Proc. of the Intl. Conf. on Advances in Computing and Information Technology-- ACIT 2014

ISBN: 978-981-07-8859-9

Sentiment Analysis using Naïve Bayes with Bigrams

Mohd Abdul Hameed Adnan Rashid Hussain S. Fouzia Sayeedunnissa

***Abstract*— With the rapid growth of reviews, ratings, recommendations and other forms of online expression, online opinion has turned into a kind of virtual currency for businesses looking to market their products, identify new opportunities and manage their reputations. Sentiment analysis extracts, identifies and measures the sentiment or opinion of documents as well as the topics within these documents. The Naïve Bayes algorithm performs a boolean classification i.e. it classifies a document as either positive or negative according to its sentiment. We have already seen by Sayeedunnisa et al [1], that the application of Naïve Bayes trained on high value features, extracted from a bag-of-words model, yields an accuracy of 89.2%. This paper studies the application of Naïve Bayes technique for sentiment analysis by including training of bigram features to improve accuracy and the overall performance of the classifier. We also evaluate the impact of selecting low vs. high value features, calculated using the concepts of Information Gain. Our dataset constitutes of tweets containing movie reviews retrieved from the Twitter social network, which were obtained and analyzed on a cloud computing platform. Our experiment is divided into three steps; the first step constitutes of selecting high value features (words) from our bag-of-words model. The next step involves the identification and calculation of the probability of co-occurrence of words within the bag-of-words to derive a set of bigrams. We then used this set and the original features to re-train and test our classifier. In the final step, we selected the most informative features (unigrams + bigrams) using a Chi-Square scoring function, which yielded the best result with accuracy at 98.2%, positive precision 98%, positive recall 98.4% and negative recall 98%. It is evident from the results, that Naïve Bayes performs the best when including only the most informative (high value) features which constitute of both unigrams and bigrams for training.**

Keywords— ***Naïve Bayes, Information Gain, Sentiment Analysis, Social Network, Twitter, Cloud Computing***

Mohd Abdul Hameed
Department of Computer Science & Engineering, University College of Engineering (A), Osmania University, Hyderabad, India

Adnan Rashid Hussain
Research & Development, Host Analytics, Inc, 101 Redwood Shores Pkwy, Ste 101, Redwood City, CA 94065

S. Fouzia Sayeedunnissa
Department of Information Technology, M.J. College of Engineering & Technology, Sultan-ul-Uloom Education Society, Hyderabad, India

I. Introduction

In recent years there has been rapid growth in on-line discussion groups and review sites where a compelling feature of the posted articles is their sentiment, or overall opinion about the post. Identifying these articles with their sentiment would provide concise summaries to readers. These articles are part of the appeal and add value to such sites.

Sentiment analysis is a topic within information extraction. It plays a major role in marketing research, where companies wish to find out what the world thinks of their product. It is also used in monitoring newsgroups and forums, where fast and automatic detection of scintillating flames is necessary and for analysis of customer feedback. Sentiment classification is helpful in business intelligence applications [6] and recommender systems [7] (e.g., Terveen et al. (1997), Tatemura (2000)), [8] where user input and feedback could be quickly summarized according to their sentiment. The sentiments can be categorized in two categories: positive and negative. In this respect, a sentiment analysis task can be interpreted as a classification task where each category represents a sentiment.

This paper presents the experimental results of a comparative study that evaluates the effectiveness of Naïve Bays Classifier and shows that the use of Information Gain and Chi Square can improve the effectiveness of sentiment analysis. We have addressed the problem of accurately classifying the sentiment in posts from micro-blogs such as Twitter. Opinion mining is having a great demand in research arena due to its difficulty as well as potential benefits in trend analysis [2][5]. Early work at the document level involved applying different methods for classifying a document's polarity as positive or negative [2][9]. Determining the general attitude of users towards a product or service, for example, can help a business measure overall consumer attitudes and customer satisfaction, and it can also provide a warning when there is a sudden change in sentiment.

II. Corpora

Our objective was to perform sentiment analysis on messages (tweets) from twitter containing mentions of movies. Twitter is a micro-blogging site, in which messages shared called tweets have a limit of 140 characters. We selected the top movies based their box-office performance using the Internet Movie Review Database (IMDB) website as reference. We then identified the official twitter accounts or hash-tags representing these movies. We then short-listed those accounts and hash-tags which were published as trending by twitter, thus ensuring that we would be able to extract sufficient content for our analysis. Using these keywords, we queried twitter using the Twitter Search API and collected the tweets. We analyzed the collected tweets and prepared a selection/transformation process to prepare the dataset for our experiment.

