CUSTOMER SENTIMENT ANALYSIS

ANIKET P. KUMAR[1] SHALINI D. PATHAK[2] JAYSHREE M. PATIL[3] SPARSH R. SINGH[4]
DR. AVISHEK RAY[5]

[1-5]Department of Electronics and Telecommunication
K.C. College of Engineering & Management Studies & Research, Kopri, Thane (E)-400 603, India.

Abstract - Sentiment analysis or opinion mining is the computational study of people's opinions, sentiments, attitudes, and emotions expressed in written language. It is one of the most active research areas in natural language processing and text mining in recent years. Its popularity is mainly due to two reasons. First, it has a wide range of applications because opinions are central to almost all human activities and are key influencers of our behaviors. Whenever we need to make a decision, we want to hear others' opinions. Second, it presents many challenging research problems, which had never been attempted before the year 2000. Part of the reason for the lack of study before was that there was little opinionated text in digital forms. It is thus no surprise that the inception and the rapid growth of the field coincide with those of social media on the Web. In fact, the research has also spread outside of computer science to management sciences and social sciences due to its importance to business and society as a whole. In this talk, I will start with the discussion of the mainstream sentiment analysis research and then move on to describe some recent work on modeling comments, discussions, and debates, which represents another kind of analysis of sentiments and opinions.

I. INTRODUCTION

Sentiment is an attitude, thought, or judgment prompted by feeling. Sentiment analysis, which is also known as opinion mining, studies people's sentiments towards certain entities. From a user's perspective, people are able to post their own content through various social media, such as forums, micro-blogs, or online social networking sites. From a researcher's perspective, many social media sites release their application programming interfaces (APIs), prompting data collection and analysis by researchers and developers. However, those types of online data have several flaws that potentially hinder the process of sentiment analysis. The first flaw is that since people can freely post their own content, the quality of their opinions cannot be guaranteed. The second flaw is that the ground truth of such online data is not always available. A ground truth is more like a tag of a certain opinion, indicating whether the opinion is positive, negative, or neutral. "It is quite a boring movie…….. But the scenes were good enough." The given line is a movie review that states that "it" (the movie) is quite boring but the scenes were good. Understanding such sentiments requires multiple tasks. Hence, Sentiment analysis is a kind of text classification based on Sentimental Orientation (SO) of opinion they contain. Sentiment analysis of product reviews has recently become very popular in text mining and computational linguistics research. Firstly, evaluative terms expressing opinions must be extracted from the review. Secondly, the SO, or the polarity, of the opinions must be determined. Thirdly, the opinion, strength, or the intensity, of an opinion should also be determined. Finally, the review is classified with respect to sentiment classes, such as Positive and Negative, based on the SO of the opinions it contains.

2. LITERATURE REVIEW

Here we provide you insight of some of the research work which helps us to understand the topic deeply. Customer sentiment analysis is an important aspect of customer relationship management that involves analyzing the opinions and feelings of customers towards a product or service.

We have referred to different international papers and journals in which we came to know that various methods were used by different authors. With the help of the previous work from the above papers we were able to build this project. Below are some authors and their proposed methodology and their limitations.

In order to classify text by overall sentiment and not just by text [1] P. Pang, L. Lee, S. Vaithyanathan et al applied machine learning algorithms on movie review databases which gave results that these algorithms out-perform human produced algorithms. The machine learning algorithms they use are Naïve-Bayes, maximum entropy, and support vector machines. They also conclude by examining various factors that classification of sentiment is very challenging. They show supervised machine learning algorithms are the base for sentiment analysis.

[2] H. Wang, D. Can, F. Bar, S. Narayana et al were the researchers who proposed a system for real time analysis of public responses for 2012 presidential elections in the U.S. They collected the responses from Twitter, a micro blogging platform. Twitter is one of the social network sites where people share their views, thoughts and opinions on any trending topic. People's responses on Twitter for election candidates in the U.S. created a large amount of data, which helped them to create a sentiment for each candidate and also to create a prediction of who would win.

[3] O. Almatrafi, S. Parack, B. Chavan et al are the researchers who proposed a system based on location. According to them, Sentiment Analysis is carried out by Natural Language Processing (NLP) and machine learning algorithms to extract a sentiment from a text unit which is from a particular location. They study various applications of location based sentiment analysis by using a data source in which data can be extracted from different locations easily. In Twitter, there is a field of tweet location which can easily be accessed by a script and hence data (tweets) from particular locations can be collected for identifying trends and patterns.

