Results and Analysis

1 Pre-processing Techniques Experiments

Table 11 summarises all experiments performed to identify the optimum set of pre-processing techniques for SVM and NB sentiment classification in this project. As Case Folding and Removing Unwanted Elements have been decided to be reliable NLP pre-processing techniques, these two techniques are used as starting experiments.

Experiment	Case Folding	Remove Unwanted Elements	Negation Transformation	Spelling Check	Repeated Letters Normalisation	Accuracy	AUC
SVM							
Starting	✓	✓				0.75	0.75254
1a	✓	✓	✓			0.75	0.75893
1b	✓	✓		✓		0.73	0.72912
1c	✓	✓			✓	0.75	0.75359
1d	✓	✓	✓	✓		0.73	0.72974
1e	✓	✓	✓		✓	**0.76**	**0.76943**
1f	✓	✓		✓	✓	0.74	0.74601
1g	✓	✓	✓	✓	✓	0.68	0.68793
NB							
Starting	✓	✓				0.74	0.74041
1a	✓	✓	✓			0.74	0.74980
1b	✓	✓		✓		0.73	0.73409
1c	✓	✓			✓	0.75	0.74802
1d	✓	✓	✓	✓		0.73	0.72493
1e	✓	✓	✓		✓	**0.75**	**0.74918**
1f	✓	✓		✓	✓	0.72	0.72018
1g	✓	✓	✓	✓	✓	0.70	0.69210

Table 1. Pre-processing Techniques Experiments

The three techniques experimented with are Negation Transformation, Spelling Check, and Repeated Letters Normalisation. Amongst the techniques, it is observe that implementing Spelling Check consistently lowers performance of sentiment classification. This can be explained such that the dataset used in this project may contain many short forms and slangs. After spelling correction, the meaning of these words and slangs may be altered, leading to a decrease in performance. For instance, Figure 25 shows a Tweet from the dataset where the Tweet author employed ‘for realsies’ as an internet slang for ‘for real’. This has mistakenly been autocorrected to ‘for realises’ which causes loss to meaning of original Tweet.

```
input_str = test_df['tweet'].iloc[1590]

output_str = spellingCorrection(input_str)

print ('original tweet: ' + input_str)
print ('autocorrected tweet: ' + output_str)
```

```
original tweet: i'm going to join a gym when i get back from florida. for realsies.
autocorrected tweet: i'm going to join a gym when i get back from florida. for realises.
```

Figure 1. Example of Tweet with inaccurate autocorrect

However, Negation Transformation and Repeated Letters Normalisation increases the performance of SVM and NB. By using only Negation Transformation in Experiment 1a with the starting pre-processing techniques, AUC for SVM sentiment classifier increases by 0.006 while AUC for a NB sentiment classifier increases by 0.009. Through Repeated Letters Normalisation in Experiment 1c with the starting pre-processing techniques, AUC for SVM sentiment classifier increases by 0.001 while accuracy of NB sentiment classifier increases by 0.01.

These two techniques are combined and implement in Experiment 1e to produce optimum results such as performance of SVM and NB are highest with the combined pre-processing techniques, in particular order of: Case Folding, Remove Unwanted Elements, Negation Transformation, Repeated Letters Normalisation.

2 Feature Extraction and Sentiment Classifiers Experiments

Table 12 summarises all experiments performed to observe the effect of feature extraction technique upon model performance. According to preliminary observations in Section 5.1, the most appropriate combination of pre-processing techniques are: Case Folding, Remove Unwanted Elements, Negation Transformation, Repeated Letters Normalisation. These pre-processing techniques are used across all following experiments.

Experiment	N-Grams	TFIDF	Accuracy	AUC
SVM				
2a	**1**	-	**0.78**	**0.77614**
2b	2	-	0.69	0.69192
2c	3	-	0.54	0.53102
2d	1,2	-	0.77	0.77446
2e	2,3	-	0.67	0.66320
2f	1,2,3	-	0.77	0.76693
2g	**-**	✓	**0.78**	**0.77705**
NB				
2a	1	-	0.75	0.75031
2b	2	-	0.71	0.70879
2c	3	-	0.60	0.60346
2d	1,2	-	0.77	0.76903
2e	2,3	-	0.67	0.70647
2f	**1,2,3**	-	**0.77**	**0.77033**
2g	**-**	✓	**0.76**	**0.75724**

Table 2. Feature Extraction and Sentiment Classifiers Experiments

The objectives for the experiments carried out in this section are:

- **To identify the optimum N-Gram techniques**

Experiments 2a, 2b, 2c

Unigrams (1-Gram), Bigrams (2-Gram), and Trigrams (3-Gram) are implemented as feature vectorisation techniques in Experiments 2a, 2b, and 2c respectively. The vectorised text computed from each technique are used to train SVM and NB sentiment classifiers to observe model performance. Amongst the three n-grams techniques, unigrams generated optimum results for both SVM and NB sentiment classifiers whereas classifiers built upon trigrams performed the worst.