Our first step was to extract tweets which were in English and discarded the ones which were in foreign languages. Although our experiment was targeted towards tweets in English, the approach can also be used for other languages. We then identified the emotional tone of the tweets (i.e. objective vs. subjective) and discarded the objective tweets. We found that a large portion of the collected tweets were objective in nature such as "Planning to see #inceptionfilm this weekend" and did not convey any sentiment. The remaining tweets were then hand labeled for supervised training, based on the understanding of the English language and using a Thesaurus as a reference. We hand labeled tweets as either “positive” or “negative” using a custom web form of our cloud application. Since we needed to train the model appropriately for both positive and negative tweets, we chose equal amount of tweets for both classes so as to prevent over-training for any one class. Using the above parameters our dataset constituted of 5000 tweets, of which 75% were used for training and 25% were used for testing. Using this corpus we applied the 10-fold cross-validation technique for training the model to achieve the best results. The advantage of this technique over repeated random subsampling is that all observations are used for both training and testing, and each observation is used for validation exactly once.

III. Architecture

The application was developed in Python using Google's Platform-as-a-Service (PaaS) offering called Google App Engine (GAE). We used various services from GAE such as Task Queues, Backends, etc. which aided in the implementation of the sentiment analysis process workflow. As discussed in the corpora section, we prepared a list of keywords for movies which we wanted to collect information for and queried the Twitter Search API.

The resultant tweets were stored as a bag-of-words along with its hand-tagged sentiment class label. We then used conditional frequency to identify the most co-occurring pair of words and converted the unigram features to bigrams. Using this data as baseline, our experiment was conducted by modifying the parameters of the Feature Selection process. Once the appropriate features were selected for training we used a Naïve Bayes algorithm to train a sentiment classification model. This trained model was then used to verify the results on the test dataset and calculate and compare the performance of the model using various metrics such as Accuracy, Precision and Recall

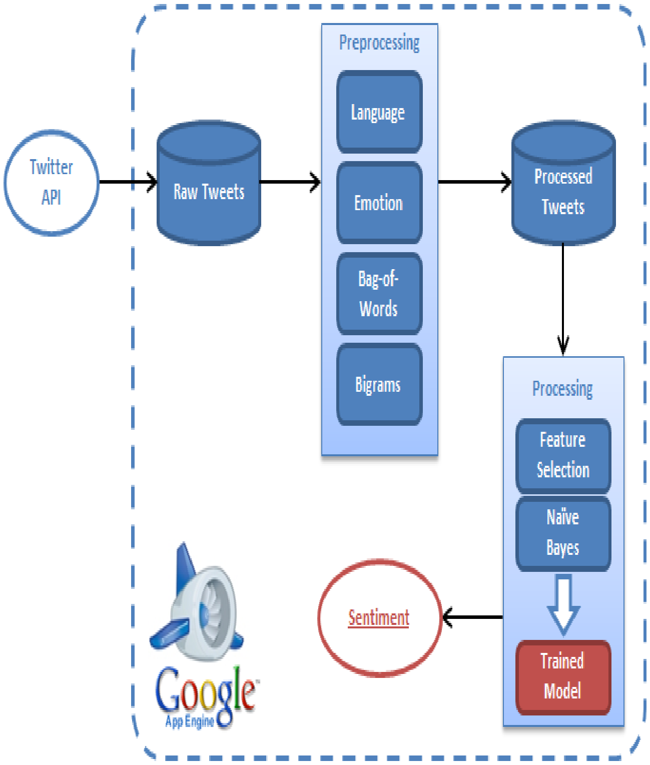

Figure 1. Architecture Diagram

IV. Experiment

In this experiment we use several feature selection techniques and apply the same to the bag-of-words model to achieve higher accuracy, precision and recall. Our intent is to eliminate low information features and thus give the model clarity by removing the noisy data. Since we are dealing with a high volume of data (words), the dimensional space greatly increases causing the available data to be sparse and less useful. Thus we needed to avoid over-fitting and the curse of dimensionality, by decreasing the size of the model which ultimately improves the performance. We have already seen that selecting high information features for training the Naïve Bayes classifier using an Information Gain scoring technique, yields better accuracy, precision and recall [1]. We then proceeded to identify bigram features and also apply our high information feature selection technique to achieve a much higher and profound performing model.

