Sentiment Classification and Analysis in Twitter

Justin Cosentino
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
jcosent1@swarthmore.edu

Emanuel Schorsch
Department of Computer Science
Swarthmore College
Swarthmore, PA 19081
eschors1@swarthmore.edu

Abstract

As the popularity of microblogging drastically increases, it is evident that there is a vast amount of opinionated data and information available to assess the general sentiment of millions of Twitter users in regards to products, events, and people. We consider the problem of classifying the overall sentiment of subjects contained within tweets and the sentiment of a marked instance of a word or phrase within a given tweet as either positive, negative, neutral, or objective. In completing these tasks, we hope to further develop a public twitter sentiment corpus.

Using two Titter corpora provided by SemEval-2013, we implemented and trained a nave Bayes classifier, a decision list classifier, and a subjectivity lexicon classifier. By using a bag-of-words framework and limited training data, we are able to successfully identify the sentiment of tweets. Our data suggests that the accuracy of our models will further increase as the size of our training corpora increase, and suggests that both naive Bayes and decision list classifiers trained on n-gram feature sets become far superior to the subjectivity lexicon classifier as the size of the training corpora increases. We find that that lack of slang and abbreviated words in the subjectivity lexicon account for this decrease in performance, and we propose creating a subjectivity lexicon that contains such language for the classification of tweets. We then conclude by examining and discussing the factors that make the sentiment classification of tweets so difficult.

1 Introduction

2 Related Works

3 Methodology

3.1 Data

A total of three different corpora were used in our experiments. Two of these corpora contained Twitter data that was used in the training and testing of our classifiers. The third data set was comprised of a sentiment lexicon which was used as a form of message classification.

3.1.1 Twitter Corpus

As previously stated, Twitter is a microblogging service that allows users to post short, 140-character messages called tweets. The two Twitter corpora used in this study were comprised of such tweets and acquired from the SemEval-2013 Competition site[1] using the provided script, which downloaded the tweets from the Twitter webpage. Some of these tweets were no longer available due to a user changing their privacy settings, deleting the designated tweet, or deleting their account. These tweets

[1] Available at http://www.cs.york.ac.uk/semeval-2013/task2/

were ignored and not used within our study. Additionally, some tweets contained newline characters, which were removed from the tweet during the download process. Although the data found within these tweets was formatted differently, each corpus contained tweets covering a wide range of topics including entities, products, and events. The tweets found in the corpora were also exclusively written in English. The first corpus contained full tweets and the polarity tag for a marked instance of a word or phrase within each tweet while the second corpus contained full tweets and the polarity tag of a given topic found within the content of each tweet. There were a total of 2,376 tweet segments or phrases within the first corpus and 591 full tweets in the second and the polarity tags for each corpus were limited to 'positve', 'negative', 'neutral' and 'objective'. Although some of the tweets contained within each corpus were identical, the first corpus prompted users to only look at the given segment of the tweet. Each data set was used individually as both training and testing data throughout the study. The first 80% of each corpus was constituted as training data while the remaining 20% of the tweets were reserved for test data.

3.1.2 Subjectivity Lexicon

The third corpus used in our experiments contained subjectivity data relating to roughly 6,500 words. This lexicon was acquired from OpinionFinder, a system that performs subjectivity analysis (Wilson et al., 2005). Each word within this corpus has a corresponding polarity, strength of subjectivity, part-of-speech tag, stem value, and word length. For the purposes of this study, the strength of subjectivity, stem value, and word length were ignored. The polarity and part-of-speech tag were then used as means of classifying the sentiment value of tweets.

3.2 Features

In order to implement the sentiment classifiers such that they can label the given data, we used a number of features built from the bag-of-words model. This model is implemented as an unordered listing of features and does not take into account word order or word location. Using this model we create feature vectors that contain n-gram features present in a given tweet. The frequency of these features is also represented by the vector. In the implementation of the decision list and naive Bayes classifiers, unigrams, bigrams, and trigrams were used to train and classify data. In the subjectivity lexicon classifier, the individual words making up the tweet and their associated polarity make up the features of the vector.

