

# TJP: Using Twitter to Analyze the Polarity of Contexts

**Tawunrat Chalothorn**

University of Northumbria at Newcastle  
Pandon Building, Camden Street  
Newcastle Upon Tyne, NE2 1XE, UK  
Tawunrat.chalothorn@unn.ac.uk

**Jeremy Ellman**

University of Northumbria at Newcastle  
Pandon Building, Camden Street  
Newcastle Upon Tyne, NE2 1XE, UK  
Jeremy.ellman@unn.ac.uk

## Abstract

This paper presents our system, TJP, which participated in SemEval 2013 Task 2 part A: Contextual Polarity Disambiguation. The goal of this task is to predict whether marked contexts are positive, neutral or negative. However, only the scores of positive and negative class will be used to calculate the evaluation result using F-score. We chose to work as ‘constrained’, which used only the provided training and development data without additional sentiment annotated resources. Our approach considered unigram, bigram and trigram using Naïve Bayes training model with the objective of establishing a simple-approach baseline. Our system achieved F-score 81.23% and F-score 78.16% in the results for SMS messages and Tweets respectively.

## 1 Introduction

Natural language processing (NLP) is a research area comprising various tasks; one of which is sentiment analysis. The main goal of sentiment analysis is to identify the polarity of natural language text (Shaikh et al., 2007). Sentiment analysis can be referred to as opinion mining, as study peoples’ opinions, appraisals and emotions towards entities and events and their attributes (Pang and Lee, 2008). Sentiment analysis has become a popular research area in NLP with the purpose of identifying opinions or attitudes in terms of polarity.

This paper presents TJP, a system submitted to SemEval 2013 for Task 2 part A: Contextual Polar-

ity Disambiguation (Wilson et al., 2013). TJP was focused on the ‘constrained’ task, which used only training and development data provided. This avoided both resource implications and potential advantages implied by the use of additional data containing sentiment annotations. The objective was to explore the relative success of a simple approach that could be implemented easily with open-source software.

The TJP system was implemented using the Python Natural Language Toolkit (NLTK, Bird et al., 2009). We considered several basic approaches. These used a preprocessing phase to expand contractions, eliminate stopwords, and identify emoticons. The next phase used supervised machine learning and n-gram features. Although we had two approaches that both used n-gram features, we were limited to submitting just one result. Consequently, we chose to submit a unigram based approach followed by naive Bayes since this performed better on the data.

The remainder of this paper is structured as follows: section 2 provides some discussion on the related work. The methodology of corpus collection and data classification are provided in section 3. Section 4 outlines details of the experiment and results, followed by the conclusion and ideas for future work in section 5.

## 2 Related Work

The micro-blogging tool Twitter is well-known and increasingly popular. Twitter allows its users to post messages, or ‘Tweets’ of up to 140 characters each time, which are available for immediate

download over the Internet. Tweets are extremely interesting to marketing since their rapid public interaction can either indicate customer success or presage public relations disasters far more quickly than web pages or traditional media. Consequently, the content of tweets and identifying their sentiment polarity as positive or negative is a current active research topic.

Emoticons are features of both SMS texts, and tweets. Emoticons such as :) to represent a smile, allow emotions to augment the limited text in SMS messages using few characters. Read (2005) used emoticons from a training set that was downloaded from Usenet newsgroups as annotations (positive and negative). Using the machine learning techniques of Naïve Bayes and Support Vector Machines Read (2005) achieved up to 70 % accuracy in determining text polarity from the emoticons used.

Go et al. (2009) used distant supervision to classify sentiment of Twitter, as similar as in (Read, 2005). Emoticons have been used as noisy labels in training data to perform distant supervised learning (positive and negative). Three classifiers were used: Naïve Bayes, Maximum Entropy and Support Vector Machine, and they were able to obtain more than 80% accuracy on their testing data.

Aisopos et al. (2011) divided tweets in to three groups using emoticons for classification. If tweets contain positive emoticons, they will be classified as positive and vice versa. Tweets without positive/negative emoticons will be classified as neutral. However, tweets that contain both positive and negative emoticons are ignored in their study. Their task focused on analyzing the contents of social media by using n-gram graphs, and the results showed that n-gram yielded high accuracy when tested with C4.5, but low accuracy with Naïve Bayes Multinomial (NBM).

