ReelTalk:
An Interactive Sentiment Analysis Application

Anne Flinchbaugh
anne.flinchbaugh@gmail.com

Eric Latimer
e@utexas.edu

Abstract

In this report, we present the implementation and application of a sentiment analysis classifier to architect the behavior of a Twitter response bot. Upon each Twitter message received, the bot provides responses that have the same sentiment as the input tweet. Additionally, the reply aims to be related to the input tweet to better resemble actual human to human conversation.

1 Introduction

Growing in importance and popularity, social network sites and microblogging, namely Twitter, are becoming staples in many people's everyday lives. Many Twitter users perceive the site as a community where they can express feelings and opinions with other like-minded users. Because of this increasing popularity and usage, more and more textual data consisting of peoples opinions is becoming available for interesting language processing tasks. One such task that would contribute to the sense of community is a Twitter account that automatically replies to a user in an interesting and reciprocal fashion.

Twitter sentiment analysis has been the focus of previous research [1], [2], [7], [8]. Pak and Paroubek's research evaluates the performance of a sentiment classifier trained on tweets [1]. However, the lack of grammatical structure in a tweet and the limited supply of tweets annotated with their sentiments restrict the performance of such classifiers. As noted by Yessenov and Misailovic, movie reviews "provide good material for analyzing ... opinions of the authors" [2]. Therefore, to maximize the effectiveness of the bot, codenamed ReelTalk, the domain for this study is restricted to movies. More specifically, the user sends a tweet about their opinion of a movie to ReelTalk, who then determines the movie title of interest, collects movie reviews online that exhibit the same sentiment, and returns a review to the user. The movie reviews are obtained from the Rotten Tomatoes API[1]. This website serves as a good source for review responses as they are clearly labeled with a sentiment, "fresh" or "rotten," and tend to follow the character limit imposed by Twitter.

2 Initialization

ReelTalk relies on various objects that are generated offline and made statically available to minimize the execution time required to reply to a given input tweet. One resource used is the list of movie titles. The titles are collected from the all-movies-ever repository[2] and stored in a text file. Another required resource is a map of each movie title to their frequency count in a general domain. These counts are calculated by searching the Enron email corpus[3] for each complete movie title string and recording the number of occurrences. In this section, we elaborate on the usage of these files. A key mechanism of ReelTalk is a classifier which uses various features to determine the polarity of the input tweet. The classifier was implemented using the Nak Scala/Java library[4], which provides an API to train and evaluate classfiers. Because the training set is large and requires more time than any other ReelTalk component, the classifier is created offline to minimize the bot's response time. Lastly, two files required by the classifier, a list of positive words and a list of negative words, are also stored statically to be referenced during execution.

The classifier is trained on three sets of data with sentiment annotation, all of which relate to

[1] http://developer.rottentomatoes.com/
[2] https://github.com/samet/all-movies-ever
[3] http://www.cs.cmu.edu/~enron/
[4] https://github.com/scalanlp/nak

tweets in general or movie reviews and are, therefore, appropriate for the domain:

1. 33,000 Rottentomatoes.com movie reviews
2. 3,000 random variety tweets[5]
3. 1,000 tweets on the 2008 US Presidential debates generated for [13]

Refer to Sections 4.2 and 6 for a detailed description of the classifier and its performance.

3 Workflow

The workflow from the user tweeting at ReelTalk to the user receiving a corresponding reply consists of four main steps:

1. Extracting the movie title from the tweet.
2. Determining the sentiment of the tweet.
3. Collecting movie reviews of the desired title.
4. Designing the reply message.

4 Implementation

4.1 Movie Title Extraction

ReelTalk assumes the user will include a movie title in their tweet. Instead of restricting the format of the input tweet to simplify the title extraction, the format was left open to allow for a more interesting interaction. Upon receiving the tweet, RealTalk uses the static list of movies to find any matches in the tweet text. With this strategy, multiple title matches can occur as many movie titles are also commonly used words or phrases. Therefore, to extract the single, most likely subject of the tweet, the bot uses the stored map of frequency counts to determine the title that is least likely to occur in a general domain. The movie title with the lowest frequency is selected. If a unique lowest frequency does not exist, ReelTalk chooses the title with the longest character length.

4.2 Sentiment Analysis

Next the tweet needs to be labeled as positive or negative (fresh or rotten). First, the movie title is removed from the text before featurization as the title may consist of words with a positive or negative sentiment, adding unwanted noise to the classification. For example, the word "lovely" in the movie title, *The Lovely Bones*, would incorrectly add a positive sentiment to what may have been a negative tweet. The resulting string is then sent to the classifier. There are two features used by the classifier: bag of words (unigram model) and polarity. Narr shows that simple unigram based sentiment tagging yields good results [6]. However, in our testing we found that combining unigrams with a polarity feature provided slightly better results (see Section 6 for evaluation).