3. PROPOSED METHOD/SYSTEM

Sentiment Analysis is a process of extracting features from user’s thoughts, views, feelings and opinions which they post on any social network websites. The result of sentiment analysis is classification of natural language text into classes such as positive, negative and neutral. The amount of data generated from social network sites is huge; this data is unstructured and cannot give any meaningful information until it is analyzed. Thus, to make this huge amount of data useful we perform sentiment analysis, i.e. extracting features from this data and classify them. The problem at hand is to perform customer sentiment analysis to understand the opinions, feelings, and attitudes of customers towards a product, service, or brand. This information can be used to make data-driven decisions on improving the product, service, or brand's overall customer experience. Hence the methodology that we have followed involves four major steps particularly i.e. Data collection, Data Processing, Sentiment Analysis and Data visualization.

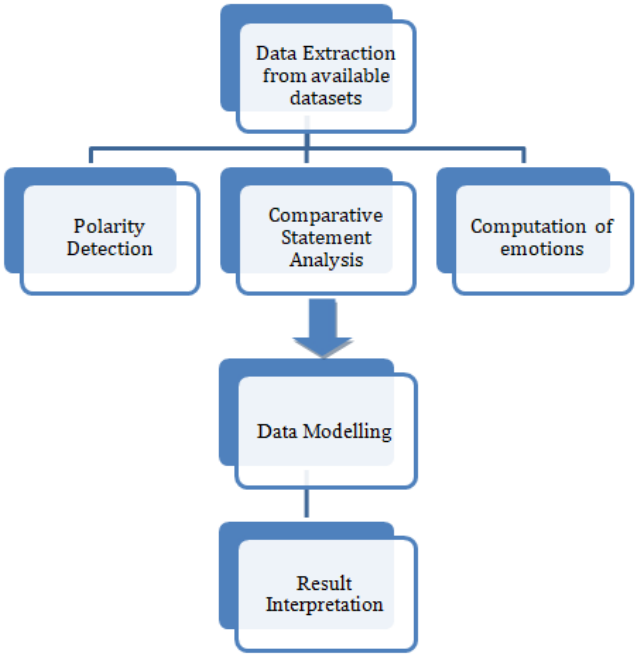

Fig 3.1 Proposed Architecture using Multinomial Bayes Classifier

As shown in Figure 3.1 the first step i.e. Data collection refers to Gathering customer feedback from various sources such as social media, customer reviews, surveys, and customer service interactions. Data collection is not a simple task, as it may seem. Various decisions have to be made for collecting data. For our thesis we maintain dataset for training, testing and for Datasets sentiment analysis while Data Processing deals with Cleaning the data, removing irrelevant information, and formatting the data to be analysed. The next step is using natural language processing techniques to classify the text as positive, negative, or neutral sentiment and the last step is visualizing the data to gain insights and identify patterns.

4.. CLASSIFIERS USED

MultinomialNB expands the use of the NB algorithm. It implements NB for data distributed multinomial, and also uses one of its versions for text classification (in which word counts are used to represent data, and also tf-idf works extremely well in regular practice). We parameterized the distribution data by vectors for every , where 'n' gives the total features (which means, the size of vocabulary for text classification) and probability of each that appears in the sample of class 'y' is We use smoothed version of maximum likelihood for estimation of parameters , which is relative frequency of counting: where Nyi represents number of times 'i' appeared in any sample of class 'y' which belongs to training sample , and gives the total number of 4 features in class 'y'. To prevent zero probabilities for further calculations, we add smoothing priors' for features that are not present in any learning samples. If, smoothing is termed as Laplace and for the smoothing is termed as Lidstone.

Multinomial Naive Bayes classification algorithm tends to be a baseline solution for sentiment analysis tasks. The basic idea of Naive Bayes technique is to find the probabilities of classes assigned to texts by using the joint probabilities of words and classes. Assume you wish to categorize user reviews as good or bad. Sentiment Analysis is a popular job to be performed by data scientists. This is a simple guide using Naive Bayes Classifier and Scikit-learn to create a Google Play store reviews classifier (Sentiment Analysis) in Python. Naive Bayes is the simplest and fastest classification algorithm for a large chunk of data. In various applications such as spam

filtering, text classification, sentiment analysis, and recommendation systems, Naive Bayes classifier is used successfully. It uses the Bayes probability theorem for unknown class prediction.
The Naive Bayes classification technique is a simple and powerful classification task in machine learning. The use of Bayes' theorem with a strong independence assumption between the features is the basis for naive Bayes classification. When used for textual data analysis, such as Natural Language Processing, the Naive Bayes classification yields good results.