### 3 Methodology

#### 3.1 Corpus

The training data set for SemEval was built using Twitter messages training and development data. There are more than 7000 pieces of context. Users usually use emoticons in their tweets; therefore, emoticons have been manually collected and labeled as positive and negative to provide some

context (Table 1), which is the same idea as in Aisopos et al. (2011).

Negative emoticons	:( :-(: :d :< D: :\/: etc.
Positive emoticons	:) ;) :-) ;-) :P ;P (: (; :D ;D etc.

Table 1: Emoticon labels as negative and positive

Furthermore, there are often features that have been used in tweets, such as hashtags, URL links, etc. To extract those features, the following processes have been applied to the data.

1. Retweet (RT), twitter username (@panda), URL links (e.g. y2u.be/fiKKzdLQvFo), and special punctuation were removed.
2. Hashtags have been replaced by the following word (e.g. # love was replaced by love, # exciting was replaced by exciting).
3. English contraction of ‘not’ was converted to full form (e.g. don’t -> do not).
4. Repeated letters have been reduced and replaced by 2 of the same character (e.g. happpppppy will be replaced by happy, coollllll will be replaced by cooll).

#### 3.2 Classifier

Our system used the NLTK Naïve Bayes classifier module. This is a classification based on Bayes’s rule and also known as the state-of-art of the Bayes rules (Cufoglu et al., 2008). The Naïve Bayes model follows the assumption that attributes within the same case are independent given the class label (Hope and Korb, 2004).

Tang et al. (2009) considered that Naïve Bayes assigns a context  $X_i$ (represented by a vector  $X_i^*$ ) to the class  $C_j$  that maximizes  $P(C_j|X_i^*)$  by applying Bayes’s rule, as in (1).

$$P(C_j|X_i^*) = \frac{P(C_j)P(X_i^*|C_j)}{P(X_i^*)} \quad (1)$$

where  $P(X_i^*)$  is a randomly selected context  $X$ . The representation of vector is  $X_j^*$ .  $P(C)$  is the random select context that is assigned to class  $C$ .

To classify the term  $P(X_i^*|C_j)$ , features in  $X_i^*$  were assumed as  $f_j$  from  $j = 1$  to  $m$  as in (2).

$$P(C_j|X_i^*) = \frac{P(C_j) \prod_{j=1}^m P(f_j|C_j)}{P(X_i^*)} \quad (2)$$

There are many different approaches to language analysis using syntax, semantics, and semantic resources such as WordNet. That may be exploited using the NLTK (Bird et al. 2009). However, for simplicity we opted here for the n-gram approach where texts are decomposed into term sequences. A set of single sequences is a unigram. The set of two word sequences (with overlapping) are bigrams, whilst the set of overlapping three

term sequences are trigrams. The relative advantage of the bi-and trigram approaches are that coordinates terms effectively disambiguate senses and focus content retrieval and recognition.

N-grams have been used many times in contents classification. For example, Pang et al. (2002) used unigram and bigram to classify movie reviews. The results showed that unigram gave better results than bigram. Conversely, Dave et al. (2003) reported gaining better results from trigrams rather than bigram in classifying product reviews. Consequently, we chose to evaluate unigrams, bigrams and trigrams to see which will give the best results