3.3 Sentiment Classifiers

In order to determine the sentiment of a given tweet, three algorithms were implemented: a naive Bayes classifier, a decision list classifier, and a subjectivity lexicon classifier.

3.3.1 Naive Bayes

Using the Naive Bayes classifier, a given feature vector $\vec{f}$ is assigned the polarity ŝ such that $\hat{s} = \arg\max\limits_{s \in S} P(s|\vec{f})$. This sentiment classifier is based upon and derived from a fundamental statistical rule called Bayes' law:

$$\hat{s} = \arg\max_{s \in S} \frac{P(\vec{f}|s)P(s)}{P(\vec{f})}.$$

Because $P(\vec{f})$ remains the same value across all possible polarities, it does not assist in determing the value of $\hat{s}$. We then have:

$$\hat{s} = \arg\max_{s \in S} P(\vec{f}|s)P(s).$$

However, given a feature vector $\vec{f}$ it is very unlikely that we will see this exact feature again. Thus we naively assume that each $\vec{f_i}$ of $\vec{f}$ is independent of all other $\vec{f_i}$. Making this assumption allows us to approximate $P(\vec{f}|s)$:

$$P(\vec{f}|s) \approx \prod_{i=1}^{n} P(f_i|s).$$

Thus, in estimating the probability of the vector $\vec{f}$ by finding the product of the probabilities pertaining to each individual feature within $\vec{f}$ given the polarity $\hat{s}$, we find the final equation for the naive Bayes classifier:

$$\hat{s} = \arg\max_{s \in S} \prod_{i=1}^{n} P(f_i|s)P(s).$$

The maximum likelihood estimate of the probability of each possible sentiment polarity is calculated by taking the count of the number of times a given feature occurs given the polarity over the number of times the feature occurs. This is represented by:

$$P(s_i) = \frac{count(s_i, w_j)}{count(w_j)}.$$

The probability of each feature given a sense is calculated in a similar manner such that:

$$P(f_i|s) = \frac{count(f_i, s)}{count(s)}.$$

Each of these equations utilizes Laplace add-one smoothing, allowing for non-zero probabilities to be assigned to words found in the test data that were not seen in the training sample.

3.3.2 Decision List

A decision list classifier is equivalent to simple case statements or an extended if-else statement. Decision lists generate a set of rules or conditions, for which there is a single classification associated. These rules are generated from tagged feature vectors and then scored and ordered based on their associated scores. Similar to the approach used by Yarowsky, each feature and value pair is treated as a rule (Yarowsky, 1994). In order to generate the best rule for each possible classification, the following equation is used:

$$\left| Log \left(\frac{P(Sense_i|f_i)}{P(\text{All other senses}|f_i)} \right) \right|.$$

Once these rules have been generated from the given feature sets and scored, the decision list creates a list of rules similar to those seen in Table 1. The decision list will then use these rules as long as their respective scores remain greater than or equal to one. At this point, the decision list will then proceed to classify words based on the most frequent sense of all tweets in the given training corpus.

Rule		Polarity
love	⇒	Positive
can't wait	⇒	Positive
looking forward	⇒	Positive
confirmed	⇒	Objective
crash	⇒	Negative
...	⇒	...
Score < 1	⇒	MFS

Table 1: Example rules generated by the decision list after training on the second Twitter corpus.

3.3.3 Subjectivity Lexicon

A subjectivity lexicon was used to classify and determine the polarity of tweets. The subjectivity lexicon was acquired from OpinionFinder, a list containing roughly 6,500 words, their polarity, and the strength of their subjectivity. This lexicon was then used to determine the polarity of each word within the given tweet or tweet segment. The strength of each word, which was labeled as either strong or weak, was ignored for testing.