The polarity of each token is calculated by looking it up in the two aforementioned lists of positive and negative words. If the word is found in the negative list, it is labeled with -1, and similarly the words found in the positive list are labeled with 1. Otherwise, it is considered neutral and labeled with a 0. This worked well enough; however, during testing it was found that simple negated sentences, such as, "I did not like the movie" were wrongly classified as positive. According to Weigand, negation is "a very common linguistic construction that affects polarity"; therefore, during the evaluation of the classifier, particular interest was put on its performance on text with forms of negation [4].

Weigand and Pang both present several approaches to handling negation: extending bag of words features to include negation words when they precede a token (i.e. "like-not" is a different feature than "like"), using a lexicon-based sentiment analysis and assigning the opposite polarity to a negated polar expression, using POS tag patterns, and using various, more complex features to represent negation [3],[4]. ReelTalk's new method of measuring polarity with negation incorporates a few of the strategies presented in [4]. The polarity measurer was extended to incorporate trigrams in the classification of each token. If "not" or a word ending in "n't" was the previous or second previous token, the current token's polarity would be negated. For example, "I did not like the movie" and "It wasn't that good" would both be switched from positive to negative because both positive tokens have negations in their previous two tokens.

4.3 Movie Review Selection

Next, now that the movie title has been extracted and the sentiment chosen, ReelTalk queries Rotten Tomatoes for reviews for the extracted movie title. The results may consist of reviews for multiple movies, instead of a single match. This occurs

[5] `http://data.dai-labor.de/corpus/sentiment`

when the movie is part of a series, has been remade in multiple years, and when the movie title is included in another movies title. To handle this disambiguation task, the results are first searched for an exact match to the extracted movie title, as the exact match is most likely to be the correct movie to review. Because Rotten Tomatoes does not provide a list of valid or exact search terms for their movie database, the exact match is not always found in the search results. If the exact match is not found, the reviews on the first movie title returned are used.

4.4 Format Candidate Replies

The acquired reviews are sorted into a list of candidate Twitter replies based on the sentiment of the input tweet determined by the classifier. The first reply in the final list is selected as the response. The reviews with a sentiment matching the input tweet are considered first and those opposing fall to end of the list. Ideally, a review with the appropriately matching sentiment is always used as a reply message. However, there is no guarantee that every movie will have both positive and negative reviews. Additionally, the rate limit enforced by the Rotten Tomatoes API limits the quantity of reviews returned per search and may prevent access to a review with the desired sentiment. When either scenario occurs, ReelTalk replies with a review of the opposing sentiment. In order to be able to verify the correct functionality of the classifier, the string "I disagree..." is prepended to the candidate reply, acknowledging the opposition. To avoid violating the 140 character limit enforced by Twitter, any candidate reply exceeding that length is trimmed appropriately. To provide the user access to the missing text, a shortned url, via the Bit.ly API[6], to the remaining review is appended to the reply.

If no reviews are available for the extracted movie title, a default response behavior occurs which queries all of Twitter for a status that match two criteria: the tweet contains at least one of the words from the original input tweet and the sentiment of the tweet matches the sentiment of the original input tweet. To increase the relevance of the returned tweet, only non stop-words are considered for the word match. Therefore, small functional words, like "the" and "a", are removed from the tweet so that only the more meaningful words are included in the query. The returned tweet in this case usually is not related to the movie, but does provide a human-generated tweet of the same sentiment.

[6] `http://dev.bitly.com/`

5 Interactive Classifier Training

A feedback mechanism was implemented to allow ReelTalk to retrain its classifier by learning from any tweets that it assigns a sentiment to incorrectly. In the case of a misassigned sentiment, the interactive message protocol requires the user to respond to ReelTalk with the desired sentiment, "fresh" or "rotten," immediately following the incorrect response[7]. ReelTalk incorporates this correct labeling and the original input tweet into its training data[8] to avoid repeating the same error.

6 Evaluation

Two components of ReelTalk's performance were evaluated independently. First, the accuracy and F1 score of the classifier when determining the correct sentiment of a tweet or movie review was evaluated. Also, the rate at which the input tweet and the resulting reply share the same sentiment, as determined by the classifier, was calculated.

