearson score of 0.305 and Spearman value of 0.298. As for the word-to-sense subtask, we scored 0.155 for Pearson and 0.150 for Spearman.

Overall, with the submitted results for two subtasks described in Section 8.1, our system's runs ranked 20th, 21st and 22nd among 38 participating systems. However, by taking into account the un-submitted results for the two other subtasks, our best run obtained 1.95 (Pearson correlation) and 1.913 (Spearman correlation), which can be ranked in the top 10 among 38 systems (figures are reported in Table 3 and 4).

9 Conclusions and Future Work

In this paper, we describe our system participating in the Task 3, at SemEval 2014. We present a compact system using machine learning approach (particularly, SVMs) to learn models from a set of lexical and semantic features to predict the degree of similarity between different-sized texts. Although we only submitted the results for two out of four subtasks, we obtained competitive results among the other participants. For future work, we are planning to increase the number of topics in LDA, as more fine-grained topics should allow predicting better similarity scores. Finally, we will investigate more on the use of syntactic features.

References

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *The Journal of Machine Learning research*, 3:993–1022.

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27.

Corinna Cortes and Vladimir Vapnik. 1995. Support-

Vector Networks. *Machine learning*, 20(3):273–297.

Christiane Fellbaum. 1999. *WordNet*. Wiley Online Library.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370.

Spandana Gella, Bahar Salehi, Marco Lui, Karl Grieser, Paul Cook, and Timothy Baldwin. 2013. Unimelb_nlp-core: Integrating predictions from multiple domains and feature sets for estimating semantic textual similarity. *Atlanta, Georgia, USA*, page 207.

Chih-Wei Hsu, Chih-Chung Chang, Chih-Jen Lin, et al. 2003. A Practical Guide to Support Vector Classification.

David Jurgens, Mohammad Taher Pilehvar, and Roberto Navigli. 2014. Semeval-2014 Task 3: Cross-Level Semantic Similarity. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval-2014) August 23-24, 2014, Dublin, Ireland*.

Thomas K Landauer, Peter W Foltz, and Darrell Laham. 1998. An Introduction to Latent Semantic Analysis. *Discourse processes*, 25(2-3):259–284.

Vladimir I Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. In *Soviet physics doklady*, volume 10, page 707.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. In *ICML*, volume 98, pages 296–304.

Andrew Kachites McCallum. 2002. Mallet: A Machine Learning for Language Toolkit.

Charles T Meadow. 1992. *Text Information Retrieval Systems*. Academic Press, Inc., Orlando, FL, USA.

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.

David Milne and Ian H Witten. 2013. An Open-Source Toolkit for Mining Wikipedia. *Artificial Intelligence*, 194:222–239.

Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi. 2004. Wordnet::similarity - Measuring the Relatedness of Concepts. In *Demonstration Papers at HLT-NAACL 2004*, pages 38–41.

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *arXiv preprint cmp-lg/9511007*.

Frane Šarić, Goran Glavaš, Mladen Karan, Jan Šnajder, and Bojana Dalbelo Bašić. 2012. Takelab: Systems for Measuring Semantic Text 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*, pages 441–448.

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of international conference on new methods in language processing*, volume 12, pages 44–49. Manchester, UK.