Application of Data Mining on Car Brands Using Twitter

Sai Duth Deekshit Garimella

Dept. Computer Science

Kent State University

[sgarimel@kent.edu](mailto:sgarimel@kent.edu)

Rohith Varma J

Dept. Computer Science

Kent State University

[rjampana@kent.edu](mailto:rjampana@kent.edu)

Kavya Amireddy

Dept. Computer Science

Kent State University

[kamiredd@kent.edu](mailto:kamiredd@kent.edu)

Saaketh Lingamneni

Dept. Computer Science

Kent State University

[slingamn@kent.edu](mailto:slingamn@kent.edu)

***Abstract -* In this project tweets were collected on car brands from the twitter API and cleaned to perform the sentiment analysis. This was done using two approaches and after which Naïve Bayes and decision tree classification algorithms were applied to find the sentiment of the tweet. Finally, the car brands were ranked based on the number of positive and negative tweets that resulted from executing the above-mentioned method.**

1. **Introduction**

In this modern era, technology is shifting from traditional to automated ways. Using traditional methods for classification of car brands is very time consuming and expensive process. Also, it’s very difficult to consider all of the factors that are crucial in classification. Hence, to overcome this, certain modern techniques were followed. Since nowadays people widely use social media to share their ideas, interests and opinions about different things we thought, considering user opinion in classification of car brands would yield better and genuine results.

Among all the social media platforms, Twitter is one of the largest social networking site with around 313 million active users. This motivated us to consider user posts (tweets) from twitter for the brand classification. Also, twitter provides an API which provides access to the data from users, tweets, and timelines. Relevant tweets can also be obtained by using keywords or hash tags. There are many other social media sites which are more popular than twitter but they restrict the access to the data. While on the other hand, twitter provides access to the data that can be used by both marketers and consumers.

In [1] Bing Liu defined the sentiment analysis and Opinion Mining as the study through which we examine people’s feeling, assessments, opinions, sentiments, evaluations, attitudes and emotions from composed dialects. It is the process of using NLP (Natural Language Processing), text analysis and other features to obtain the characteristics in a sentence that are useful in finding its sentiment.

Tweets were collected related to different car brands and classified into positive or negative sentiment. Then the brands were ranked based on the number of positive and negative opinions (tweets) that resulted on the car brands from the analysis. In doing so, two different approaches were considered for classifying the sentiment.

` One is the feature vector method [2] with which each tweet was tokenized and each word was added into the feature vector. Earlier to performing this, all the stop words, punctuation marks and other special characters were removed, which don’t play any role in finding the sentiment [3]. The presence of the word in the tweet was considered as 1 and absence as 0 and labelled it as positive or negative.

Then machine learning algorithms like Naïve Bayes algorithm and decision tree algorithm were applied for classification. For a new tweet, tokenization was done to remove the unnecessary words and symbols. The tweet was then classified as positive or negative using the previous patterns of the trained classifier. But when this algorithm was applied on a new tweet, and if the words were not in the feature vector of the training set then the tweet was assigned with default label i.e., Positive. This method worked well with large data and with unigrams, but negated statements like ‘not good’ could not be handled which completely change the sentiment of the tweet.

In the second method, a set of pre-defined positive and negative words were collected from the web. Then an algorithm was implemented in python, which initially tokenized the tweet and then removed the stop words. Next it compared each word with the predefined list of positive and negative word list and labelled the words with their sentiment. After this, the number of positive words, negative words and inverters were counted and labelled as positive or negative. Next, Machine learning algorithms were applied to the data to classify the tweets as positive or negative.

1. **Literature Survey**

In [4] Aron Culotta and Jennifer Cutler proposed a similar idea about “Mining Brand Perceptions from twitter” in which they have considered the structure of brands social network instead of user generated text.

Kennedy, A, & Inkpen, D. In [5] proposed a method for finding the sentiment analysis of movie reviews in which they considered negation, intensifier and diminishes, which can reverse the polarity as well as affect the positive or negative polarity of the sentence. They found out that along with other features, considering these three kinds of shifters improved the result of the classifier than previous results.

