Tackling the OnForumS Challenge

Hristo Tanev
Joint Research Centre, EC
Ispra, Italy
htanev@gmail.com

Alexandra Balahur
Joint Research Centre, EC
Ispra, Italy
alexandra.balahur@jrc.ec.europa.eu

Abstract

In this paper we describe the participation of the Joint Research Centre in the OnForumS shared task. We experimented with distributional similarity to measure the distance between comments and sentences from the corresponding articles. Using distributional term similarity for English resulted in a small improvent with respect to the lexical-based baseline (0.03). For detecting sentiment and attitude we used statistical ngram models, which gave relatively good performance in both languages - English and Italian.

1 Introduction

The shared task OnForumS was quite challenging and on the other hand this task has practical importance: Finding the link between a comment and the sentence to which it refers, as well as the sentiment and the attitude of the comment, can be used in different applications - summarization of discussions, navigation inside them, etc.

The problem of finding a semantic relation betweet text fragments is similar to the textual entailment and question-answering tasks. Therefore, finding a viable solution to the OnForumS task may inspire algorithms in other spheres of NLP. Sentiment detection in online fora is important, since it can highlight certain comments.

In our participation we carried out comment-sentence linking and sentiment analysis. We did not implement a special algorithm for the detection of attitude (argument), but we rather used the output of the sentiment analysis.

For linking the comments to the article sentences we used two algorithms for the English language: a baseline one which finds the cosine similarity between the comments and the sentences and a more elaborated one which uses distributional similarity and co-occurrence between words. For Italian we used the baseline algorithm only, since we had no corpus from which to estimate the word similarity. The distributional similarity gave improvement of 0.03 with respect to the baseline.

For sentiment analysis we used statistical multilingual ngram models.

For the English language our system ranked 2nd in the comment linking task and it was 3rd in sentiment and argument detection. For Italian our performence was lower for linking and argument detection, however we were 2nd in sentiment detection.

2 Related Work

Sentence based similarity was addressed in different works. For example, (Li et al., 2006) uses semantic information and word order to find similarity of sentences. Semantic similarity between words was exploited in a sentence-similarity algorithm, described in (Islam and Inkpen, 2008). Short text similarity was discussed also in the context of textual entailment. For example (Wang and Neumann, 2007) recognizes textual entaiment by considering similarity between sentences. A survey of text similarity approaches is presented in (Gomaa and Fahmy, 2013).

There are different methods for sentiment analysis: statistical and lexicon-based ones. A survey of sentiment-analysis algorithms is presented in (Liu and Zhang, 2012).

Related to our work are the different papers which address the analysis of online activity.Among the others, the work presented in (Hassan et al., 2012) analyses the polarity of the relations between the participants in online groups and (Dasigi et al., 2012) presents detection of attitude in online fora.

3 Linking Comments to Sentences from News Article

The subtask of linking comments to news article sentences was of a particular interest for us, since we are currently experimenting with similarity measures between short texts. We used two methods to find the similarity between a comment and an article sentence: the baseline one uses classical lexical based cosine similarity, while the more complex method exploits distributional similarity and term co-occurrence statistics between words. For each comment we find the sentence in the article which is most similar to it and if the similarity is above a certain threshold, defined empirically, we link the comment to that sentence.

The cosine similarity baseline is defined in the following way:

$$sim(v_1, v_2) = \frac{\sum_{t \in v1 \bigcap v2} idf(t).idf(t)}{\|v1\| \, \|v2\|}$$

where v_1 and v_2 are the term vectors of the two sentences.

3.1 Exploiting similarity between terms

The problem with the lexical similarity, introd measure is that it cannot account for semantic similarities between different terms. For example, the phrases *expert in computer science* and *specialist in information technology* have no common terms, but in practice they are synonyms. In order to address this phenomena, we exploited an efficient algorithm for calculation of distributional similarity between pairs of terms, as well as term co-occurrences. The algorithm is based on efficient association network of words, presented in (Tanev, 2014).

We defined the following ad-hoc similarity measure to find the similarity between two term vectors v_1 and v_2:

$$sim(v_1, v_2) = \frac{sim_1(v_1, v_2) + sim_1(v_2, v_1)}{\sum_{t1 \in v1} idf(t1) + \sum_{t2 \in v2} idf(t2)}$$

$$sim_1(v_1, v_2) = \sum_{t_1 \in v1} idf(t_1).vsim(t_1, v_2)$$
$$vsim(t_1, v_2) = Min(\sum_{t2 \in v2} sim(t_1, t_2), 1)$$

Similarity between terms t_1 and t_2 is measured, taking into account their co-occurrence and their distributional similarity, as it is described in (Tanev, 2014). More precisely:

$$sim(t_1, t_2) = cooc(t_1, t_2)/4 + distsim(t_1, t_2)$$

$$cooc(t_1, t_2) = \frac{p(t_1, t_2)}{p(t_1).p(t_2)}$$

where $p(t_1, t_2)$ is the probability that t_1 and t_2 co-occur and $p(t)$ is the probability to find t in the text.

