FBK-TR: SVM for Semantic Relatedness and Corpus Patterns for RTE

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

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

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

Abstract

This paper reports the description and scores of our system, FBK-TR, which participated at the SemEval 2014 task #1 "Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Entailment". The system consists of two parts: one for computing semantic relatedness, based on SVM, and the other for identifying the entailment values on the basis of both semantic relatedness scores and entailment patterns based on verb-specific semantic frames. The system ranked 11^{th} on both tasks with competitive results.

1 Introduction

In the Natural Language Processing community, meaning related tasks have gained an increasing popularity. These tasks focus, in general, on a couple of short pieces of text, like pair of sentences, and the systems are required to infer a certain meaning relationship that exists between these texts. Two of the most popular meaning related tasks are the identification of Semantic Text Similarity (STS) and Recognizing Textual Entailment (RTE). The STS tasks require to identify the degree of similarity (or relatedness) that exists between two text fragments (sentences, paragraphs, ...), where similarity is a broad concept and its value is normally obtained by averaging the opinion of several annotators. The RTE task requires the identification of a directional relation between a pair of text fragments, namely a text (T) and a hypothesis (H). The relation (T $\rightarrow$ H) holds whenever the truth of H follows from T.

At SemEval 2014, the Task #1 "Evaluation of Compositional Distributional Semantic Models on Full Sentences through Semantic Relatedness and Entailment" (Marelli et al., 2014a) primarily aimed at evaluating Compositional Distributional Semantic Models (CDSMs) of meaning over two subtasks, namely semantic relatedness and textual entailment (ENTAILMENT, CONTRADICTION and NEUTRAL), over pairs of sentences (Marelli et al., 2014b). Concerning the relatedness subtask, the system outputs are evaluated against gold standard ratings in two ways, using Pearson correlation and Spearman's rank correlation (rho). The Pearson correlation is used for evaluating and ranking the participating systems. Similarly, for the textual entailment subtask, system outputs are evaluated against a gold standard rating with respect to accuracy.

Our team, FBK-TR, participated in both subtasks with five different runs. In this paper, we present a comprehensive description of our system which obtained competitive results in both tasks and which is not based on CDSMs. Our approach for the relatedness task is based on machine learning techniques to learn models from different lexical and semantic features from the train corpus and then to make prediction on the test corpus. Particularly, we used support vector machine (SVM) (Chang and Lin, 2011), regression model to solve this subtask. On the other hand, the textual entailment task uses a methodology mainly based on corpus patterns automatically extracted from annotated text corpora.

The remainder of the paper is organized as follows: Section 2 presents the SVM system for semantic relatedness. Section 3 describes the methodology used for extracting patterns and computing the textual entailment values. Finally, Section 4 discusses about the evaluations and Section 5 presents conclusions and future work.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 289–293,
Dublin, Ireland, August 23-24 2014.

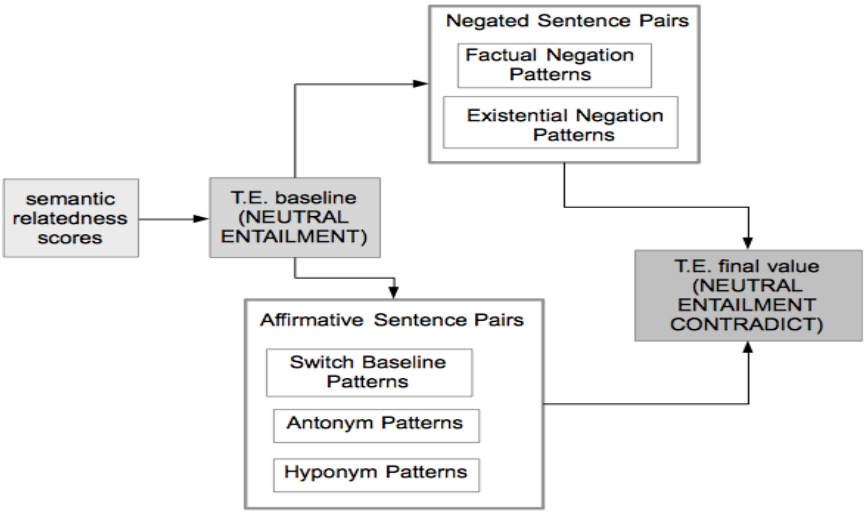

Figure 1: Schema of the system for computing entailment.

