Sentiment Analysis using Natural Language Processing

Prithvisinh Jhala
Student ID: 1137435
Dept. of Computer Science
Lakehead University
Thunder Bay, Ontario
Email: jhalap@lakeheadu.ca

***Abstract*—Sentiment analysis is text mining that finds and extracts subjective information from source material like product reviews, social media, news, tweets, etc., allowing a company to better understand the social sentiment of its brand, product, or service. Performing Sentiment analysis, a company can serve its customers better by learning from their reviews and comments. This detailed project report provides a brief survey on the methods used for Sentiment analysis using Machine Learning and Natural Language Processing concepts.**

Index terms- Sentiment Analysis, Natural Language Processing, Data Analysis, Machine Learning, Decision Making, Neural networks, Business Intelligence.

I. Introduction

Due to the tremendous growth in online marketplaces over the last several decades, online vendors and merchants now invite their customers to give their thoughts on the things they've purchased. As a result, millions of evaluations are generated every day, making it difficult for a potential customer to decide whether or not to purchase a product. For product manufacturers, analyzing such a large number of reviews and comments is challenging and time-consuming. Therefore, they use Sentiment analysis to make this task efficient and easy-going. The ability of algorithms to analyze text has greatly increased as a result of recent developments in deep learning. Advanced artificial intelligence algorithms used creatively can be a valuable tool for conducting in-depth research [1]. When these fundamental notions are combined, they form a powerful tool for evaluating millions of brand reviews and comments with human-level accuracy.

II. Sentiment Analysis

Sentiment Analysis, is the systematic identification, extraction, quantification, and the study of subjective information using text analysis and natural language processing. It is the classification of sentiment characteristics from real-world data like review board comments, social media news, search engine results, television, stock forum, internet news etc. Sentiment analysis is commonly used in customer service, clinical medicine and marketing to analyse reviews and survey answers, as well as online and social media and healthcare materials. It is crucial in data analytics for delivering accurate prediction with machine learning techniques. The sentiment is classified into three types: positive, negative, and neutral. There are two primary activities. (1) Product features are classified from user feedback, and (2) the statements are labelled as positive or negative. These are extremely difficult tasks. This is done at the file, paragraph, and word levels, with the help of machine learning technologies including unsupervised and supervised algorithms [2].

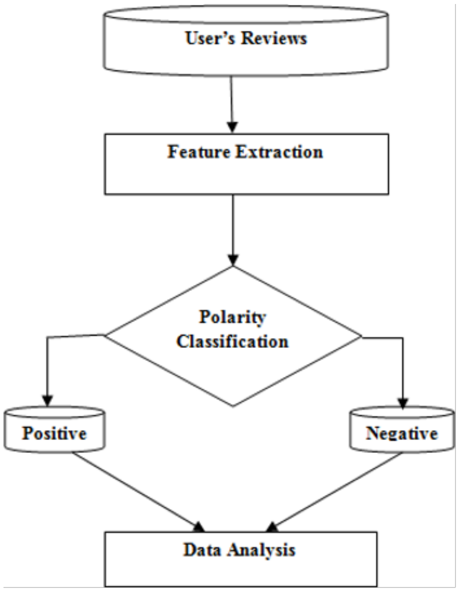


Fig. 1. Sentiment Analysis Workflow

III. Methodology

There are various methods used to classify Sentiments based on Natural language processing. Some of them are described below. Lexicon based approach can be followed to perform Sentiment Analysis on product reviews. It is the sentiment conveyed by each word and phrase. The lexicon-based framework is utilized to extract sentiment. This method is a straightforward strategy for Sentiment Analysis of data that does not require training. It is primarily intended to perform tasks involving opinion-bearing words. Opinion words are the words which are commonly used to articulate positive or negative opinions (or sentiments), words including ”excellent”, ”good”, ”poor” and ”bad”. The number of negative and positive opinion words in each sentence of the evaluation is used to evaluate the characteristic of the product. If the negative opinion words are more than the positive opinion words, then the final conclusion of the feature will be negative and otherwise positive.