Distributional similarity $distsim(t_1, t_2)$ takes into account the common ajacent words to t_1 and t_2. Distributionally similar words are usually synonyms or semantically related ones. They tend to appear in similar contexts. More formally, $distsim(t_1, t_2)$ value is calculated as the cosine similarity between the context feature vectors of t_1 and t_2. Each context feature of a term t is a term adjacent in the training corpus to t, the position (left or right) w.r.t t, as well as the stop words between these terms. For example, one of the most outstanding feature of car is $driving - a \rightarrow X$. This is also a good feature for *truck, bus, lorry, Toyota*, etc. Features are scored, on the basis of their co-occurrence with t. If the cosine similarity between the context feature vectors of two terms is high, this means that they will appear in similar contexts and according to the Harris' distributional hypothesis, their semantics will be similar.

In order to calculate co-occurrence and distributional similarity in real time, we use an association network of words, extracted from a training corpus of news articles. The association network is a graph in which two words are connected, if they are adjacent in the training corpus. Each edge is labeled with the stop-words which appear between co-occurring words, as well as with the frequency of this co-occurrence. More detailed description is provided in (Tanev, 2014).

4 Sentiment Analysis

Our sentiment analysis system is based on a hybrid approach, which employs supervised learning with a Support Vector Machines Sequential Minimal Optimization linear kernel, on unigram and bigram features, but exploiting as features sentiment dictionaries, emoticon lists, slang lists and other features specific for fora and social media. We do not employ any specific language analysis software. The aim is to be able to apply, in a straightforward manner, the same approach to as many languages as possible.

The sentiment analysis process contains two stages: pre-processing and sentiment classifica-

tion. Pre-processing involves tokenization, normalization of language (only done for English) and the addressing of special signals of emotion in informal texts emoticons, punctuation signs, and capitalization (which are marked correspondingly). Once the text is pre-processed, it is passed on to the sentiment classification module.

For the sentiment classification, we employ supervised learning using the Support Vector Machines Sequential Minimal Optimization implementation(Platt and others, 1999) in Weka, with a linear kernel, based on boolean features - the presence or absence of n-grams (unigrams, bigrams and unigrams plus bigrams) determined from the training data (tweets that were previously pre-processed as described above). Bigrams are used specifically to spot the influence of modifiers (negations, intensifiers, diminishers) on the polarity of the sentiment-bearing words. This approach was successfully employed for English and although for other languages other (additional or slightly different) features might be useful to be included, at this point we employ the same approach for all the languages considered.

The multilingual model is created from SemEval 2013 Task 2 Sentiment Analysis in Tweets (Wilson et al., 2013).

4.1 Detecting the comments' attitudes

We did not have a specific algorithm to estimate the comment's attitude (argument). We rather used the detected sentiment polarity of the comment: If the comment's sentiment was detected to be positive, we marked the comment as being *in favor*. If the sentiment was negative, the comment was assumed to be *against*. Otherwise, the comment was considered to be *impartial*.

5 Experiments and discussion

We have participated in the shared task both for English and Italian. The obtained results are promising, but still they can be improved. Regarding the linking between comments and sentences, for English we submitted 2 different runs - in the first we used term similarity, in the second, we used the baseline vector similarity. The first run turned out to give better results, which proves the suitability of our similarity measure. Regarding the sentiment abnalysis, we ranked 3rd in argument and sentiment analysis detection in English and 2nd in sentiment detection in Italian. This shows that our sentiment detection models work well across languages.

In our future work we intend to experiment with different text similarity measures and heuristics for connecting comments to sentences.

The lessons learned from the OnForumS shared task teach us how to analyse short texts such as social media messages. Similarity algorithms which were developed for this task can be used to link and cluster tweets, Facebook messages and comments

References

Pradeep Dasigi, Weiwei Guo, and Mona Diab. 2012. Genre independent subgroup detection in online discussion threads: a pilot study of implicit attitude using latent textual semantics. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 65–69. Association for Computational Linguistics.

Wael H Gomaa and Aly A Fahmy. 2013. A survey of text similarity approaches. *International Journal of Computer Applications*, 68(13):13–18.

Ahmed Hassan, Amjad Abu-Jbara, and Dragomir Radev. 2012. Detecting subgroups in online discussions by modeling positive and negative relations among participants. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 59–70. Association for Computational Linguistics.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(2):10.

Yuhua Li, David McLean, Zuhair A Bandar, James D O'shea, and Keeley Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *Knowledge and Data Engineering, IEEE Transactions on*, 18(8):1138–1150.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In *Mining text data*, pages 415–463. Springer.

John Platt et al. 1999. Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methodssupport vector learning*, 3.

Hristo Tanev. 2014. Learning textologies: Networks of linked word clusters. In *Text Mining*, pages 25–40. Springer.

Rui Wang and Günter Neumann. 2007. Recognizing textual entailment using sentence similarity based on dependency tree skeletons. In *Proceedings of the*

ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, pages 36–41. Association for Computational Linguistics.

Theresa Wilson, Preslav Nakov, Zornitsa Kozareva, Alan Ritter, Sara Rosenthal, and Veselin Stoyanov. 2013. Semeval-2013 task 2: Sentiment analysis in twitter.