Sentiment categorization of Beer Reviews

Sunny Changediya (A20353568)
Department of Computer science,
Illinois Institute of Technology,
schanged@hawk.iit.edu

Abstract:

In the era of the web, a huge amount of information is now flowing over the network. Since the range of web content covers subjective opinion as well as objective information, it is now common for people to gather information about anything that they wish for. We analyze over 5000 beer reviews from a week period and find that tracking small number of reviews allows us to identify actual sentiment towards beer with accuracy of 73%. However since a considerable amount of beer reviews exists as text-fragments and most of them with tokens such as bitter, malty, sour, drinkable etc. which are negative in general, but considered as positive while describing beer[1]. This makes it hard to achieve higher accuracy and sometimes without having any kind of numerical scales, it is hard to classify their evaluation efficiently without reading full text. This paper focuses on extracting text fragments from the web and suggests various experiments in order to improve the quality of a classifier. We find that this sentiment classifier can reduce error rates by over half in parse tree generation experiment, though more research is needed to develop methods that are robust in cases of extremely vague and fragmented text[2].

1 Introduction:

There has been growing interest in the quality and taste of Beer. Given easy access to social network, the immense amount of subjective content such as place, bars, and beer reviews on the web today, we can derive a great deal of benefit by being able to interpret the opinions and sentiment put forth in these reviews. Analyzing beer reviews generally tends to be more difficult because of completely different term analogy used to describe beers[1]. Eg. Bitter and sour are generally used to mark good beer but it is not the same case with other reviews. After careful analysis, it is found that, unlike other product reviews, beer reviews are more complex to analyze and make sense out of it because they generally tend to be more fragmented and most of the text describe place, ambience, location, service and color rather than actual taste. This brought us to classify beer reviews for sentiment analysis.

In this project, we apply natural language processing techniques to classify a set of beer reviews based on text fragments. More specifically,

- We develop Logistic Regression classifier to classify and predict reviews as an indicator of beer sentiment.
- We implement a set of features that we believe to be relevant to the sentiment expressed in reviews and analyze their effect on performance, providing insights into what works and why sentiment categorization can be so difficult specifically on Beer reviews where there is so much of noise in terms of different word terminology, noisy texts related to description of place, location, crowd, and Beer.

- We also examine effect of POS Tagger, and Parse tree generation on overall sentiment classification.

In this paper, we report results of our analysis over 7000 Beer reviews collected from RateBeer.com and BeerAdvocate.com.

2 Dataset Analysis:

We begin with a description of the data used in all experiments.

2.1 AFINN Sentiment:

AFINN is a list of English words rated for valence with an integer between minus five (negative) and plus five (positive). The file is tab-separated. This data is used to calculate afinn score of a review and classify them as positive or negative based on afinn score[6].

2.2. Beer Reviews:

Our dataset consists of around 10,000+ beer reviews from RateBeer.com, an online social network for finding good beers. Each of the rating page has 15 beer profiles and there are 100 most recent rating pages. So we crawl for 1500 beer profiles out of which 1300 are unique profiles. Each Beer profile in turn has 6-7 average reviews per page and can span up to 100 pages of reviews per Beer profile. To balance dataset, only 6-7 reviews per beer profile are crawled.

Dataset also consists of reviews from BeerAdvocate.com, another social network which maintains beer database. This dataset is crawled for most-recent reviews which are used as test data for predicting classifier output.

The RateBeer and BeerAdvocate datasets are crawled using custom developed crawlers in python. The class RateBeer() and BeerAdvocate() provides necessary functions to crawl for reviews.

For the purpose of development, we use only a small portion of training and testing set reviews. For training set, we use 2000 reviews from RateBeer and for testing set, we use 3000 reviews from BeerAdvocate.

As a final note, the beer profiles and reviews are mixed with all languages and countries. So we only select reviews which are written in language ‘en’.

2.3 Analysis:

It is very difficult to handle beer reviews in general. We noticed that mostly beer reviews are rated using multi-class labeling. Which means their average rating depends on taste, place, aroma, appearance, and palate. This adds considerable noise in overall beer ratings and it becomes difficult to classify which beer is actually good.

Consider following review:

{"review": "Draught at Haket Batch#2 Beer Festival Cons 2017-04-15 Göteborg AR dried tropical fruity AP cloudy bronze wee white head F mandarin dried tropicak fruity grainy", "name": "Magic Rock Cannonball", "type_score": {"appearance": 3.0, "palate": 3.0, "taste": 2.5, "aroma": 3.5, "overall": 4.0}}

- The reviewer has given rating for taste = 2.5 and overall = 4.0. The taste rating for beer is very less and can be considered bad beer taste compared to mean ratings of all beers. But the overall rating has score = 4.0 which suggests beer is good.

Sometimes user reviews are nonsensical as in the review below.

