TJP: Identifying the Polarity of Tweets from Context

Tawunrat Chalothorn
Department of Computer Science and Digital Technologies
University of Northumbria at Newcastle, Pandon Building, Camden Street
Newcastle Upon Tyne, NE2 1XE, UK
tawunrat.chalothorn@northumbria.ac.uk

Jeremy Ellman
Department of Computer Science and Digital Technologies
University of Northumbria at Newcastle, Pandon Building, Camden Street
Newcastle Upon Tyne, NE2 1XE, UK
jeremy.ellman@northumbria.ac.uk

Abstract

The TJP system is presented, which participated in SemEval 2014 Task 9, Part A: Contextual Polarity Disambiguation. Our system is 'constrained', using only data provided by the organizers. The goal of this task is to identify whether marking contexts are positive, negative or neutral. Our system uses a support vector machine, with extensive pre-processing and achieved an overall F-score of 81.96%.

1 Introduction

The aim of sentiment analysis is to identify whether the subject of a text is intended to be viewed positively of negatively by a reader. Such emotions are sometimes hidden in long sentences and are difficult to identify. Consequently sentiment analysis is an active research area in natural language processing.

Sentiment is currently conceived terms of polarity. This has numerous interesting applications. For example, Grabner et al. (2012) used sentiment analysis to classify customers' reviews of hotels by using a star rating to categorize the reviews as bad, neutral and good. Similarly, Tumasjan et al. (2010) tried to predict the outcome of the German federal election through the analysis more than 100,000 tweets posted in the lead up. Sentiment analysis has also used to classify whether dreams are positive or negative (Nadeau et al. 2006).

This paper presents the TJP system which was submitted to SemEval 2014 Task 9, Part A: Contextual Polarity Disambiguation (Rosenthal et al., 2014). TJP focused on the 'Constrained' task.

The 'Constrained' task only uses data provided by the organizers. That is, external resources such as sentiment inventories (e.g. Sentiwordnet (Esuli, and Sebastiani 2006) are excluded. The objective of the TJP system was to use the results for comparison with our previous experiment (Chalothorn and Ellman, 2013). More details of these can be found in section 5.

The TJP system was implemented using a support vector machine (SVM, e.g. Joachims, 1999) with the addition of extensive pre-processing such as stopword removal, negation, slang, contraction, and emoticon expansions.

The remainder of this paper is constructed as follows: firstly, related work is discussed in section 2; the methodology, the experiment and results are presented in sections 3 and 4,

Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), pages 657–662,
Dublin, Ireland, August 23-24, 2014.

respectively. Finally a discussion and future work are given in section 5.

2 Related Work

Twitter is a popular social networking and microblogging site that allows users to post messages of up to 140 characters; known as 'Tweets'. Tweets are extremely attractive to the marketing sector, since tweets may be searched in real-time. This means marketing can find customer sentiment (both positive and negative) far more quickly than through the use of web pages or traditional media. Consequently analyzing the sentiment of tweets is currently active research task.

The word 'emoticon' is a neologistic contraction of 'emotional icon'. It refers specifically to the use of combinations of punctuation characters to indicate sentiment in a text. Well known emoticons include :) to represent a happy face, and :(a sad one. Emoticons allow writers to augment the impact of limited texts (such as in SMS messages or tweets) using few characters.

Read (2005) used emoticons from a training set downloaded from Usenet newsgroups as annotations (positive and negative). Using the machine learning techniques of Naïve Bayes and SVM, Read (2005) achieved up to 61.50 % and 70.10%, accuracy respectively in determining text polarity from the emoticons used.

Go et al. (2009) used distant supervision to classify sentiment of Twitter, similar to Read (2005). Emoticons were used as noisy labels in training data. This allowed the performance of supervised learning (positive and negative) at a distance. Three classifiers were used: Naïve Bayes, Maximum Entropy and SVM. These classifiers were able to obtain more than 81.30%, 80.50% and 82.20%, respectively accuracy on their unigram testing data .

Aramaki et al. (2011) classified contexts on Twitter related to influenza using a SVM. The training data was annotated with the polarity label by humans, whether they are positive or negative. The contexts will be labelled as positive if the contexts mention the user or someone close to them has the flu, or if they mention a time when they caught the flu. The results demonstrated that they obtained a 0.89 correction ratio for their testing data against a gold standard.

