A Comparative study of identifying hate and offensive comments on social media platforms using text classification models and NLP techniques

- By Eranki Vasistha

Project progress report

Overview and Key changes made

The main goal of this particular phase of the project was to train the dataset and build a simple naïve bayes model to classify a comment (posted on twitter or any social media platform) in one of the three labels : 'offensive comment', 'hate comment' or 'neither'. Before training and building the model, data preprocessing was performed on the dataset and the resultant dataset along with a few properties will be described in the subsequent sections. Another model was built using the same naïve bayes algorithm but the feature space was reduced to 2000 features using feature selection method called 'Information gain'. I talk about this in more detail as well later in the paper. One key change which I made when compared to proposal is that I will now perform a comparative study on various machine learning methods and some NLP techniques to identify hate text rather than processing a two step hate text classification which I mentioned in the proposal earlier and that's why I changed the project name as well.

1 Literature Review

1.1 Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection

I used this paper to understand how to preprocess the dataset which includes a lot of social media *slangs* and a format of text which is exclusively used only in social media platforms such as usage of hashtag ('#') phrases. Another major learning from this paper was deciding which class labels to use and thus I am using the same three classes to label my dataset as this paper.

1.2 Effective hate-speech detection in Twitter data using recurrent neural networks

This research paper helped me further understand that how to transform my dataset effectively. Classifying the label, removing special characters and removing usernames were again inferred from this paper. I used the most common data preprocessing techniques for all the papers I read and this helped me feel more confident about my choices in general.

This paper also used neural networks to identify hate comments in a given datasets which I intend to use in the future as one of my primary models and compare it with the naïve bayes model where in I will try to understand which model is better in which aspect and why?

1.3 Thumbs up? Sentiment Classification using Machine Learning Techniques

This paper was given as a reading task in one of the previous weeks but I found it extremely helpful because this paper made me understand that while implementing text classification using any machine learning algorithms using unigram feature to represent our bag of words feature space is more effective than any of the bigram, trigram or n gram features. In the world of machine learning, almost always the most simple solution is generally the best solution to our problem.

1.4 A comparative study on feature selection in text categorization.

This paper helped me choose a good feature selection for my model. Information gain, density frequency and so on are some feature selection techniques which are generally used in the text classification process. Out of these, information gain has proven to be the most popular and effective feature selection methods where each feature has a value called *information gain*. The higher the value of the information gain, the more influential or important a feature becomes when it comes to classifying the text data.

1.5 Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text

This paper is again one of the previous week's reading task but this is very closely related to the kind of project I am working on here. Reading this made me understand that if I want to use Natural Language Processing (NLP) toolkits on the dataset and try and compare the results with machine learning techniques or model I build and use, it's probably easier better to use corpus dedicated to the linguistics used in the social media most often. My ultimate goal with this project is to obtain and report a comprehensive understanding of how different machine learning algorithms and NLP toolkits work on hate speech detection on social media platform like twitters.

2 Dataset description

The dataset taken for this project has 24783 unique tweets and retweets. There are 7 columns in the dataset and we mainly deal only with tweets and class columns. For better understandability of dataset and results, I programmatically converted class label 0 as *hate comment* , class label 1 as *Offensive comment* and class label 2 as *Neither* . Figure-1 shows the dataset distribution of the entire dataframe according to class labels.

This entire dataset was again converted into training and testing dataset by assigning 70% of the rows to training set and remaining rows to testing dataset. This resulted in training set containing 17348 unique rows and testing dataset having 7435 tuples of data. Figure-2 and Figure-3 shows the data output after dividing the entire dataset into training and testing sets and showing the amount of tuples or rows each class has in the training dataset.

We can clearly see from the figures that this current dataset is skewed towards offensive comment label. Thus my aim as the project progresses is to find a bit more balanced dataset and train and test the model using that dataset.

```
The number of offensive comments in the whole dataset are :  19190
The number of hate comments in the whole dataset are :  1430
The number of okay comments in the whole dataset are :  4163
```

Figure -1 : Dataset label wise data distribution

```
Training dataset length is :  17348  and testing dataset length is :  7435
```

Figure-2: Dataset distribution after dividing it into training and testing set

```
The number of offensive comments in the training dataset are :  13319
The number of hate comments in the training dataset are :  1103
The number of okay comments in the training dataset are :  2926
```

Figure -3 : Training dataset label wise data distribution

3 Preprocessing and Operations

The dataset underwent several preprocessing techniques to prepare it for modeling. URLs, special characters (@, !, &, etc.), and hashtags were removed. Although hashtags could potentially yield valuable features if deconstructed into separate words, they were omitted to assess the Naïve Bayes model's performance without advanced preprocessing. This simplification allowed us to evaluate whether the model could achieve satisfactory results without extensive data refinement. Additionally, users and usernames were removed, as they were deemed less relevant in identifying text classification. By taking this approach, we aimed to determine the baseline performance of the Naïve Bayes model and avoid unnecessary complexity. These changes were then written back to the csv file of the dataset as a new column called 'cleaned_tweets' and this column was used to train the model. Figure-4 shows a random example statement before and after preprocessing.

```
One random raw tweet is :  "@Crhedrys: Pussy licking pussy.... meow meow #StopWhite
&#8221;""""

&#128533;
 One random pre processed statement is :     pussy licking pussy  meow meow
```

