

Twitter Based System: Using Twitter for Disambiguating Sentiment Ambiguous Adjectives

Alexander Pak, Patrick Paroubek

Université de Paris-Sud,
Laboratoire LIMSI-CNRS, Bâtiment 508,
F-91405 Orsay Cedex, France
alexpak@limsi.fr, pap@limsi.fr

Abstract

In this paper, we describe our system which participated in the SemEval 2010 task of disambiguating sentiment ambiguous adjectives for Chinese. Our system uses text messages from Twitter, a popular microblogging platform, for building a dataset of emotional texts. Using the built dataset, the system classifies the meaning of adjectives into positive or negative sentiment polarity according to the given context. Our approach is fully automatic. It does not require any additional hand-built language resources and it is language independent.

1 Introduction

The dataset of the SemEval task (Wu and Jin, 2010) consists of short texts in Chinese containing target adjectives whose sentiments need to be disambiguated in the given contexts. Those adjectives are: 大 big, 小 small, 多 many, 少 few, 高 high, 低 low, 厚 thick, 薄 thin, 深 deep, shallow, 重 heavy, light, 巨大 huge, 重大 grave.

Disambiguating sentiment ambiguous adjectives is a challenging task for NLP. Previous studies were mostly focused on word sense disambiguation rather than sentiment disambiguation. Although both problems look similar, the latter is more challenging in our opinion because impregnated with more subjectivity. In order to solve the task, one has to deal not only with the semantics of the context, but also with the psychological aspects of human perception of emotions from the written text.

In our approach, we use Twitter¹ microblogging platform to retrieve emotional messages and form two sets of texts: messages with positive emotions and those with negative ones (Pak and Paroubek,

¹<http://twitter.com>

2010). We use emoticons² as indicators of an emotion (Read, 2005) to automatically classify texts into positive or negative sets. The reason we use Twitter is because it allows us to collect the data with minimal supervision efforts. It provides an API³ which makes the data retrieval process much more easier than Web based search or other resources.

After the dataset of emotional texts has been obtained, we build a classifier based on n-grams Naïve Bayes approach. We tested two approaches to build a sentiment classifier:

1. In the first one, we collected Chinese texts from Twitter and used them to train a classifier to annotate the test dataset.
2. In the second one, we used machine translator to translate the dataset from Chinese to English and annotated it using collected English texts from Twitter as the training data.

We have made the second approach because we were able to collect much more of English texts from Twitter than Chinese ones and we wanted to test the impact of machine translation on the performance of our classifier. We have experimented with Google Translate and Yahoo Babelfish⁴. Google Translate yielded better results.

2 Related work

In (Yang et al., 2007), the authors use web-blogs to construct a corpora for sentiment analysis and use emotion icons assigned to blog posts as indicators of users' mood. The authors applied SVM and CRF learners to classify sentiments at the sentence level and then investigated several strategies to determine the overall sentiment of the document. As

²An emoticon is a textual representation of an author's emotion often used in Internet blogs and textual chats

³<http://dev.twitter.com/doc/get/search>

⁴<http://babelfish.yahoo.com/>

the result, the winning strategy is defined by considering the sentiment of the last sentence of the document as the sentiment at the document level.

J. Read in (Read, 2005) used emoticons such as “:-)” and “:-(” to form a training set for the sentiment classification. For this purpose, the author collected texts containing emoticons from Usenet newsgroups. The dataset was divided into “positive” (texts with happy emoticons) and “negative” (texts with sad or angry emoticons) samples. Emoticons-trained classifiers: SVM and Naïve Bayes, were able to obtain up to 70% accuracy on the test set.

In (Go et al., 2009), authors used Twitter to collect training data and then to perform a sentiment search. The approach is similar to the one in (Read, 2005). The authors construct corpora by using emoticons to obtain “positive” and “negative” samples, and then use various classifiers. The best result was obtained by the Naïve Bayes classifier with a mutual information measure for feature selection. The authors were able to obtain up to 84% of accuracy on their test set. However, the method showed a bad performance with three classes (“negative”, “positive” and “neutral”).

In our system, we use a similar idea as in (Go et al., 2009), however, we improve it by using a combination of unigrams, bigrams and trigrams (Go et al., 2009) used only unigrams). We also handle negations by attaching a negation particle to adjacent words when forming ngrams.

3 Our method

3.1 Corpus collection

Using Twitter API we collected a corpus of text posts and formed a dataset of two classes: positive sentiments and negative sentiments. We queried Twitter for two types of emoticons considering eastern and western types of emoticons⁵:

- Happy emoticons: :-), :), ^_^, ^o^, etc.
- Sad emoticons: :-), :(, T_T, ;-;, etc.

We were able to obtain 10,000 Twitter posts in Chinese, and 300,000 posts in English evenly split between negative and positive classes.

The collected texts were processed as follows to obtain a set of n-grams:

1. Filtering – we remove URL links (e.g. <http://example.com>), Twitter user names (e.g.

⁵http://en.wikipedia.org/wiki/Emoticon#Asian_style

@alex – with symbol @ indicating a user name), Twitter special words (such as “RT”⁶), and emoticons.

