Blinov: Distributed Representations of Words for Aspect-Based Sentiment Analysis at SemEval 2014

Pavel Blinov, Eugeny Kotelnikov
Vyatka State Humanities University
{blinoff.pavel, kotelnikov.ev}@gmail.com

Abstract

The article describes our system submitted to the SemEval-2014 task on Aspect-Based Sentiment Analysis. The methods based on distributed representations of words for the aspect term extraction and aspect term polarity detection tasks are presented. The methods for the aspect category detection and category polarity detection tasks are presented as well. Well-known skip-gram model for constructing the distributed representations is briefly described. The results of our methods are shown in comparison with the baseline and the best result.

1 Introduction

The sentiment analysis became an important Natural Language Processing (NLP) task in the recent few years. As many NLP tasks it's a challenging one. The sentiment analysis can be very helpful for some practical applications. For example, it allows to study the users' opinions about a product automatically.

Many research has been devoted to the general sentiment analysis (Pang et al., 2002), (Amine et al., 2013), (Blinov et al., 2013) or analysis of individual sentences (Yu and Hatzivassiloglou, 2003), (Kim and Hovy, 2004), (Wiebe and Riloff, 2005). Soon it became clear that the sentiment analysis on the level of a whole text or even sentences is too coarse. General sentiment analysis by its design is not capable to perform the detailed analysis of an expressed opinion. For example, it cannot correctly detect the opinion in the sentence "*Great food but the service was dreadful!*". The sentence carries opposite opinions on two facets of a restaurant. Therefore the more detailed version of the sentiment analysis is needed. Such a version is called the aspect-based sentiment analysis and it works on the level of the significant aspects of the target entity (Liu, 2012).

The aspect-based sentiment analysis includes two main subtasks: the aspect term extraction and its polarity detection (Liu, 2012). In this article we describe the methods which address both subtasks. The methods are based on the distributed representations of words. Such word representations (or word embeddings) are useful in many NLP task, e.g. (Turian et al., 2009), (Al-Rfou' et al., 2013), (Turney, 2013).

The remainder of the article is as follows: section two gives the overview of the data; the third section shortly describes the distributed representations of words. The methods of the aspect term extraction and polarity detection are presented in the fourth and the fifth sections respectively. The conclusions are given in the sixth section.

2 The Data

The organisers provided the train data for restaurant and laptop domains. But as it will be clear further our methods are heavily dependent on unlabelled text data. So we additionally collected the user reviews about restaurants from *tripadviser.com* and about laptops from *amazon.com*. General statistics of the data are shown in Table 1.

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 140–144,
Dublin, Ireland, August 23-24, 2014.

Table 1: The amount of reviews.

Domain	The amount of reviews
Restaurants	652 055
Laptops	109 550

For all the data we performed tokenization, stemming and morphological analysis using the FreeLing library (Padró and Stanilovsky, 2012).

3 Distributed Representations of Words

In this section we'll try to give the high level idea of the distributed representations of words. The more technical details can be found in (Mikolov et al., 2013).

It is closely related with a new promising direction in machine learning called the deep learning. The core idea of the unsupervised deep learning algorithms is to find automatically the "good" set of features to represent the target object (text, image, audio signal, etc.). The object represented by the vector of real numbers is called *the distributed representation* (Rumelhart et al., 1986). We used the skip-gram model (Mikolov et al., 2013) implemented in Gensim toolkit (Řehůřek and Sojka, 2010).

In general the learning procedure is as follows. All the texts of the corpus are stuck together in a single sequence of sentences. On the basis of the corpus the lexicon is constructed. Next, the dimensionality of the vectors is chosen (we used 300 in our experiments). The greater number of dimensions allows to capture more language regularities but leads to more computational complexity of the learning. Each word from the lexicon is associated with the real numbers vector of the selected dimensionality. Originally all the vectors are randomly initialized. During the learning procedure the algorithm "slides" with the fixed size window (it's algorithm parameter that was retained by default – 5 words) along the words of the sequence and calculates the probability (1) of context words appearance within the window based on its central word under review (or more precisely, its vector representation) (Mikolov et al., 2013).

$$p(w_O \mid w_I) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W} \exp({v'_w}^T v_w)}, \qquad (1)$$

where v_w and v'_w are the input and output vector representations of *w*; w_I and w_O are the current and predicted words, W – the number of words in vocabulary.

The ultimate goal of the described process is to get such "good" vectors for each word, which allow to predict its probable context. All such vectors together form the vector space where semantically similar words are grouped.

4 Aspect Term Extraction Method

We apply the same method for the aspect term extraction task (Pontiki et al., 2014) for both domains. The method consists of two steps: the candidate selection and the term extraction.