Finally, a well known paper by Bollen and Mao (2011) identified a correlation between the movements of the Dow Jones stock market index, and prevailing sentiment as determined from twitter's live feed. This application has prompted considerable work such as Makrehchi et al (2013) that has attempted to create successful trading strategies from sentiment analysis of tweets.

These work both the wide ranging applications of analysing twitter data, and the importance of Sentiment Analysis. We now move on to look at our approach to SemEval 2014 task 9.

3 Methodology

3.1 Corpus

The training and development dataset of SemEval was built using Tweets from more than one thousand pieces of context. The contexts have various features often used in Tweets, such as emoticons, tags, usernames etc. These features were extracted from the datasets before training for the supervised machine learning model.

During initial pre-processing of the datasets, emoticons were labelled by matching with the emoticons that have been collect manually from the dataset. Those labelled were matched against a well-known collection of emoticons[†].

Subsequently, negative contractions[‡] were expanded in place and converted to full form (e.g. don't -> do not). Moreover, the features of

[†]http://en.wikipedia.org/wiki/List_of_emoticons

[‡]http://en.wikipedia.org/wiki/English_auxiliaries_and_contractions#Negative_contractions

twitters were also removed or replaced by words such as twitter usernames, URLs and hashtags.

A Twitter username is a unique name that shows in the user's profile and may be used for both authentication and identification. This is shown by prefacing the username with an @ symbol. When a tweet is directed at an individual or particular entity this can be shown in the tweet by including @username. For example a tweet directed at 'tawunrat' would include the text @tawunrat. Before URLs are posted in twitter they are shortened automatically to use the t.co domain whose modified URLs are at most 22 characters. However, both features have been removed from the datasets. For the hashtags, they are used for represent keyword and topics in twitter by using # follow by words or phrase such as #newcastleuk. This feature has been replaced with the following word after # symbol. For example, #newcastleuk was replaced by newcastleuk.

Frequently repeated letters are used in tweets for emphasis. These were reduced and replaced using a simple regular expression by two of the same character. For example, happppppppy will be replaced with happy, and coolllllll will be replaced by cooll. Next, special character such as [,],{,},?,and ! were also removed. Slang and contracted words were converted to their full form. E.g. 'fyi' was converted to 'for your information'. Finally, NLTK (Bird et al. 2009) stopwords such as 'a', 'the', etc., were removed from the datasets.

3.2 Classifier

Our system uses the SVM classifier model (Hearst et al., 1998, Cristianini and Shawe-Taylor, 2000), which is based on SVM-light (Joachims, 1999). SVM is a binary linear classification model with the learning algorithm for classification and regression analyzing the data and recognizing the pattern.

Training SVMLight requires data to be formulated into vectors of attribute value pairs preceded by a numeric value. For example,

<target> <feature>:<value> <feature>:<value> ... <feature>:<value> # <info>

Here, 'target' represents the polarity of a sentence or tweet; 'feature' refers to a term in the document, and 'value' refers to a feature weight. This could be used as the relative frequency of a term in the set of documents, or Tf-Idf. Tf-idf is the combination of term frequency (tf) and inverse document frequency (idf), is a weight value often used in text mining and information retrieval. This weight is a statistical measure used to evaluate the relative important of word in a document in the collection (Manning et al., 2008).

$$tf - idf_{t,d} = tf_{t,d} * idf_t \quad (1)$$

where $tf - idf_{t,d}$ is the weighting the scheme assigns to term t in document d

Term frequency (tf) is used to measure how frequent the term appears in the document.

$$tf_{t,d} = \frac{n_{t,d}}{\sum_k n_{k,d}} \quad (2)$$

where $n_{t,d}$ is the number of term t appears in a document d. $\sum_k n_{k,d}$ is the total number of terms k in the document d.

Inverse document frequency (idf) is used to measure how important the term is – i.e. whether the term is common or rare in the collection.

$$idf_t = \log \frac{D}{d_t} \quad (3)$$

where D is the total number of documents in the collection in corpus. d_t is the number of documents d which term t appears.

Therefore, we chose to work with both of these to observe which yielded the best results in the polarity classification.