6.1 Accuracy/F1 Score

The performance of ReelTalk's classifier was tested on six datasets: four corpora of annotated tweets (Debate08, HCR, Stanford , and Emoticon), a disjoint set of movie reviews from Rotten Tomatoes, and a small handmade set of fundamental (basic) positive/negative statements, collectively labeled as "gold". The first four corpora contain tweets that are not movie-domain specific. The Debate08 set consists of tweets from the 2008 Obama-McCain presidential debate ([10],[11]). The HCR set consists of tweets regarding health care reform [12]. The Stanford set contains about 400 manually annotated tweets from the research of [13]. The Emoticon set consists of tweets that are annotated with sentiment based on the use of emoticons. For this experiment, only the positive and negative tweets from these copora were included, the neutral tweets were omitted. The last dataset mentioned was necessary in order to test the classifier on simple short

[7] The user's response must be unique. It is suggested to append the current date+time to the sentiment

[8] 33 instances of the new tweet and sentiment are added to the training data as the appropriate weighting for immediate correction.

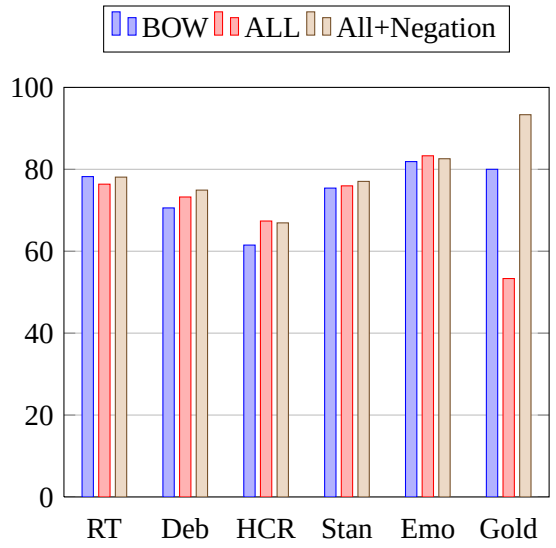

Figure 1: Accuracy of classifier per corpus on each feature set.

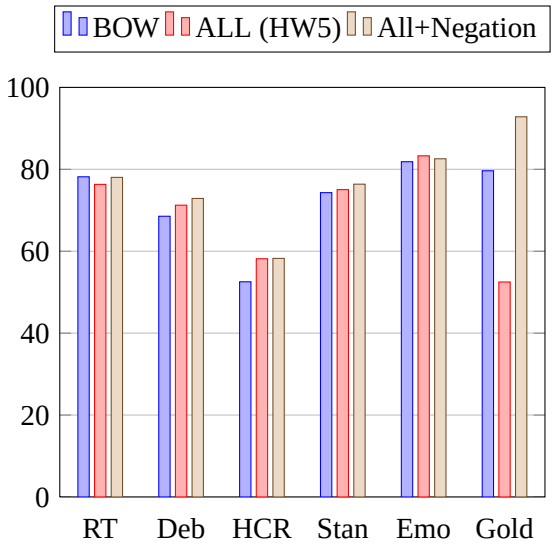

Figure 2: F1 Score of classifier per corpus on each feature set.

positive/negative statements frequently occurring in opinionated tweets that earlier iterations of the classifier were labeling incorrectly.

The results of this experiment are shown in Figures 1 and 2. The figures not only present the classifier's performance on each test dataset (named above), but also compare the performance of the classifier when using different feature sets: bag of words (BOW), bag of words and word polarity (All), and bag of words, word polarity, and the negation technique (All+Negation). See Section 4.2 for more discussion.

As seen in the figures, the classifier performs well on the movie-domain datasets. Because the classifier is trained to be movie-domain specific, it is expected that it performs well on the Rotten Tomatoes and "gold", as they both consist of only movie related tweets. However, the classifier showed nearly equal performance on the Stanford and Emoticon test datasets when compared to Rotten Tomatoes, even though they do not contain movie-specific tweets. Because the tweets from the Stanford and Emoticon corpora are not restricted to any domain, the classifier is able to recognize the polarity of the words and label with a high accuracy. The slightly worse performance seen on the Debate08 and HCR corpora is expected. These tweets are not only non-movie related, but are restricted to specific political topics. Because the language used in these tweets differs from movie reviews, the classifier does not assign the sentiment as well.

For most of the test datasets, the use of more extended features does not result in significant improvement of accuracy or F1 Score. However, the advantage of using the negation technique is clear when considering the performance on the "gold" dataset. This dataset is a simple but meaningful representation of likely input tweets. Therefore, the decision to use the All+Negation feature set was obvious.

