FBK-HLT: An Application of Semantic Textual Similarity for Answer Selection in Community Question Answering

Ngoc Phuoc An Vo
University of Trento,
Fondazione Bruno Kessler
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 #3 "Answer Selection in Community Question Answering" for English, for both subtasks. We submit two runs with different classifiers in combining typical features (lexical similarity, string similarity, word n-grams, etc.) with machine translation evaluation metrics and with some ad hoc features (e.g user overlapping, spam filtering). We outperform the baseline system and achieve interesting results on both subtasks.

1 Introduction

Answer selection is an important task inside the wider task of question answering that represents at the moment a topic of great interest for research and business as well. Analyzing social data like answers given inside a forum is a way to maximize the value of this type of knowledge source that is usually affected by a very noisy information due to out of topic spam, double posting, cross posting or other issues. Recognizing useful posts from bad ones, and automatically detecting the main polarity of answers to a given question is a way to treat an amount of data that otherwise might be difficult to handle.

A promising way to provide insight into these questions was brought forward as Shared Task #3 in the SemEval-2015 campaign for "Answer Selection in Community Question Answering" (Màrquez et al., 2015) for English and Arabic languages. In the Subtask A, each system is given a set of questions in which each one contains some data like posting date, author's Id, a set of comments, at least one, but usually more; then the participating the system has to classify comments as *good*, *bad* or *potential* according to their relevance with the question. In Subtask B, a subset of these questions are predefined as *yes/no questions*, system has to classify them into *yes*, *no* or *unsure* classes based on the individual good answers. We participate in this shared task (only in English) with a system composing several different features using a multiclass classifier. We are interested in finding out whether similarity, machine translation evaluation metrics and task specific techniques could increase the accuracy of our system. In this paper, we outline our method and present the results for the answer selection task; the paper is organized as follows: Section 2 presents the System Description, Section 3 describes the Experiment Settings, Section 4 reports the Evaluations, Section 5 is the Error Analysis and finally, Section 6 presents the Conclusions and Future Work.

2 System Description

In order to build our system, we extract and adopt several different linguistic features from a Semantic Textual Similarity (STS) system (Vo et al., 2015) and then consolidate them by a multiclass classifier. Different features can be used independently or together with others to measure the semantic similarity and recognize the paraphrase of a given sentence pair as well as to evaluate the significance of each feature to the accuracy of system's predictions. Hence, the system is expandable and scalable for adopting more useful features aiming for improving the accuracy.

Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), pages 231–235,
Denver, Colorado, June 4-5, 2015.

2.1 Data Preprocessing

As data preprocessing is a crucial step for preparing useful information to be learned by the system, we focus the beginning of our work trying to simplify data without losing information. Our system is based on semantic similarity, so it needs pairs of sentences to compare; we pair-up every question with all of its comments, one by one, e.g. a question with five comments becomes five pairs of sentences composed by the question and five different comments. Questions and comments are composed by subject and body, so for questions, we merge the subject and body together if the subject does not occur inside the body; and for comments, we also check if the comment's subject is not identical to question's subject with the prefix *RE:*. As the forum data also contains lot of informal writing, we normalize them by applying a simple filter that substitutes common abbreviation: *"u - you", "r - are", "ur - your", "Iam - I am", "any1 - anyone"*.

2.2 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.[1] It measures the similarity between two sentences by finding a set of maximal common subtree for a pair of parse trees, 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 constituent trees for generalization (chunker and parser).[2] As it is an unsupervised approach, we apply the tool directly to the preprocessed texts to compute the similarity of syntactic structure of sentence pairs.

2.3 Machine Learning Evaluation Metric - METEOR

We also use evaluation metrics for machine translation as suggested in (Madnani et al., 2012) for paraphrase recognition on Microsoft Research paraphrase corpus (MSRP) (Dolan et al., 2004). In machine translation, the evaluation metric scores the hypotheses by aligning them to one or more reference translations. We take into consideration to use all the eight metrics proposed, but we find that adding some of them without a careful process of training on the dataset may decrease the performance of the system.