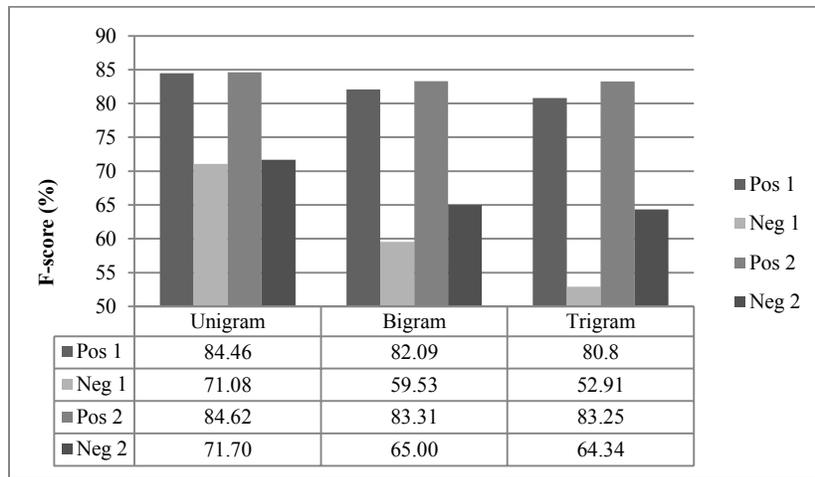


Figure 1: Comparison of Twitter messages from two approaches

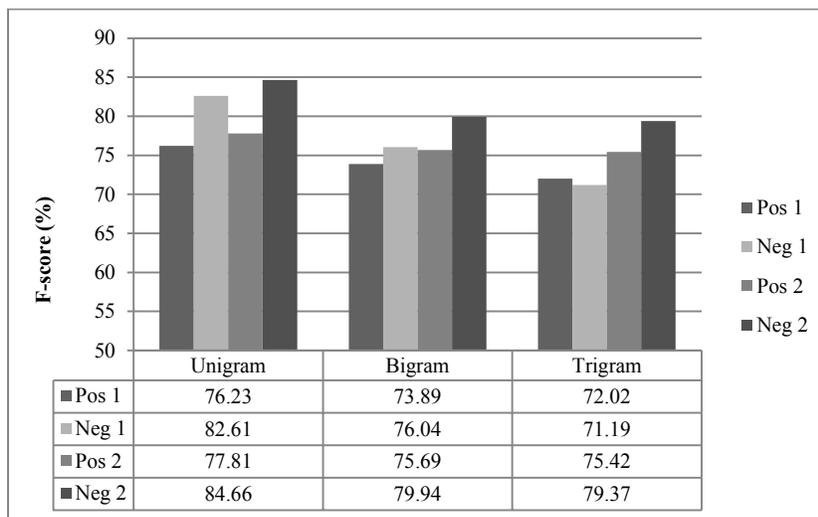


Figure 2: Comparison of SMS messages from two approaches

in the polarity classification. Our results are described in the next section.

#### 4 Experiment and Results

In this experiment, we used the distributed data from Twitter messages and the F-measure for system evaluation. As at first approach, the corpora were trained directly in the system, while stopwords (e.g. a, an, the) were removed before training using the python NLTK for the second approach. The approaches are demonstrated on a sample context in Table 2 and 3.

After comparing both approaches (Figure 1), we were able to obtain an F-score 84.62% of positive and 71.70% of negative after removing stopwords. Then, the average F-score is 78.16%, which was increased from the first approach by 0.50%. The results from both approaches showed that, unigram achieved higher scores than either bigrams or trigrams.

Moreover, these experiments have been tested with a set of SMS messages to assess how well our system trained on Twitter data can be generalized to other types of message data. The second approach still achieved the better scores (Figure 2), where we were able to obtain an F-score of 77.81% of positive and 84.66% of negative; thus, the average F-score is 81.23%.

The results of unigram from the second approach submitted to SemEval 2013 can be found in Figure 3. After comparing them using the average F-score from positive and negative class, the results showed that our system works better for SMS messaging than for Twitter.

gonna miss some of my classes.		
Unigram	Bigram	Trigram
gonna miss some of my classes	gonna miss miss some some of of my my classes	gonna miss some miss some of some of my of my classes

Table 2: Example of context from first approach

gonna miss ( <i>some of</i> ) my classes.		
Unigram	Bigram	Trigram
gonna miss my classes	gonna miss miss my my classes	gonna miss my miss my classes

Table 3: Example of context from second approach. Note ‘some’ and ‘of’ are listed in NLTK stopwords.