2 System Overview for Semantic Relatedness Subtask

Concerning the Semantic Relatedness subtask our SVM system is built on different linguistic features, ranging from relatedness at the lexical level (WordNet based measures, Wikipedia relatedness and Latent Semantic Analysis), to sentence level, including topic modeling based on Latent Dirichlet allocation (LDA) and string similarity (Longest Common Substring).

2.1 Lexical Features

At the lexical level, we built a simple, yet effective Semantic Word Relatedness model, which consists of 3 components: WordNet similarity (based on the Lin measure as implemented in Pedersen package `WordNet:Similarity` (Pedersen et al., 2004), Wikipedia relatedness (as provided by the Wikipedia Miner package (Milne and Witten, 2013)), and Latent Semantic Analysis (Landauer et al., 1998), with a model trained on the British National Corpus (BNC) [1] and Wikipedia. At this level of analysis, we concentrated only on the best matched (lemma) pairs of content words, i.e. Noun-Noun, Verb-Verb, extracted from each sentence pair. The content words have been automatically extracted by means of part-of-speech tagging (TreeTagger (Schmid, 1994)) and lemmatization.

For words which are not present in WordNet, the relatedness score has been obtained by means of the Levenshtein distance (Levenshtein, 1966).

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

2.2 Topic Modeling

We have applied topic modeling based on Latent Dirichlet allocation (LDA) (Blei et al., 2003) as implemented in the MALLET package (McCallum, 2002). The topic model was developed using the BNC and Wikipedia (with the numbers of topics varying from 20 to 500 topics). From the proportion vectors (distribution of documents over topics) of the given texts, we apply 3 different measures (Cosine similarity, Kullback-Leibler and Jensen-Shannon divergences) to compute the distances between each pair of sentences.

2.3 String Similarity: Longest Common Substring

As for the string level, two given sentences are considered similar/related if they are overlapping/covering each other (e.g sentence 1 covers a part of sentence 2, or otherwise). Hence, we considered the text overlapping between two given texts as a feature for our system. The extraction of the features at the string level was computed in two steps: first, we obtained Longest Common Substring between two given sentences. After this, we also considered measuring the similarity for the parts before and after the LCS between two given texts, by means of the Lin measure and the Levenshtein distance.

3 System Overview for RTE Subtask

The system for the identification of the entailment values is illustrated in Figure 1. Entailment values are computed starting from a baseline (only ENTAILMENT and NEUTRAL values) which relies on the output (i.e. scores) of the semantic relatedness system. After this step, two groups of entailment patterns are applied whether the surface form of a sentence pair is affirmative (i.e. absence of negation words) or negative. Each type of pattern provides in output an associated entailment value which corresponds to the final value assigned by the system.

The entailment patterns are based on verb-specific semantic frames that include both syntactic and semantic information. Hence, we have explicit access to the information that individual words have and to the process of combining them in bigger units, namely phrases, which carry out meanings. The patterns have two properties: i.) the senses of the words inside the pattern are stable, they do not change whatever context is added to the left, right or inside the phrase matching the pattern, and ii.) the replacement of a word with another word belonging to a certain class changes the senses of the words. Patterns with these properties are called Sense Discriminative Patterns (SDPs). It has been noted (Popescu et al., 2011) that we can associate to a phrase that is matched by an SDP a set of phrases for which an entailment relationship is decidable showing that there is a direct relationship between SDPs and entailment .