We use the latest version of METEOR (Denkowski and Lavie, 2014) that finds alignments between sentences based on exact, stem, synonym and paraphrase matches between words and phrases. We used the system as distributed on its website, using only the "norm" option that tokenizes and normalizes punctuation and lowercase as suggested by documentation.[3] We compute the word alignment scores between questions and comments.

2.4 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 Latent Semantic Analysis (LSA) and Latent Dirichlet Allocation (LDA) typically overlook, is explicitly modeled. We use the pipeline to compute the similarity scores for question-comment pairs.[4]

2.5 User Overlapping

We extract a simple binary feature focused on comment's author. We suppose that question's author is not usually as same as comment's author, so if a question has one or more comments associated with the same question's author, these comments are probably descriptions or explanations about the question. We label 1 for comments made by the same question's author and 0 otherwise.

2.6 Spam Filtering - JFilter

Recognizing good comments from bad comment is a task somehow similar to spam filtering, to capture this feature, we use a Java implementation, Jfilter (Francesco Saverio Profiti, 2007), based on a fuzzy version of the Rocchio algorithm (Rocchio, 1971). This system uses a classifier that needs training, so to avoid overfitting, from the training and development datasets, we randomly choose a subset of *good*

[1] https://code.google.com/p/relevance-based-on-parse-trees/
[2] https://opennlp.apache.org
[3] http://www.cs.cmu.edu/%7Ealavie/METEOR/index.html
[4] http://www.cs.columbia.edu/%7Eweiwei/code.html

	Accuracy	F1 (G)	F1 (B)	F1 (D)	F1 (P)	F1 (NE)	F1 (O)	F1 WM
Baseline	53.19	0.694	0	0	0	0	0	0.369
1-against-all	**60.06**	0.731	0.189	0.545	0	0	0	0.523
Random Correction Code	59.02	0.722	0.319	0.540	0	0	0	**0.539**
Exhausted Correction Code	60.00	0.731	0.18	0.545	0	0	0	0.521

Table 1: Result obtained using different classification algorithms for Subtask A (G good; B bad; D dialog; P potential; NE not-English; O other; WM Weighted Mean) on Development dataset.

	Accuracy	F1 (Yes)	F1 (No)	F1 (Unsure)	F1 (Not-Applicable)	F1 WM
Standard Features	44.4444	0.316	0	0.077	0.593	0.355
Standard Features + Subtask A output	45.4444	0.327	0	0.08	0.589	0.358
Standard Features + Subtask A gold-standard labels	70.7071	0.667	0	0.069	1	0.635

Table 2: Subtask B system performances on Development dataset.

comments to use as non-spam dataset; in contrast, we select a subset of *bad* and *potential* to use as spam dataset to train JFilter. This configuration was used to train our system during development; for the final run with test dataset, we train JFilter with both development and training datasets. JFilter gives a binary judgment (HAM or SPAM) which is used as a feature for our system in Subtask A.

3 Experiment Settings

We use the machine learning toolkit WEKA (Hall et al., 2009) to obtain robust and efficient implementation of different classifiers, as well as to reduce develop time of the system. For Subtask A, we build one model using all the features described in Section 2. Table 1 reports some experiments in which we select a good classifier to optimize both the Accuracy and F1-score of the system. During the development, we select the default implementation "1-against-all" classification algorithm (with logistic regression) for both subtasks.

For Subtask B, we make some modifications to the system due to some important differences between two subtasks. As the question classification depends on the quality of its comments, we substitute the spam filtering feature by the comments' labels from Subtask A system's output. In order to examine this hypothesis, we firstly use the gold-standard labels of comments from Subtask A as a feature for the question classification in Subtask B. The high Accuracy and F1-score from this setting proves our hypothesis correct. To avoid the overfitting, we again use only the label predictions from Subtask A as a feature for our Subtask B system. Table 2 shows that a precise output from Subtask A can significantly benefit the performance of Subtask B system.

As Subtask B does not focus on comment labeling, but question labeling, to achieve this purpose after classifying all comments as *yes*, *no*, *unsure* or *Not Applicable*, we simply aggregate comments of every question with a majority vote. We label a question as *yes* if the majority of its comments are classified as *yes*, the same for *no*; if there no major judgment of either *yes* or *no*, the question is classified as *unsure*.

