COMP30027 Report:
NLP research on Tweet's emotion

Yuntao(Lulu) Lu,
Jiahe(Grace) Liu

1. Introduction

With the exponential development of technology, people are able to express their opinions and attitudes through social media. At the same time, a massive number of messages are broadcasted (Perera & Karunanayaka, 2022). To study these data, sentiment analysis, also known as opinion mining, is introduced to analyze people's opinion and attitudes in the form of written text through computational analysis (Liu, 2015). This method can help us understand the general idea behind the text and improves the decision making in many fields (Mansouri et al., 2022). Therefore, this study aims to develop a sentiment classification model that classifies tweets' attitude into "positive", "negative" and "natural" labels. The provided datasets of this study include a *Training* dataset and *Testing* dataset where the *Training* dataset is composed of the tweet texts and labels whilst the *Testing* dataset only contained the tweet texts.

In order to achieve the study aim, we proposed the following hypothesis:

- For data splitting, in general, models with Cross-fold Validation generate a higher accuracy than Random Hold-out.
- For vectorization, in general, models with TFIDF vectorization process outweigh Bag of Words (BoW).
- Feature selection, in general, improves the training speed and overall accuracy.
- Support Vector Machine (SVM) outweighs Logistic Regression and Random Forest in general cases.
- Stacking will consolidate these models and generate an optimal result.

2. Method

This study will follow the general NLP principles with the steps in Figure 1:

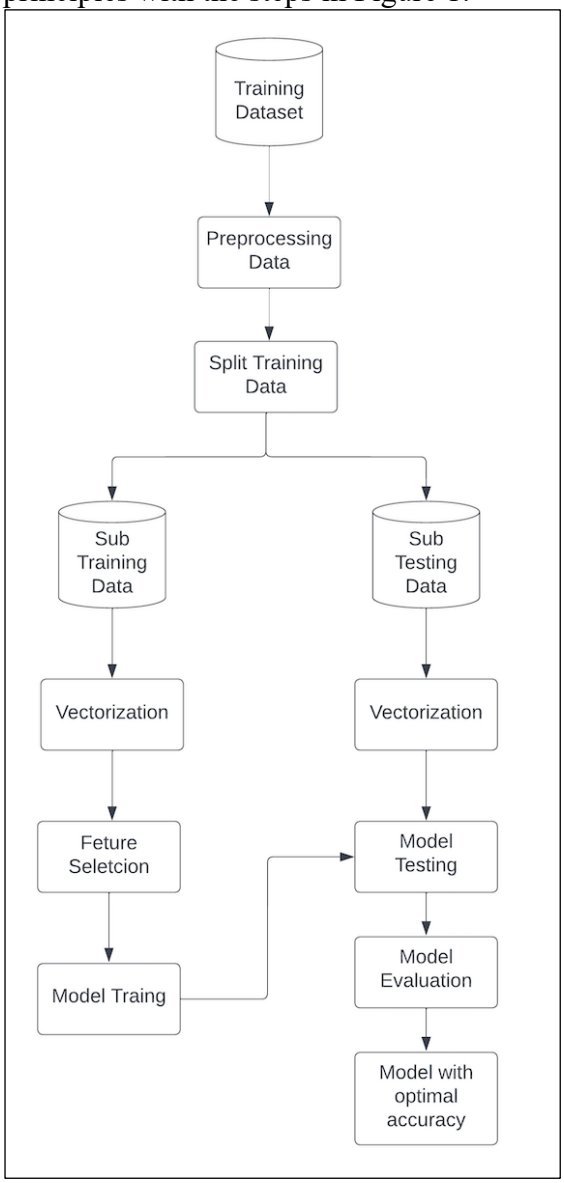

Figure 1- Approach Overview

To test our hypothesis, we designed our study with the following procedures to collect relative data:

- Conduct Cross-fold Validation and Random Hold-out to split the *Training* dataset respectively.
- Vectorize datasets with BoW and TFIDF correspondingly. For each vectorized dataset generate 1-gram and 2-gram word-grouping respectively.
- For each 1-gram and 2-gram models, compare models with no feature selection and with K-Best feature selection
- Train classification models including Baseline Model, Logistic Regression, Random Forest, SVM, and Stacking respectively.

