Customer feedback analysis by detecting emotion from text

Kim, Da Ye, *8971-8937*

Abstract

—**Businesses should be responsible for customers' feedback, so the analysis of reviews from the customer is a fundamental step for all businesses since it provides valuable insight and information to companies. This project will discuss how to extract valuable information from reviews. The most commonly used way is that according to keywords, the output shows relevant reviews and identifies the sentiment opinion such as if it is positive or negative. However, this is not directly helpful to give the company bright insight into their products. Therefore, a more detailed analysis is required. This project went one step forward than originally exiting solution. The categories of keywords are more broken down and they bring and organize the relevant specific words repeatedly mentioned in reviews with the sentiment opinions as well. This project implemented application for detecting emotions from text data.**

***Index Terms*— Text analysis, Sentiment analysis, Key word Categorization, Text Classification, Emotion Recognition from Text**

I. Introduction

Classifying the text or sentence into positive class or negative class does provides the limited information although huge amount of size of data set are analyzed. In order to extract more valuable information, the more classifications are required to detect informative text. The more classifications provide more information with the same size of datasets, comparing to detecting whether text represents positive or negative. However, there are few problems to build more classifications because each class should be well defined and avoid duplication or overlapping among different classes. In addition, the character of each class should be clearly distinct from one another. The other problem is to require more various words in dataset since this dataset can be used for training the computer to learn and detect the pattern. In order to solve these problems, detecting emotion from text can be suggested. First, emotions can be labeled in various class and distinguished clearly from different emotion classes. There are tons of datasets containing enormous size of words which represents emotions such as novel. To get high accuracy of detecting emotion from text, Naïve Bayes theorem and modes are used in this project.

II. Description

A. Explore datasets

To construct well defined classes for emotions, I classify seven different emotions for each class:
'joy', 'fear', 'anger', 'sadness', 'disgust', 'shame', 'guilt'. These datasets are found from relevant text data mining websites. Among the found emotion text dataset, to implement this project, I chose the seven categories. Each emotion dataset has synonyms which represents same or similar expression with one of the set emotions. Because the test data is used for training the computer (or functions) and for testing to detect emotions from text, it should contain emotional words in each sentence.

B. Modifying datasets

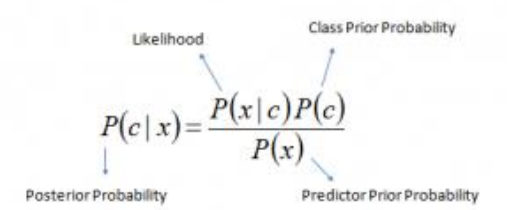
It is hard for the computation operations to interpret human languages and even harder to detect targeted words from natural language. Even if datasets are digitalized for computers to be able to read, they are required to replace the informative text data for the computer to interpret them accurately. There are various methods to modify text data.

- Tokenization – this function breaks the text and sentences into the formative data size. This tokenized data can be chunks of words or singular word.
- Stemming – this detects the root form of words and replace them into the original form. For example, a word 'having' will be replaced with 'have'.
- Lemmatization -this function is more sophisticated rather than stemming since it uses vocabulary and morphological analysis. For example, a word 'better' is normalized into 'good'.
- Stopwords – this is the list of meaningless word or text such as is, are, the, a, etc.
- Negation – this is also the list of words which seem to express negative but do not represent any meaning for emotion detecting algorithm. For example, words 'not, neither and however' do not show emotional expression.
- Pos tagging – this function classifies words in grammatical elements based on context. It analyzes words using the syntax tree.

C. *Naïve Bayes classifier*

Python is thought as the best way to implement Natural Language processing since it has Natural Language Tool Kit(NLTK) platform. In addition, this NLTK provides few types of analysis strategy based on Naïve Bayes modes. Naïve Bayes modes are based on Naïve Bayes theorem and it calculates probability unknown datasets belonging to each given class.

- Naïve Bayes Theorem



$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

  - P(c|x) is the posterior probability of class given predictor: c is target and x is attributes.
  - P(c) is the prior probability of class.
  - P(x|c) is the likelihood which is the probability of predictor given class.
  - P(x) is the prior probability of predictor

- Multinomial Naïve Bayes

  -this mode calculates the average probability of each word for given class. Due to his calculation, this shows how each word is distributed after analyzing the ratio the number of classes which contains words belonging to and the total number of the class.

$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$

  If α is 0, it is called Laplace smoothing.
  The parameter θ is the distribution ratio in each class. It is analyzed by the smoothed version of max likelihood.

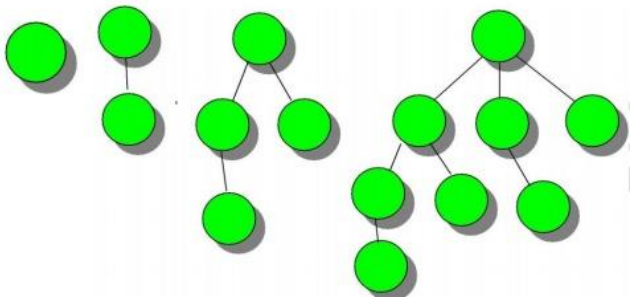
NLTK libraries provides Multinomial Naïve Bayes modes classification so this is efficient to predict dataset in various classes. detecting emotion from text is performed efficiently based on this classification functions.

III. EVALUATION

There are two implementations for this project, and each has different approach. One is to implement analysis of word frequency and the other is to detect emotion from text which are explained above and previous section. These two implementations show the distinct character and features in running time, accuracy and usage.

