FBK-HLT: A New Framework for Semantic Textual Similarity

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

Simone Magnolini
University of Brescia,
Fondazione Bruno Kessler
Trento, Italy
magnolini@fbk.eu

Octavian Popescu
IBM Research, T.J. Watson
Yorktown, US
o.popescu@us.ibm.com

Abstract

This paper reports the description and performance of our system, FBK-HLT, participating in the SemEval 2015, Task #2 "Semantic Textual Similarity", English subtask. We submitted three runs with different hypothesis in combining typical features (lexical similarity, string similarity, word n-grams, etc) with syntactic structure features, resulting in different sets of features. The results evaluated on both STS 2014 and 2015 datasets prove our hypothesis of building a STS system taking into consideration of syntactic information. We outperform the best system on STS 2014 datasets and achieve a very competitive result to the best system on STS 2015 datasets.

1 Introduction

Semantic related tasks have been a noticed trend in Natural Language Processing (NLP) community. Particularly, the task Semantic Textual Similarity (STS) has captured a huge attention in the NLP community despite being recently introduced since SemEval 2012 (Agirre et al., 2012). Basically, the task requires to build systems which can compute the similarity degree between two given sentences. The similarity degree is scaled as a real score from 0 (no relevance) to 5 (semantic equivalence). The evaluation is done by computing the correlation between human judgment scores and system scores by the mean of Pearson correlation method.

At SemEval 2015, Task #2 "Semantic Textual Similarity (STS)", English STS subtask (Agirre et al., 2015) evaluates participating systems on five test datasets: image description (*image*), news headlines (*headlines*), student answers paired with reference answers (*answers-students*), answers to questions posted in stach exchange forums (*answers-forum*), and English discussion forum data exhibiting commited belief (*belief*). As being inspired by the UKP system (Bär et al., 2012), which was the best system in STS 2012, we build a supervised system on top of it. Our system adopts some word and string similarity features in UKP, such as string similarity, character/word n-grams, and pairwise similarity; however, we also add other distinguished features, like syntactic structure information, word alignment and semantic word similarity. As a result, our team, FBK-HLT, submitted three runs and achieve very competitive results in the top-tier systems of the task.

The remainder of this paper is organized as follows: Section 2 presents the System Description, Section 3 describes our Experiment Settings, Section 4 reports the Evaluations of our system. Finally, Section 5 is Conclusions and Future Work.

2 System Description

We describe our system, which is built from different linguistic features. We construct a pipeline system, in which each component produces different features independently and at the end, all features are consolidated by a machine learning tool, which learns a regression model for predicting the similarity scores from given sentence-pairs. On top of this, the system is expandable and scalable for adopting more useful features aiming for improving the accuracy. The System Overview in Figure 1 shows the logic and design processes in which different com-

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 102–106,
Denver, Colorado, June 4-5, 2015.

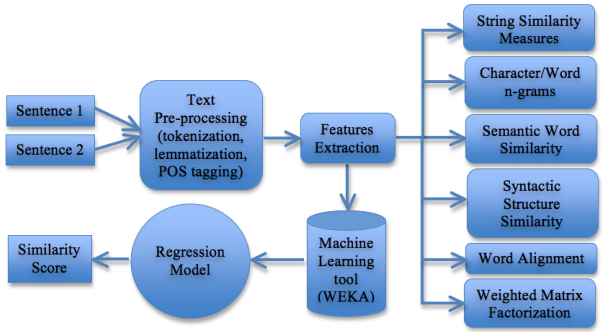

Figure 1: System Overview.

ponents connect and work together.

2.1 Data Preprocessing

The input data undergoes the data preprocessing in which we use Tree Tagger (Schmid, 1994) to perform tokenization, lemmatization, and Part-of-Speech (POS) tagging. On the other hand, we use Stanford Parser (Klein and Manning, 2003) to obtain the dependency parsing from given sentences.