2. Tokenization – we segment text by splitting it by spaces and punctuation marks, and form a bag of words. For English, we kept short forms as a single word: “don’t”, “I’ll”, “she’d”.
3. Stopwords removal – in English, texts we removed articles (“a”, “an”, “the”) from the bag of words.
4. N-grams construction – we make a set of n-grams out of consecutive words.

A negation particle is attached to a word which precedes it and follows it. For example, a sentence “I do not like fish” will form three bigrams: “I do+not”, “do+not like”, “not+like fish”. Such a procedure improves the accuracy of the classification since the negation plays a special role in opinion and sentiment expression (Wilson et al., 2005). In English, we used negative particles ‘no’ and ‘not’. In Chinese, we used the following particles:

1. 不 – is not + noun
2. 未 – does not + verb, will not + verb
3. 莫 (別) – do not (imperative)
4. 無 (沒有) – does not have

3.2 Classifier

We build a sentiment classifier using the multinomial Naïve Bayes classifier which is based on Bayes’ theorem.

$$P(s|M) = \frac{P(s) \cdot P(M|s)}{P(M)} \quad (1)$$

where s is a sentiment, M is a text. We assume that a target adjective has the same sentiment polarity as the whole text, because in general the lengths of the given texts are small.

Since we have sets of equal number of positive and negative messages, we simplify the equation:

$$P(s|M) = \frac{P(M|s)}{P(M)} \quad (2)$$

⁶An abbreviation for retweet, which means citation or reposting of a message

$$P(s|M) \sim P(M|s) \quad (3)$$

We train Bayes classifiers which use a presence of an n-grams as a binary feature. We have experimented with unigrams, bigrams, and trigrams. Pang et al. (Pang et al., 2002) reported that unigrams outperform bigrams when doing sentiment classification of movie reviews, but Dave et al. (Dave et al., 2003) have obtained contrary results: bigrams and trigrams worked better for the product-review polarity classification. We tried to determine the best settings for our microblogging data. On the one hand high-order n-grams, such as trigrams, should capture patterns of sentiments expressions better. On the other hand, unigrams should provide a good coverage of the data. Therefore we combine three classifiers that are based on different n-gram orders (unigrams, bigrams and trigrams). We make an assumption of conditional independence of n-gram for the calculation simplicity:

$$P(s|M) \sim P(G1|s) \cdot P(G2|s) \cdot P(G3|s) \quad (4)$$

where $G1$ is a set of unigrams representing the message, $G2$ is a set of bigrams, and $G3$ is a set of trigrams. We assume that n-grams are conditionally independent:

$$P(Gn|s) = \prod_{g \in Gn} P(g|s) \quad (5)$$

Where Gn is a set of n-grams of order n .

$$P(s|M) \sim \prod_{g \in G1} P(g|s) \cdot \prod_{g \in G2} P(g|s) \cdot \prod_{g \in G3} P(g|s) \quad (6)$$

Finally, we calculate a log-likelihood of each sentiment:

$$L(s|M) = \sum_{g \in G1} \log(P(g|s)) + \sum_{g \in G2} \log(P(g|s)) + \sum_{g \in G3} \log(P(g|s)) \quad (7)$$

In order to improve the accuracy, we changed the size of the context window, i.e. the number of words before and after the target adjective used for classification.

4 Experiments and Results

In our experiments, we used two datasets: a trial dataset containing 100 sentences in Chinese and

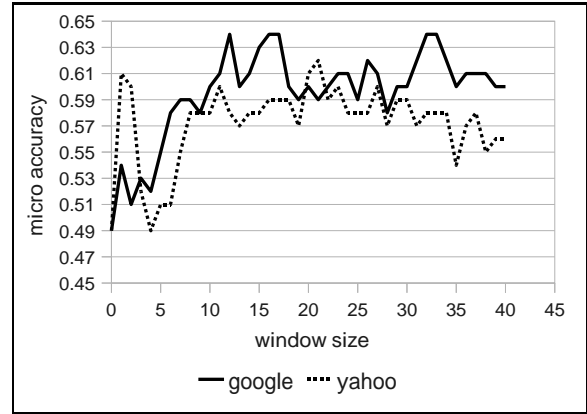


Figure 1: Micro accuracy when using Google Translate and Yahoo Babelfish

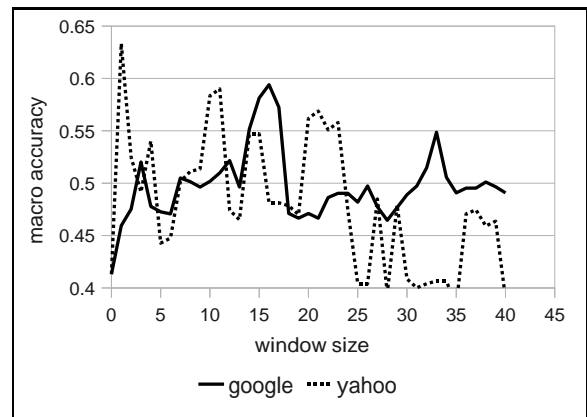


Figure 2: Macro accuracy when using Google Translate and Yahoo Babelfish

a test dataset with 2917 sentences. Both datasets were provided by the task organizers. Micro and macro accuracy were chosen as the evaluation metrics.