The generated data will enable us to make the comparative analysis in the following aspects in Table 1:

Data Splitting (BoW & TFIDF)	Model Accuracy (in average)	Cross-validation	Random Hold-out
	LG		
	SVM		
	RF		

Data Vectorization	Model Accuracy (in average)	BoW	TFIDF
	LG		
	SVM		
	RF		

N-gram Vectorization	Model Accuracy (in particular)	BoW 2-gram no feature seletcion	BoW 1-gram no feature seletcion
	LG		
	SVM		
	RF		

Feature Selection	Model Accuracy (in particular)	BoW 1-gram with feature selection	BoW 1-gram no feature selection
	LG		
	SVM		
	RF		

Table 1 - Data to be Collected

2.1 Preprocessing Data

2.1.1 Data Cleaning

Data cleaning modifies data which is irrelevant, repetitive, or inappropriately formatted. In our study, to clean the unstructured tweet texts, we applied the following data cleaning approaches in Figure2:

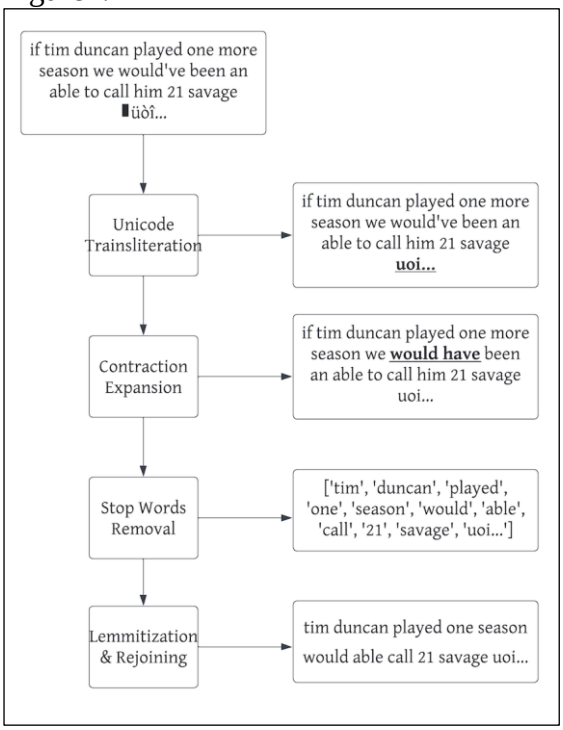

Figure 2- Data Cleaning

2.1.2 Data Splitting

Since the given *Testing* dataset's true labels were hidden, we split the *Training* dataset into a *Sub-training* dataset and a *Sub-testing* dataset where both datasets are now labelled. Moreover, we placed this procedure prior to vectorization so that the *Sub-testing* dataset can be treated as unseen testing data.

In our study, we compared the overall accuracy for Random Hold-out and Cross-fold Validation in different models respectively to test whether Cross-fold Validation would provide a better result. Random Hold-out was conducted with the splitting ratio of 3.58: 1 and Cross-fold Validation repeated the splitting process for 3 times. The splitting ratio for Random Hold-out was the same as the ratio between the given *Training* and *Testing* dataset to mimic the similar context.

2.1.3 Vectorization

The reason for vectorization is that machine learning models do not recognize texts directly. Nevertheless, they accept the input as a set of vectorized numerical

features (Juluru et al., 2021). Therefore, BoW and TFIDF are used to establish the connection between texts and models.

We performed TFIDF and BoW on selected models respectively and compared the accuracy. The comparison focused on detecting the significance of weighing approach as BoW distributed the same weight for all features whilst TFIDF weighed each feature according to its appearing frequencies. Thus, we were expected to test whether the weighing strategy was more prominent for this study.

Moreover, we applied and compared the word grouping of 1-gram and 2-gram respectively for BoW to verify the hypothesis that 2-gram would outperform 1-gram. As relevant work suggested that extracting N-gram features provides a better result for model training (Kaur et al., 2021), regarding each word as a feature may lose some important information.

2.1.4 **Feature Selection**

Feature Selection is an approach to reduce data dimension by selecting the best features through their statistical test rankings.

