Evaluating classical ML-NLP algorithms on Parts of Speech and Text Classification Tasks

Undergraduate Student Name: Harshil Shah
Ph.D. Mentors: Daniel Nkemelu
Research Advisor: Dr. Michael Best

Github Link: https://github.com/TID-Lab/Burmese-Content-Analysis

ABSTRACT

There have been numerous recent developments in the domain of creating language models that can generate impressive performance metrics on various Natural Language Processing (NLP) tasks like Question-Answering, Named Entity Recognition, Sentiment Analysis, Parts of Speech (PoS) tagging etc. These language models, pre-trained on extremely huge datasets, learn to effectively model language and its nuances which are then used for various downstream NLP tasks.

However, the impressive performances of these models are highly dependent on the availability of digital language data and the intrinsic features of the target language. For many low resource languages, the abilities of these models to yield better performance metrics than the existing classical Machine Learning(ML) algorithms is untested.

In this report, we evaluate the performance of the classical ML algorithms on a low resource language, Burmese. We use Parts of Speech tagging and Text classification to evaluate the performance of the classifiers. This report is divided into two main parts: Supervised learning using Artificial Neural Network (ANN) on the Burmese Treebank Parts of Speech data and Supervised learning using Multinomial Naive Bayes(MNB) classifier and Support Vector Machine on the Burmese BBC news data.

GOAL

We aim to create a Burmese Bert model that would perform classification tasks on unseen data with high accuracy. To understand the accuracy and the advantages in using the Burmese BERT model, we need to compare it against the existing best models in the supervised learning domain. This is the strategy used by many researchers to evaluate the performance of the BERT models developed for other languages like French, Arabic, Dutch etc. This compiled document lists the data sources, experiment design, and evaluation tasks from research papers published for BERT models in other languages: BERT reference.

This Burmese BERT model would later be used in our Social-Media tracking system Aggie to identify instances of Hate Speech during the Myanmar elections in November 2020.

SUPERVISED LEARNING ON THE BURMESE TREEBANK DATA

PROBLEM DEFINITION

Given a set of training data containing burmese sentences and Parts of Speech (PoS) annotations for each word in the sentence, design an efficient supervised learning model that predicts the PoS annotation of a word in the test set based on learned patterns such as placement of the word in the sentence, past usages, neighbors etc.

DATA

The data for this experiment is obtained from Burmese Treebank Data. This dataset contains 20,106 sentences with syntactic tree annotations. For this experiment, only the Height 1 annotations from the syntactic tree were utilized as these annotations represented the PoS of a word in the tokenized sentence. More information on Height 1 syntactic annotations and the dataset can be found here.
The initial distribution of annotations in the original data is shown in **Figure 1**.