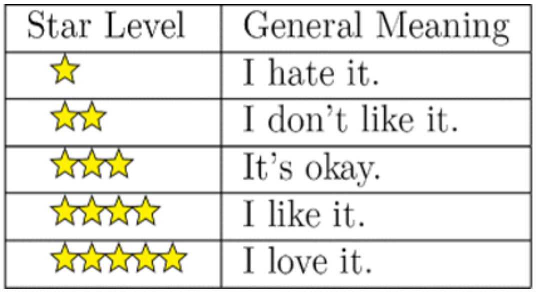
Another method used in Sentiment analysis is Tokenization. Tokenization is the process of breaking down large pieces of text into smaller ones. Tokenization divides the text data into words and sentences, which are referred to as tokens. These tokens aid in the comprehension of the context or the development of the NLP model. By evaluating the sequence of words, tokenization aids in interpreting the meaning of the text. For example, the text “The product is good.” can be tokenized as, “The”, “product”, “is”, “good”. Gensim, Keras and NLTK are some of the libraries that can be used to accomplish the task. The words that does not add any meaning in the sentence are known as “STOP words”. Words like “is”, “am”, “are”, “the”, “but”, etc. are removed from the sentence. The remaining tokens take part in POS tagging. Part-of-speech (POS) taggers have been developed in natural language processing to classify words based on their parts of speech. A POS tagger is particularly beneficial for sentiment analysis for the following two reasons: 1) Nouns and pronouns, for example, are frequently devoid of sentiment. With the help of a POS tagger, it can filter out such words; 2) It can also be used to differentiate between words that can be used in various parts of speech.

Sentiment Analysis can also be done simply by providing customers with a method like Star review method to review a product. It is very easy to extract reviews based on this approach as each number of stars contain a specific review. The number of stars determine how good or bad the product is. If the product is given 1 star then it is not really liked by the customers, however, if it is given 5 stars, then the customers loved it. If it is given 3 stars then the reaction is considered to be neutral. So analysing sentiments following this approach is very easy but, nowadays people provide written reviews instead of just star rating the product so, it becomes essential to use Natural Language processing and the above mentioned methods to analyse the sentiment of the customers [3].

However, in this project, I have used only 2 classes (Positive and Negative) to maintain the accuracy. The more the number of classes, the lesser will be accuracy of the model because it is very difficult to train the model with more classes. So I have removed the neutral class (3), and combined the classes 1 and 2 as Negative sentiment and 4 and 5 as positive sentiment.

IV. Dataset

There are several datasets which can be used in Sentiment analysis which depends on the requirements of the company or organization performing this task. As explained above, SA can be done on product reviews, book reviews, tweets, movie reviews, stock market prediction, etc. The format of data used generally is .json file.[4]



Star Level	General Meaning
★	I hate it.
★★	I don’t like it.
★★★	It’s okay.
★★★★	I like it.
★★★★★	I love it.

Fig. 2. Amazon Reviews

The dataset that I have used is Amazon Fine Food reviews, which can be download from an open source website Kaggle.com. The dataset is in .csv format. The size of the dataset is around 643 MBs. And it contains over 500,000 customer reviews of fine food products from Amazon up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories [5].

V. Data Pre-processing

Prep-rocessing of the data is a crucial part of the data mining process. Knowledge discovery during the training phase is more challenging if there is a lot of irrelevant and repetitive information or noisy and inaccurate data. The stages of preprocessing and filtering can take a long time to complete. Cleaning, normalisation,instance selection, feature extraction, transformation and selection, and so on are all examples of data preparation. The final training set is the result of data preparation.

A. *Negative Phrase Identification*

This is a very crucial step to follow in SA. With the use of negative prefixes, words like adjectives and verbs can communicate the opposite sentiment. Consider the following line from a review of an electronic device: ”The built-in speaker also has its uses, although nothing revolutionary so far.” According to the list, the word ”revolutionary” is a positive word. However, the statement ”nothing revolutionary” conjures up mixed emotions. As a result, it is critical to recognise such sentences. Two types of phrases have been identified in this study: negation-of-adjective (NOA) and negation-of-verb (NOV).

B. *Noise Removal*

Any text that is unrelated to the context of the data and the final result might be designated as noise. The general technique to noise removal is to create a dictionary of noise entities, then loop through the text object by tokens (or words), removing those tokens that are in the noise dictionary.