{'type_score': {'overall': 4.25, 'aroma': 3.0, 'palate': 5.0, 'appearance': 4.0, 'taste': 2.5}, 'name': 'North Coast Brother Thelonious', 'review': 'Bottle Dark brown colour with ruby reflexes rocky head that does not stay long Aroma is quite much candy sugar raisin pleasant warming alcohol Medium sweetness not bitter Soft carbonation smooth not cloying Not very complex but well balanced and easy to drink Good'}

- The review has scored taste = 2.5 and overall = 4.25 which is quite absurd. The review text says little about beer taste in general but more about place, aroma and other noisy stuff. It is very difficult to classify beer sentiment from noisy text because text is very much about appearance of beer and place but nothing about taste. Still reviewer ends up saying ‘easy to drink’ and score taste = 2.5 but overall score suggests this is worth a try.

Some reviews are much fragmented, in the sense that they does not form a sensible review to understand.

{'type_score': {'overall': 4.25, 'aroma': 3.5, 'palate': 4.0, 'appearance': 4.0, 'taste': 4.0}, 'name': 'Marché du Village - Série Impériale - Barley Wine', 'review': 'Aroma sugar,vanilla,malty Appearance:clear cooper brown,light white-beige head Flavor:strong caramel-malt,good bitters,vanilla oak...nice'}

Most of the Beer reviews are described with completely different terminologies that other product reviews like restaurant reviews.

{"name": "Delirium Tremens", "review": "Yellow straw lightly hazy soapy white head Aroma is yeasty bananas spice similar to weizens but more intense Taste has sweetness sourness and bitterness in good balance honey fruit spices and ethers Body is rather thick soft oily texture Very good", "type_score": {"palate": 4.0, "aroma": 3.5, "overall": 3.75, "appearance": 3.0, "taste": 4.0}}

The beer review here is quite predictive of particular user’s choice of taste. Some individual may like bitter beer, and some may like sweet one. The ‘bitter, sour’ terms are usually indicate positive beers for most of the user’s while this is not the case with other product reviews such as restaurant reviews.

The general analysis shows that, Beer reviews are most fragmented, with lots of noise in terms of beer, place, aroma description and little sentiment about actual beer taste. Also most of the terms used to describe beer sentiment have different terminologies compared to other product reviews in general.

3 Feature Selection:

Here we describe the techniques employed to classify the sentiment of reviews. We describe the successes and failures of the various techniques, present insights we learned regarding multi-class sentiment categorization, and propose future improvements to our methods. The most important part of successful classification is selection of appropriate features. We attempted numerous different standard methods of deriving a useful set of features.

3.1 Stop Words:

We recognized that classifying bag-of-words, many words provide no useful information for sentiment analysis. Articles such as ‘a’ and ‘the’ and pronouns such as ‘he,’ ‘they,’ and ‘I’ provide little or no information about sentiment. So we decided to remove stopwords from reviews which help reduce vocabulary size. We use nltk stopwords library to remove stopwords. There are 153 unique stopwords in the library.[7]

3.2 Pruning:

One problem with beer database was dealing with wide variety of review sizes. Sometimes, beer reviews are long and they tend to explain place, location, and service in brief. Others are short reviews which are mostly fragmented with general description about aroma, and good/bad about beer. Two examples are as follows:

“A Mes rate Cask in the Cask Cutler in Sheffield.Black brown colour with a thin off white head Excellent coffee malt aroma with some awesome chocolate The flavour is more of the same but slightly thin and rather minerally one of those rate words that you just have to make up sometimes A little raw in the finish Not the best Pictish but still better than plenty of other brewers offerings"

"Bottle at home a while back Pale clear golden pour Fruity aroma and taste with banana citrus some yeast spices hay and honey Nice balance Great beer”

Both the reviews are different with small review giving more information and long one just noise. We implement pruning feature to prune vocab based on frequency in the whole document.

3.3 TF-IDF Score:

It is revealed that most of beer reviews use words food, bitter, sour, aroma, pale, drink, pint etc. while most of important tokens suggesting beer sentiment are very rare in overall document. To account for this imbalance, we use tf-idf score as a feature for each token in a document. The tf-idf value increases proportionally to the number of times a word appears in the document, but is often offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

- *TF(t) = (# term t appears in a document) / (# terms in the document).*
- *IDF(t) = log(# of documents / # of documents containing term ‘t’).*

3.4 POS Tagger:

We implement NLP'ish feature representation using Stanford POS tagger to tag each token with part-of-speech tagging. Given most of the reviews are fragments of terms, POS tagger helps tagging important parts of text and relate them easily for sentiment classification. The rules extracted from POS tagger includes trigram pair of tokens which follow: [8]

- *JJ-->NN,*
- *RB-->NN,*
- *NN-->VBZ-->JJ,*
- *NN-->IN-->JJ*

Careful analysis shows that there rules generally describe meaningful information about sentiment in general. The meaning for rules is described in Stanford POS tagger manual.