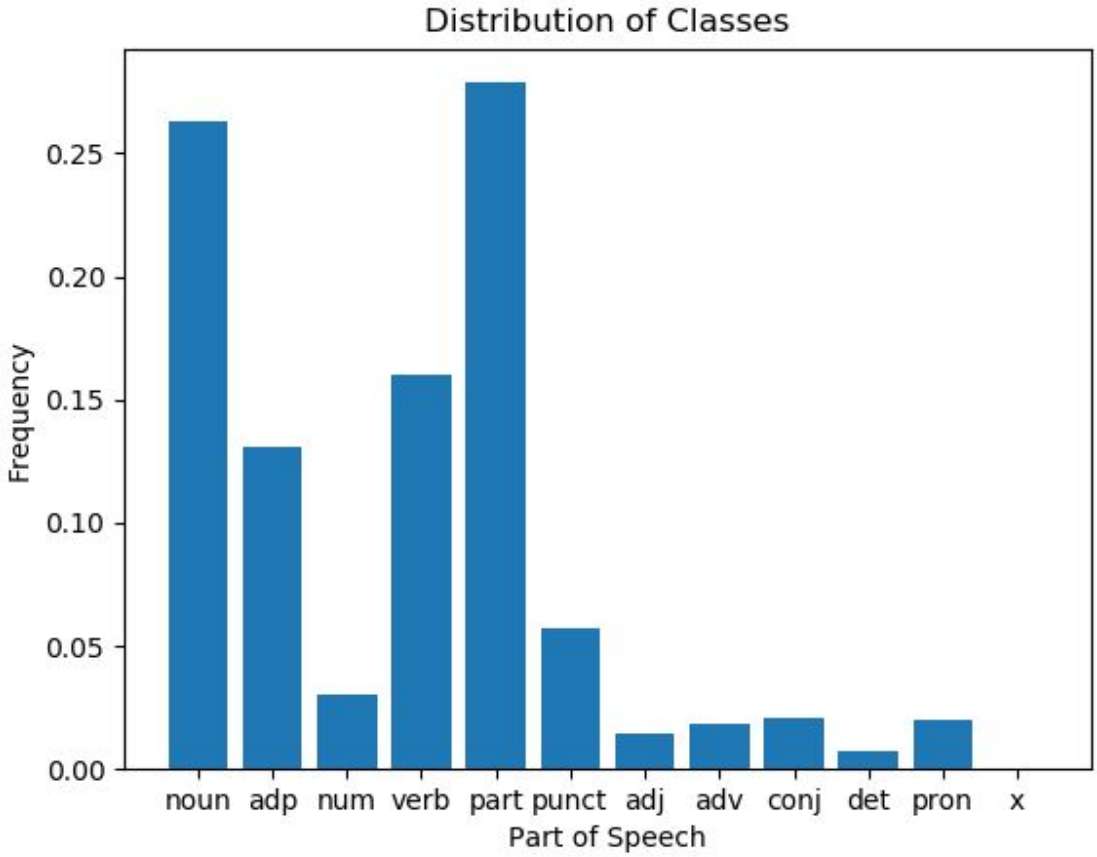


Figure 1. Initial class distribution

As seen in the data, nouns, adpositions (adp), verb, and particles(part) cover about 80% of the tokens in the corpus**[1]**. The presence of the adjective(adj) tag is very sparse in the Height 1 annotations. We contacted our Burmese Language experts to provide some insight on why the presence of the adjectives is so less considering how amply they are used in other languages and the reason is as follows: "In Burmese, the adjective annotations make most sense when used to tag a combination of tokens(expressions) rather than tagging individual tokens". So at Height 1. where all annotations are at token-level, we do not see too many adjective annotations.

PREPROCESSING

The preprocessing step involved creating a simpler JSON file containing each token and its annotation by removing the syntactic tree annotations present in the original data. This step ensured that only Height 1 token-level annotations are being used for the learning process. This JSON file can be found here..

EXPERIMENT DESIGN

We used a supervised learning approach for this task of multi-class classification. We decided to use an Artificial Neural Network(ANN) for this PoS tag prediction task. The main reason for using Artificial Neural Network for this task of PoS tagging of Burmese tokens is because of the demonstrated success of RNNs(Recurrent Neural Network) and CNNs(Convolutional Neural Networks) in PoS tagging tasks in other languages like English, German etc **[2]**.

The high level design of the experiment includes separating the original dataset into training, validation and test data samples, adding additional features to each token such as the word position, neighbors, etc., vectorizing the training data with the added features using Dictionary Vectorizer[3], and supplying that vectorized transformed dataset to the artificial neural network. The ANN uses the training data samples to tune the neural network's weights and biases, validation data to get an estimate of model's performance while training, and the test dataset to get an unbiased estimate of the model's final performance.

The distribution of the training, testing and validation set is shown in Table 1.

Dataset	**Size**
Training	16084
Test	4022
Validation	4021

Table 1. The distribution of the training, testing and validation set

FEATURE ENGINEERING

For each token in our original dataset, we add additional features in a dictionary format to improve the performance of the ANN. These features include the position of the token in the sentence, left and right neighbors, prefixes and suffixes of the token, and the length of the sentence. The choice of these additional features is dependent on the structure of the language and can be arbitrary to some extent. Positional information of the token is important because most languages including Burmese contain a pattern where certain PoS tags occur at certain places in the sentence. Similarly, learning about the left and right neighbors also helps to generate a pattern to accurately predict the PoS tag of a token. Variations in performance in the presence and absence of these features will be discussed in the Results section. Features that were added to each token in the code are as follows:.

```
{
 'nb_terms': len(sentence_terms),
 'token': token,
 'index': index,
 'is_first': index == 0,
 'is_last': index == len(sentence_terms) - 1,
 'prefix-1': token[0],
 'prefix-2': token[:2],
 'suffix-1': token[-1],
 'suffix-2': token[-2:],
 'prev_token': '' if index == 0 else sentence_terms[index - 1],
 'next_token': '' if index == len(sentence_terms) - 1 else
sentence_terms[index + 1]
}
```

After feature engineering, each token with the additional features represented in the dictionary format as shown above was vectorized using the DictVectorizer function **[3]**. The vectorized data was then used for the training of the ANN.