2.2 Word and String Similarity Features

We adopt some word and string similarity features from the UKP system (Bär et al., 2012), which are briefly described as follows:

- String Similarity: we use Longest Common Substring (Gusfield, 1997), Longest Common Subsequence (Allison and Dix, 1986) and Greedy String Tiling (Wise, 1996) measures.

- Character/Word n-grams: we compare character n-grams (Barrón-Cedeno et al., 2010) with the variance n=2, 3, ..., 15. In contrast, we compare the word n-grams using Jaccard coefficient done by Lyon (Lyon et al., 2001) and containment measure (Broder, 1997) with the variance of n=1, 2, 3, and 4.

- Semantic Word Similarity: we use the pairwise similarity algorithm by Resnik (Resnik, 1995) on WordNet (Fellbaum, 1998), and the vector space model Explicit Semantic Analysis (ESA) (Gabrilovich and Markovitch, 2007) which is constructed by two lexical semantic resources Wikipedia [1] and Wiktionary [2].

2.3 Syntactic Structure Features

We exploit the syntactic structure information by the mean of three different toolkits: Syntactic Tree Kernel, Distributed Tree Kernel and Syntactic Generalization. We describe how each toolkit is used to learn and extract the syntactic structure information from texts to be used in our STS system.

2.3.1 Syntactic Tree Kernel

Syntactic Tree Kernel (Moschitti, 2006) is a tree kernels approach to learn the syntactic structure from syntactic parsing information, particularly, the Partial Tree (PT) kernel is proposed as a new convolution kernel to fully exploit dependency trees. We use the open-source toolkit "Tree Kernel in SVM-Light[3]" to learn this syntactic information.

Having assumed that paraphrased pairs would share the same content and similar syntactic structures, we decide to choose the Microsoft Research Paraphrasing Corpus (Dolan et al., 2005) which contains 5,800 sentence pairs extracted from news sources on the web, along with human annotations indicating whether each pair captures a paraphrase/semantic equivalence relationship. This corpus is split into Training set (4,076 pairs) and Testing set (1,725 pairs).

We use Stanford Parser (Klein and Manning, 2003) to obtain the dependency parsing from sentence pairs. Then we use the machine learning tool svm-light-tk 1.2 which uses Tree Kernel approach to learn the similarity of syntactic structure to build a binary classifying model on the Train dataset. The output predictions are probability confidence scores in [-1,1], corresponds to the probability of the label to be positive. According to the assumption above, we label paraphrased pairs as 1, -1 otherwise. We obtain the Accuracy of 69.16% on the Test set.

2.3.2 Distributed Tree Kernel

Distributed Tree Kernel (DTK) (Zanzotto and Dell'Arciprete, 2012) is a tree kernels method using a linear complexity algorithm to compute vectors for trees by embedding feature spaces of tree fragments

[1] http://en.wikipedia.org/wiki/Main_Page
[2] http://en.wiktionary.org
[3] http://disi.unitn.it/moschitti/Tree-Kernel.htm

Settings	deft-forum	deft-news	headlines	images	OnWN	tweet-news	Mean
Baseline	0.353	0.596	0.510	0.513	0.406	0.654	0.507
DLS@CU (ranked 1st)	0.4828	0.7657	0.7646	0.8214	0.8589	0.7639	0.761
Word/String Sim (1)	0.4314	0.7089	0.6887	0.7671	0.8125	0.6932	0.7008
Syntactic Features (2)	0.2402	0.3886	0.3233	0.2419	0.4066	0.4489	0.3441
(1) & (2)	0.4495	0.7032	0.6902	0.7627	0.8115	0.6974	0.7026
All Features	**0.5076**	0.7616	**0.7647**	0.8182	**0.8953**	0.7485	**0.7672**

Table 1: Evaluation Results on STS 2014 datasets.