SDP patterns have been obtained from large parsed corpora. To maximize the accuracy of the corpus we have chosen sentences containing at maximum two finite verbs from BNC and Annotated English Gigaword. We parsed this corpus with the Stanford parser, discarding the sentences from the Annotated English Gigaword which have a different parsing. Each words is replaced with their possible SUMO attributes (Niles and Pease, 2003). Only the following Stanford dependencies are retained as valid *[n, nsub]sbj, [d,i,p]obj, prep, [x,c]comp*. We considered only the most frequent occurrences of such patterns for each verb. To cluster into a single SDP pattern, all patterns that are sense auto-determinative, we used the OntoNotes (Hovy et al., 2006) and CPA (Hanks, 2008) lexica. Inside each cluster, we searched for the most general hypernyms for each syntactic slot such that there are no common patterns between clusters (Popescu, 2013). However, the patterns thus obtained are not sufficient enough for the task. Some expressions may be the paraphrasis a word in the context of an SDP. To extract this information, we considered all the pairs in training that are in an ENTAILMENT relationship, with a high relatedness score (4 to 5), and we extracted the parts that are different for each grammatical slot. In this way, we compiled a list of quasi synonym phrases that can be replaced inside an SDP without affecting the replacement. This is the only component that depends on the training corpus. Figure 2 describes the algorithm for computing entailment on the basis of the SDPs. The following subsections illustrate the identification of entailment relation for affirmative sentences and negated sentences.

```
Algorithm Entailment via SDPs
Require: Parsed pairs of phrases
Require: Synonymy repository
Require: Hyponym, Hypernym repository
Require: NegativeWordsList
Require: DepSet dependency set
Ensure: {ENTAILMENT, CONTRADICTION, NEUTRAL}
Consider the SDP for phrase_1 and phrase_2
flNeg ← 0;
if phrase_1 xor phrase_2 contain words ∈ NegativeWordsList then
    generate EA sets
    flNeg ← 1;
end if
match ← 1
while match do
    consider each syntactic position in SDPs
    if word_1 and word_2 ∉ repositories then
        match ← -1
    end if
end while
if match == 1 then
    if flNeg == 1 then
        return CONTRADICTION
    end if
    return ENTAILMENT
end if
return NEUTRAL
```

Figure 2: Algorithm for computing entailment.

3.1 Entailment on Affirmative Sentences

Affirmative sentences use three types of entailment patterns. The switch baseline and hyponym patterns works in this way: If two sentences are matched by the same SDP, and the difference between them is that the second one contains a hypernym on the same syntactic position, then the first one is entailed by the second (i.e. ENTAILMENT). If the two SDPs are such that the difference between them is that the second contains a word which is not synonym, hypernym or hyponym on the same syntactic position, then there is no entailment between the two phrases (i.e. NEUTRAL). The entailment direction is from the sentence that contains the hyponym toward the other sentence. The antonym patterns check if the two SDPs are the same, with the only difference being in the verb of the second sentence being an antonym of the verb in the first sentence (i.e. CONTRADICTION).

3.2 Entailment on Negative Sentences

As for negated sentences, we distinguish between existential negative phrases (i.e. *there is no* or *there are no*) and factual negative ones (presence of a negative polarity word). An assumption related to each SDP is that it entails the existence of any of the component of the pattern which can be expressed by means of dedicated phrases. A SDP of the kind "*[Human] beat [Animal]*", entails both phrases, namely *there is a [Human]* and *there is a [Animal]*. We call this set of associated existential phrases, Existential Assumptions (EAs). This type of existential entailment obtained through the usage of SDP has a direct consequence for handling the ENTAILMENT, CONTRADICTION and NEUTRAL types of entailment when one of the phrases is negated. If the first phrase belongs to the EA of the second one, then the first phrase is entailed by the second phrase; if the first phrase is an existential negation of a phrase belonging to the EA set of the second phrase, meaning that it contains the string *there is/are no*, then the first one is a contradiction of the second phrase; if neither the first phrase, nor its negation belong to the EA set of the second phrase, then the two sentences are neutral with respect to the entailment. The general rule described in 3.1 applies to these types of phrases as well: replacing a word on the same syntactic slot inside a phrase that is matched by a SDP leads to a CONTRADICTION type of entailment, if the replacement is a hypernym of the original word. Similarly, the approach can be applied to factual negative phrases. The scope of negation is considered to be the extension of the SDP and thus the negative set of EAs.