NEURAL NETWORK ARCHITECTURE

As mentioned earlier, we use an Artificial Neural Network(ANN) for the task of PoS tagging. This ANN consists of one input layer, two hidden layers, and one output layer. Each hidden layer is applied to the ReLU activation function (Rectified Linear Unit). ReLU activation function yields “equal or better performance than hyperbolic tangent networks”**[4]**. Dropout layers are added after each hidden layer for regularization and to avoid overfitting. The dropout rate is set to 0.2. The softmax activation function

is used on the output layer which is a standard choice for calculating a probability distribution over multiple classes in multi-class models**[5]**. The neural network architecture can be visualized in **Figure 2**.

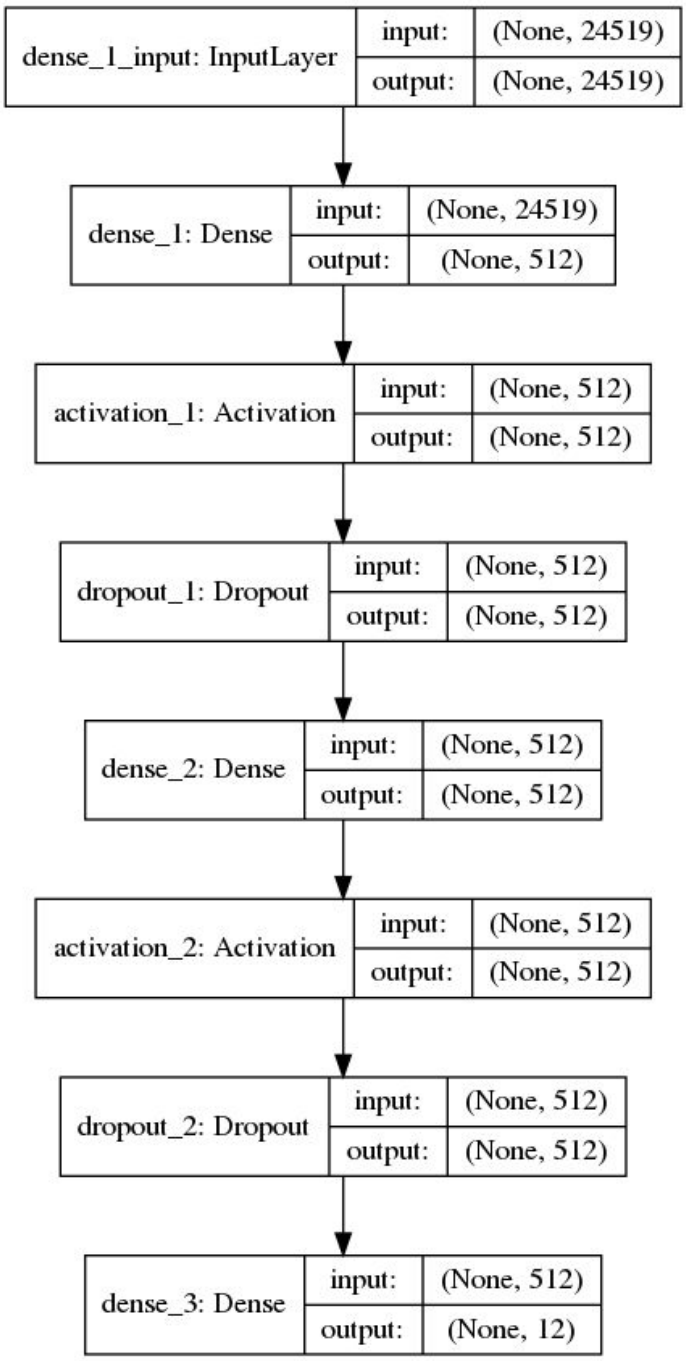


Figure 2. ANN architecture..

The size of each hidden layer is 512 neurons. The number of input samples supplied to the model is variable hence it's represented as "?" in **Figure 2**. The output layer produces a vector of size twelve

which is the probability distribution over the twelve PoS tags. The entry with the highest probability is the predicted PoS tag for the token.