4.1 Candidate Selection

First of all we collect some statistics about the terms in the train collection. We analysed two facets of the aspect terms: the number of words and their morphological structure. The information about the number of words in a term is shown in Table 2.

Table 2: The statistics for the number of words in a term.

	Domain	
Aspect term	**Restaurant, %**	**Laptop, %**
One-word	72.13	55.66
Two-word	19.05	32.87
Greater	8.82	11.47

On the basis of that we've decided to process only single and two-word aspect terms. From the single terms we treat only singular (NN, e.g. *staff, rice, texture, processor, ram, insult*) and plural nouns (NNS, e.g. *perks, bagels, times, dvds, buttons, pictures*) as possible candidates, because they largely predominate among the one-word terms. All conjunctions of the form NN_NN (e.g. *sea_bass, lotus_leaf, chicken_dish, battery_life, virus_protection, customer_disservice*) and NN_NNS (e.g. *sushi_places, menu_choices, seafood_lovers, usb_devices, recovery_discs, software_works*) were candidates for the two-word terms also because they are most common in two-word aspect terms.

4.2 Term Extraction

The second step for the aspect term identification is the term extraction. As has already been told the space (see Section 3) specifies the word groups. Therefore the measure of similarity between the words (vectors) can be defined. For NLP tasks it is often the cosine similarity measure. The similarity between two vectors $\vec{a} = (a_1,...,a_n)$ and $\vec{b} = (b_1,...,b_n)$ is given by (Manning et al., 2008):

$$\cos(\theta) = \frac{\sum_{i=1}^{n} a_i b_i}{\sqrt{\sum_{i=1}^{n} a_i^2} \cdot \sqrt{\sum_{i=1}^{n} b_i^2}}, \qquad (2)$$

where θ – the angle between the vectors, n – the dimensionality of the space.

In case of the restaurant domain the category and aspect terms are specified. For each category the seed of the aspect terms can be automatically selected: if only one category is assigned for a train sentence then all its terms belong to it. Within each set the average similarity between the terms (the threshold category) can be found.

For the new candidate the average similarities with the category's seeds are calculated. If it is greater than the threshold of any category than the candidate is marked as an aspect term.

Also we've additionally applied some rules:

- Join consecutive terms in a single term.
- Join neutral adjective ahead the term (see Section 5.2 for clarification about the neutral adjective).
- Join fragments matching the pattern: <an aspect term> of <an aspect term>.

In case of the laptop domain there are no specified categories so we treated all terms as the terms belonging to one general category. And the same procedure with candidates was performed.

4.3 Category Detection

For the restaurant domain there was also the aspect category detection task (Pontiki et al., 2014).

Since each word is represented by a vector, each sentence can be cast to a single point as the average of its vectors. Further average point for each category can be found by means of the sentence points. Then for an unseen sentence the average point of its word vectors is calculated. The category is selected by calculating the distances between all category points and a new point and by choosing the minimum distance.

4.4 Results

The aspect term extraction and the aspect category detection tasks were evaluated with Precision, Recall and F-measure (Pontiki et al., 2014). The F-measure was a primary metric for these tasks so we present only it.

The result of our method ranked 19 out of 28 submissions (constrained and unconstrained) for the aspect term extraction task for the laptop domain and 17 out of 29 for the restaurant domain. For the category detection task (restaurant domain) the method ranked 9 out of 21.

Table 3 shows the results of our method (**Bold**) for aspect term extraction task in comparison with the baseline (Pontiki et al., 2014) and the best result. Analogically the results for the aspect category detection task are presented in Table 4.

Table 3: Aspect term extraction results (F-measure).

	Laptop	**Restaurant**
Best	0.7455	0.8401
Blinov	**0.5207**	**0.7121**
Baseline	0.3564	0.4715

Table 4: Aspect category detection results (F-measure).

	Restaurant
Best	0.8858
Blinov	**0.7527**
Baseline	0.6389

5 Polarity Detection Method

Our polarity detection method also exploits the vector space (from Section 3) because the emotional similarity between words can be traced in it. As with the aspect term extraction method we follow two-stage approach: the candidate selection and the polarity detection.

5.1 Candidate Selection

All adjectives and verbs are considered as the polarity term candidates. The amplifiers and the negations have an important role in the process of result polarity forming. In our method we took into account only negations because it strongly affects the word polarity. We've joined into one unit all text fragments that match the following pattern: not + <JJ | VB>.

5.2 Term Polarity Detection

At first we manually collected the small etalon sets of positive and negative words for each domain. Every set contained 15 words that clearly identify the sentiment. For example, for the positive polarity there were words such as: great, fast, attentive, yummy, etc. and for the negative polarity there were words like: terrible, ugly, not_work, offensive, etc.