In [6] Luciano Barbosa and Junlan Feng proposed a paper on “Robust sentiment detection on twitter from biased and noisy data”. They put forward a method which classified the sentiment of the text messages based on the syntax of the tweets and meta information about the words that make up the tweets.

Abhishanga Upadhyay, Luis Mao and Malavika Goda Krishna in [7] wrote a paper on “Mining Data from Twitter” which described the detail use of twitter API and how python can be used to fetch the required data from twitter.

In [8] pang and lee proposed a paper on “sentiment classification using Machine Learning Techniques” in which they used the unigram model for classification of sentiment on movie reviews. They worked on different features and found out that unigrams are viable for sentiment analysis in regular reviews.

1. **Approach**

As discussed in the introduction of the paper, our approach to the problem started with collecting the data. Data was obtained from Twitter API using an Application Programming Interface provided by twitter, which allowed access to all the tweets that are posted in twitter. Python programming language was used to interact with the Twitter API. When a search keyword was queried for the Twitter API, the result was a list of documents in the JSON format which contained several attributes like username, ID, location, text, language, Geo location etc with the keyword in the text field.

To start with, around twenty-four car brands were selected. Then a program was written which took these car brand names as input and provided a list of documents in the JSON format as output. These documents were stored in Mongo dB in the form of objects. In the next step, the text part of the documents in English language was retrieved and the rest was eliminated. The output of this step was a tweet that was taken as input to find the sentiment. An example format of the tweet is below:

*“@ghijk no response. Such poor customer service. It's a sad situation when you feel your only recourse is social media. #ghijk #poorservice”*

Among the tweets that were retrieved, the tweets that contained keywords like safety, performance, luxury, style, fuel economy were extracted which resulted in the tweets related to cars (These words were considered since they were mentioned in an online study as the key characteristics that are considered for car brand classification).

*“RT @xyz: @abc no response. Such poor customer service. It's a sad situation when you feel your only resource is social media. #abc #xyz http://www.xyz.com”.*

A tweet syntax is as mentioned above. It contains RT which represents that a tweet is replying to another tweet. ’@’ is placed in front of a username to refer a person with that username. A Hashtag in a tweet is used to describe a topic. It also contains an URL, which is related to a specific topic or an advertisement.

Natural Language Tool Kit (NLTK) which is a set of libraries and programs that are used with python programming language, was used for performing operations on natural language. The usage of NLTK is described more in detail in upcoming sections of the paper.

After obtaining relevant tweets, a set of steps was followed to find the sentiment of the tweets which are as mentioned below:

1. Data Preprocessing.
2. Feature Selection.
3. Application of machine learning techniques(classification).

**Data Preprocessing**

The tweet that is obtained after removing unwanted tweets was noisy and contained lots of unnecessary symbols or words that played no importance in finding the sentiment of the tweet [3]. Hence preprocessing steps in the data were applied to optimize it for further analysis [9]. The Initial tweet before preprocessing was as given below.

*“Never getting a #xyz again. My local @abc dealer won't give me a loaner because it wasn't bought there. Bad customer service.”*

Initially, each tweet was converted into its lower case. Next, the URL in the tweet was converted to ‘URL’ and word that started with ‘@’ to ‘U’ to indicate that it was a user name. Also, every hashtag word was replaced by its original word and ‘not’ with ‘NOT’. All these operations were performed using Regular expressions. Python provides us with a library of regular expressions that can be used by importing it. After this step of preprocessing the tweet looked like below.

*“never getting a xyz again. my local ||U|| dealer wo NOT give me a loaner because it was NOT bought there. bad customer service.”*

The next step in data preprocessing is tokenization. In this process, we divided the tweet into tokens (words). This was done using the wordtokenize library of Natural Language Tool Kit (NLTK) package. The output of tokenization was a collection of Unigrams (single words) [9]. While performing tokenization, all the punctuation marks like ('**\'",.:`()**') and any words that started with number and stop words like (a, am, are, have) etc., were removed as they played no part in finding sentiment of the tweet[3]. A set of stop words was collected that are available on the web. Along with this, words like ‘coooooool’ were replaced with ‘cool’ by replacing multiple repeated characters with three characters to keep the intensity of the word. Next, words like haven’t, wouldn’t, were replaced with have not and would not. After the above preprocessing steps the tweet looked like below.