3.5 Word2vector:

In this feature analysis, each token is represented as sparse vector of probabilities using genism library for word2vector generation. The word2vec model is generated with parameters hs=1 and negative=0 to get score probability of whole sentence as a review text. Sentence score is used as a feature to represent probability of sentence[9].

3.6 Parse Tree (Dependency Parser):

This is the most important feature as we generate parse trees for all reviews using Stanford DependencyParser and add feature for words which modifies meaning of sentence. The rules for feature extraction follows:

- *(NN, amod, JJ)*
- *(## , nsubj, NN)*
- *(NN, amod , JJ)*
- *(RB, advmod, JJ)*
- *(VBD, nmod , NN)*

This feature help us identify modifiers to actual beer sentiment such as:

"that beer, was bitter, but with great taste". The beer(NN) modifies (great taste) and "great(NN) modifies taste(adv)".

3.7 Afinn:

AFINN feature set is used to identify positive and negative words score in a sentence. The review is featurized with overall positive and negative score of tokens in a sentence. This helped us identify overall sentiment of a review.

3.8 Token Pair:

In token pair features, we iterate over tuples of trigrams and tried to capture information such as *"but great taste", "not that good".* Token pair features are implemented using window size of k=3 to iterate over each review in a sentence.

4 Sentiment Analysis:

To evaluate our accuracy with respect to the classification task, we use a number of metrics that reflect the quality of our predictions. The first of these is simple training and testing accuracy of various combination, which we calculate using cross-validation accuracy. Second we use k-fold cross-validation accuracy on test data set, which measures accuracy of training set over K-folds of iteration.

We find it important to consider both of the above metrics when evaluating the performance. As a simple example, a rating of overall=5 is the most common among the reviews so we can get a precision of 0.464 by blindly labeling each review as positive; however, the cross-validation accuracy for labeling all reviews with a rating of overall=5 is 0.852. Thus, labeling all reviews as positive gives us better precision, but also means that we are further off on our incorrect predictions. Given these baselines of 0.464 for precision and 0.852 for cross-validation accuracy, we attempt to improve our sentiment categorization performance through machine learning and natural language processing techniques, which we detail below.

4.1 Implementation:

We experimented with different methods of preprocessing the data. Because the reviews are unstructured in terms of user input, reviews can look like anything from a paragraph of well-formatted text to a jumble of seemingly unrelated words to a run-on sentence with no apparent regard for grammar or punctuation.

1. Our initial pass over data tokenized reviews based on whitespace, asci characters, and numbers and treated each token as a unigram. Generally integers such as "33ml bottle", "2 1/2 – glass beer" does not contribute to overall sentiment of review. So we tokenize text and replace integers with <NUM> token. It is noticed that we were able to improve performance by removing punctuation in addition to the whitespace and converting all letters to lowercase. So we treat occurrences of "Awesome" and "awesome" as equal.
2. Then we implement all feature mentioned in section 2.
3. The parse tree generation is kept as an optional feature extraction as it takes a while to generate parse tree for a sentence. It is observed that, it takes on average 500 seconds per 100 reviews to generate parse trees with each text length <= 100 tokens. Here we generate parse tree for all reviews at once and store them in a file for faster and improved feature extraction.

We adapt to logistic regression classifier to implement sentiment classification for beer reviews data. We train classifier using different combinations of settings of feature function implementation and pruning frequencies and find best model setting with highest cross-validation accuracy.

The settings include combinations of feature functions as follows;

E.g. [(token_pair, pos_tagger, word2vec), 5], [(word2vec, token_pair), 4], [word2vec, 5] where numeric term indicates pruning frequency to vectorize method.

We find best setting of model with highest cross-validation accuracy to fit the model again for training samples of reviews data and predict on test review data. The final analysis involves predicting probability of classification on test data and reporting top-misclassified reviews with highest probabilities for analysis.