RESULTS

The ANN produced an overall average accuracy of 0.97, average precision of 0.96, and average recall of 0.94 across all 12 classes on the test dataset. The class-wise scores for precision and recall are as follows.

	precision	recall	f1-score	support
adj	0.97	0.88	0.92	2070
adp	0.96	0.98	0.97	18332
adv	0.97	0.94	0.96	2467
conj	0.91	0.75	0.82	2932
det	0.90	0.93	0.91	987
noun	0.99	0.98	0.99	37207
num	1.00	0.99	0.99	4426
part	0.97	0.97	0.97	38559
pron	0.96	0.98	0.97	2812
punct	1.00	1.00	1.00	8009
verb	0.96	0.99	0.98	21952
accuracy			0.97	139753
macro avg	0.96	0.94	0.95	139753
weighted avg	0.97	0.97	0.97	139753

As seen in the figure, the ANN model is very successful in identifying the punctuations in the text which is plausible since the punctuations have distinct representations and can only be labeled as punctuations. Same is true for the numbers classified by the model. The model is also very successful in classifying the nouns in the test dataset. We can also see that even though certain classes dominated the original dataset(nouns, adpositions, verb, and particles), the classifier does not suffer from the class imbalance problem. This can be explained by the fact that we have enough samples even for the infrequent class that the ANN is able to learn its intrinsic qualities and thus, successfully tag the tokens belonging to that particular class in the test dataset.

Figure 3 shows the performance of the model on the validation set and the training set during the learning phase. The validation set is withheld during the training period and is only used at the end of every epoch to get an estimate of model's performance during the learning phase. As we can see, the accuracy on the validation set remains almost constant throughout the learning phase while the accuracy on the training set increases at each epoch. However, this is not enough to conclude that the

ANN model is overfitting the training data because if that were the case, the accuracy on the validation set would decrease and more importantly, the accuracy on the test dataset would not be so close to the accuracy of the training and validation set. Since the accuracies of training, validation, and test set align with each other, we can safely conclude that the ANN model does not overfit but learns the intrinsic patterns associated with each class and uses those to make predictions on the test set.

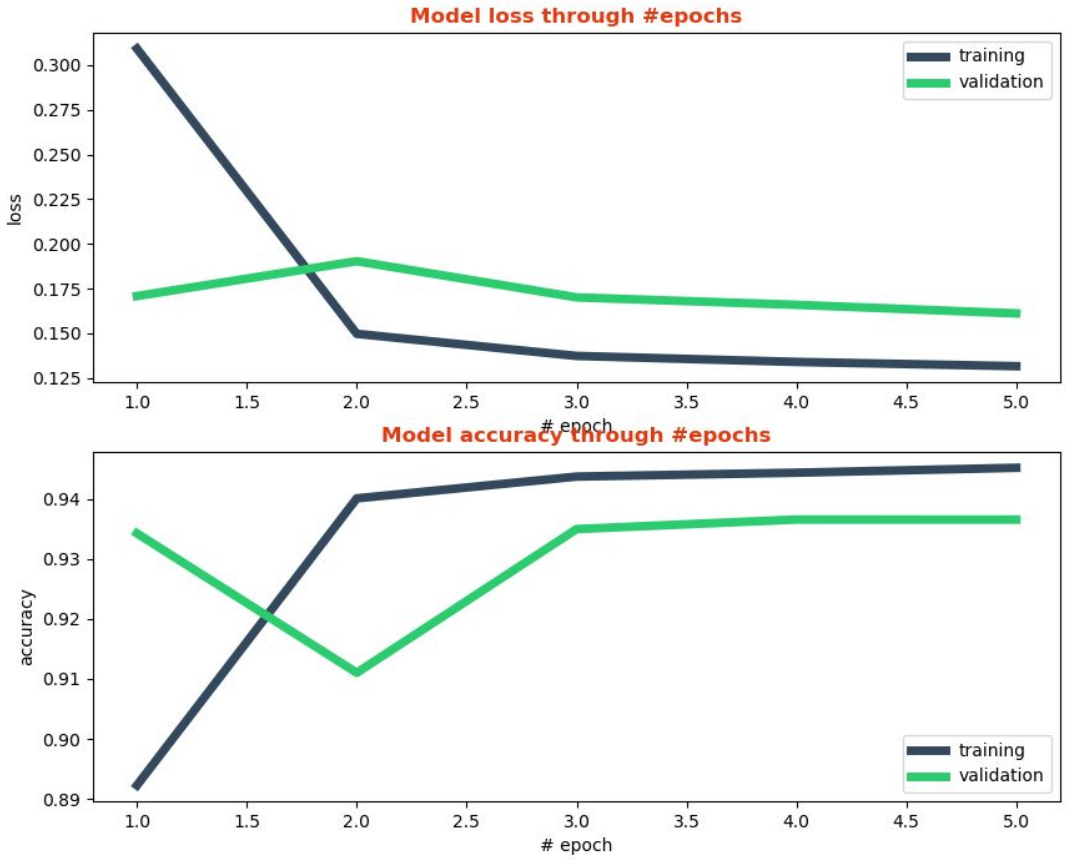


Figure 3. The trend in accuracy and loss over training and validation set over five epochs during the learning phase

OBSERVATIONS

We also trained the ANN over a different set of features associated with each token. Since each token is the smallest unit in Burmese language, using prefixes and suffixes of the token to improve the performance did not make much sense. Training the model without prefixes and suffixes did not change the accuracy/precision/recall of the model significantly as expected.

