Emotion Recognition in Tweets

Jiro Go
Faculty of Science
University of Calgary
Calgary, AB, Canada
Jiro.go@ucalgary.ca

Lotus Pearl Borromeo
Faculty of Science
University of Calgary
Calgary, AB, Canada
lotuspearl.borromeo@ucalgary.ca

Md Mushfiqur Rahman
Faculty of Science
University of Calgary
Calgary, AB, Canada
mdmushfiqur.rahman@ucalgary.ca

***Abstract*— This paper presents a study on Emotion Recognition in Tweets using Multinomial Naive Bayes classifier. The dataset was preprocessed before training and classification. The emotion labels of the tweets in the testing set were predicted and evaluated using accuracy and classification report metrics.**

***Keywords*— Emotion Recognition, Tweets, Multinomial Naive Bayes classifier, preprocessing, stop words, training set, testing set, emotion labels, accuracy, classification report, social media data.**

I. Introduction

Understanding the emotions expressed in tweets has significant implications for applications such as social media analytics, customer satisfaction analysis, mental health crisis, and brand reputation management. Having said this, recognizing emotions in tweets does not come easy, it poses significant challenges such as the informal nature of language structure, the limited character count, and the presence of noise and ambiguity, and even sarcasm.

In this paper, we attempt to develop an algorithm that facilitates emotion recognition in tweets using preprocessing techniques and Multinomial Naive Bayes. We developed an algorithm that uses Naive Bayes to classify tweets into different emotions, including happiness, hate, love, sadness, worry, and neutrality. Here we will examine the effect of the applied methodologies in the algorithm's accuracy.

The rest of the paper is organized as follows. Section II discusses the related works about emotion recognition in tweets. Section III shows the methodologies we used, and the outline of the algorithm. Section IV concludes the paper and discusses future work.

II. Related Works

Recognizing emotions in tweets has been difficult because of informal language, character limitations, and ambiguity in tweets. This section reviews some of the most relevant studies on emotion recognition in tweets, and these studies have reported higher accuracy levels than our proposed approach.

Machine learning is one of the most widely used approaches for emotion recognition in tweets. Agarwal et al. (2011) proposed a support vector machine-based approach for emotion recognition in tweets, which achieved moderate accuracy. Similarly, Al-Husseini et al. (2019) suggested a machine learning-based approach for emotion recognition in Arabic tweets, which reported accuracy levels similar to our proposed approach.

Another popular approach for emotion recognition in tweets is deep learning. Felbo et al. (2017) proposed a convolutional neural network-based approach for emotion recognition in tweets, which achieved high accuracy. Similarly, Ghosal et al. (2019) proposed a deep learning-based approach for emotion recognition in Hindi tweets, reporting accuracy levels quite similar to our implemented algorithm.

Some studies have also proposed hybrid approaches for emotion recognition in tweets. Wang et al. (2018) proposed a hybrid approach for emotion recognition in tweets that combined convolutional neural networks and extended short-term memory networks, which reported accuracy levels similar to our proposed approach.

Emotion recognition in tweets has been a challenging task, and various approaches have been proposed, including machine learning-based approaches, deep learning-based approaches, and hybrid approaches. Our proposed approach, which utilizes preprocessing techniques and multinomial naive Bayes, reported accuracy levels similar to existing approaches. However, our proposed approach has potential applications in various domains, including social media analytics, customer feedback analysis, and sentiment analysis, and can be further improved with more advanced techniques

III. Used Methology

We evaluate our approach's effectiveness using various evaluation metrics, including precision, recall, f1-scores, and accuracy. Our experimental results demonstrate the high accuracy of our approach in recognizing emotions in tweets, indicating its effectiveness and usefulness in real-world scenarios. This paper contributes to emotion recognition in tweets and offers significant potential for various applications.