In this study, we compared models with and without features selection to test whether feature selection improves the training speed and overall accuracy. We conducted a Chi-square statistical testing to select the top K vectorized features with highest correlation to the label. The reason for conducting the Select K-best feature selection method on the *Sub-training* dataset was that it specified the exact number of selected features so that the *Sub-testing* dataset could be adjusted to have the same dimension. Thus, the reason not choosing Variance Threshold was that we couldn't specify the exact number of selected features directly to keep the dimensions fit.

2.2 **Model Selection**

Selected models for this study were Baseline Model, Logistic Regression, SVM, Random Forest and Stacking. The reason for eliminating KNN was because KNN did not perform well with large datasets of high dimension. For large datasets, the distance between the chosen point and other points are similarly large which may impair the algorithm's accuracy.

As each model was consisted of different hyperparameters, the optimization process for each model lay in finding the most promising hyperparameters. In our study, we firstly evaluated important hyperparameters and then applied *Grid Search* to generate the optimal combination of the potential values of hyperparameters.

2.2.1 **Baseline Model**

Firstly, we applied the Baseline Model to identify important patterns of the dataset using the most frequent label as the dummy classifier. This approach assisted us to discover the distribution of class labels for our *Training* dataset and make further manipulations for a more accurate prediction result.

2.2.2 **Logistic Regression**

We chose Logistic Regression because it is time-efficient when dealing with large datasets. In this study, as we aimed to classify tweets into three classes, a Multinomial Logistic Regression could be applied. Moreover, due to its binary algorithm, Logistic Regression generates a high processing speed and is very time efficient when dealing with large datasets which highly suit this study context.

Solver is an important hyperparameters that decides the choice of algorithm based on whether dataset is multiclass. In this study, since we have three classes "positive", "negative", and "neutral", a multinomial logistic regression was applied to observe the correlation between each feature and class. Thus, we chose the

lbfgs solver. Meanwhile, we set the *max iteration number* equal to 10,000 which enabled us to generate a more generalized result.

2.2.3 **Supported Vector Machine (SVM)**

SVM was chosen because it can be applied to highly dimensional data and understand complex relationships. This is because when dealing with data of high dimension, SVM is capable of transforming data into required form through suitable kernel functions. In this study, since each unique word or phrase represented a feature, the dataset was of large dimension. Thus, SVM was suitable.

For important hyperparameters, we used Grid Search to find the optimal *C value* and *kernel function* where *C value* represents the inverse of regularization power and the *kernel function* represents the mathematical approach used to transform the data point's dimension. In general, lower *C value* enables higher error tolerance, resulting in a more general model. After experimenting we found and used the optimal *C value* of 0.01 and the kernel *function* of 'linear'.

2.2.4 **Random Forest**

We also applied Random Forest which is a tree-based algorithm consisting of many decision trees built through sets of randomly selected features. The reason for choosing this model instead of decision tree was that Random Forest could control overfitting effectively and generate a higher prediction accuracy. As our highly dimensional dataset might result in a complex model, overfitting was very likely to occur if only decision tree was applied.

Crucial hyperparameters included the *number of decision trees*, *node splitting methods*, *max depth*, *minimal sample to be at a leaf node*, *minimal sample to be split at a leaf node*, and *the maximum number of features considered for best split.* Grid Search was applied beforehand to find the optimal hyperparameter combination.

2.2.5 **Stacking**

Stacking is a heterogeneous meta classifier combining different individual classifiers which learns the optimal combination through using Logistic Regression to determine the weight of each sub-models (Zhou, 2012). In this study, we used a stacking model to combine the mentioned models.

2.3 **Evaluation**

For models mentioned above, we evaluated each model in terms of its accuracy and time complexity. In addition, for models with the highest accuracy, we analyzed its macro-averaging and weighted-averaging Precision and Recall. Analysis was made based on the dataset features discovered through Baseline Model.

3. Result

Through the Baseline Model, as demonstrated in Figure 3, we discovered that the class labels were unevenly distributed. Neutral labels indicated dominance across *Sub-testing* dataset and *Training* dataset.