A. *Quantitative analysis*

Analysis of word frequency is implemented through JAVA and designed to use social media because the input data consists of hashtag (#), string and integer. This implementation detects the hashtag and analyzes the word following after the hashtag while calculating the frequency of the word. The essential concept for building this algorithm is using Max Fibonacci Heap and hash table.



Each node is labeled with the string following after hashtag and has value that is the number of appearances. There is a function detecting hashtag, extracting the string and checking if hash table has the string. If there is the string in the hash table, the value of the string's node is increased, and the node is moved in Fibonacci Heap. This Implementation performs in efficient time since it takes O(1) amortized time complexity for operations.

The main implementation for detecting emotion from text includes calculation of word frequency as well but it plays a trivial roll comparing to other functions. This implementation takes polynomial time. However, it takes mostly the cube of the size of all input data. Because all datasets are filtered and created as detectable and informative data for the classification function to train and to test. All normalization and extracting meaningless functions should performs input data in order of functions. In addition, this shows relatively low accuracy: in this test, it shows 63 percent of accuracy.

IV. RELATED WORK

Text analysis is used in various area. The first implementation which detect word frequency is one of the most performed in real fields. Clustering text is also actively used and this is useful to detect spam mails or messages on the phone since it detects the pattern in spams and making them in a cluster.

Most text analysis is performed after modifying or analyzing text data using Tokenization, Normalizing, stopwords, and Pos tagger. In addition, text is analyzed in similar classifier which performing with Naïve Bayes theorem.

Naïve Bayes classifier has pros and cons. The benefit of it is easy to detect and classify in various classes at the same time. However, if the training dataset has no word in some of given classes, it is impossible to extract text belonging to the classes. Simply it cannot predict the output. In addition, according to the result of the implantation, the accuracy is not high enough

since the value or meaning is changed depending on the context.

V. Summary and Conclusions

A. *Detecting Patterns & Ambiguity of human language*

- Naive Bayes theorem needs well distributed input dataset. After performing this implementation, it shows low accuracy which has a little above 60 percent. This low accuracy value occurs because it has low value of alpha for Laplace smoothing there are words which counts 0. In order to perform high accuracy, we need enormous size of text data which has the huge amount of emotion representing words. However, with this gigantic size of dataset, the implementation runs in inefficient time to get informative value.

B. *Quantitative analysis*

- Using the efficient data structure and mathematical concepts, it performs timely to show informative data from analyzing. This quantitative analysis takes usually polynomial time and especially, max Fibonacci heap takes O(1) in amortized time complexity. This performance hardly requires the enormous size of dataset. However, it efficiently performs and provide high accuracy since it is based on fundamental arithmetic operations instead of high abstract mathematical theory.

C. *Sophisticated classification*

-The most important criteria to have accuracy of interpretation human language is to construct highly detailed classified dataset. The more densely the dataset is collected, the more accurate output can come out. Naïve Bayes theorem implements classifications with well distributed data. However, the accuracy depends on how appropriate Naïve Bayes model is used in each different condition such as the size of data, the number of detectable words, and the distribution of data.

VI. REFERENCES

[1] Walaa Medhat, Ahmed Hassan, Hoda Korashy, “Sentiment analysis algorithms and applications: A survey (unpublished work style)” Ain Shams University, September 2013.

[2] Charu C. Aggarwal, ChengXiang Zhai, “Mining Text Data”, University of Illinois at Urban-Champaign.

[3] James D Thomas, Katia Sycara, “Integrating Genetic Algorithms and Text Learning for Financial Prediction”, Department of Computer Science Carnegie Mellon University

[4] Armin Seyeditabari, Narges Tabari, Wlodek Zadrozny, “Emotion Detection in Text: a Review” UNC Charlotte, June 2018.

[5] Ricardo Mendes, Joao P. Vilela, “Privacy-Preserving Data Mining: Methods, Metrics, and Applications”, Department of Informatics Engineering, University of Coimbra https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7950921

[6] Farhad Malic, “NLP: Text Mining Algorithms”, Jun 2019. https://medium.com/fintechexplained/nlp-text-mining-algorithms-4546c6ca30a

[7] Vishal Gupta, Gurpreet S. Lehal,,“A Survey of Text Mining Techniques and Applications (published)” University Institute of Engineering & Technology, Panjab University Chandigarh, India, August 2009

[8] M. Rajman, R. Besançon, “Text Mining: Natural Language techniques and Text Mining applications (published)”, Artificial Intelligence Laboratory, Computer Science Department, 1998

[9] Text classification algorithm https://developers.google.com/machine-learning/guides/text-classification

[10] Javed Shaikh,“Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK”, July 2017 https://towardsdatascience.com/machine-learning-nlp-text-classification using-scikit-learn-python-and-nltk-c52b92a7c73a

[11] Shivam Bansal, “A Comprehensive Guide to Understand and Implement Text Classification in Python”, April 2018 https://www.analyticsvidhya.com/blog/2018/04/a-comprehensive-guideto-understand-and-implement-text-classification-in-python/

[12] Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L. E., & Brown, D. E., “Text Classification Algorithms: A Survey (published)”, University of Virginia, April 2019

[13] Alyona Medelyan, “Text Analytics Approaches: A Comprehensive Review (article)”, October 2018. https://getthematic.com/insights/5-text-analytics-approaches/ [14] Rui Xia, Chengqing Zong, Shoushan Li, “Ensemble of feature sets and classification algorithms for sentiment classification”, Chinese Academy of Sciences (CASIA), Beijing, November 2010.

[15] Shiv Naresh Shivhare, Prof. Saritha Khethawat, “Emotion Detection from Text”, Department of CSE and IT, Maulana Azad National Institute of Technology, Bhopal, Madhya Pradesh, India