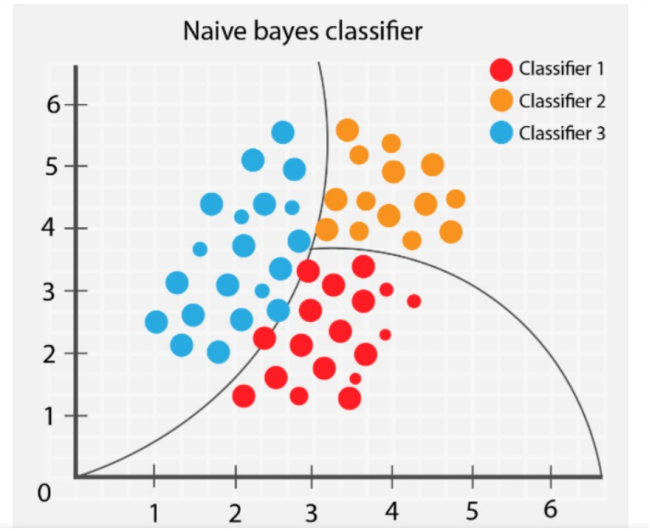

Fig 4.1. Naive Bayes Classification

Simple Bayes or independent Bayes models are other names for naive Bayes models. All of these terms refer to the classifier's decision rule using Bayes' theorem. In practice, the Bayes theorem is applied by the Naive Bayes classifier. The power of Bayes' theorem is brought to machine learning with this classifier. Let's have a brief look at maths.

Given the dependent feature vector (x_1 , …, xn) and the class Ck. Bayes' theorem is stated mathematically as the following relationship:

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)P(x_1, \ldots, x_n \mid C_k)}{P(x_1, \ldots, x_n)}$$

`According to the "naive" conditional independence assumptions, for the given class Ck each feature of vector xi is conditionally independent of every other feature xj for i≠j..`

$$P(x_i \mid C_k, x_1, \ldots, x_n) = P(x_i \mid C_k)$$

Thus, the relation can be simplified to

$$P(C_k \mid x_1, \ldots, x_n) = \frac{P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)}{P(x_1, \ldots, x_n)}$$

Since P(x_1 , …, xn) is constant, if the values of the feature variables are known, the following classification rule can be used:

$$P(C_k \mid x_1, \ldots, x_n) \propto P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

$$\Downarrow$$

$$\hat{y} = \arg\max_k P(C_k)\prod_{i=1}^{n} P(x_i \mid C_k)$$

To avoid underflow, log probabilities can be used.

$$\hat{y} = \underset{k}{\operatorname{argmax}}(\ln P(C_k) + \sum_{i=1}^{n} \ln P(x_i \mid C_k))$$

The variety of naive Bayes classifiers primarily differs between each other by the assumptions they make regarding the distribution of P(xi|Ck), while P(Ck) is usually defined as the relative frequency of class Ck in the training dataset.

The multinomial distribution is parametrized by vector θk=(θk1,…,θkn) for each class Ck, where n is the number of features (i.e. the size of the vocabulary) and θki is the probability P(xi|Ck) of features appearing in a sample that belongs to the class Ck.

The parameters θk is estimated by a smoothed version of maximum likelihood, i.e. relative frequency counting:

$$\hat{\theta}_{ki} = \frac{N_{ki} + \alpha}{N_k + \alpha n}$$

where Nki is the number of times feature i appears in a sample of class k in the training set T, and Ny is the total count of all features for class Ck. The smoothing priors α≥0 accounts for features not present in the learning samples and prevents zero probabilities in further computations. Setting α=1 is called Laplace smoothing, while α<1 is called Lidstone smoothing.

Thus, the final decision rule is defined as follows:

$$\hat{y} = \underset{k}{\operatorname{argmax}}(\ln P(C_k) + \sum_{i=1}^{n} \ln \frac{N_{ki} + \alpha}{N_k + \alpha n})$$

5. RESULTS

The plot in Fig 5.1 typically shows a curve that starts at a low accuracy rate and increases as the number of training samples increases. At some point, the curve may start to level off, indicating that additional training data will not significantly improve the model's accuracy. This information can be valuable in determining the best trade-off between the amount of training data and the model's accuracy, particularly in cases where the training data is limited or expensive to acquire.

