Stance Classification

Nils Dunlop, 20010127-2359,
mailto:gusdunlni@student.gu.se

Francisco Erazo Piza, 19930613-921
guserafr@student.gu.se

Chukwudumebi Ubogu, 19810624-5320
mailto:gusuboch@student.gu.se

February 18, 2024

Abstract

Stance or sentiment, classification can be used as a tool to assess the temperature of a situation, through societal reflections on the situation. In an era of social media, these reflections can be collated through comments on various platforms, Reddit, YouTube, and Twitter to name a few. The ability to assess public sentiment promotes both an understanding of feelings on matters and where possible can allow for course correction depending on the objective of the sentiment assessment.

After running a few classification models on crowd-sourced annotated pro or against comments on the COVID-19 vaccination, the Multinominal Naive Bayes classifier, using the CountVectoriser found precision, recall, and F1 scores of 0.81, after tuning the parameters. Similarly when compared to to a basic dummy classifier model, using the CountVectoriser, the model was found to have 0.515% accuracy. Only guessing the most frequent stance being pro-vaccination hence showing the strength of the Multinominal Naive Bayes compared to simply guessing. In addition, after assessing a confusion matrix it was further confirmed that compared to other ML models considered the Naive Bayes Model coupled with the CountVectorizer led to the lowest confusion rates.

1 Introduction

Crowd-sourcing annotation is a method used to gather data/information by outsourcing tasks to a large group of people. It typically involves multiple individuals contributing to the labelling or tagging of data. For this project, students were recruited as annotators for COVID-19 data on pro and anti-vaccine sentiments expressed as comments. This collection effort was particularly used to leverage the power of the crowd to generate large datasets quickly and cost-effectively. It was unclear which platform was used by annotations, but a general allowance was for comments from social media.

This report demonstrates ***Stance classification*** where using supervised machine learning, using annotated data collected by students a text classifier is developed to determine whether a given textual comment expresses a positive or negative sentiment about COVID-19 vaccination. Crowd-sourcing annotation offers several advantages, including scalability, cost-effectiveness, and the ability to handle large volumes of data quickly. This notwithstanding, challenges remain, especially around annotation quality and the management of diverse annotator backgrounds and biases. However, crowd-sourcing annotation remains a respectable tool in machine-learning applications.

2 Assessing Consensus between data annotators

Annotation consensus aims to evaluate and reconcile multiple labels to a data point in a dataset. The process seeks to resolve conflicts in determining a single label that best represents a ”true” label of a data point. Inconsistency in error labeling can lead to poor model performance which can be costly to fix at a later stage. Industry practice, when using annotators appears to be several rounds of annotation on a single dataset to ensure label accuracy and validity.

2.1 Extent of consensus amongst annotators

While there are several methods to assess consensus amongst annotators, in this project it was kept simple using ***Majority Vote***.Looking at the labels, ***Negative (0) labels***: with 0/0 annotation, 19248 records were found. ***Positive(1) labels***: with 1/1 annotations, 18221 records were found. Therefore, of a total of 50 068 records, a total of 3*7 469, 74.8%* were found to be in *full* agreement (across annotators) on sentiment of the comments.

To settle the case of partial or no consensus, a function, *“extract-majority-label”*, was defined. This function returned the most frequent sentiment in cases of partial or consensus amongst annotators. This function was then applied to the training dataset. If after applying the “majority-label” function, and if there was still no consensus sentiment, these rows were dropped.

In the end, only training data with either full consensus or majority consensus were used in the model training process. The final dataset used for training had 42,840 records, implying an attrition rate of 15% from the original training dataset. This is only marginally better than the original training dataset with 84.38% sentiment with full consensus.

Given that there were different numbers of annotations per comment, on some there were only two, while in an extreme case one had 57 annotators, any comparison amongst annotators, that needed comparison of a fixed set of annotators could not be varied out.