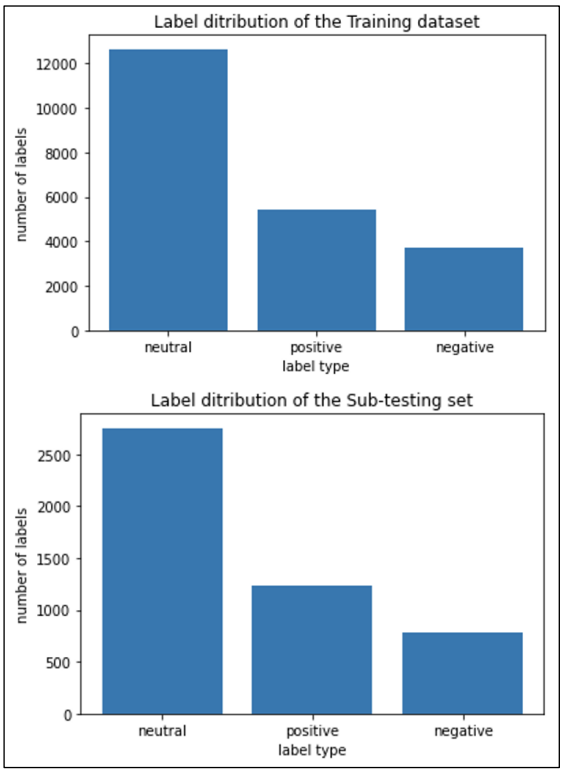

Figure 3- Data Distribution

Table 2 suggests that Cross-validation generated a higher averaged accuracy than Random Hold-out across all models. This predominance was valid under both BoW and TFIDF cases. Another overwhelming advantage was observed in N-gram vectorization where 1-gram outweighed 2-gram under all studied cases.

On the other hand, in most cases BoW demonstrated a better performance than TFIDF. However, this superiority was not absolute as TFIDF outperformed BoW with a 0.1% higher average accuracy under RF cases. Similarly, data preprocessed with feature selection demonstrated a generally higher accuracy in LG and RF whereas SVM generated a slightly better performance where no feature selection was applied.

Data Splitting (BoW)	Model Accuracy (in average)	Cross-validation	Random Hold-out	**Cross-validation > Random Hold-out**
	LG	0.6486	0.6259	**Yes**
	SVM	0.6331	0.6306	**Yes**
	RF	0.5883	0.5872	**Yes**
Data Splitting (TFIDF)	Model Accuracy (in average)	Cross-validation	Random Hold-out	**Cross-validation > Random Hold-out**
	LG	0.63	0.6249	**Yes**
	SVM	0.5876	0.5863	**Yes**
	RF	0.5887	0.5882	**Yes**
Data Vectorization	Model Accuracy (in average)	BoW	TFIDF	**BoW > TFIDF**
	LG	0.63725	0.62745	**Yes**
	SVM	0.63185	0.58695	**Yes**
	RF	0.58775	0.58845	**No**
N-gram Vectorization	Model Accuracy (in particular)	BoW 2-gram no feature seletcion	BoW 1-gram no feature seletcion	**2-gram > 1-gram**
	LG	0.6045	0.6416	**No**
	SVM	0.6107	0.6601	**No**
	RF	0.5814	0.5831	**No**
Feature Selection	Model Accuracy (in particular)	BoW 1-gram with feature selection	BoW 1-gram no feature selection	**Feature Selection > No Feature Selection**
	LG	0.6565	0.6416	**Yes**
	SVM	0.6523	0.6601	**No**
	RF	0.599	0.5828	**Yes**

Table 2 – Grouped Data

Model	**Vectorization**	**N-gram**	**Feature selection**	**Random Holdout Testing Accuracy**	**Cross Validation Testing Accuracy**
LR	BoW	1 gram	None	0.646	0.6416
		2 gram	None	0.6115	0.6045
		1 gram	Chi2 Select K Best	**0.6877**	0.6565
		2 gram	Chi2 Select K Best	0.6492	0.6011
	TFIDF	1 gram	None	0.6543	0.6548
		2 gram	None	0.602	0.5986
		1 gram	Chi2 Select K Best	**0.6618**	0.6519
		2 gram	Chi2 Select K Best	0.602	0.5942
SVM	BoW	1 gram	None	0.6595	0.6601
		2 gram	None	0.6034	0.6107
		1 gram	Chi2 Select K Best	**0.6624**	0.6523
		2 gram	Chi2 Select K Best	0.6071	0.599
	TFIDF	1 gram	None	**0.5959**	0.5925
		2 gram	None	0.582	0.581
		1 gram	Chi2 Select K Best	0.5903	0.5911
		2 gram	Chi2 Select K Best	0.582	0.5806
RF	BoW	1 gram	None	0.5814	0.5831
		2 gram	None	0.5781	0.5814
		1 gram	Chi2 Select K Best	**0.6012**	0.5971
		2 gram	Chi2 Select K Best	0.5926	0.5871
	TFIDF	1 gram	None	0.5816	0.5828
		2 gram	None	0.5781	0.5814
		1 gram	Chi2 Select K Best	**0.6014**	0.599
		2 gram	Chi2 Select K Best	0.5939	0.5896
Stacking			0.5857		