Individual counts of the number of positive and negative words for each tweet were than kept. If a tweet or tweet segment contained more positive words than negative words, the tweet was then defined as having a positive polarity. If the tweet contained more negative words than positive words, the tweet was determined to have a negative polarity. However, if a tweet contained neither any positive words nor any negative words or if the count of positive and negative words were equal, the tweet was labeled as objective. Within the subjectivity lexicon classifier, words were not labeled as neutral.

Because the subjectivity classifier requires no labeled tweets to be used as training data, the classifier was tested on all tweets within each corpus. However, in order to compare the results of the subjectivity lexicon classifier to the results of our other classifiers, the lexicon was also used to label only the test data used by all other classifiers.

3.3.4 Most Frequent Sense

The most frequent sense (MFS) classifier is used as a baseline for comparison in our experiments. For each corpus, the classifier counts the number of occurrences for each polarity classification and then chooses that most frequently occurring tag. The classifier then labels all test data using this classification. This method is also used in the decision list if the list has exhausted all possible rules with a score greater than or equal to one.

4 Results

5 Analysis

6 Conclusion

6.1 Future Work

7 Credits

This document has been adapted from the instructions for HLT-NAACL proceedings, which in turn was based on the formats of earlier ACL and EACL Conference proceedings. Those versions were written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence*.

8 Introduction

The following describes the formatting instructions for both the midterm and final project. You are required to adhere to these specifications. At submission time, you are required to provide the complete TeX source, including any supporting external files, as well as a Portable Document Format (PDF) of your report.

9 General Instructions

Manuscripts must be in two-column format. Exceptions to the two-column format include the title, authors' names and complete addresses, which must be centered at the top of the first page, and any full-width figures or tables (see the guidelines in Subsection 9.4). **Type single-spaced.** Start all pages directly under the top margin. See the guidelines later regarding formatting the first page.

9.1 Electronically-available resources

This description is provided in LaTeX (cs65f12.tex) along with the LaTeX style file used to format it (cs65f12.sty). In addition, there is a bibliography style (cs65f12.bst) and sample bibliography file (cs65f12.bib). These files are all in the `cs65/labs/04-05/` directory.

9.2 Format of Electronic Manuscript

The easiest way to turn this LaTeX into a PDF, is to use the Makefile found (also in `cs65/labs/04-05/`). The Makefile will compile your file (and your bibliography file) and turn it into a PDF.

Here are the basic instructions:

- `make` will create a PDF file from your LaTeX document.
- `make view` will display the PDF file.
- `make clean` will clean up some files you might not need
- `make cleanall` will clean up all non-source files

For reasons of uniformity, Adobe's **Times Roman** font should be used. In LaTeX this is accomplished by putting

```
\usepackage{times}
\usepackage{latexsym}
```

in the preamble, as was done in this file.

Print-outs of the PDF file should look like the present document, which conforms to the formatting requirements. If you cannot meet the above requirements, please contact me as soon as possible.

9.3 Layout

Format manuscripts two columns to a page, in the manner these instructions are formatted. The exact dimensions for a page on US-letter paper are:

- Left and right margins: 1in
- Top margin:1in
- Bottom margin: 1in
- Column width: 3.15in
- Column height: 9in
- Gap between columns: 0.2in

Papers should not be submitted on any other paper size.

9.4 The First Page

Center the title, author's name(s) and affiliation(s) across both columns. Do not use footnotes for affiliations. Use the two-column format only when you begin the abstract.

Title: Place the title centered at the top of the first page, in a 15-point bold font. A long title should be typed on two lines without a blank line intervening. Approximately, put the title at 1in from the top of the page, followed by a blank line, then the author's names(s), and the affiliation on the following line. Do not use only initials for given names (middle initials are allowed). The affiliation should contain the author's complete address, and an email address. Leave about 0.75in between the affiliation and the body of the first page.