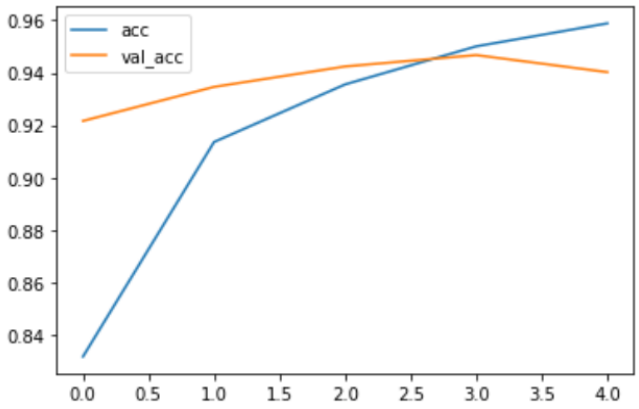

Fig 5.1. Accuracy Plot for Sentiment analysis

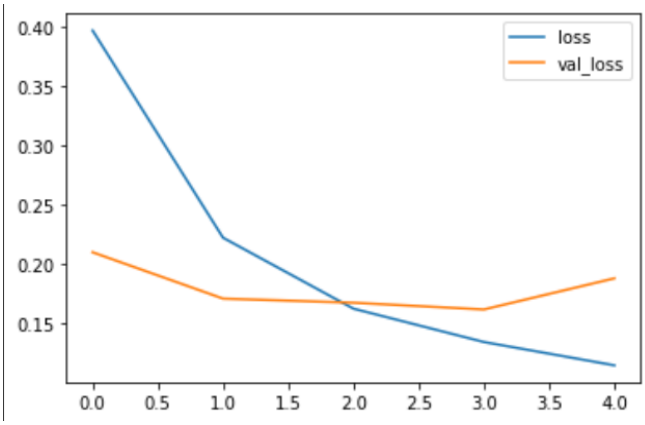

Fig 5.2 Accuracy Plot for Sentiment analysis

The plot in Fig 5.2 shows a curve that starts at a high value and decreases over time as the model learns to make better predictions. The shape of the curve can provide insights into the behavior of the model during training, such as whether it is overfitting or underfitting the training data. By analyzing the loss plot, machine learning practitioners can adjust the model's hyperparameters and make other changes to improve its performance.

```
predict_sentiment("This is the worst movie I have ever seen")

1/1 [==============================] - 0s 38ms/step
Predicted label:  negative
```

Fig 5.3 Output of Negative Statement

```
predict_sentiment("This was the best flight of my life")

1/1 [==============================] - 0s 36ms/step
Predicted label:  positive
```

Fig 5.4 Output of Positive Statement

The above Figures 5.3 and 5.4 demonstrate our sentiment analysis model in which we pass a sentence into a function named predict_sentiment, and it returns the predicted label of the sentence as positive or negative.

6. CONCLUSION

In conclusion, customer sentiment analysis is a valuable tool for businesses looking to understand their customers better and improve their products and services. By analysing customer feedback and sentiment, businesses can gain insights into customer needs, preferences, and pain points, which can help them make data-driven decisions that improve customer satisfaction, retention, and overall business performance. Customer sentiment analysis can also help businesses identify potential issues early and take proactive measures to address them, as well as improve their marketing and messaging strategies. Ultimately, businesses that prioritize customer sentiment analysis are better positioned to meet customer needs and expectations, differentiate themselves from competitors, and achieve long-term success.

7. REFERENCES

[1] S. M. Vohra and J. B.Teraiya, "A Comparative Study of Sentiment Analysis Techniques", Journal of Information, Knowledge and Research in Computer Engineering, 2012.

[2] Ahmad Kamal and Muhammad Abulaish, "Statistical Features Identification for Sentiment Analysis using Machine Learning Techniques", International Symposium on Computational and Business Intelligence, 2013.

[3] Neethu M.S and Rajasree R., "Sentiment Analysis in Twitter using Machine Learning Techniques", Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013.

[4] Erik Cambria, "An Introduction to Concept-Level Sentiment Analysis", 2013.

[5] R. Feldman, "Techniques and applications for sentiment analysis", Proc. ACM, pp. 56-82, 2009.

[6] B. Sun and TY. V. Ng, "Analyzing Sentimental Influence of Posts on Social Networks", Proc. The 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, 2014.

[7] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou and P. Li, "User-level sentiment analysis incorporating social networks", Proc. The 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1397- 1405, 2011.