The default settings of SVMLight were used throughout. This meant that we used a linear kernel that did not require any parameters.[§]

4 Experiment and Results

In our experiment, we used the datasets and evaluated the system using the F-score measurement. During pre-processing features were extracted from both datasets. First, we used a frequency of word as a featured weight by calculating the frequency of word in the dataset and, during pre-processing, we labelled the emotions in both datasets. The results revealed a lower than average F-score at 34.80%. As this was quite low we disregarded further use of term frequency as a feature weight. We moved on to use Tf-Idf as the feature weight and, again, emoticons in both datasets were labelled. The score of 78.10% was achieved. Then, we kept the prepossessing of the training set stable by combining the features to extract from the testing data. These results are presented in Table 1[**].

The highest score of 81.96% was recorded when all the features were combined and extracted from both datasets.

The lowest score of 36.48% was recorded when emoticons were extracted from testing data and all features were extracted from training datasets. The results of the highest scoring experiment were submitted to the task organizers.

Following solution submissions, the task organizers announced the scores by separating the data into the following five groups: LiveJournal2014; SMS2013; Twitter2013; Twitter2014; and Twitter2014 Sarcasm. This would allow the identification of any domain dependent effects. However, the results showed that we achieved above average in all the datasets, as illustrated in Figure 1.

[§]Based on SVMLight

[**]The results in the table are from the test set 2014 in task 2A.

5 Conclusion and Future Work

The TJP system participated in SemEval 2014 Task 9, Part A: Contextual Polarity Disambiguation. The system exploited considerable pre-processing, before using the well known, SVMLight machine learning algorithm (Joachims. 1999). The pre-processing used several twitter specific features, such as hashtags and ids, in addition to more traditional Information Retrieval concepts such as the Tf-Idf heuristic (Manning et al., 2008). The results showed that the combination of all features in both datasets achieved the best results, at 81.96%.

An aspect of this contribution is the comparative analysis of feature effectiveness. That is, we attempted to identify which factor(s) made the most significant improvement to system performance. It is clear the pre-processing had a considerable effect on system performance. The use of a different learning algorithm also contributed to performance since, on this task, SVMLight performed better than the Naive Bayes algorithm that was used by our team in 2013.

Sentiment resources was not been used in our system in SemEval 2014 as same as in SemEval 2013 whilst other user groups have employed a variety of resources of different sizes, and accuracy (Wilson et al., 2013). These points lead to the following plan for future activities.

Our future work is to rigorously investigate the success factors for sentiment analysis, especially in the twitter domain. More specifically, we have formulated the following research questions as a result of our participation in SemEval

- Are Sentiment resources essential for the Sentiment Analysis task?
- Can the accuracy and effectiveness of sentiment lexicons be measured? If so, which feature of the resource (accuracy vs. coverage) is the most effective metric.
- Might it be more effective to use a range of sentiments (e.g. [-1.0 .. 1.0]), rather

than binary approach(e.g. positive and negative) taken in SemEval 2013, and 2014?

- Is one machine learning algorithm sufficient, and if so which is it? Or, alternately would an ensemble approach (Rokach, 2005) significantly improve performance?

Training \ Testing	Emoticon	Negation	@user URL	HashTag	Repeated letters	Special characters	Slang	Stopwords
Emoticon	78.10%	75.18%	75.18%	75.25%	75.25%	76.35%	76.26%	68.19%
Negation	63.56%	79.06%	79.06%	75.25%	79.14%	80.07%	80.00%	69.70%
@user, URL	63.54%	79.05%	79.05%	79.12%	79.14%	80.07%	80.00%	69.70%
HashTag	63.59%	79.08%	79.08%	79.11%	79.18%	80.10%	80.03%	69.67%
Repeated letters	63.60%	79.08%	79.10%	79.14%	79.18%	80.11%	80.02%	69.74%
Special characters	67.87%	79.10%	78.55%	79.17%	78.62%	80.82%	80.69%	69.62%
Slang	68.39%	78.39%	78.39%	78.62%	78.45%	80.70%	80.85%	69.56%
Stopwords	36.48%	64.67%	64.67%	78.45%	64.67%	64.82%	64.82%	81.96%