Table3 – Ungrouped Data

Table 3 suggests that among all model combinations, the optimal classifier was LR with 1-gram and feature selection where the accuracy was 66.37%. Moreover, it was found that for each model with different vectorization techniques, the highest accuracy appeared to be among models where data was split with Random Hold-out.

However, Cross-fold Validation still demonstrated an averagely higher accuracy.

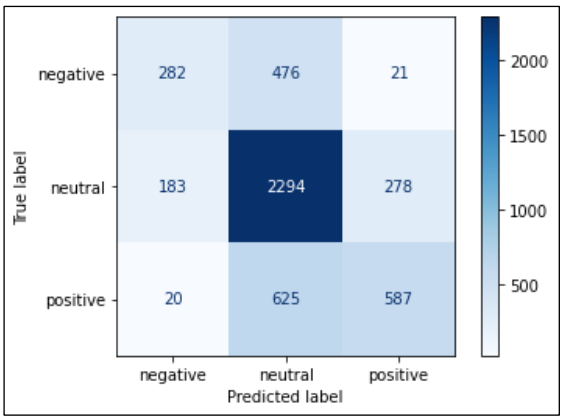

Table3 – Confusion Matrix for Chosen Model

For the model with highest accuracy, we conducted further evaluation in terms of Macro and Weighted Precision Recall and F1 score and Table 4 demonstrated that the weighted indices were generally higher than the macro indices.

LR with 1-gram BoW and Feature Selection			
	Precision	Recall	F1 score
Macro	0.6399	0.557	0.5822
Weighted	0.6569	0.6637	0.6475
Accuracy		0.6637	

Table4 – Precision Recall and F1 score on Macro and Weighted averaging

In addition, the study also found that in terms of computation time, LR was the fastest followed by RF and Stacking whilst SVM appeared to be the slowest.

4. Discussion

4.1 TFIDF versus BoW

The TFIDF vectorization did not demonstrate better performance than BoW in general. The mismatch between the result and our hypothesis may be that TFIDF failed to scale up the weight of words that are key for labelling. For instance, verbs or adjectives with strong emotions such as “hate” or “happy” might appear with a higher frequency in the corpus than some unimportant but rare words such as a person’s name. However, TFIDF may give higher weight to the name word as it was rarer, resulting in poor prediction accuracy. This indicates that for NLP with datasets consisting of a large number of varied features, the TFIDF’s weighting characteristic may backfire.

4.2 2-gram versus 1-gram

It was observed that under all studied cases, 1-gram generated higher accuracy than 2-gram. This discovery did not follow our original hypothesis either. The reason for this may the dramatic increase in the dataset’s dimension after using 2-gram where it proliferated from 36,182 to 149,048, enhancing calculation complexity.

Moreover, as combining neighbouring words introduced new features, this may lower the weight of key features highly correlated with label prediction. Since the key features had already been a minority, introducing 2-gram may exacerbate this disadvantage.

4.3 Models

4.3.1 SVM

Though SVM had good generalization performance and the overfitting risk was lowered, the drawback was evident. Since SVM can only directly make binary classification, for this multiclass classification problem, SVM had to break down the calculation with one-to-one or one-to-all approaches, resulting in a long calculation time of around 1016s. Thus, with the complex kernel function, it was difficult to calculate the probability of each feature and directly detect the decision boundary for higher dimensional datasets.