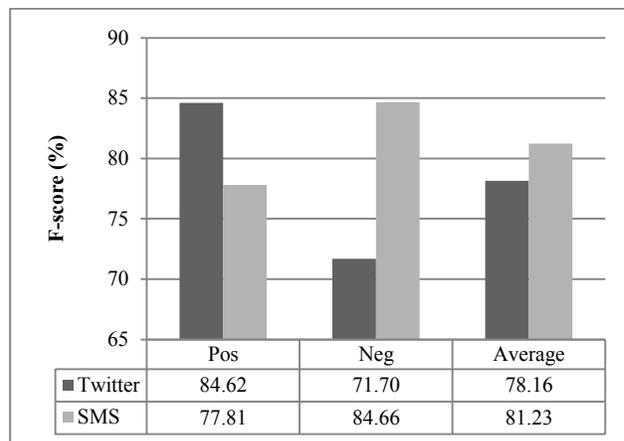


Figure 3: Results of unigram of Twitter and SMS in the second approach

#### 5 Conclusion and Future Work

A system, TJP, has been described that participated in SemEval 2013 Task 2 part A: Contextual Polarity Disambiguation (Wilson et al., 2013). The system used the Python NLTK (Bird et al 2009) Naive Bayes classifier trained on Twitter data. Furthermore, emoticons were collected and labeled as positive and negative in order to classify contexts with emoticons. After analyzing the Twitter message and SMS messages, we were able to obtain an average F-score of 78.16% and 81.23% respectively during the SemEval 2013 task. The reason that, our system achieved better scores with SMS message than Twitter message might be due to our use of Twitter messages as training data. However this is still to be verified experimentally.

The experimental performance on the tasks demonstrates the advantages of simple approaches. This provides a baseline performance set to which more sophisticated or resource intensive techniques may be compared.

For future work, we intend to trace back to the root words and work with the suffix and prefix that imply negative semantics, such as ‘dis-’, ‘un-’, ‘-ness’ and ‘-less’. Moreover, we would like to collect more shorthand texts than that used commonly in microblogs, such as gr8 (great), btw (by the way), pov (point of view), gd (good) and ne1 (anyone). We believe these could help to improve our system and achieve better accuracy when classifying the sentiment of context from microblogs.

## References

- Alec Go, Richa Bhayani and Lei Huang. 2009. Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford, 1-12.
- Ayse Cufoglu, Mahi Lohi and Kambiz Madani. 2008. *Classification accuracy performance of Naive Bayesian (NB), Bayesian Networks (BN), Lazy Learning of Bayesian Rules (LBR) and Instance-Based Learner (IB1) - comparative study*. Paper presented at the Computer Engineering & Systems, 2008. ICCES 2008. International Conference on.
- Bo Pang, Lillian Lee and Shivakumar Vaithyanathan. 2002. *Thumbs up?: sentiment classification using machine learning techniques*. Paper presented at the Proceedings of the ACL-02 conference on Empirical methods in natural language processing - Volume 10.
- Fotis Aisopos, George Papadakis and Theodora Varvarigou. 2011. *Sentiment analysis of social media content using N-Gram graphs*. Paper presented at the Proceedings of the 3rd ACM SIGMM international workshop on Social media, Scottsdale, Arizona, USA.
- Huifeng Tang, Songbo Tan and Xueqi Cheng. 2009. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7), 10760-10773.
- Jonathon. Read. 2005. *Using emoticons to reduce dependency in machine learning techniques for sentiment classification*. Paper presented at the Proceedings of the ACL Student Research Workshop, Ann Arbor, Michigan.
- Kushal Dave, Steve Lawrence and David M. Pennock. 2003. *Mining the peanut gallery: opinion extraction and semantic classification of product reviews*. Paper presented at the Proceedings of the 12th international conference on World Wide Web, Budapest, Hungary.
- Lucas R. Hope and Kevin B. Korb. 2004. *A bayesian metric for evaluating machine learning algorithms*. Paper presented at the Proceedings of the 17th Australian joint conference on Advances in Artificial Intelligence, Cairns, Australia.
- Mostafa Al Shaikh, Helmut Prendinger and Ishizuka Mitsuru. 2007. *Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis*. Paper presented at the Proceedings of the 2nd international conference on Affective Computing and Intelligent Interaction, Lisbon, Portugal.
- Pang Bo and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Found. Trends Inf. Retr.*, 2(1-2), 1-135.
- Steven Bird, Ewan Klein and Edward Loper. 2009. *Natural language processing with Python*: O'Reilly.
- Theresa Wilson, Zornitsa Kozareva, Preslav Nakov, Sara Rosenthal, Veselin Stoyanov and Alan Ritter. 2013. SemEval-2013 Task 2: Sentiment Analysis in Twitter *Proceedings of the 7th International Workshop on Semantic Evaluation*: Association for Computational Linguistics.