C. Lexicon Normalization

Another sort of textual noise concerns the many representations that a single word can exhibit. For example, the words ”player”, ”play”, ”playing”, ”played” and ”plays” are all versions of the word ”play”. The step transforms a word’s discrepancies into their normalised version (also known as lemma). Normalization is an important step in text feature engineering because it reduces high-dimensional (N number of features) features to a low-dimensional space (1 feature), which is perfect for any machine learning model.

Common Lexicon normalization techniques are as follows.

1) Stemming: Stemming is a simple rule-based technique for removing suffixes from a word (e.g., ”es”, ”s”, ”ly”, ”ing”, and so on).

2) Lemmatization: Lemmatization is a way of acquiring the root form of a word that uses lexicon and morphological analysis.

3) Stop Word Removal: Some NLP applications, such as sentiment analysis and text summarization, require stop word removal as a preprocessing step. A basic yet crucial step is to remove stop words as well as regularly recurring terms.

VI. Sentiment Classification Algorithms

There are various Machine Learning and Natural language processing algorithms used to classify Sentiments, once feature extraction is done, and datasets are analysed. The ones I used in this project are described briefly below.

A. Random Forest

Because it outperformed a single decision tree in terms of accuracy, the random forest classifier was chosen. It’s essentially a bagging-based ensemble approach. The following is how the classifier works: Given D, the classifier first generates k bootstrap samples of D, each of which is labelled Di. A Di has the same amount of tuples as D and is sampled using D’s replacement. Because sampling with replacement is used, some of the original D tuples may not appear in Di, while others may appear many times. After then, the classifier creates a decision tree based on each Di. As a result, a ”forest” made up of k decision trees emerges. Each tree gives its class prediction as one vote to classify an unknown tuple, X. The final decision of X’s class is allocated to the one that gets the most votes [6].

B. Feature Extraction

Feature Extraction is a technique for reducing the amount of features in a dataset by generating new ones from current ones (and then discarding the original features). The previous feature set should then be able to summarise the majority of the data in the new reduced set of features. From a mixture of the original set, a simplified version of the original features can be generated.

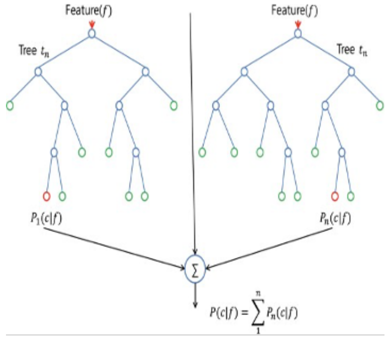


Fig. 3. Random Forest model

C. LSTM

LSTM (Long Short term memory) networks are a sort of deep neural network that may learn order dependence in sequence prediction challenge. This is a requirement in a variety of complicated issue domains, including machine translation, speech recognition, and others.

D. RNN

A recurrent neural network (RNN) processes sequences one element at a time, whether it’s daily stock prices, phrases, or sensor measurements, while preserving a recollection (called a state) of what’s come before. Recurrent indicates that the current time step’s output becomes the next time step’s input. The model analyses not just the current input, but also what it knows about the previous elements at each element of the sequence.

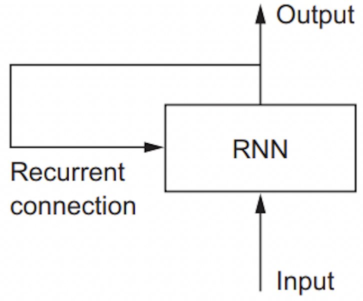


Fig. 4. RNN Model

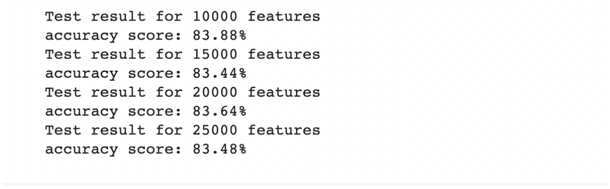
E. Keras

Keras is a free open source Python framework for constructing and evaluating deep learning models that is both powerful and simple to use. Keras has an Embedding layer for text data that may be utilised with neural networks. It necessitates integer encoding of the input data, with each word represented by a distinct number. The Tokenizer API, which comes with Keras, can be used to execute this data preparation step. The Embedding layer starts with random weights and learns an embedding for every word in the training dataset.