2.2 Assessment of Data reliability

Pre-processing steps As part of the pre-processing step for this project, the data is split into comments and sentiment. Although most comments were annotated by at least 2 annotators some were annotated by more than two. The most extreme case in annotator numbers was **(57)**. In total, 50,068 comments were in the full training dataset. Of these, 84.38% of them had full consensus amongst the annotators, and 15.62% with partial or no consensus amongst the annotators.

Moreover, several preprocessing steps were implemented to ensure both training and test comments were optimally formatted for textual analysis. These preprocessing functions include::

- **Simple Preprocessing**: This function converts comments into strings, removes URLs, and replaces newline characters and multiple white spaces with a single space, thereby standardizing the text format.
- **Text Normalization**: Special characters and emojis, such as /@;:¡¿+= —.!?,‘, are addressed and normalized to ensure consistency across the dataset. This step is crucial for reducing noise and standardizing text representation.
- **Majority Sentiment Extraction**: For comments with more than two stances, the sentiment that represents the majority view of the annotators is extracted and assigned to the comment. This approach ensures the sentiment of a comment accurately reflects the consensus among annotators, providing a clearer understanding of the overall sentiment.
- **Clarification of Stance Ambiguity**: Following the majority vote removal, we further refined the dataset by eliminating the small subset of -1 stance instances totaling 22 cases. This decisive action was essential to enhance the dataset's clarity and ensure the reliability of its sentiment labels effectively reducing ambiguity.
- **Enhancement of Stop Words and Lemmatization**: In the concluding phase of data preprocessing, we extended the SpaCy library's list of stop words to better tailor the textual analysis to our specific context. The additional words added were *'vaccine', 'vaccination', 'vaccinate', 'vaccinated', 'use', 'people', 'person', 'like', 'think', 'know', 'case', 'want', 'mean', 'find', 'read', 'point'*. Additionally, we applied lemmatization to normalize the words, further refining the text for analysis.
- **TextBlob for Stance Classifictation**: To leverage text processing library for a robust comparison of this Stance Classification project, the *TextBlob library was used* to crosscheck the stance label of the data set with general sentiment extraction from sentences. As a simplified text processing library, it lends itself to speech tagging. As such, when the stance classification data is compared under Textblob, it is reveal that some "negative" classifications might in fact be neutral. However, the share of positive sentiments between TextBlob and project's were largely similar.

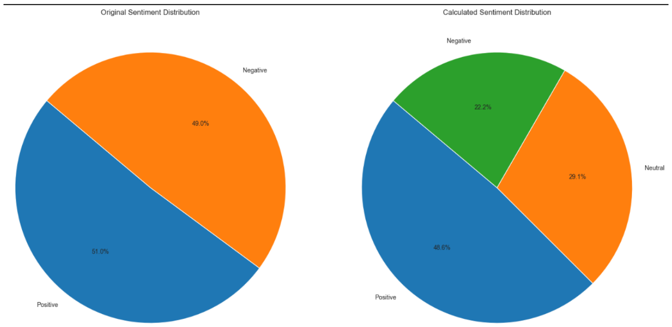


Figure 1: Comparison of label split vs Textblob sentiment classifier.

2.3 Processing of features

For feature extraction in stance classification, the nuanced composition and sequence of sentences play a crucial role. Given this complexity, basic feature extraction methods fall short. The TfidfVectorizer was primarily employed for this project emphasizing the relevance of words within documents. It operates by calculating Term Frequency (TF): the occurrence rate of words in a document, and Inverse Document Frequency (IDF): which assesses a word's significance across a document corpus. This approach de-emphasizes common terms while highlighting unique ones by integrating stopwords treatment to minimize their influence.

Additionally, the CountVectorizer was a supplementary tool focused on the raw occurrence count of words offering a straightforward measure of textual features absent in the nuanced weighting of TfidfVectorizer. This direct counting complements TfidfVectorizer's depth by providing a baseline frequency perspective, essential for analyzing text for stance classification without the complexity of Tfidf's scoring mechanism.
By integrating both, harnessed the TfidfVectorizer's ability to evaluate word importance and CountVectorizer's straightforward frequency counts for a balanced feature extraction strategy that aligns with the intricate demands of stance classification.