However, adding information about the left and the right neighbors of a given token in the feature set significantly increased the performance of the model. This tells us that the neighboring tokens provide vital context to our model to determine the PoS of the given token at least in the Burmese language.

Another interesting observation is how the validation set accuracy and loss stay almost constant during the five epochs. This tells us that even though our model is learning new patterns in the training set (as evidenced by the increase in the training set accuracy), there are certain nuances about the validation and test set that are not captured by the training set. These nuances could be that a certain word is used in multiple different contexts and is tagged with a different PoS tag every time. From the analysis of the original dataset, Table 2 shows some cases where a same word is tagged with multiple different PoS tags at different places in the original data. This can lead to the constant trend in validation set accuracy and loss especially if all the different usages of the same word are not captured enough times in the training set. The Table 2 shows instances of tokens assigned either 4 or 5 different PoS tags. Further analysis shows that there are 103 tokens with 3 different PoS tags in the original data and 943 tokens with 2 different PoS tags.

Tokens	**PoS Tags**
သည်	adp, det, noun, part, pron
လေး	adj, noun, num, part, verb
မ	adj, adv, noun, part
ပြီး, ရာ, ရင်	conj, noun, part, verb
တိုင်း	adp, noun, part, verb
အောင်	adj, conj, part, verb
တချို့	adj, det, noun, pron
လိုက်	adv, noun, part, verb

Table 2. This table shows the tokens and the assigned tags in the original data.

DISCUSSION

Overall, the ANN model performs exceptionally well on the Burmese language Parts of Speech (PoS) tagging task. When comparing this performance against the novel Burmese BERT model, the main area to pay attention to is how well the Burmese BERT model is able to correctly identify PoS tags for tokens that are assigned three or more PoS tags in the original data depending on the context and its usage.

CONCLUSION

We used supervised learning to create an Artificial Neural Network(ANN) to perform the task of Parts of Speech(PoS) tagging on the Burmese language. The model performed exceptionally well yielding an average accuracy of 97% with high precision and recall. We made interesting observations on why the accuracy and loss of the validation set stays constant after each epoch and provided a possible explanation for such a trend. Future work in this area involves comparing the novel Burmese BERT model against the Neural Network and evaluating the differences in performance metrics of both the models.

SUPERVISED LEARNING ON THE BURMESE BBC DATA

PROBLEM DEFINITION

Given a set of training data containing burmese BBC articles and their categories, design an efficient supervised learning classifier that predicts the category of the article based on learned patterns in the test dataset.

We also trained several classifiers using different vectorization methods and varying training sizes: `{200, 1000, 2000, 10000, All samples}` with almost equal samples per class (proportionate to sample count per class in the original data). The reason for creating classifiers for different training sizes is to identify the crossover point where the Burmese BERT model starts to outperform the traditional ML classifiers.

DATA

We used the Corpus Crawler tool to collect Burmese news articles from the Myanmar BBC website **[7]**. We collected a total of 24,787 articles dating back to 2009. However, the Corpus Crawler tool did not extract information about the category of the article. Hence, we edited the Corpus Crawler's script to also extract the article's category along with the text. We will be releasing the BBC burmese article text and category data for public use. The news articles were categorized into one of the following categories:


```
{
  "unknown":887,
  "articles":1,
  "burma":13696,
  "economy":110,
```

```
  "ethnic":38,
  "help":5,
  "in":353,
  "in_depth":1,
  "indepth":11,
  "institutional":120,
  "interactivity":183,
  "live":15,
  "media":113,
  "mobi_geno":3,
  "multimedia":810,
  "news":3431,
  "programmes":430,
  "services":3,
  "specials":19,
  "sport":488,
  "world":10493
}
```


As seen above, we can clearly notice that some classes like world and burma contain significantly more number of articles than other classes like media and programmes. Additionally, some web-links parsed by the corpus crawler do not even represent news articles like the ones contained in the "help" and "in" category. Certain news articles did not contain any category information, hence, those are included in the "unknown" category. Considering these issues, we narrowed down the number of classes to four containing burma, world, sports, and economy. The reasoning to choose these four categories for the classifier was that these categories are clearly distinct from one another and there is an informational value associated with classifying articles in these categories than the other categories present in the dataset.