4 Evaluation and Ranking

Table 1 illustrates the results for Pearson and Spearman correlations for the relatedness subtask on the test set. Table 2 reports the Accuracy values for the entailment subtask on the test set.

Concerning the relatedness results our systems ranked 11^{th} out of 17 participating systems. Best score of our system is reported in Table 1. One of the main reason for the relatively low results of the systems for this subtask concerns the fact that it is designed for a general-level of texts (i.e. compositionality is not taken into account).

Team	**Pearson**	**Spearman**
ECNU_run1 (ranked 1^{st})	0.82795	0.76892
FBK-TR_run3	0.70892	0.64430

Table 1: Results for semantic relatedness subtask.

Team	**Accuracy**
Illinois-LH_run1 (ranked 1^{st})	84.575
FBK-TR_run3	75.401
∗FBK-TR_baseline	64.080
∗FBK-TR_new	**85.082**

Table 2: Results for entailment subtask.

As for the entailment subtask, our system ranked 11^{th} out of 18 participating systems. The submitted results of the system are illustrated in Table 2 and are compared against the best system, our baseline system (∗FBK-TR_baseline) as described in Figure 1, and a new version of the participating system after fixing some bugs in the submitted version due to the processing of the parser's output (∗FBK-TR_new). The new version of the system scores in the top provides a new state of the art result, with an improvement of 10 points with respect to our submitted system.

5 Conclusion and Future Work

This paper reports the description of our system, FBK-TR, which implements a general SVM semantic relatedness system based on distributional features (LSA, LDA), knowledge-based related features (WordNet and Wikipedia) and string overlap (LCS). On top of that, we added structural information at both semantic and syntactic level by using SDP patterns. The system reached competitive results in both subtasks. By correcting some bugs in the entailment scripts, we obtained an improvement over our submitted systems as well as for the best ranking system. We plan to improve and extend the relatedness system by means of compositional methods. Finally, the entailment system can be improved by taking into account additional linguistic evidences, such as the alternation between indefinite and definite determiners, noun modifiers and semantically empty heads.

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.

Patrick Hanks. 2008. Mapping meaning onto use: a Pattern Dictionary of English Verbs. In *Proceedings of the AACL 2008*.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% Solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*, pages 57–60.

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.

M Marelli, L Bentivogli, M Baroni, R Bernardi, S Menini, and R Zamparelli. 2014a. Semeval-2014 Task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval 2014: International Workshop on Semantic Evaluation, August 23-24, 2014, Dublin, Ireland*.

M Marelli, S Menini, M Baroni, L Bentivogli, R Bernardi, and R Zamparelli. 2014b. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC 2014, Reykjavik (Iceland): ELRA*.

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

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

Ian Niles and Adam Pease. 2003. Mapping WordNet to the SUMO Ontology. In *Proceedings of the IEEE International Knowledge Engineering Conference*, pages 23–26.

Ted Pedersen, Patwardhan Siddharth, and Michelizzi Jason. 2004. Wordnet::Similarity: Measuring the Relatedness of Concepts. In *Proceedings of the HLT-NAACL 2004*.

Octavian Popescu, Elena Cabrio, and Bernardo Magnini. 2011. Textual Entailment Using Chain Clarifying Relationships. In *Proceedings of the IJCAI Workshop Learning by Reasoning and its Applications in Intelligent Question-Answering*.

Octavian Popescu. 2013. Learning Corpus Patterns Using Finite State Automata. In *Proceedings of the ICSC 2013*.

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