3 Learning Algorithm selection

3.1 Models Tested

To tackle the sentiment analysis task, we carefully selected models based on their compatibility and performance in similar contexts. Our models include **Logistic Regression**, **Multinomial Naive Bayes**, **Linear Support Vector Machine** and **Random Forest**. Here's a brief overview of each model's relevance to our project:

Logistic Regression is renowned for its simplicity and efficiency in binary classification problems. Given the sentiments are binary (0 or 1) the fit was natural. Logistic Regression excels where there is a roughly linear relationship between text features and target outcome. It offers clear insights on what words hold more weight in classification, and is scalable to large datasets.
Multinomial Naive Bayes is specifically designed for text classification tasks with discrete features (e.g., word counts or frequencies). MNB is efficient and requires a small amount of training data to estimate the parameters, and is suitable for text classification with large datasets. It is simple in practice, with surprising efficiency in text classification tasks.
Linear Support Vector Machines constructs a hyperplane in a high-dimensional space to separate different classes. It works for text classification when the data is linearly separable or can be transformed into a linearly separable space. While it was unclear if the dataset is linearly separable, if it was it would be in a high-dimensional spaces SVM would thrive.
Random Forest is an ensemble learning method that constructs multiple decision trees during training and combines their predictions through voting or averaging. It is appropriate for text classification tasks with complex nonlinear relationships between features and target classes. It also is resilient to over fitting, and this suitable for text classification tasks with high-dimensional, noisy feature spaces.

These models were chosen for their proven track records in text analysis, each bringing unique strengths to the sentiment classification task from interpretability and efficiency to handling complex feature interactions. Below are the best accuracy scores achieved on the training set by the baseline models showing their effectiveness in the sentiment classification task:

Model	Best Accuracy on Training Set
Logistic Regression	0.806
Multinomial Naive Bayes	0.811
Linear Support Vector Machines	0.815
Random Forest	0.791

Table 1: Best Accuracy Scores of Baseline Models on Training Set

3.2 Hyperparameter tuning process

After establishing baseline models, we proceeded to fine-tune their hyperparameters to tailor their performance more closely to our dataset. Hyperparameter fine-tuning is a crucial step in optimizing a machine learning model to enhance its effectiveness in tasks like document classification. These parameters, set before the training starts play a essential role in guiding the learning process but are not directly learned through training. The adjustment of hyperparameters affects several aspects of the model: it modulates the models complexity, adjusts the learning rate, and calibrates the strength of regularization.

Hyperparameter tuning aims to enhance model performance, prevent overfitting, boost robustness, and optimize computational resource use, ensuring accurate, generalizable, and efficient sentiment analysis.

Through hyperparameter tuning we attempted to strike an optimal balance that leverages the strengths of each model, ensuring they are well-suited to tackle the nuances of our specific sentiment analysis task. Below are the best accuracy scores achieved on the test set by these hyperparameter-tuned models:

Model	Best Accuracy on Training Set (Tuned)
Logistic Regression	0.655
Multinomial Naive Bayes	0.808
Linear Support Vector Machines	0.708
Random Forest	0.711