However, initial testing of the Support Vector Machine(SVM) and Naive Bayes(NB) classifier with optimal hyperparameters on the dataset containing these four categories revealed how severely the class imbalance issue was affecting the results. The results are as follows:

```
Num articles: 24787

classifier: Naive Bayes Classifier

              precision    recall  f1-score   support
      burma       0.83      0.92      0.88      2760
    economy       1.00      0.10      0.18        20
      sport       0.95      0.36      0.52       107
      world       0.85      0.77      0.81      2071

   accuracy                           0.84      4958
  macro avg       0.91      0.54      0.60      4958
weighted avg      0.84      0.84      0.84      4958

classifier: SVM Classifier

              precision    recall  f1-score   support
      burma       0.93      0.92      0.92      2760
    economy       0.88      0.35      0.50        20
      sport       0.86      0.63      0.72       107
      world       0.88      0.91      0.89      2071

   accuracy                           0.91      4958
  macro avg       0.89      0.70      0.76      4958
weighted avg      0.91      0.91      0.90      4958
```

As shown above, recall scores for economy and sports are extremely low for both classifiers. These results were not very useful for us since our goal is to create a well-performing classifier that can effectively be compared against the Burmese BERT model that we plan to use. Since, having the economy and sports categories is not vital to our end goal and since the classifier's performance was not improving beyond this threshold, we decided to drop those two categories entirely to solve the class imbalance issue.

Hence, the dataset that we used contains articles only belonging to the burma and the world categories. In the following sections, the term dataset refers to the collection of 24,189 "burma" and "world" articles.

PREPROCESSING

Preprocessing step included removing stop words, non-burmese text, and HTML tags from the parsed data **[9]**. We performed tokenization of the Burmese text by segmenting it using Myanmar Language Tool **[8]**.

FEATURE EXTRACTION

We used different strategies for vectorizing the input data before supplying that information to the classifiers. We used Term Frequency- Inverse Document Frequency (Tf Idf) approach, unigrams, bigrams, trigrams, and combinations of those to achieve the best performance in our classifiers. Detailed analysis of which approach yielded the best results will be discussed in the Results section.

Previous research has shown that N-gram language models perform well for Burmese segmentation tasks **[10]**. Following those results, we decided to explore how well N-grams vectorization approach along with an additional Tf Idf approach performs in the Burmese text classification task.

MULTINOMIAL NAIVE BAYES

Naive Bayes ML models are widely used for text categorization problems because of the high computational efficiency achieved on high-dimensional problems **[12]**. We chose Multinomial NB (MNB) in particular, because of its proven high performance in NLP problems involving feature counts/fractional counts **[13]**.

HYPERPARAMETERS

We used the Cross Validation Grid Search to find the optimal hyperparameters **[11]**. The hyperparameters to fine-tune included the alpha value, which is the Laplace smoothing parameter. The range for the alpha value was selected using the standard practice to select the optimal range based on the U-shaped generalization curve **[14]**.

Example: For the following range of alpha {0.001, 0.01, 0.1, 0.2, 0.4, 0.5, 0.7, 0.9, 1} and Tf Idf vectorization method on training size of 1000 samples, the generalization curve of the validation set is shown in **Figure 4.**

Figure 4. Graph showing generalization error for different alpha values

The graph follows an almost "U shaped curve" showing that the optimal value for alpha lies in the selected range at the minima on the graph. For the MNB classifier trained on 1000 samples with Tf Idf vectorization method, the optimal alpha value occurs when the generalization error is at minimum i.e. the minima on the graph where alpha = 0.2. The ranges for each classifier were adjusted similarly until the fairly U-shaped generalization error curve was obtained.

The range of the alpha values tested were as follows: {0.001, 0.01, 0.1, 0.2, 0.4, 0.5, 0.7, 0.9, 1}. The best alpha value differed for each MNB classifier trained of varying number of samples and different vectorization approach. The optimal values in most cases were 0.2 and 0.4.

RESULTS

The MNB classifier yielded the best results overall when all training samples were used. The vectorization scheme did not play a huge role in improving the performance of the model trained on larger training sets. For medium sized training sets (4000-12000 samples in this case), bigrams and a combination of bigrams and trigrams yielded the best results. The trends can be observed in the following figures.