Abstract: Type the abstract at the beginning of the first column. The width of the abstract text should be smaller than the width of the columns for the text in the body of the paper by about 0.25in on each side. Center the word **Abstract** in a 12 point bold font above the body of the abstract. The abstract should be a concise summary of the general thesis and conclusions of the paper. It should be no longer than 200 words.

Text: Begin typing the main body of the text immediately after the abstract, observing the two-column format as shown in the present document.

Indent when starting a new paragraph. For reasons of uniformity, use Adobe's **Times Roman** fonts, with 11 points for text and subsection headings, 12 points for section headings and 15 points for the title. If Times Roman is unavailable, use **Computer Modern Roman** (LaTeX's default; see section 9.2 above). Note that the latter is about 10% less dense than Adobe's Times Roman font.

9.5 Sections

Headings: Type and label section and subsection headings in the style shown on the present document. Use numbered sections (Arabic numerals) in order to facilitate cross references. Number subsections with the section number and the subsection number separated by a dot, in Arabic numerals. Do not number subsubsections.

Citations: Citations within the text appear in parentheses as (Harris, 1955) or, if the author's name appears in the text itself, as Harris (1955). Append lowercase letters to the year in cases of ambiguities. Treat double authors as in (Hafer and Weiss, 1974), but write as in (Hana et al., 2006) when more than two authors are involved. Collapse multiple citations as in (Harris, 1967; Déjean, 1998).

References: Gather the full set of references together under the heading **References**; place the section before any Appendices, unless they contain references. Arrange the references alphabetically by first author, rather than by order of occurrence in the text. Provide as complete a citation as possible, using a consistent format.

The LaTeX and BibTeX style files provided roughly fit the American Psychological Association format, allowing regular citations, short citations and multiple citations as described above.

Appendices: Appendices, if any, directly follow the text and the references (but see above). Letter them in sequence and provide an informative title: **Appendix A. Title of Appendix**.

Acknowledgement sections should go as a last section immediately before the references. Do not number the acknowledgement section.

9.6 Footnotes

Footnotes: Put footnotes at the bottom of the page. They may be numbered or referred to by asterisks or other symbols.[2] Footnotes should be separated from the text by a line.[3]

9.7 Graphics

Illustrations: Place figures, tables, and photographs in the paper near where they are first discussed, rather than at the end, if possible. Wide illustrations may run across both columns. Do not use color illustrations as they may reproduce poorly.

Captions: Provide a caption for every illustration; number each one sequentially in the form: "Figure 1. Caption of the Figure." "Table 1. Caption of the Table." Type the captions of the figures and tables below the body, using 11 point text.

[2] This is how a footnote should appear.

[3] Note the line separating the footnotes from the text.

References

Hervé Déjean. 1998. Morphemes as necessary concept for structures discovery from untagged corpora. In *Proceedings of the ACL-98 Workshop on New Methods in Language Processing and Computational Natural Language Learning*.

Margaret A. Hafer and Stephen F. Weiss. 1974. Word segmentation by letter success varieties. *Information Storage and Retrieval*, 10:371–385.

Jirka Hana, Anna Feldman, Luiz Amaral, and Chris Brew. 2006. Tagging Portuguese with a Spanish tagger. In *Proceedings of the EACL-06 Workshop on Cross-Language Knowledge Induction*.

Zellig Harris. 1955. From phoneme to morpheme. *Language*, 31:190–222.

Zellig Harris. 1967. Morpheme boundaries within words: Report on a computer test. In *Transformations and Discourse Analysis Papers*. Department of Linguistics, University of Pennsylvania.

T. Wilson, P. Hoffmann, S. Somasundaran, J. Kessler, J. Wiebe, Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Opinionfinder: A system for subjectivity analysis. In *Proceedings of HLT/EMNLP on Interactive Demonstrations*, pages 34–35. Association for Computational Linguistics.

D. Yarowsky. 1994. Decision lists for lexical ambiguity resolution: Application to accent restoration in spanish and french. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 88–95. Association for Computational Linguistics.