Table 1: The results of each feature analyzed in the approach of TF-IDF

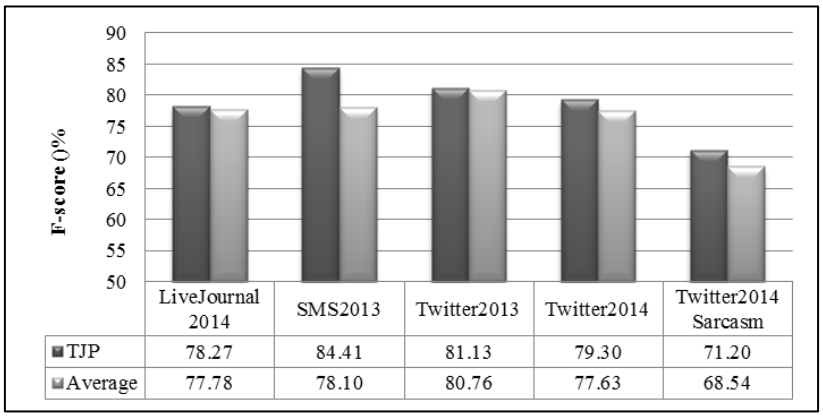

Figure 1: The comparison of TJP and average scores

References

Alec Go, Richa Bhayani and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford***,** 1-12.

Andrea Esuli, Fabrizio Sebastiani 2006 "SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining" in Proceedings of the 5th Conference on Language Resources and Evaluation, LREC (2006), pp. 417-422

Andranik Tumasjan, Timm O. Sprenger, Philipp G. Sandner and Isabell M. Welpe. 2010. "Predicting elections with twitter: What 140 characters reveal about political sentiment," in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, pp. 178-185.

Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze, *Introduction to Information Retrieval*, Cambridge University Press. 2008. ISBN: 0521865719.

David Nadeau, Catherine Sabourin, Joseph De Koninck, Stan Matwin and Peter D. Turney. 2006.

"Automatic dream sentiment analysis," presented at the In: *Proceedings of the Workshop on Computational Aesthetics at the Twenty-First National Conference on Artificial Intelligence*, Boston, Massachussetts, USA.

Dietmar Grabner, Markus Zanker, Gunther Fliedl and Matthias Fuchs. 2012."Classification of customer reviews based on sentiment analysis," *presented at the 19th Conference on Information and Communication Technologies in Tourism (ENTER)*, Helsingborg, Sweden.

Eiji Aramaki, Sachiko Maskawa and Mizuki Morita. 2011. Twitter catches the flu: detecting influenza epidemics using Twitter. *Proceedings of the Conference on Empirical Methods in Natural Language Processing.* Edinburgh, United Kingdom: Association for Computational Linguistics.

Johan. Bollen and Huina. Mao. Twitter mood as a stock market predictor. IEEE Computer, 44(10):91–94.

Jonathon Read. 2005. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. *Proceedings of the ACL Student Research Workshop.* Ann Arbor, Michigan: Association for Computational Linguistics.

Lior Rokach. 2005. Chapter 45 Ensemble Methods for Classifiers. *In:* Oded Maimon and Lior Rokach (eds.) *Data Mining and Knowledge Discovery Handbook.* Springer US.

Marti A. Hearst, Susan T. Dumais, Edgar Osman, John Platt and Bernhard Scholkopf . 1998. Support vector machines. *IEEE, Intelligent Systems and their Applications*, 13**,** 18-28.

Masoud Makrehchi, Sameena Shah and Wenhui Liao. 2013. Stock Prediction Using Event-Based Sentiment Analysis. Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on,. 337-342.

Nello Cristianini and John Shawe-Taylor. 2000. *An introduction to support vector machines and other kernel-based learning methods*, Cambridge university press.

Sara Rosenthal, Preslav Nakov, Alan Ritter and Veselin Stoyanov. 2014. SemEval-2014 Task 9: Sentiment Analysis in Twitter. *International Workshop on Semantic Evaluation (SemEval-2014).* Dublin, Ireland.

Steven Bird, Ewan Klein and Edward Loper. 2009. *NLTK: Natural language processing with Python*, O'Reilly.

Takeshi Sakaki, Makoto Okazaki and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. *Proceedings of the 19th international conference on World wide web.* Raleigh, North Carolina, USA: ACM.

Tawunrat Chalothorn and Jeremy Ellman. 2013. TJP: Using Twitter to Analyze the Polarity of Contexts. *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013).* Atlanta, Georgia, USA: Association for Computational Linguistics.

Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Alan Ritter, Sara Rosenthal and Veselin Stoyanov. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter. *Proceedings of the 7th International Workshop on Semantic Evaluation.* Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large-scale support vector machine learning practical. *Advances in kernel methods.* MIT Press.