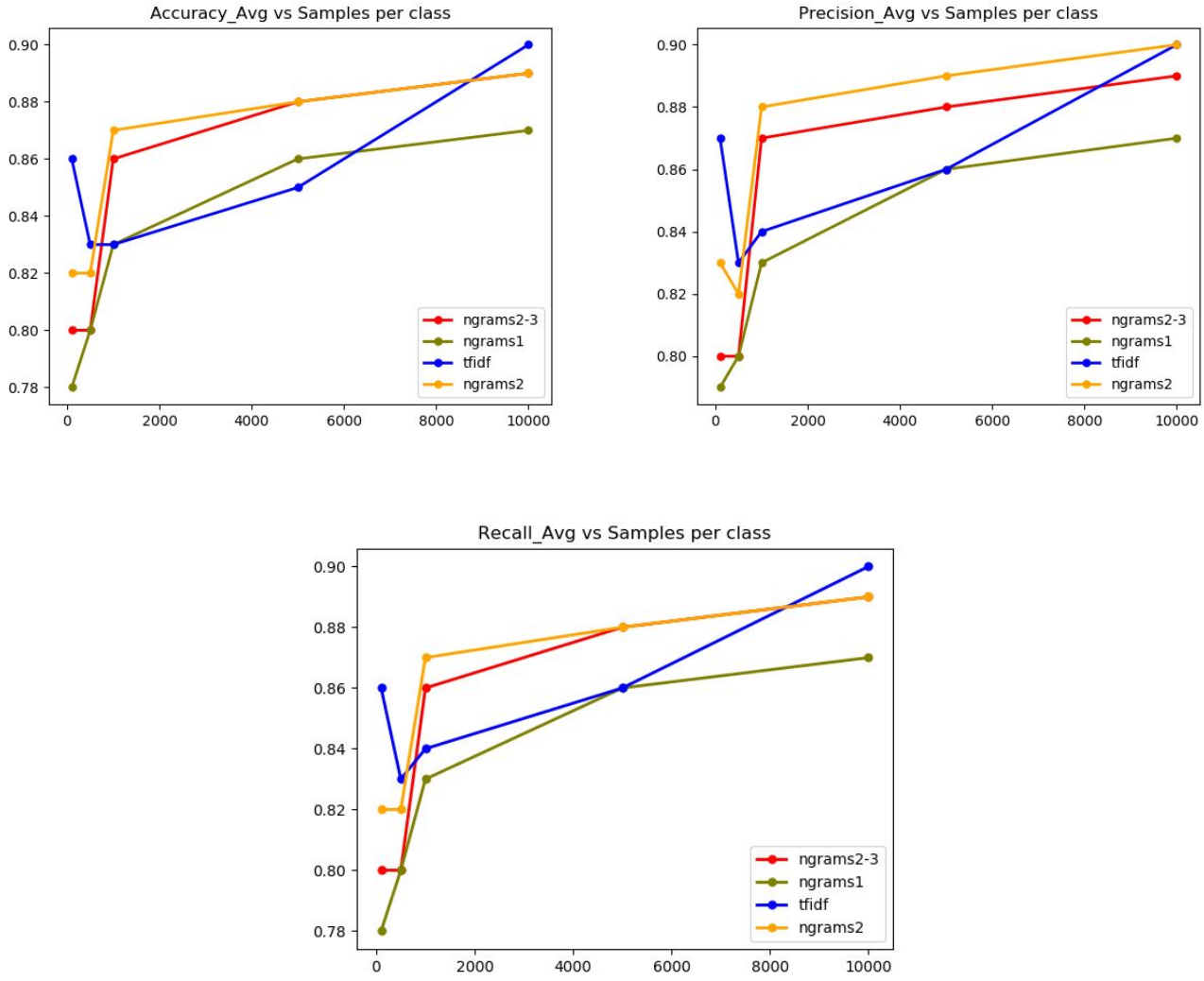


Figure 5. Graphs showing the trend in average Accuracy, Precision, and Recall for different vectorization methods with increasing size of the training set. The X-axis shows the approximate number of training samples per class.