System	ans-forums	ans-students	belief	headlines	images	Mean
Baseline	0.4453	0.6647	0.6517	0.5312	0.6039	0.5871
DLS@CU-S1 (ranked 1st)	0.739	0.7725	0.7491	0.825	0.8644	**0.8015**
FBK-HLT Run1	0.7131	0.7442	0.7327	0.8079	0.8574	**0.7831**
FBK-HLT Run2	0.7101	0.7410	0.7377	0.8008	0.8545	0.7801
FBK-HLT Run3	0.6555	0.7362	0.7460	0.7083	0.8389	0.7461

Table 2: Evaluation Results on STS 2015 datasets.

in low-dimensional spaces. Then a recursive algorithm is proposed with linear complexity to compute reduced vectors for trees. The dot product among reduced vectors is used to approximate the original tree kernel when a vector composition function with specific ideal properties is used.

Firstly, we use Stanford Parser (PCFG Parser) trained on Penn TreeBank (Klein and Manning, 2003) to obtain the dependency parsing of sentences, and feed them to the software "distributed-tree-kernels" to produce the distributed trees.[4] Then, we compute the Cosine similarity between the vectors of distributed trees of each sentence pair. This cosine similarity score is converted to the scale of STS and SR for evaluation.

2.3.3 Syntactic Generalization

Given a pair of parse trees, the Syntactic Generalization (SG) (Galitsky, 2013) finds a set of maximal common subtrees. The toolkit "relevance-based-on-parse-trees" is an open-source project which evaluates text relevance by using syntactic parse tree-based similarity measure.[5] Given a pair of parse trees, it measures the similarity between two sentences by finding a set of maximal common subtrees, using representation of constituency parse trees via chunking. Each type of phrases (NP, VP, PRP etc.) will be aligned and subject to generalization. It uses the OpenNLP system to derive dependency trees for generalization (chunker and parser).[6] This tool is made to give as a tool for text relevance which can be used as a black box, no understanding of computational linguistics or machine learning is required. We apply the tool on the STS datasets to compute the similarity of syntactic structure of sentence pairs.

2.4 Further Features

We also deploy other features which also may help in identifying the semantic similarity degree between two given sentences, such as word alignment in machine translation evaluation metric and the vector space model Weighted Matrix Factorization (WMF) for pairwise similarity.

2.4.1 Machine Translation Evaluation Metric - METEOR

METEOR (Metric for Evaluation of Translation with Explicit ORdering) (Banerjee and Lavie, 2005) is an automatic metric for machine translation evaluation, which consists of two major components: a flexible monolingual word aligner and a scorer. For machine translation evaluation, hypothesis sentences are aligned to reference sentences. Alignments are then scored to produce sentence and corpus level scores. We use this word alignment feature

[4]https://code.google.com/p/distributed-tree-kernels

[5]https://code.google.com/p/relevance-based-on-parse-trees

[6]https://opennlp.apache.org

to learn the similarity between words, phrases in two given texts in case of different orders.

2.4.2 Weighted Matrix Factorization (WMF)

WMF (Guo and Diab, 2012) is a dimension reduction model to extract nuanced and robust latent vectors for short texts/sentences. To overcome the sparsity problem in short texts/sentences (e.g. 10 words on average), the missing words, a feature that LSA/LDA typically overlooks, is explicitly modeled. We use the pipeline to compute the similarity score between texts.

3 Experiment Settings

We generate and select 25 optimal features, ranging from lexical level to string level and syntactic level. We deploy the machine learning toolkit WEKA (Hall et al., 2009) for learning a regression model (*GaussianProcesses*) to predict the similarity scores. We build three models based on three sets of features to verify our hypothesis in which we augment that computing semantic similarity degree is not only about lexical similarity and string similarity, but also taking into consideration a deeper level at syntactic structure where more semantic information is embedded.

In the system development process, we train our system on the given datasets of STS 2012, 2013 and use the STS 2014 datasets for evaluating the system. In Table 1, we also examine the contribution of different features to the overall accuracy of system, and prove that syntactic structure information also has some impact to the performance of our system. Our model using all features described above outperform the best system DLS@CU in STS 2014 evaluation.