4.2 Evaluation:

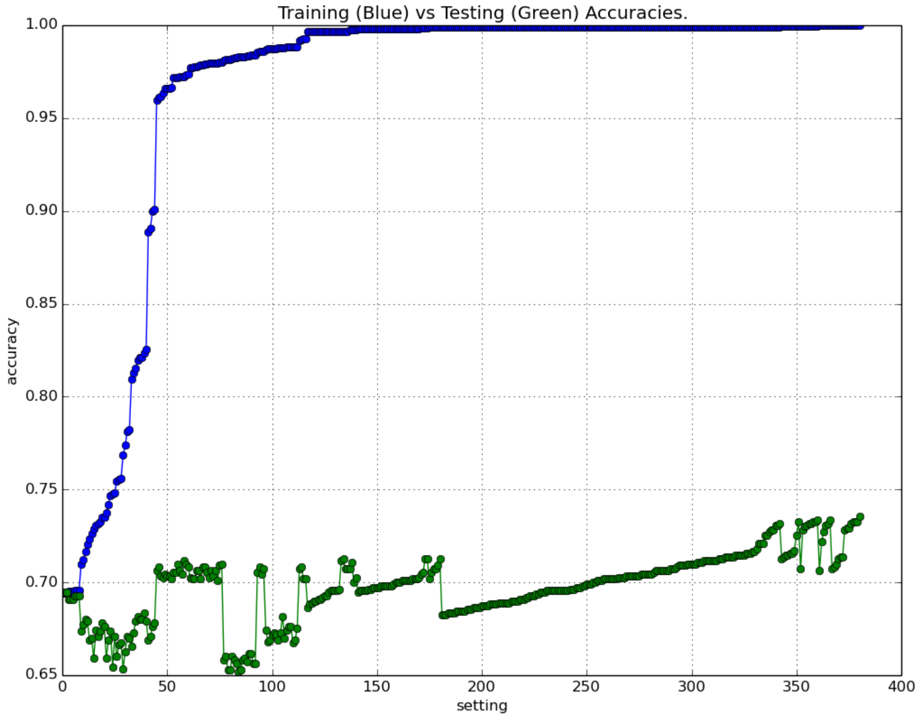

Figure-1: Training(Blue) vs Testing(Green) Accuracy

To evaluate results, results of 10-fold cross validation accuracy is shown in figure-1. The test set accuracy with best setting of model, turns out to be 74%, as shown in figure-1. This suggests that our classifier performed reasonably well compared to other classifiers. The results are quite disappointing in the sense that POS tagger and Parse Tree generation doesn't improved accuracy to a greater extent but bag-of-words feature set still contributes largely towards correct classification of sentiments. The fact that beer reviews are mostly fragments of words and with completely different word terminology, suggests that there is a need of filtering out spurious word sentiments which can affect classifier analysis. As explained earlier, review with "bitter, sour, pale' might be classified as negative by the classifier but in the training set they may be labelled as positive review.

5 Conclusion:

From above feature implementation, we come across very firm conclusion about sentiment classification on data which is noisy and fragmented in nature.

- We observed that, most of the beer reviews are fragmented. This affects POS tagger and Parse tree feature implementation, as accuracy remains low compared to simple bag of words.
- The sentence structure and word terminology differs from that of other product reviews. The top misclassified reviews include words like bitter, sour, not bad etc. This suggests that classifier is misguided and treats bitter, sour as negative words.
- First, stopwords proved to be useful initially in improving our classification performance, but turned out to be harmful when used in conjunction with a sentiment model.
-

The problem of generating a high quality sentiment categorization system is a complex and intricate one with lots of exciting applications. Sentiment is expressed in so many complex ways that it is impossibly difficult to model every minute detail, but with the right approach and enough patience, we can build a classifier that performs reasonably well at predicting the rating for a review given its text. In most aspects of implementation, each change that we make can help us predict sentiment for some reviews that we previously missed, but can also cause us to miss reviews that we had previously categorized correctly. By tinkering with the many tradeoffs and complexities of sentiment categorization, we have arrived at a reasonably well performing model that can successfully predict more than 60% of our reviews; however, as our evaluation shows, there is still a long way to go and there is always much that can be done to improve.

6 Acknowledgements

We would like to thank Prof. Aron Culotta (aronwc@iit.edu), for valuable suggestions throughout the implementation phase and, more importantly, for his guidance in the conception and execution of this project.

7 References:

1. Amir Ghazvinian. Star Quality: Sentiment Categorization of Restaurant Reviews
2. Aron Culotta. Detecting influenza outbreaks by analyzing Twitter messages
3. Dave K, Lawrence S, Pennock DM. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In Proceedings of WWW. 2003; 519-528.
4. Kennedy A, Inkpen D. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. Computational Intelligence. 2006
5. Pang B, Lee L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In Proceedings of the ACL.2005; 115-124.
6. Finn Årup Nielsen. "A new ANEW: Evaluation of a word list for sentiment analysis in microblogs", Proceedings of the ESWC2011 Workshop on 'Making Sense of

Microposts': Big things come in small packages 718 in CEUR Workshop Proceedings: 93-98. 2011 May. http://arxiv.org/abs/1103.2903

7. Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. In LREC 2006.
8. Kristina Toutanova and Christopher D. Manning. 2000. Enriching the Knowledge Sources Used in a Maximum Entropy Part-of-Speech Tagger. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC-2000), pp. 63-70.
9. Gensim: Petr Sojka, proceedings of the LREC 2010 workshop on new challenges for NLP Framework.