A. *High Information Bag-of-words*

Feature selection is an important part of the process. The classifier contains hundreds or thousands of features and it needs to identify low information feature and eliminate them. These are features that are common across both sentiment classes and therefore contribute little information to the classification process. Individually they are harmless, but in aggregate low information features can decrease the classifier performance.

To find the highest information features, we need to calculate information gain for each word. One of the best metrics for information gain is chi square. To use it, first we need to calculate a certain statistics for each word: its overall frequency and its frequency within each class. This is done with a Frequency Distribution for overall frequency of words, and a Conditional Frequency Distribution where the conditions are the class labels. Once we have those numbers, we can score words with the Chi Square function, then sort the words by score and select the features which have a minimum score. We then prepare a set of these words, and use it as a reference set for including any words that we train the model with. Now each document (tweet) is classified based on the presence of these high information words. Using this technique, we again trained the Naïve Bayes classifier and following are the test results:

TABLE I. HIGH INFORMATION BAG-OF-WORDS

Metric	Score
Accuracy	0.892
Positive Precision	0.837
Positive Recall	0.976
Negative Precision	0.966
Negative Recall	0.812

These results will be considered as baseline metrics for our further experiments comparison. From the above results, it is evident that the classifier performs better when classifying positive documents compared to the negative documents. One possible reason for this result could be the usage of a combination of words which gets classified incorrectly. Consider an example in which the word “like” gets identified with a positive polarity and even “not like” gets identified as positive. This is fundamentally due to the nature of the bag-of-words model, which assumes that every word is independent, and the model is never able to learn for the co-occurrence of such words and its implication to the classification process. Thus we proceed to train our model by also including multiple words.

B. *Bigram Collocations*

We proceeded to identify bigrams by calculating certain statistics. We calculate the frequencies for each word and the conditional frequencies of all combination of words. This is done to ensure that the identified bigram occurs more often than expected and not by chance. We then trained the classifier using the individual words (unigrams) and our extracted bigram features, with the following result:

TABLE II. BIGRAM COLLOCATIONS

Metric	Score
Accuracy	0.746
Positive Precision	0.669
Positive Recall	0.972
Negative Precision	0.948
Negative Recall	0.52

As expected by our hypothesis, the positive precision had reduced considerably. As we concluded in our earlier experiment, this was due to the fact that we included all bigrams with our words for training. In the next step of our experiment we proceeded to include only significant bigrams along with our high value words (unigrams).

C. *High Information Bigrams and Unigrams*

Using the same technique as detailed in our first step of the experiment (6.1), our objective was to prepare a set of the most informative features which could include both bigrams and unigrams. Using these high information features ensures that the accuracy of prediction improves drastically. Using the Information Gain technique, we applied the Chi-Square function or all our features. We then extracted the features which had a minimum score value of 3, sorted them as per their score and prepared a reference set. By including on the training features which were members of the reference set, we then trained our Naïve Bayes classifier and ache vied the following test results:

TABLE III. HIGH INFORMATION BIGRAMS AND UNIGRAMS

Metric	Score
Accuracy	0.982
Positive Precision	0.980
Positive Recall	0.984
Negative Precision	0.983
Negative Recall	0.98

As shown, there is a considerable increase in all the metrics indicating that the classifier performs the best, when trained using only the most informative bigram and unigram features.

V. Graphs

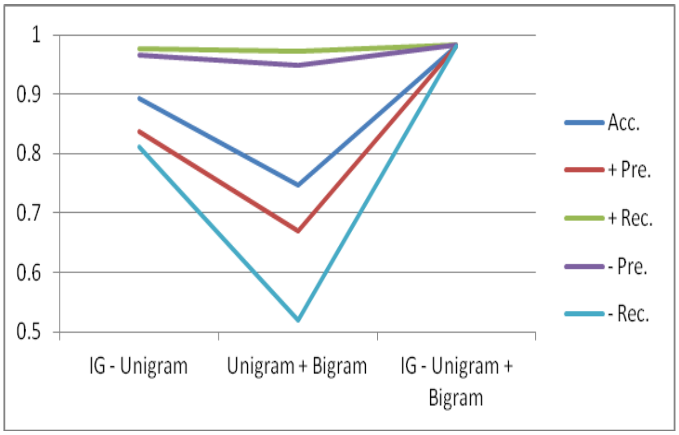

Figure 2. Accuracy vs. Precision vs. Recall

The evaluation metrics used for this experiment are Accuracy, Positive Precision, Positive Recall, Positive & Negative Precision and Negative Recall. These metrics have been plotted for high value features (unigrams) by applying Information Gain, all unigrams + bigrams features and for high value features (unigrams + bigrams) by applying Information Gain. The performance is compared on the prediction over its classes; positive and negative.