It's a multipurpose layer that may be utilised in a number of ways, including: It can be used on its own to learn a word embedding that can then be stored and reused in a subsequent model. It can be used as part of a deep learning model that learns the embedding as well as the model. It can be used to load a word embedding model that has already been trained, which is a sort of transfer learning.

VII. Results and conclusion

We need a lot of features to perform effective sentiment analysis or solve any NLP challenge. It's difficult to determine the exact amount of features required. So I tried 10,000 to 30,000 features. And printed out the accuracy scores that correspond to the number of features.



```
Test result for 10000 features
accuracy score: 83.88%
Test result for 15000 features
accuracy score: 83.44%
Test result for 20000 features
accuracy score: 83.64%
Test result for 25000 features
accuracy score: 83.48%
```

Fig. 5. Results

Precision is the percentage of relevant examples found among the retrieved examples, whereas recall is the percentage of relevant instances found among the total number of relevant examples. Precision and recall are thus founded on a grasp of and measurement of relevance.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

Fig. 6. Precision And Recall

The F1 score is a metric for determining the correctness and accuracy of a test. To compute the score, it takes into account both the precision p and the recall r of the test: The F1 score is the harmonic mean of precision and recall, with 1 being the highest and 0 being the worst.

$$F_1 = \frac{2}{\frac{1}{\text{recall}} + \frac{1}{\text{precision}}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

Fig. 7. F1 Score

Overall, the accuracy of the model is between 80 and 88% which can be showcased in the code itself.

VIII. Future Work

In this model, I have used only two classes (Positive and Negative) for the sentiment analysis for better accuracy. I will work on this project in future to improve the accuracy and increase the number of classes. I will add a class for a neutral sentiment, because customers can have a neutral review on a food product. And this model can be used in other datasets too, so I am working on creating a generalised UI for end users and I will add more than one datasets to the model to train it better, but that requires a large amount of RAM and processing power. Training the model using this very dataset took me around 5-6 hours, so if we add more datasets, then it will take a very long time to train the model. So using deep learning and neural network models, we can improve the accuracy by using more than one datasets.

IX. Acknowledgements

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my course instructor, Dr. Trevor Tomesh, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report. Dr. Trevor Tomesh helped me with the basic concepts of conducting a research and preparing a well formatted standard report document. It will be very useful to me in the future as well.

I would also like to thank Mr. Amine Trabelsi who provided me brief knowledge of the concepts of Natural Language Processing this semester, with the help of which I was able to conduct a research in this vast field Sentiment Analysis.

References

[1] P. Chitra and e. a. Karthik, ”Sentiment analysis of product feedback using natural language processing,” ELSEVIER, 24 December 2020.

[2] X. Fang and J. Zhan, ”Sentiment analysis using product review data,” Journal of Big Data, vol. 2, no. 5, pp. 1-14, 2015.

[3] A. Dhara and S. e. a. Arkadeb, ”Sentiment Analysis of Product-Based Reviews Using Machine Learning Approaches,” RCC INSTITUTE OF INFORMATION TECHNOLOGY, BELIAGHATA, KOLKATA, 2014.

[4] X. W. e. a. Wanliang Tan, ”Sentiment Analysis for Amazon Reviews,” Stanford, Stanford.

[5] J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. WWW, 2013.

[6] S. PAKNEJAD, ”Sentiment classification on Amazon

reviews using machine learning approaches," KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, Stockholm, Sweden, 2018

X. Supporting Material

I referred to the following books and websites for assistance and support that I required to complete this project. These websites are open source and are really helpful for computer scientists who are on a beginner level in coding.

Python for Data Analysis by Wes McKinney
https://stackoverflow.com/
https://www.kaggle.com/
https://www.overleaf.com/project
https://towardsdatascience.com/
https://www.youtube.com/
https://colab.research.google.com/
https://devguide.python.org/
https://www.nltk.org/
https://keras.io/