First, we compared the performance of our method when using Google Translate and Yahoo Babelfish for translating the trial dataset. The results for micro and macro accuracy are shown in Graphs 1 and 2 respectively. The x-axis represents a context window-size, equal to a number of words on both sides of the target adjective. The y-axis shows accuracy values. From the graphs we see that Google Translate provides better results, therefore it was chosen when annotating the test dataset.

Next, we studied the impact of the context window size on micro and macro accuracy. The impact of the size of the context window on the accuracy of the classifier trained on Chinese texts is depicted in Graph 3 and for the classifier trained on English texts with translated test dataset

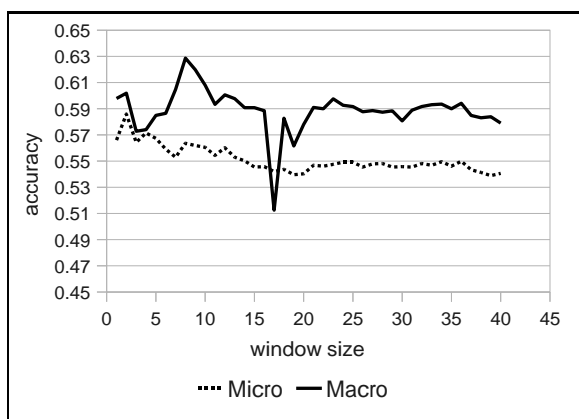


Figure 3: Micro and macro accuracy for the first approach (training on Chinese texts)

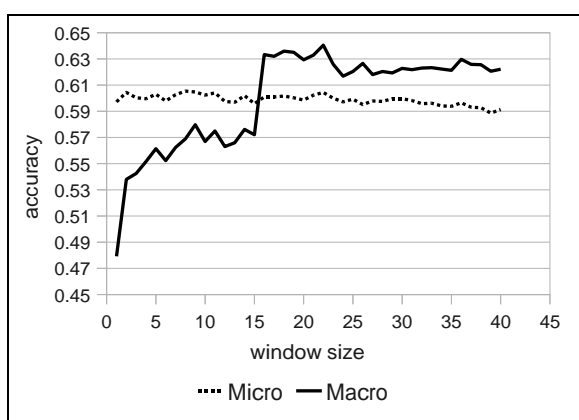


Figure 4: Micro and macro accuracy for the second approach (training on English texts which have been machine translated)

in Graph 4.

The second approach achieves better results. We were able to obtain 64% of macro and 61% of micro accuracy when using the second approach but only 63% of macro and 61% of micro accuracy when using the first approach.

Another observation from the graphs is that Chinese requires a smaller size of a context window to obtain the best performance. For the first approach, a window size of 8 words gave the best macro accuracy. For the second approach, we obtained the highest accuracy with a window size of 22 words.

5 Conclusion

In this paper, we have described our system for disambiguating sentiments of adjectives in Chinese texts. Our Naïve Bayes approach uses information automatically extracted from Twitter mi-

croblogs using emoticons. The techniques used in our approach can be applied to any other language. Our system is fully automate and does not utilize any hand-built lexicon. We were able to achieve up to 64% of macro and 61% of micro accuracy at the SemEval 2010 task

For the future work, we would like to collect more Chinese texts from Twitter or similar microblogging platforms. We think that increasing the training dataset will improve much the accuracy of the sentiment disambiguation.

References

- Kushal Dave, Steve Lawrence, and David M. Pennock. 2003. Mining the peanut gallery: opinion extraction and semantic classification of product reviews. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 519–528, New York, NY, USA. ACM.
- Alec Go, Lei Huang, and Richa Bhayani. 2009. Twitter sentiment analysis. Final Projects from CS224N for Spring 2008/2009 at The Stanford Natural Language Processing Group.
- Alexander Pak and Patrick Paroubek. 2010. Twitter as a corpus for sentiment analysis and opinion mining. In *Proceedings of LREC 2010*.
- 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 (EMNLP)*, pages 79–86.
- Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. In *Proceedings of the ACL Student Research Workshop*, pages 43–48.
- Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT '05: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 347–354, Morristown, NJ, USA. Association for Computational Linguistics.
- Yunfang Wu and Peng Jin. 2010. Semeval-2010 task 18: Disambiguating sentiment ambiguous adjectives. In *SemEval 2010: Proceedings of International Workshop of Semantic Evaluations*.
- Changhua Yang, Kevin Hsin-Yih Lin, and Hsin-Hsi Chen. 2007. Emotion classification using web blog corpora. In *WI '07: Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pages 275–278, Washington, DC, USA. IEEE Computer Society.