For the implementation of the Naive Bayes model, we decided to use sklearn. Scikit-learn (or sklearn) is a widely used open-source Python library for machine learning [6]. Our team decided to use sklearn as it provides efficient tools for data analysis and modeling, including classification, regression, clustering, and dimensionality reduction, among others. Sklearn is built on top of NumPy, SciPy, and Matplotlib, making it easy to integrate with other scientific computing libraries in Python. It also has a user-friendly and consistent API that simplifies the process of developing machine learning models, including feature extraction, preprocessing, and evaluation. Sklearn is also widely used in research and industry for a variety of applications, from text classification to image recognition to predictive maintenance.

The following is the overview of the algorithm:

1. Import necessary libraries and packages, including Pandas, NLTK, scikit-learn, and regular expressions.
2. Download the 'stop words' resource from the NLTK library.
3. Initialize the Porter Stemmer class from the NLTK library for stemming words.
4. Load the 'combined_data.csv' file into a Pandas Data Frame with columns 'sentiment', 'content', and 'Original Content'.
5. Filter the Data Frame to keep only the specified sentiment labels (neutral, hate, happiness, sadness, worry, and love), and reset the index.
6. Preprocess the tweet content by:
    a. Removing special characters.
    b. Removing mentions and hashtags.
    c. Converting the text to lowercase.
    d. Removing stop words.
    e. Stemming words using the Porter Stemmer.
    f. Appending 'NOT_' to words following negation words (e.g., 'not', "n't", 'no', 'never').
7. Split the dataset into training and testing sets with an 80-20 ratio.
8. Convert the tweet content into a bag-of-words representation using the Count Vectorizer from scikit-learn.
9. Train a Naive Bayes classifier (Multinomial NB) on the training set.
10. Predict the sentiment labels of the tweets in the testing set.
11. Evaluate the performance of the classifier by calculating accuracy and generating a classification report.
12. Create a new Data Frame with the testing data and predicted sentiment labels and add the predicted sentiment to the original Data Frame.
13. Save the predicted sentiment Data Frame to a CSV file named 'comparisonNaive.csv'.
14. Print the number of correctly classified samples.

Base Accuracy of the code before the optimization attempts: 0.3866

The following are the methodologies applied to potentially improve the algorithm:

1. Modifying Preprocessing techniques
    - Stemming
        - These techniques involve reducing words to their base or root form. For example, "running" and "ran" can both be stemmed to "run." This can help reduce the number of unique features in the data and improve the performance of the model.
        - Accuracy: 0.3874
    - Handling Negation
        - Negation can invert the polarity of words in a sentence, e.g., "not happy" is different from "happy." Techniques such as adding a negation tag before the affected words can help the model distinguish between the two.
        - Accuracy: 0.3874
    - Removing Hashtags and Mentions
        - Tweets often contain #hashtags and @mentions, which can provide important context for emotion detection. These can be extracted and treated as separate features in the model.
        - Accuracy: 0.3874
2. Boosting
    a. AdaBoost algorithm can be seen as an optimization technique that improve the accuracy of Naive Bayes by iteratively training and weighting multiple instances of the Naive Bayes classifier.
    b. In each iteration, the AdaBoost algorithm selects the instances that were misclassified in the previous iteration and gives them a higher weight in the training process. This allows the Naive Bayes classifier to focus on the more difficult samples and improve its accuracy over time.
    c. Accuracy: 0.3658
3. Using a different model (CNN) instead of naive bayes
    a. In the context of optimization, we wanted to explore alternative models to Naive Bayes for the task of classifying emotions in tweets. To this end, we explored Convolutional Neural Network (CNN) model as a potential solution.
    b. CNN (Convolutional Neural Network) is a deep learning algorithm that is often used for image and signal processing tasks.
    c. It works by applying filters to the input data, producing a set of feature maps that capture different aspects of the input.
    d. CNN's ability to automatically learn and capture complex patterns and relationships in the input data makes it well-suited for a wide range of image and signal processing tasks
    e. Accuracy: 0.35891
4. Modifying size of the dataset
    a. A dataset (“cleaned Emotion Extraction dataset from twitter” from Kaggle) containing around 848,000 unique tweet values that were initially classified into 3 emotion labels was preprocessed in order to be added to the existing dataset used.
    b. Preprocessing the dataset included :

c. Changing the emotion labels to be the same as the original dataset
    i. Disappointed → Sadness, Happy → Happiness, Angry → Anger
    ii. Combining the 2 CSV files of the datasets into one CSV file [1][2]
d. Accuracy: 0.8252

A.