Experiment 2d, 2e, 2f

Further optimisation through N-Gram feature extraction is attempted through hybrid N-Gram techniques. Varying combinations of unigrams, bigrams, and trigrams are experimented to observe the effect of an increased number of features per datapoint upon sentiment classification. Using a hybrid method could also improve context inclusivity leading to better performance. The results tabulated in Table 12 shows that a hybrid technique of unigrams, bigrams, and trigrams and improved the performance of the NB sentiment classifier by 2%. However, performance of SVM is consistently impaired through hybridised feature extraction. It is also worth noting that the hybrid unigram- bigram-trigram technique generated 220890 features which increased SVM training time up to 20 minutes.

The accuracy for SVM and NB according to feature space dimension is plotted in Figure 26. From left to right, the x-axes represents an increasing feature space dimension such that trigrams have the lowest feature space dimensions and uni-bi-trigrams have the highest feature space dimensions. The plot shows a general trend where accuracy of model increases along with an increase in feature space dimension.

According to the Bias-Variance Tradeoff, a model that is too simple with few parameters may have high bias and low variance, whereas a model that is too complex will have high variance and low bias. As it is seen that model performance increases along with feature space dimension, it can be deduced that the dataset performs better with models with higher complexity.

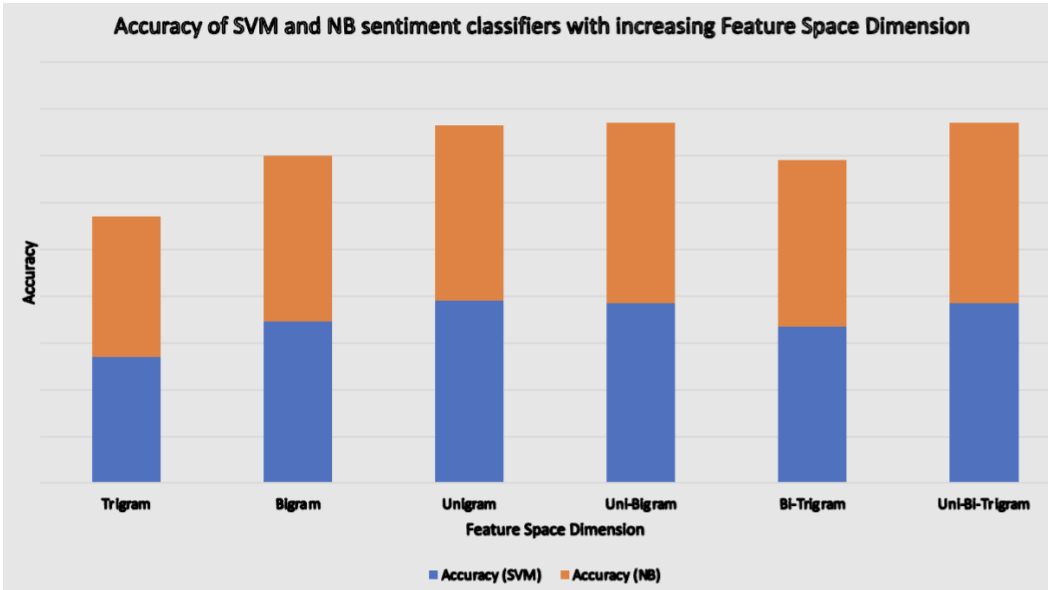

Figure 2. Accuracy against Feature Space Dimension defined by N-Grams

Results: Optimum N-Grams technique for SVM is unigrams which yields accuracy of 78% while optimum N-Grams technique for NB is a hybrid of unigrams, bigrams and trigrams which yields accuracy of 77%

- **To implement TF-IDF feature vectorisation**

Experiment 2g

SVM and NB model trained with TF-IDF vectorised texts achieved accuracy of 0.78 and 0.76 respectively. It is observed that TF-IDF achieves highest performance with an SVM model as compared to a NB model.