[('never'), ('getting'), (‘xyz’), ('local'), ('||u||'), ('dealer'), ('wo'), ('not'), ('give'), ('loaner'), ('not'), ('bought'), ('bad'), ('customer'), ('service')]

Now, the sentiment of each word was obtained as output from the tokenization process and each word was labelled with its sentiment. Set of positive words and negative words were collected in different files from the web files given in [10]. From the output after the tokenization process, each word was compared with the positive and negative words and labelled as positive or negative. If the word was not available in both of the files, then it was labelled as none. The output of this step was as below.

“[('never', 'none'), ('getting', 'none'), (‘xyz', 'none'), ('local', 'none'), ('||u||', 'none'), ('dealer', 'none'), ('wo', 'none'), ('not', 'negative'), ('give', 'none'), ('loaner', 'none'), ('not', 'negative'), ('bought', 'none'), ('bad', 'negative'), ('customer', 'none'), ('service', 'none')]”.

Tweets dB

Pre-Processing

Feature selection

classification

*Fig.1 Sentiment Analysis steps*

**Feature Selection**

Two different approaches were used in the process of Feature Selection.

The first method was using Unigrams in a tweet as feature [2][8]. In the initial step, the tweets for training data were collected and labelled as positive or negative manually. Then preprocessing steps were applied on each tweet. After the tokenization process, stop words, URL and username were removed. Now the output of this process which was a list of words was added to the feature list. Finally, a feature vector was obtained with all the words that occurred in a tweet. So, whenever a new tweet was tested for the sentiment, it was preprocessed as mentioned above and the presence of the words was checked with the words in the feature vector. If the word was present, then it was marked as 1 otherwise it was marked as 0. Based on the pattern of the 1’s and 0’s that was obtained, the tweet was labelled as positive or negative using the training data.

But, there are some limitations in this method. It would not be able to handle the Bi-grams like ‘not bad’ which completely change the sentiment of the tweet and only works on individual words. Also, when a new word appears in the test data that is not in the feature vector of the training data, it is labelled as positive irrespective of the sentiment of the tweet. It works well if there is a dataset of around 10k tweets [2].

In the second method, along with unigrams, bigrams and other different features were taken into consideration. Features like a count of positive words, count of negative words, count of inverters (negated statements), count of intensifiers, and number of comparative words [11] were considered. For each tweet, we have collected the feature information and then we create a ARFF file for that model and apply Machine Learning algorithms using WEKA. This method works better than previous model. We have gone through many papers which considered polarity of the emoticons as one of the features. But, we found that very few tweets contain emoticons embedded in the tweets on car brands. So, we avoided considering it in the classification of the sentiment.

**Classification**

Classification is a form of Data Analysis which plays a vital role in getting accuracy of trained data. It is a two-stage process. In the initial stage, we train the classifier and in the next stage we perform the classification. Among various classification methods, naïve Bayes and decision tree are used in our project. Decision

Naïve Bayes is an essential probabilistic model. In [12] they compared learning algorithms including decision tree algorithm with Naïve Bayes algorithm and found that sometimes Naïve Bayes performs better than others. When we used Naïve Bayes classification we got an accuracy percentage of 91.35%. It works well for small data set and as the dataset increases the accuracy goes down as high biased classifiers are not effective in providing accurate result [13].

Decision tree is one of the most used machine learning algorithms. It is used to obtain knowledge from given data and represent it in tree structure. It converts the obtained knowledge from given data into set of if-then conditions which makes it easily understandable by the user [12]. Attribute with maximum information gain will be root node followed by rest. The class is represented with the leaf node. The accuracy results are obtained by tracing a path from route to the leaf node where it holds the class prediction for that tuples [14]. The accuracy obtained by Decision tree is 91.85%. The performance of decision tree is better than Naïve Bayes algorithm.

1. **CONCLUSION**

In this paper, we tried to put forward the method we followed to perform sentiment analysis on tweets obtained from twitter on car brands followed by application of Machine Learning techniques. Even though similar type of brand perceptions was already done by Aron Culotta in [4], we have used sentiment analysis for finding the opinion on car brand. It provided us an opportunity to apply modern approaches in sentiment analysis along with machine learning techniques to find the opinion on the user towards car brand.  Initially the tweets were collected and then they were cleaned to find those that are related to car brands which is done by use of key words. After obtaining the relevant tweets a set of steps that include data preprocessing, feature selection and application of machine learning techniques were followed to classify the sentiment of the tweet. Then the data set was divided into training data and test data and then manually labelled the sentiment of the tweet.