Table 2: Best Accuracy Scores of Hyperparameter-Tuned Models on Test Set

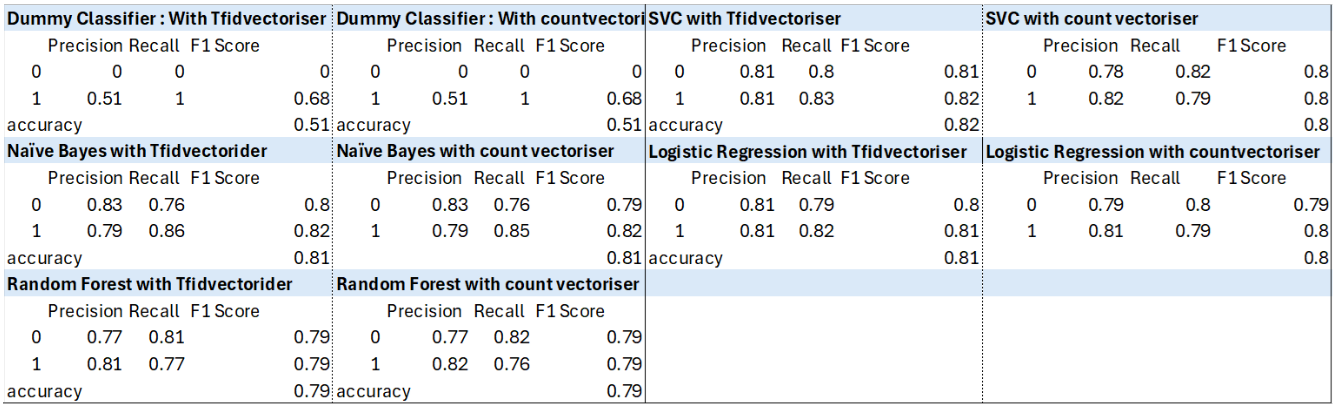
4 Quality Evaluation

4.1 Comparison to the trivial baseline

To determine the effectiveness of the models, benchmark was established a dummy classifier, and configured to predict the most frequent label which would simulate a naive approach that uniformly assigns the predominant class to all instances. This baseline classifier achieved an accuracy of 51.4% by uniformly predicting comments as positive. This comparison framework enables a more meaningful evaluation of our models performance relative to a simplistic strategy. Notably, our hyperparameter-tuned Multinomial Naive Bayes model significantly outperformed this trivial baseline, achieving an impressive accuracy of 83% on the final test dataset compared to the baseline's modest 49%. This strong contrast underscores the efficacy of our modeling approach and the value of sophisticated machine learning techniques in sentiment analysis tasks.

4.2 System error assessment

A confusion matrix categorizes predictions into true positives, true negatives, false positives, and false negatives therefore showing the accuracy of the model's predictions in comparison to actual sentiments. The fewer the misclassifications, the more effectively the model generalizes across data. Presented below are the confusion matrices for the evaluated models showcasing their predictive performance in detail.



Dummy Classifier : With Tfidvectoriser

	Precision	Recall	F1 Score
0	0	0	0
1	0.51	1	0.68
accuracy			0.51

Dummy Classifier : With countvectori

	Precision	Recall	F1 Score
0	0	0	0
1	0.51	1	0.68
accuracy			0.51

SVC with Tfidvectoriser

	Precision	Recall	F1 Score
0	0.81	0.8	0.81
1	0.81	0.83	0.82
accuracy			0.82

SVC with count vectoriser

	Precision	Recall	F1 Score
0	0.78	0.82	0.8
1	0.82	0.79	0.8
			0.8

Naïve Bayes with Tfidvectorider

	Precision	Recall	F1 Score
0	0.83	0.76	0.8
1	0.79	0.86	0.82
accuracy			0.81

Naïve Bayes with count vectoriser

	Precision	Recall	F1 Score
0	0.83	0.76	0.79
1	0.79	0.85	0.82
			0.81

Logistic Regression with Tfidvectoriser

	Precision	Recall	F1 Score
0	0.81	0.79	0.8
1	0.81	0.82	0.81
accuracy			0.81

Logistic Regression with countvectoriser

	Precision	Recall	F1 Score
0	0.79	0.8	0.79
1	0.81	0.79	0.8
			0.8

Random Forest with Tfidvectorider

	Precision	Recall	F1 Score
0	0.77	0.81	0.79
1	0.81	0.77	0.79
accuracy			0.79

Random Forest with count vectoriser

	Precision	Recall	F1 Score
0	0.77	0.82	0.79
1	0.82	0.76	0.79
accuracy			0.79

Figure 2: Baseline Models Classification Report

The table below showcases a selection of instances where the preferred model incorrectly predicted the sentiment. These examples highlight potential areas for model improvement:

Predicted	Actual	Text
Positive	Negative	step daughter is in hospital on the stroke ward. she's had the AZ vaccine. she's 25 years old I'm beside myself
Positive	Negative	I have covid right now and I was never vaccinated. Had a fever for the first two days but now I'm fine. My mom got the vaccine and booster (she has covid too) and shes a lot worse than me
Positive	Negative	My youngest son became autistic after having the MMR vaccine. They are full of it
Negative	Positive	Well everyone needs to get vaccinated to stop the spread, that's the only solution to wiping out these variants, make it law, get vaccinated or go to jail lol
Positive	Negative	URGENT! No more vaccination pass obtained with an infection! Olivier Véran announced this evening that it will be necessary to have received at least one dose of vaccine and an infection to obtain the vaccine pass. It's becoming more and more clear, Pfizer runs this country!

Table 3: Sample of Incorrect Predictions

These mispredictions could stem from various factors, such as the models inability to grasp the context fully or interpret the sentiment conveyed through sarcasm or nuance in language. Furthermore, the presence of keywords typically associated with positive sentiments in negative contexts and vise versa could mislead the model, showcasing the challenge of accurately categorizing sentiments in complex or emotionally charged statements.

5 Conclusion

The comparative analysis of various models using TfidfVectorizer and CountVectorizer revealed significant performance differences.Notably, the Multinomial Naive Bayes model, when fine-tuned and paired with the CountVectorizer, emerged as the top performer, achieving an impressive accuracy, precision, and recall score of 0.82 on the final test dataset.

The journey towards achieving accurate stance classification was not without its challenges, particularly in data annotation and preprocessing. Ambiguities and inconsistencies in labeling, along with the diligent selection of stopwords, underscored the critical importance of data quality and preprocessing techniques. To ensure the integrity of our dataset, we prioritized comments with clear consensus and majority classification, reinforcing the significance of data quality in obtaining reliable classification outcomes.

Inter-annotator Consensus: The task would have benefited from a more streamlined approach to annotator consensus. Implementing measures such as Cohen's Kappa for two raters or Fleiss's Kappa for a fixed number of raters could enhance the robustness of inter-annotator agreement. However, due to the variable number of annotators per comment applying these measures would have resulted in a considerable reduction in dataset size a compromise we were not prepared to make.

Vectorization Techniques: Our exploration of vectorization techniques underscored the distinct advantages of Tfidf and Count vectorizers. Looking forward, the incorporation of advanced models like BERT offers exciting future analysis. BERT's ability to comprehend the nuanced context of text through bidirectional understanding positions it as a state-of-the-art tool for enhancing the depth of text analysis, especially in complex classification tasks like stance detection.

As the relevance of stance classification continues to ascend, particularly in arenas like social media monitoring, opinion mining, and misinformation detection, the stakes have never been higher. The alarming role of platforms like Facebook in disseminating hate speech during events like the 2017 Rohingya crisis in Myanmar accentuates the potential of stance classification in preempting and mitigating the spread of harmful narratives. Looking ahead, the deployment of advanced stance classification algorithms could very well be a pivotal tool in safeguarding against the escalation of online hate speech into real-world violence, underscoring the profound societal impact of this field of study.

References

Crystal, C., 2023, The Carnegie Endowment for International Peace, *Facebook, Telegram, and the Ongoing Struggle Against Online Hate Speech*, online at `https://carnegieendowment.org/2023/09/07/facebook-telegram-and-ongoing-struggle-against-online-hate-speech-pub-90468#:~:text=In%202018%2C%20the%20United%20Nations,hate%20speech%20in%20the%20country`.

Yantseva, V., and Kucher, K., 2021, Linneuniversitetet Publications, *Machine Learning for Social Sciences: Stance Classification of User Messages on a Migrant-Critical Discussion Forum*, online at `https://lnu.diva-portal.org/smash/record.jsf?pid=diva2%3A1616667&dswid=5316`.

Wikipedia, The Free Encyclopedia, *Inter-rater reliability*, online at `https://en.wikipedia.org/wiki/Inter-rater_reliability`.