As expected, the classifier performed best when dealing with high value features (unigrams + bigrams), with an accuracy of 98.2%. This was achieved by calculating the score of the unigrams + bigrams using Information Gain, and selecting only high value features based on a certain criteria. The high information unigram features bag-words-model (Step 1) gave an accuracy of 89.2%. In this model, the classifier showed better performance in classifying positive documents, than compared to negative documents. When including the bigram features (step 2) into the training set, the accuracy of the classifier dropped to 74.6%. There was also considerable decrease in Positive Precision which dropped from 83.7% to 66.9% and Negative Recall which dropped from 81.2% to 52%. The performance for other metrics also decreased, but it was relatively very less. This was a clear indication that the classifier suffered from over-fitting and the curse of dimensionality. We also noticed that the predictions were not equally balanced for both the classes and this definitely required improvement. When selecting only high value features by applying the concepts of Information Gain using a Chi-Square scoring function, we were able to tremendously improve performance on all the metrics (step 3). In this experiment we achieved performance of 98.2% Accuracy, 98% Positive Precision, 98.4% Positive Recall, 98.3% Negative Precision and 98% Negative Recall. Considering these results, we can notice that although positive documents predictions were more accurate compare to negative documents, in the final step the performance for both the classes is almost equal and concluding that the model is balanced. We can also conclude that by removing low value features, it brings clarity to the model by removing noisy data and saves it from over-fitting.

VI. Conclusion

In this paper, we evaluated the implementation of several feature extraction and selection techniques to improve the performance of the Naïve Bayes classifier. The classifier was trained and tested on the social network Twitter data, and aimed to extract the sentiment / opinion of the individual Twitter messages (tweets). By selecting and using only high information features (bigrams and unigrams) for training the classifier, we were able to achieve good performance for the specific used training set.

Using the above technique, the classifier achieved an accuracy of 98%, while also displaying equally promising results for precision and recall over both the classification classes. We can also consider the implementation of semantic orientation as an extension of this study for future research.

References

[1] Sayeedunnisa, S.F., Hussain, A.R., Hameed, M.A.: Supervised Opinion Mining of Social Network data using a Bag of word Approach on the cloud. In: Seventh International Conference on Bio-Inspired Computing: Theories and Application (2012)

[2] Hussain, A.R., Hameed, M.A., Hegde, N.P.: Mining Twitter using Cloud Computing. In: World Congress on Information and Communication Technologies, pp. 187-190 (2011)

[3] Pang, B., Lee, L.: Opinion Mining and Sentiment Analysis. In: Foundations and Trends in Information Retrieval (2008)

[4] Das, S.R., Chen, M.Y.: Yahoo! for Amazon: Sentiment Extraction from Small Talk on the Web. In: Management Science, pp. 1375-1388 (2007)

[5] Forman, G.: An extensive empirical study of feature selection metrics for text classification. In: The Journal of Machine Learning Research, pp. 1289-1305 (2003)

[6] Burgoon, J.K., Blair, J.P., Qin, T., Nunamaker, J.F.Jr.: Detecting deception through linguistic analysis. In: Proceedings of the 1st NSF/NIJ conferences on Intelligence and Security Informatics, pp. 91-101 (2003)

[7] Terveen, L., Hill, W., Amento, B., McDonald, D., Creter, J.: PHOAKS: A system for sharing recommendations. In: Communications of the ACM, pp. 59-62 (1997)

[8] Tatemura, J.: Virtual reviewers for collaborative exploration of movie reviews. In: Proceedings of the 5th International conference on Intelligent user interfaces, pp. 272-275 (2000)

[9] Pandey, V., Iyer, C.: Sentiment analysis of microblogs. Technical Report, Stanford University (2009)

[10] Lewin, J. S., Pribula, A.: Extracting Emotion from Twitter. Technical Report, Stanford University (2009)

[11] Pang, B., Lee, L., Vaithyanatan, S.: Thumbs up?: sentiment classification using machine learning techniques. In: Proceedings of the ACL-02 conference on Empirical methods in natural language processing, pp. 79-86 (2002)