By using tokenization process, unnecessary characters and punctuations were removed. Python programming is used to interact with twitter API that contains NLTK to perform operations on natural language.

Two different approaches were used in feature selection. In the second approach, we combined features that are used in [5][6][11] to obtain better performance. We compared their results and found that using features like number of positive words, negative words, inverters, intensifiers, comparative words in a tweet has profound impact on accuracy when compared to using Unigram features. Further, Naïve Bayes and decision tree algorithms were used for classification of the sentiment. Though there was slight difference in the percentage of accuracy, both the algorithms did well in the classification of the sentiment.  As part of future work, we planned to build a web application which let the user enter the name of the car brand and outputs the graph of opinion of the users along with the rank among the available car brands.

1. **ACKNOWLEDGEMENT**

We would like to convey our regards to **Kent State University** and Department of Computer Science which has given us chance to acquire an expertise level technical knowledge. We extend our gratitude towards professor **Dr. Kambiz Ghazinour**, for providing an environment and platform to enrich our skills and motivating us in our endeavor and helping us to realize our full potential.

1. **REFERENCE**

[1] Bing Liu. “Sentiment analysis and Opinion Mining”. Morgan and Claypool publishers,2012.

[2] Ravikiran Janardhana. “how to build a twitter sentiment analyzer?” published. [Online] URL: https://www.ravikiran j.net/posts/2012/code/how-build-twitter-sentiment-analyzer/#motivation [Accessed: 14 – Nov – 2016].

[3] Delenn Chin, Anna Zappone, Jessica Zhao. “Analyzing Twitter Sentiment of the 2016 presidential candidates”. Stanford.edu [web].

[4] Aron Culotta and Jennifer Cutler. “Mining Brand Perceptions from twitter”, Proceedings of EMNLP 2002, pp. 79–86.

[5] Kennedy, A, & Inkpen, D. “sentiment analysis”, Sentiment Classification of Movie and Product Reviews Using Contextual Valence Shifters, University of Ottawa, Ottawa, ON, K1N 6N5, Canada.

[6] Luciano Barbosa and Junlan Feng. “Robust sentiment detection on twitter from biased and noisy data”, Coling 2010: Poster Volume, pages 36–44, Beijing, August 2010.

[7] Abhishanga Upadhyay, Luis Mao and Malavika Goda Krishna. “Mining Data from Twitter”, "Abhishanga/Twitter-API-Python." GitHub. N.p., n.d. Web. 24 Oct. 2014.

[8] Pang and Lee. “sentiment classification using Machine Learning Techniques”, Proceedings of the Conference on Empirical Methods in Natural, Language Processing (EMNLP), Philadelphia, July 2002, pp. 79-86. Association for Computational Linguistics.

[9]Gautami Tripathi and Naganna S. “Feature seolection and classification approach for sentiment analysis”. MLAIJ Vol.2, No.2, June 2015.

[10] Jeffrey Breen. “twitter-sentiment-analysis-tutorial-201107” published: [Online] URL: <https://github.com/> Jeffrey breen/twitter-sentiment-analysis-tutorial- 201107/ tree/master/data/opinion-lexicon-English [Accessed: 18-Nov-2016].

[11] Nitin Jindal and Bing Liu."Identifying Comparative Sentences in Text Documents" Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval (SIGIR-6), Seattle 2006.

[12] Tom M. Mitchell,” Machine Learning”, McGrawHill Science,1 – March – 1997.

[13] Edwin Chen.”Choosing a machine learning classifier”,[online]Available:<http://blog.echen.me/2011/04/27/choosing> -a-machine-learning-classifier/ ,April 2011,[Accessed: 12- Dec-2016].

[14] Jiawei Han, Micheline kamber and Jian Pei. “Data Mining Concepts and Techniques”. Third edition, The Morgan Kaufmann Series in Data Management Systems.