We submitted three runs with different sets of features as below:

- **Run1**: All features described in Section 2 used.
- **Run2**: The feature obtained by Distributed Tree Kernel approach is excluded as sometimes it returns negative correlation.
- **Run3**: No syntactic features are included.

4 Evaluations

In Table 2 we report the performance of our three runs achieved on the STS 2015 test datasets. Among three submitted runs, Run1 has the best score, which confirm that exploiting the syntactic structure information benefits the overall performance of our system. Besides, although occasionally the features extracted by Distributed Tree Kernel approach returns negative result, it still contributes a small positive portion in the final result, which is shown in the Run2. In contrast, the Run3 which excludes all syntactic structure features, eventually, returns 4% lower than the other two runs.

In overall, our system achieves a very competitive result compared to the best ranked system, DLS@CU-S1. Specifically, the difference between our Run1 and the DLS@CU-S1 on each test dataset of STS 2015 varies slightly 1%-2%. However, this difference is not statistically significant, as we can understand that each system may perform slightly different on different evaluation datasets. Generally, by taking into account the results of our system and DLS@CU on both STS 2014 and 2015 evaluation datasets, we can consider that we are almost equivalent in performance.

5 Conclusions and Future Work

In this paper, we describe the pipeline system FBK-HLT participating in the SemEval 2015, Task #2 "Semantic Textual Similarity", English subtask. We present a supervised system which considers multiple linguistic features from low to high language level, such as lexical, string and syntactic. We also augment that looking into the syntactic structure of text will more or less benefit the capability of predicting the semantic similarity. Among our three submitted runs, our performance is much above the baseline and very competitive to the best system; we are ranked in the top-tier (12^{th}, 13^{th}, and 23^{nd}) out of total 73 systems.

For the time being, we can see that the contribution of syntactic features is still limited (about 4%) to the overall performance. However, it does not deny the significance of syntactic information in semantic related tasks, especially, this STS task. Hence, we expect to study to exploit more useful features from the syntactic information, which intuitively, is supposed to play a significant role in semantic reasoning.

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*, pages 385–393.

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)*, Denver, CO, June.

Lloyd Allison and Trevor I Dix. 1986. A bit-string longest-common-subsequence algorithm. *Information Processing Letters*, 23(5):305–310.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

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-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 435–440.

Alberto Barrón-Cedeno, Paolo Rosso, Eneko Agirre, and Gorka Labaka. 2010. Plagiarism detection across distant language pairs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 37–45.

Andrei Z Broder. 1997. On the resemblance and containment of documents. In *Compression and Complexity of Sequences 1997*, pages 21–29. IEEE.

Bill Dolan, Chris Brockett, and Chris Quirk. 2005. Microsoft research paraphrase corpus. *Retrieved March*, 29:2008.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *IJCAI*, volume 7, pages 1606–1611.

Boris Galitsky. 2013. Machine learning of syntactic parse trees for search and classification of text. *Engineering Applications of Artificial Intelligence*, 26(3):1072–1091.

Weiwei Guo and Mona Diab. 2012. Modeling sentences in the latent space. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 864–872.

Dan Gusfield. 1997. *Algorithms on strings, trees and sequences: computer science and computational biology*. Cambridge University Press.

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.

Dan Klein and Christopher D Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430.

Caroline Lyon, James Malcolm, and Bob Dickerson. 2001. Detecting short passages of similar text in large document collections. In *Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing*, pages 118–125.

Alessandro Moschitti. 2006. Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329.

Philip Resnik. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453.

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.

Michael J Wise. 1996. Yap3: Improved detection of similarities in computer program and other texts. In *ACM SIGCSE Bulletin*, volume 28, pages 130–134. ACM.

Fabio Massimo Zanzotto and Lorenzo Dell'Arciprete. 2012. Distributed tree kernels. In *Proceedings of the 29th International Conference on Machine Learning*.