Figure-4 : Text before and after applying data preprocessing and transformation

4 Feature Extraction and Feature selection

4.1 Feature Extraction

Unigram bag of words features space representation was used to extract the features from the cleaned tweets columns. [5] made me understand that when it comes to naïve bayes feature extraction, unigram feature extraction works more effectively than the any of the other bag of words feature representation like bigram, trigram or n-gram. This is mainly because the assumption that naïve bayes model makes of conditional independency and therefore, the occurrence of one word or one feature can also influence the classification of text. There was a total of 15344 unigram features extracted from the entire training dataset. The shape of the feature space of training and testing dataset is given in the figure -5

4.2 Feature Selection

The feature selection used in this project is k-best features using information gain or mutual information. This a powerful method which gives effective results which was proven in [6]. I will talking briefly about information gain now.

4.3 Information Gain

Information gain calculates the reduction in entropy or surprise from transforming a dataset in some way. It is commonly used in the construction of decision trees from a training dataset, by evaluating the information gain for each variable, and selecting the variable that maximizes the information gain, which in turn minimizes the entropy and best splits the dataset into groups for effective classification. Information gain can also be used for feature selection, by evaluating the gain of each variable in the context of the target variable. In this slightly different usage, the calculation is referred to as mutual information between the two random variables.

Information Gain, or IG for short, measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. A larger information gain suggests a lower entropy group or groups of samples, and hence less surprise.

In my dataset, I took 2000 best features out of the 15344 unigram features, i.e, the top 2000 features which has the highest information gain. One thing I would like to mention here as a possible disadvantage of feature selection using information gain is that it took almost 15-20 minutes for the code to reduce the feature space from 15344 to 2000. Thus, as number of features which were initially extracted is very high then the time to execute the feature selection process increases exponentially. The feature space after the feature selection was executed is shown in the figure-

This vector is then again trained to on the naïve bayes model and then the results of this model were compared to the model trained on the feature space with no feature selection.

```
Train feature space before filtering:  (17348, 15344)
Train feature space after filtering:  (17348, 2000)
Test feature space before filtering:  (7435, 15344)
Test feature space after filtering:  (7435, 2000)
```

Figure -5 : Feature space before and after feature selection process

5 Building my model

Naïve bayes approach was used by me to build a text classification model. The reason for using this as my preliminary model was naïve bayes is one of the simplest and trivial models for text classification and high understandability. It is a good starting point to train a dataset which is relatively new to me. As my project work progresses and I get more familiar with the dataset being used and it's nature, I will use more advanced and powerful solutions like neural networks to implement it. Another reason for choosing this as my first model, is that naïve bayes generally has low variance which means that it underfits the data a lot. Thus starting from a low point and moving up is a good approach to solving a problem.

I will now provide a brief summary about Naïve bayes concept and how it helps in text classification.

It The Naïve Bayes classifier relies on a simple, probabilistic approach, utilizing the bag-of-words representation to classify text. This representation can take two forms:

1. *Bag-of-Words (Binary):* It is also called the multivariate Bernouli model which uses simple 0 or 1 to represent if the word is present in the document or not.
2. *Bag-of-Words (Count-Based):* Also called as the multinomial model, here, the count of each word is stored which would help us know the relevance of the data present

6 Results and error analysis

The evaluation metrics used by me to measure the performance of my two models is are given below along with a brief description of each metric:

- **Precision**

  Evaluates the quality of positive predictions. A high precision score means the model is less likely to be wrong when it predicts a positive.
- **Recall**

  Assesses how sensitive the model is to positive instances. A high recall score means the model finds more positives.
- **F1 score**

  A harmonic mean of precision and recall that balances the importance of both metrics.

There are other metrics as well like confusion matrix, micro averaging and macro averaging which I have used in my model evaluation metrics but I will not describe about them in this report as the above three metrics are my main criteria for evaluating the models and their performances.