Results: SVM sentiment classifier built upon TF-IDF vectorised texts achieved accuracy of 78%, with AUC value of 0.77033, outperforming SVM with unigrams vectorised texts. NB sentiment classifier built upon TF-IDF vectorised texts achieved accuracy of only 76%, performing worse that NB with hybridised uni-bi-trigrams technique.

- **To establish baseline models for optimisation experiments carried out in Section 5.3**

The baseline models are established through identifying the best performing N-Grams techniques for each sentiment classifier, and models of TF-IDF implemented with each sentiment classifier. These baseline models are for comparison purposes such that each optimisation technique implemented is evaluated against baseline models to identify optimisation ability.

Results: Baseline models selected are highlighted blue in Table 13 and are as below:

Model	Feature Extraction Technique	Sentiment Classifier	Accuracy	AUC
A	Unigrams	SVM	78%	0.77614
B	TF-IDF	SVM	78%	0.77705
C	Unigrams, Bigrams, Trigrams	NB	77%	0.77033
D	TF-IDF	NB	76%	0.75724

Table 3. Baseline Models

3 Optimisation Experiments

Optimisation Techniques Experiments

This section aims to analyse and discuss the results of experimenting with optimisation techniques: Hybrid feature extraction N-Grams and TF-IDF, Information Gain, and Grid Search Hyperparameter Tuning. The results of all experiments are summarised in Table 14.

Experiment	Optimisation Method	Feature Extraction	Sentiment Classifier	Accuracy	AUC	Accuracy Improvements	AUC Improvements
3a	**Hybrid N-Grams and TF-IDF**		**SVM**	**0.79**	**0.78705**	**+1%**	**+0.01%**
			NB	**0.78**	**0.77636**	**+1.5%**	**+0.01%**
3b	Information Gain (Built Upon N-Grams)	N-Grams	SVM	0.75	0.75321	-3%	-0.02%
		TF-IDF		0.77	0.76811	-1%	-0.01%
		N-Grams	NB	0.74	0.74172	-3%	-0.03%
		TF-IDF		0.74	0.74405	-2%	-0.01%
3c	Information Gain (Built Upon TF-IDF)	N-Grams	SVM	0.74	0.74089	-4%	-0.04%
		TF-IDF		0.76	0.73832	-2%	-0.03%
		N-Grams	NB	0.74	0.76196	-2%	-0.01%
		TF-IDF		0.74	0.74020	-2%	-0.02%
3d	**Hyperparameter Tuning**	N-Grams	**SVM**	**0.78**	**0.77907**	**-**	**+0.01%**
		TF-IDF		**0.79**	**0.78534**	**+1%**	**+0.01%**

Table 4. Optimisation Techniques Experiments

The experiments in this section has the following objectives:

- **To implement a hybridised N-Grams and TF-IDF feature extraction technique and deduce potential improvements in performance of sentiment classification**

Experiment 3a

A hybridised N-Grams and TF-IDF involves the computation of TF-IDF score for each n-gram phrase. The N-Gram range is defined according to optimum N-value observed in Section 5.2 such that unigrams produce optimum results with SVM classifier and uni-bi-trigrams product optimum results with NB classifiers.

Code Snippet 19 is used to perform TF-IDF feature extraction upon unigrams to train the SVM sentiment classifier while Code Snippet 20 is used to perform TF-IDF feature extraction upon unigrams, bigrams and trigrams to train the NB sentiment classifier.

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(ngram_range=(1,1), min_df=1).fit(X_train)
X_train_tfidf_vect_E = tfidf_vect.transform(X_train)
```

Code Snippet 1. TF-IDF feature extraction on unigrams

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf_vect = TfidfVectorizer(ngram_range=(1,3), min_df=1).fit(X_train)
X_train_tfidf_vect_F = tfidf_vect.transform(X_train)
```

Code Snippet 2. TF-IDF feature extraction on uni-bi-trigrams

The results of Experiment 3a shows that the SVM and NB sentiment classifier performed with an accuracy of 79% and 78%. Through comparing to the baseline models, it is evident that performance of sentiment classification has improved the accuracy of SVM model on average by 1% and the NB model on average by 1.5%*.