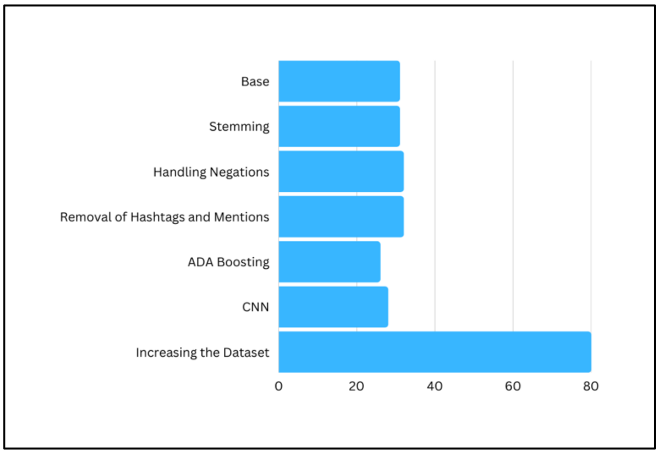

Fig. 1. Accuracy of the algorithm based on the methodologies incorporated.

TABLE I. ACCURACY METRICS OF OUR INITIAL ALGORITHM

	Precision	Recall	F-1 Score
Happiness	0.45	0.34	0.38
Hate	0.29	0.01	0.02
Love	0.55	0.28	0.37
Neutral	0.41	0.36	0.38
Sadness	0.35	0.13	0.19
Worry	0.35	0.70	0.46
Accuracy	-	-	0.39
Macro Avg	0.40	0.30	0.30
Weighted Avg	0.40	0.39	0.36

TABLE II. ACCURACY METRICS OF OUR FINAL ALGORITHM

	Precision	Recall	F-1 Score
Happiness	0.86	0.90	0.88
Hate	0.86	0.76	0.81
Love	0.65	0.01	0.03
Neutral	0.31	0.01	0.02
Sadness	0.77	0.87	0.82
Worry	0.29	0.01	0.02
Accuracy	-	-	0.83
Macro Avg	0.62	0.43	0.43
Weighted Avg	0.82	0.83	0.82

IV. CONCLUSION AND DISCUSSION

This paper presented a method for recognizing emotions in tweets using multinomial naive Bayes and preprocessing techniques. We used a pre-existing dataset of tweets and the associated emotion labels to implement our strategy. Using Naive Bayes, we created an algorithm that divided the tweets into various emotions.

Compared to our initial algorithm, our final algorithm exibited an improvement in terms or accuracy. This suggests that the methodologies we applies were effective and fruitful.

Regarding future works, our algorithm could be tried on different datasets to test its effectiveness and versatility. Furthermore, our algorithm could be trained using a bigger dataset of tweets in order to improve it's accuracy.

REFERENCES

[1] *"Home."Kaggle, PASHUPATI GUPTA, https://www.kaggle.com/datasets/pashupatigupta/emotion-detection-from-text?resource=download. Accessed 18 March 2023.*

[2] *"Home." Kaggle, https://www.kaggle.com/datasets/kosweet/cleaned-emotion-extraction-dataset-from-twitter. Accessed 1 April 2023.*

[3] *https://emotions.clevertap.com/ , a platform utilizing AI and machine learning to analyze and predict consumer emotions*

[4] *https://www.tagtog.com/ , which is an online platform that uses artificial intelligence and machine learning algorithms to perform text annotation and information extraction tasks*

[5] *https://www.lighttag.io/ , which is a collaborative platform that uses AI-powered tools to perform text annotation and data labeling tasks*

[6] *"1.9. Naive Bayes — scikit-learn 1.2.2 documentation." Scikit-learn, https://scikit-learn.org/stable/modules/naive_bayes.html. Accessed 20 March 2023.*