4.3.2 Random Forest

We observed that Random Forest was more time-efficient than SVM where the general operation time for RF is less than 10s. This may be because, similar to Bootstrap and Bagging, each decision tree was computed at the same time and did not interfere with each other. Similar to SVM, the overfitting issue is less likely to occur due to the algorithm’s randomness nature. However, a drawback of this algorithm is that we could not observe the logic behind each decision and make further analysis.

4.3.3 Logistic Regression

Logistic regression was comparatively easy to interpret as we could discover the correlation between feature and class through the output. Moreover, due to the simplicity of the algorithm, was efficient dealing with a large dataset usually with computation time less than a second. However, despite its outweighing to other models in most cases, due to its simplicity, there may be a risk of overfitting.

4.3.4 Stacking

Moreover, our last hypothesis was not guaranteed as Stacking did not generate a better output. The reason for this may be that Stacking is limited to use one preprocessed dataset. As we can see from Table 3, each model's optimal performance was generated with a different combination of preprocessing method. Thus, the chosen dataset in stacking may produce unbalanced results for each model, resulting in poorer results.

4.4 Cross validation versus Random Hold-out

The accuracy of cross-fold validation is higher than hold-out in most cases, which supports our hypothesis. To specify, as cross-fold validation involved more calculations, it produced a more accurate result but with a general 5 times of the calculation time. As demonstrated by Table 4, in all cases the optimal model occurred with Random Hold out. However, due to occasionality, this data splitting may not represent the actual status between the training and validation dataset and the accuracy is not as steady as Cross validation.

4.5 Evaluation

Though Accuracy was the highest index, since our dataset was unevenly distributed, we may infer that the weighted average is more reliable where neutral instances dominate the entire dataset. For weighted average, precision was more important to us. This is because as the model has more opportunities to learn about the texts with neutral labels, it may misclassify other labels as neutral. Therefore, to prevent this we may look for models with higher precision scores where false positive mistakes are less.

5. Conclusions

In conclusion, this study generated a classifier to predict the sentiment label for a tweet. This classifier uses 1-gram BoW vectorization technique and Random Hold-out data splitting followed by selecting K best features with Chi square mathematical analysis. It was found that, in general, 1-gram outperformed 2-gram and Cross-fold validation outweighed Random Hold-out. In most studied cases, BoW performed better than TFIDF. The study also found that LG demonstrated an overall better performance followed by SVM and RF whilst Stacking did not generate the optimal model. The limitations of this study may be that the evaluation was made mainly based on accuracy where there was a lack of consideration for precision and recall comparisons across all studied models.

6. Reference

Juluru, K., Shih, H. H., Keshava Murthy, K. N., & Elnajjar, P. (2021). Bag-of-Words technique in natural language processing: A primer for radiologists. *RadioGraphics*, *41*(5), 1420–1426. https://doi.org/10.1148/rg.2021210025

Kaur, S., Singh, S., & Kaushal, S. (2021). Abusive content detection in online user-generated data: A survey. *Procedia Computer Science*, *189*, 274–281. https://doi.org/10.1016/j.procs.2021.05.098

Liu, B. (2015). Sentiment analysis. *Cambridge University Press*, 1–17.

https://doi.org/10.1017/cbo9781139084789

Mansouri, N., Soui, M., Alhassan, I., & Abed, M. (2022). TextBlob and BiLSTM for sentiment analysis toward COVID-19 vaccines. 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA), 73–78. https://doi.org/10.1109/cdma54072.2022.00017

Mukherjee, A., Venkataraman, V., Liu, B. & Glance, N. What Yelp fake review filter might be doing? 7th International AAAI Conference on Weblogs and Social Media, 2013.

Perera, S., & Karunanayaka, K. (2022). Sentiment analysis of social media data using fuzzy-rough set classifier for the prediction of the presidential election. 2022 2nd International Conference on Advanced Research in Computing (ICARC), 188–193. https://doi.org/10.1109/icarc54489.2022.9754173

Rayana, S. & Akoglu, L. Collective opinion spam detection: Bridging review networks and metadata. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 985-994.

Zhou, Z. (2012). Ensemble Methods: Foundations and Algorithms (Chapman & Hall/CRC Machine Learning & Pattern Recognition) (1st ed.). Chapman and Hall/CRC.