	Subtask A		Subtask B	
Team	Mac F1	Acc	Mac F1	Acc
JAIST	57.19	72.52		
VectorSlu			63.7	72.0
FBK-HLT	47.32	69.13	27.8	40.0

Table 3: Evaluation Results on Subtasks A and B.

Team	Accuracy	F1 (G)	F1 (B)	F1 (D)	F1 (P)	Macro F1
JAIST (3-classes)	72.67	79.11	78.29	0	14.48	57.29
HLT-FBK (3-classes)	69.13	75.80	66.15	0	0	47.32
JAIST (4-classes)	59.62	76.52	40.38	57.21	18.41	48.13
HLT-FBK (4-classes)	62.40	75.80	43.42	51.23	0	42.61

Table 4: Subtask A - Comparison with best system for 3-classes and 4-classes evaluation (G good; B bad; D dialog; P potential; Macro F1).

Team	Accuracy	F1 (Yes)	F1 (No)	F1 (Unsure)	Macro F1
VectorSlu	72.0	83.87	57.14	50.0	63.67
FBK-HLT	40.0	50.0	0.0	33.33	27.78

Table 5: Subtask B - Comparison with best system.

4 Evaluations

We submit only one run for both subtasks (English language) using the "1-against-all" classification algorithms. In Subtask A, we achieve good results, especially, we are ranked 4th out of 12 teams in Accuracy. In Subtask B, as we only apply the simple approach "majority vote", the result is reasonable as expected. Table 3 shows our performance in both subtasks in regard to the best systems, both in Macro F1 and Accuracy measures.

5 Error Analysis

In this section, we conduct an analysis of our system's performance on test dataset. In Subtask A, our analysis consists of some comparison between our system and the best system, JAIST. According to results in Table 4, for the evaluation on 3-classes (*good*, *bad*, and *potential*), our system is dramatically penalized by low performance on detecting *bad* comments, besides, it is not able to classify the *potential* ones. This particular class of comments is very small in training dataset. There are 50.45% for *good* comments, 41.09% for *bad* and only 8.25% for *potential*. During the development, as we decide to optimize the Accuracy and F1 weighted on the number of comments, this decision misleads our system to ignore this small class. Hence, in order to improve the system performance, we may need to search for a specific feature for *potential* comments like what we did with user overlapping for *dialog* ones. For the evaluation on 4-classes (*good*, *bad*, *dialog* and *potential*), our system performance rises significantly, our system shows a good capability to distinguish between *dialog* and other comments.

In Subtask B, the performance comparison in Table 5 shows that our system achieves reasonable performance on the *Yes* and *Unsure* classes, but has no capability to capture the *No* class. Moreover, most of the instances of *No* class have been misclassified as *Unsure* class. This shows an unclear separation between these two classes which confuses the system. Thus, to fix this issue, we need to find more specific features which may help to distinguish the *No* class and others.

6 Conclusions and Future Work

In this paper, we describe our system participating in the SemEval 2015, Task #3 "Answer Selection in Community Question Answering" in English, for both subtasks. We present a supervised system which considers multiple linguistic features such as lexical, string and some task-specific features. Our performance is much above the baseline and shows some interesting properties in specific scenarios. We also show some error analysis in which we investigate the limit and drawback of our system on specific comment and question classes.

For future work, we expect to study to exploit more useful features, especially, task-related features, to improve the classification performance on *potential* labeled comments and *No* labeled questions, which will lead to a significant improvement of the overall performance.

References

Michael Denkowski and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350.

Claudio Biancalana Francesco Saverio Profiti. 2007. Jfilter: un filtro antispam intelligente in java. *Mokabyte*, (124). in Italian.

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.

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.

Nitin Madnani, Joel Tetreault, and Martin Chodorow. 2012. Re-examining machine translation metrics for paraphrase identification. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 182–190.

Lluís Màrquez, James Glass, Walid Magdy, Alessandro Moschitti, Preslav Nakov, and Bilal Randeree. 2015. Semeval-2015 task 3: Answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*.

Joseph John Rocchio. 1971. Relevance feedback in information retrieval.

Ngoc Phuoc An Vo, Simone Magnolini, and Octavian Popescu. 2015. FBK-HLT: A new framework for semantic textual similarity. *In Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015), Denver, US*.