6.2 Matching Input/Reply Sentiment

The second experiment was conducted as a way to evaluate the relevance and reasonability of the resulting tweet replies to the input tweet. To quantify this component, two sets of movie tweets and their corresponding intended sentiments were created. The first set was handmade and annotated by the authors and exhibited definitive sentiment, and the second set was gathered from a twitter user (@MovieTwoosh) who tweets movie reviews paired with a rating. If the rating was a B or higher, it was annotated as "fresh", any lower rating was considered "rotten". Each input tweet from both sets was sent to ReelTalk, and the resulting replies were recorded. To evaluate the agreement of the reply with the originally intended sentiment, the classifier was then run on both the input and the output of both sets and statistics were recorded.

This evaluation determines if, despite the fact that there is no guarantee from Rotten Tomatoes that a "fresh" review will sound positive and a "rotten" review will sound negative, the resulting

	Input F1	Input Accuracy	Output F1	Output Accuracy
Handmade	87.59	87.88	72.73	72.73
MovieTwoosh	93.89	93.94	72.63	72.73

Table 1: Performance of the classifier at labeling the input and providing expected output.

reviews still provide an appropriate response to the user. This not only further evaluates the accuracy of the sentiment classifier, but also how the entire system performs in practice toward the main goal of replying to a user in a manner resembling the input. Table 1 shows how accurately the input was labeled and also how accurately the output matched the input. The first two columns show that the classifier was very accurate in labeling the input. The latter two columns show that the resulting reply tweet matched the input tweet more than 72% of the time.

7 Future Work

The use of Rotten Tomatoes as the movie review source suffices, but introduces problems when attempting to access the reviews of the correct movie title (see Section 4.3). The reliability of ReelTalk to respond with content pertaining to the correct movie title could be improved with the use of a better movie review database.

To improve the movie title extraction task, nominal semantic role labeling, as presented in Liu et al. (2011), could be incorporated. This would involve identifying the "predicate-argument structures" of the tweet [9]. Given the restricted domain of our input tweets, the movie title is guaranteed to be an argument in that structure. This knowledge would limit the amount of text to search for movie titles in each tweet, and, therefore, also decrease the chance of error.

The classifier could be further improved by continuing to use the interactive classifier training feature. Allowing the interactive training to occur over an extended period of time, the classifier could be periodically reevaluated on the same test data (see Section 5) to verify and observe any improvements in performance. Additionally, the classifier would become more robust after longer sessions of interactive training with a variety of users. Assuming many users would participate, ReelTalk would be evaluated on several different writing styles and word usage and learn how to correctly classify any misinterpreted examples.

8 Conclusion

In this paper, we presented how sentiment analysis and other NLP mechanisms can be combined to create an interactive Twitter response bot. The bot uses a trained classifier to recognize sentiment and finds reviews online to reply to twitter users about movies. The evaluation results showed that the bot performed the sentiment classification task accurately and replied to users appropriately.

References

[1] Pak, A., and Paroubek 2010. Twitter as a corpus for sentiment analysis and opinion mining. Proceedings of LREC 2010 (2010).

[2] Yessenov, Kuat, and Saa Misailovic. Sentiment Analysis of Movie Review Comments. Methodology (2009): 1-17.

[3] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. Found. Trends Inf. Retr., 2(1-2):1135

[4] Wiegand, Michael, et al.. A survey on the role of negation in sentiment analysis. Proceedings of the workshop on negation and speculation in natural language processing. Association for Computational Linguistics, 2010.

[5] Diakopoulos. N. and Shamma, D. Characterizing Debate Performance via Aggregated Twitter Sentiment. CHI 2010, ACM, 2010.

[6] Narr, Sascha et al. Language-Independent Twitter Sentiment Analysis. 2012.

[7] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. 2011 Sentiment analysis of twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011), pages 3038, Portland, Oregon, jun. Association for Computational Linguistics.

[8] Liu, K.; Li, W.; and M. Guo. Emoticon Smoothed Language Models for Twitter Sentiment Analysis. In Proceedings of AAAI. 2012.

[9] Liu, X.; Zhang, S.; Wei, F.; and Zhou, M. 2011c. Recognizing named entities in tweets. In ACL, 359367.

[10] David A. Shamma; Lyndon Kennedy; Elizabeth F. Churchill. 2009 Tweet the Debates: Understanding Community Annotation of Uncollected Sources. ACM Multimedia, ACM.

[11] Nicholas A. Diakopoulos; David A. Shamma. 2010. Characterizing Debate Performance via Aggregated Twitter Sentiment CHI 2010, ACM.

[12] Michael Speriosu, Nikita Sudan, Sid Upadhyay, and Jason Baldridge. 2011. Twitter Polarity Classification with Label Propagation over Lexical Links and the Follower Graph. In Proceedings of the First Workshop on Unsupervised Methods in NLP. Edinburgh, Scotland.

[13] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. Unpublished manuscript. Stanford University, 2009.