6.1 Performance of my model:

The performance of the two models: one with feature selection and the other model without feature selection are shown in the figures -6 and 7 below. We can clearly see that the naïve bayes model performs pretty well on this dataset with an accuracy of around 89% without the feature selection and 91% accuracy with the information gain feature selection model. However, this just a very high level analysis. If we go a step further with our performance metrics then, it shows that f1 score and recall for label hate comment is very near to zero when model without feature selection was trained. The same metrics get better when the feature selection using information gain was used. This means that feature selection not only increased the overall accuracy of my model but also is a better classifier of the text data.

```
Accuracy score:  0.881102891728312
Individual label performance:
                   precision    recall  f1-score   support

     Hate Comment       0.43      0.04      0.07       327
          Neither       0.88      0.61      0.72      1237
Offensive Comment       0.88      0.99      0.93      5871

         accuracy                           0.88      7435
        macro avg       0.73      0.54      0.57      7435
     weighted avg       0.86      0.88      0.86      7435

[[  12   32  283]
 [   3  754  480]
 [  13   73 5785]]
(0.881102891728312, 0.881102891728312, 0.8811028917283119)
```

Figure -6 : Evaluation metrics of naïve bayes model without the feature selection. As we an see, the hate comment label has very low recall and f1 score.

```
Accuracy score:  0.9004707464694015
Individual label performance:
                   precision    recall  f1-score   support

      Hate Comment       0.42      0.14      0.21       327
           Neither       0.85      0.77      0.81      1237
 Offensive Comment       0.92      0.97      0.94      5871

          accuracy                           0.90      7435
         macro avg       0.73      0.63      0.65      7435
      weighted avg       0.88      0.90      0.89      7435

[[  46   43  238]
 [   8  958  271]
 [  56  124 5691]]

(0.9004707464694015, 0.9004707464694015, 0.9004707464694015)
```

Figure -7: Evaluation metrics of naïve bayes model with the feature selection using information gain. As we an see, the hate comment label has a better score when compared to no feature selection model

6.2 Error Analysis and inferences :

The major bottle neck or errors which my naïve based model produced when trained on the dataset was not being able to classify hate comments label properly. If we see the confusion matrix of both the trained model, a lot of hate comments were falsely classified as offensive comments class. This means that our classifier is unable to differentiate between hate comments and offensive comments effectively. Thus, a more powerful technique has to be deployed in order to overcome this problem.

Another observation which I can make is that the dataset which I used for this phase, is highly skewed towards offensive dataset comments. I have found some more datasets which are balanced in terms of class labels and will be using those as well along with this dataset to train my future model as well as these two models. They need a bit more transformation and thus for now are not being used in this phase of the project. Also, I do feel that even though the dataset is highly skewed and it's initial results might be misleading, it is still a good point to start the project and make some initial strides with it. Thus, in this project report I stuck to this particular dataset.

7 Future Work

My main aim with this project is to understand what kind of text classification methods, text preprocessing methods and feature selection methods would work the best on social media comments and detection of hate text in them. Thus, I want to use various machine learning techniques and NLP methods as well to understand this process.

References

[1]H. Watanabe, M. Bouazizi and T. Ohtsuki, "Hate Speech on Twitter: A Pragmatic Approach to Collect Hateful and Offensive Expressions and Perform Hate Speech Detection," in IEEE Access, vol. 6, pp. 13825-13835, 2018, doi: 10.1109/ACCESS.2018.2806394.

keywords: {Speech;Feature extraction;Twitter;Voice activity detection;Task analysis;Sentiment analysis;Twitter;hate speech;machine learning;sentiment analysis},Galen Andrew and Jianfeng Gao. 2007. Scalable training of L_1-regularized log-linear models. In *Proceedings of the 24th International Conference on Machine Learning*, pages 33–40.

[2]Pitsilis, G.K., Ramampiaro, H. & Langseth, H. Effective hate-speech detection in Twitter data using recurrent neural networks. Appl Intell 48, 4730–4742 (2018). https://doi.org/10.1007/s10489-018-1242-yJames W. Cooley and John W. Tukey. 1965. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation*, 19(90):297–301.

[3] G. Koushik, K. Rajeswari and S. K. Muthusamy, "Automated Hate Speech Detection on Twitter," 2019 5th International Conference On Computing, Communication, Control And Automation (ICCUBEA), Pune, India, 2019, pp. 1-4, doi: 10.1109/ICCUBEA47591.2019.9128428. keywords: {Bag of Words;TFIDF;Hate Speech},

[4] Alexandre Pinto, Hugo Gonçalo Oliveira, and Ana Oliveira Alves. Comparing the Performance of Different NLP Toolkits in Formal and Social Media Text. In 5th Symposium on Languages, Applications and Technologies (SLATE'16). Open Access Series in Informatics (OASIcs), Volume 51, pp. 3:1-3:16, Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2016)

[5] Panneerselvam, Kalaiselvan. "Thumbs up? Sentiment Classification using Machine Learning Techniques."

[6] Yang, Yiming, and Jan O. Pedersen. "A comparative study on feature selection in text categorization." *icml*. Vol. 97. No. 412-420. 1997.

Mary Harper. 2014. Learning from 26 languages: Program management and science in the babel program. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. Dublin City University and Association for Computational Linguistics, page 1. http://aclweb.org/anthology/C14-1001.