As the complexity of model and feature space has increased through this hybrid feature extraction technique, the model performance increases as the dataset used seems to have better performance for models of increasing complexity.

Results: Hybrid unigrams and TF-IDF increased accuracy of SVM sentiment classifier by 1%. Hybrid unigrams, bigrams, and trigrams and TF-IDF increased accuracy of NB sentiment classifier by 1.5%*. It is concluded that hybrid N-Grams and TF-IDF technique is successful in optimising the sentiment classifiers in this project.

** Calculation for performance improvement of Naïve Bayes:* $\frac{accuracy\ of:(model\ E - model\ D) + (model\ E - model\ C)}{2}$

- **To identify the optimum approach of implementing Information Gain (Approach 1: Compute IG threshold through texts vectorised by N-Grams, Approach 2: Compute IG threshold through texts vectorised by TF-IDF)**

Experiment 3b, 3c

The two approaches of performing Information Gain explained in Section 4.6 are implemented to ignore terms that are not of top 20 importance. The information gain threshold obtained through N-Grams vectorisation is 2.72542e-05, and through TF-IDF vectorisation is 1.62072e-04. Following this, vectorisation is performed while ignoring terms with Information Gain below the defined thresholds. The vectorised text data are used to train SVM and NB models.

The results of these experiments are observed to consistently lower the performance of SVM and NB sentiment classifiers. In Approach 1, Information Gain built upon N-Gram vectorisation has decreased accuracy of the SVM model by an average of 2%, and NB model by an average of 2.5%. In Approach 2, Information Gain built upon TF-IDF vectorisation has decreased accuracy of the SVM and NB model by an average of 2%.

Through feature selection with Information Gain, the dimension of dataset is reduced and complexity of model is decreased. As performance is lowered, it implies that retaining all features are important, regardless of term relevancy.

Results: Approach 1 for performing Information Gain resulted in a decrease in accuracy of SVM sentiment classification by 2% and NB sentiment classification by 2.5%. Approach 2 for performing Information Gain resulted in a decrease in accuracy of SVM and NB sentiment classification by 2%. It is concluded that both approaches of Information Gain is not successful as an optimisation technique in this study.

- **To implement Grid Search Technique and deduce potential improvements in performance of sentiment classification**

Experiment 3d

Hyperparameter Tuning is commonly used to optimise the parameters of SVM models. Grid Search technique is employed where each hyperparameter value is searched across a defined grid of search space. This method is exhaustive and takes a longer processing time. The parameters for the baseline SVM model and parameters identified through the grid search hyperparameter tuning technique is shown in Table 15.

Through training an SVM model with the updated hyperparameter values, AUC has improved for SVM model trained with unigrams vectorised texts by 0.01% while accuracy for SVM model trained with TF-IDF vectorised texts increased by 1%.

Parameters	**Before hyperparameter tuning**	**After hyperparameter tuning**	
		N-Grams	**TF-IDF**
c	1	10	100
gamma	0.08	0.01	1

Table 5. SVM hyperparameters before and after hyperparameter tuning

Results: Hyperparameter Tuning increased AUC for SVM with unigrams by 0.01% and accuracy for SVM with TF-IDF by 1%

Hybrid Optimisation Techniques Experiment

From preliminary results in Section 5.3.1 (Experiment 3b and 3c), Information Gain does not improve the model. Hence, it is not included in the final proposed optimisation model. This section aims to expand upon optimisation techniques through hybridisation. As the hybridised N-Grams and TF-IDF feature extraction technique and Grid Search Hyperparameter Tuning techniques yielded in successful optimisation, these two techniques are further combined to experiment if model performance can increase.

This section aims to display and analyse the effects of combining N-Grams and TF-IDF with Hyperparameter Tuning upon SVM sentiment classifier's performance. The results of the experiment are shown in Table 16.

From Table 16, it is observed that optimisation is successful as accuracy has increased by 1% while AUC has increased by 0.01%. This optimisation technique has achieved highest performance as compared to other models built within this project. It is observed that a hybrid of successful optimisation techniques has carried a good impact upon performance of sentiment classification.

Experiment	Optimisation Method	Sentiment Classifier	Accuracy	AUC	Accuracy Improvements	AUC Improvements
4a	**N-Grams + TFIDF + Hyperparameter Tuning**	**SVM**	**0.79**	**0.78801**	**+1%**	**+0.01%**

Table 6. Hybrid Optimisation Techniques Experiment