By measuring the average similarity for a candidate to the positive and the negative seed words we decided whether it is positive (+1) or negative (–1). Also we set up a neutral threshold and a candidate's polarity was treated as neutral (0) if it didn't exceed the threshold.

For each term (within the window of 6 words) we were looking for its closest polarity term candidate and sum up their polarities. For the final decision about the term's polarity there were some conditions:

- **If** sum > 0 **then** *positive*.
- **If** sum < 0 **then** *negative*.
- **If** sum == 0 **and** all polarity terms are neutral **then** *neutral* **else** *conflict*.

5.3 Category Polarity Detection

By analogy with the category detection method, using the train collection, we calculate the average polarity points for each category, i.e. there were 5×4 such points (5 categories and 4 values of polarity). Then a sentence was cast to a point as the average of all its word-vectors. And closest polarity points for the specified categories defined the polarity.

5.4 Results

The results of our method (**Bold**) for the polarity detection tasks are around the baseline results for the Accuracy measure (Tables 5, 6).

Table 5: Aspect term polarity detection results (Accuracy).

	Laptop	Restaurant
Best	0.7049	0.8095
Blinov	**0.5229**	**0.6358**
Baseline	0.5107	0.6428

Table 6: Category polarity detection results (Accuracy).

	Restaurant
Best	0.8293
Blinov	**0.6566**
Baseline	0.6566

However the test data is skewed to the positive class and for that case the Accuracy is a poor indicator. Because of that we also show macro F-measure results for our and baseline methods (Tables 7, 8).

Table 7: Aspect term polarity detection results (F-measure).

	Laptop	Restaurant
Blinov	**0.3738**	**0.4334**
Baseline	0.2567	0.2989

Table 8: Category polarity detection results (F-measure).

	Restaurant
Blinov	**0.5051**
Baseline	0.3597

From that we can conclude that our method of the polarity detection more delicately deals with the minor represented classes than the baseline method.

6 Conclusion

In the article we presented the methods for two main subtasks for aspect-based sentiment analysis: the aspect term extraction and the polarity detection. The methods are based on the distributed representation of words and the notion of similarities between the words.

For the aspect term extraction and category detection tasks we get satisfied results which are consistent with our cross-validation metrics. Unfortunately for the polarity detection tasks the result of our method by official metrics are low. But we showed that the proposed method is not so bad and is capable to deal with the skewed data better than the baseline method.

Acknowledgments

We would like to thank the organizers and the reviewers for their efforts.

Reference

Abdelmalek Amine, Reda Mohamed Hamou and Michel Simonet. 2013. Detecting Opinions in Tweets. *International Journal of Data Mining and Emerging Technologies*, 3(1):23–32.

Christopher Manning, Prabhakar Raghavan and Hinrich Schütze. 2008. Introduction to Information Retrieval. *Cambridge University Press*, New York, NY, USA.

Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 79–86.

Bing Liu. 2012. Sentiment Analysis and Opinion Mining. *Synthesis Lectures on Human Language Technologies*.

Pavel Blinov, Maria Klekovkina, Eugeny Kotelnikov and Oleg Pestov. 2013. Research of lexical approach and machine learning methods for sentiment analysis. *Computational Linguistics and Intellectual Technologies*, 2(12):48–58.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 486–497.

Joseph Turian, Lev Ratinov, Yoshua Bengio and Dan Roth. 2009. A preliminary evaluation of word representations for named-entity recognition. In *Proceedings of NIPS Workshop on Grammar Induction, Representation of Language and Language Learning*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado and Jeffrey Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Lluís Padró and Evgeny Stanilovsky. 2012. FreeLing 3.0: Towards Wider Multilinguality. In *Proceedings of the Language Resources and Evaluation Conference*, LREC 2012, pages 2473–2479.

Soo-Min Kim and Eduard Hovy. 2004. Determining the sentiment of opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING-2004.

Maria Pontiki, Dimitrios Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation*, SemEval 2014, Dublin, Ireland.

Peter Turney. 2013. Distributional semantics beyond words: Supervised learning of analogy and paraphrase. *Transactions of the Association for Computational Linguistics*, 1:353-366.

Radim Řehůřek and Petr Sojka. 2010. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 46–50.

Rami Al-Rfou’, Bryan Perozzi, Steven Skiena. 2013. Polyglot: Distributed Word Representations for Multilingual NLP. In *Proceedings of Conference on Computational Natural Language Learning*, CoNLL’2013.

David Rumelhart, Geoffrey Hintont, Ronald Williams. 1986. Learning representations by backpropagating errors. *Nature*.

Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136.