OBSERVATIONS

We see that, generally, N-grams approach outperforms Tf Idf in MNB classifier for most sets of training sample sizes; however, Tf Idf outperforms N-grams either when all samples are used or the fewest

samples are used i.e the left and right ends of the trends shown in **Figure 5**. Tf Idf assigns higher scores to terms that are used more rarely in the dataset, and often the ones which are more deterministic of the category the sample belongs to **[15]**. We observed that terms like တကသ, ခဏအက, ရန်ကုန်မြို့, ရစ, ကယ, မသ, တယ, အသက, ဆယ, တက, and ကန were assigned much higher scores compared to other terms. Many of these terms, when used in the right context, help the classifier in determining the category of the given article. For example, the word ရန်ကုန်မြို့(Yangon City), appears in 7 documents in total out of 200 randomly chosen training samples. When the model was tested on 47 unseen articles containing the word ရန်ကုန်မြို့, it yielded an accuracy score of 80.85%. This shows how few deterministic terms that appear consistently in the training set can improve the accuracy of the classifier even on smaller datasets when using Tf Idf approach.

For medium sample sizes, the trends show that N-grams approach yields much better outcomes. This indicates that with increasing sample sizes, quantitative information on word combinations helps the classifier make better decisions than just using the Tf Idf scores. This means that with increasing training size, the presence of deterministic terms across the training set increases, therefore reducing/equalizing the overall Tf Idf scores across several deterministic terms. At this point, the quantitative information on word combinations derived from N-grams helps the classifier in better categorizing the BBC articles.

Also, the average accuracy, precision, and recall are fairly in sync with another indicating that the classifier is equally good at categorizing articles belonging to each class i.e. Burma and World in this case.

SUPPORT VECTOR MACHINE

Support Vector Machines(SVM) usually outperform NB classifiers in text classification/categorization tasks; however, SVMs take extremely long time to find the optimal boundary for categorization purposes on high-dimensional problems such as this one. Training each SVM classifier using the entire training data for this problem took on average 3 hours on the Georgia Tech PACE cluster with 12 nodes and 4 cores per node. SVMs did out perform MNB classifiers on larger training sets.

HYPERPARAMETERS

We used the Cross Validation Grid Search to find the optimal hyperparameters**[11]**. The hyperparameters to fine-tune included the C value (regularization parameter), gamma (kernel coefficient), and kernel type. The ranges for each hyperparameter are as follows:

```
C       : {0.01, 0.1, 1, 10, 100}
Gamma   : {0.01, 0.1, 1, 10}
Kernel  : {linear, rbf}
#The rbf kernel is the term used by scikit learn for Gaussian kernels.
```

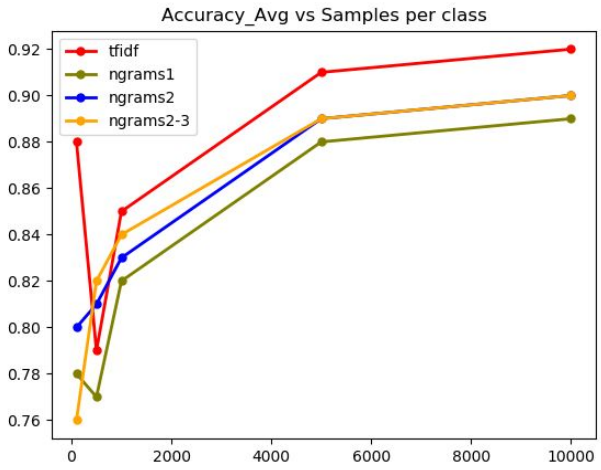
All numerical ranges are selected using the "U-shaped" generalization error curve method. The polynomial kernel is left out because of the poor performance compared to the other two kernels revealed in the initial testing.

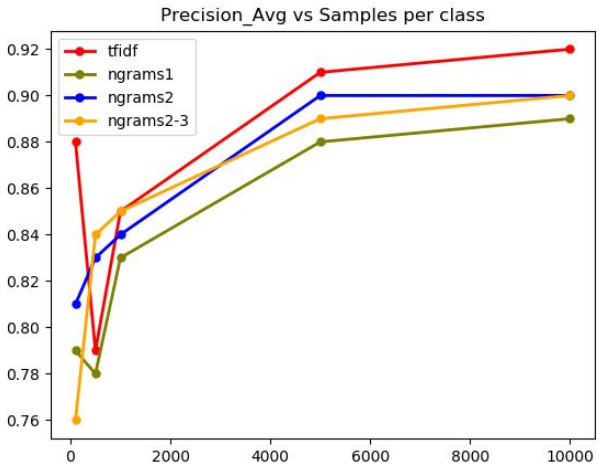
RESULTS

The common hyperparameters that appeared in most of the SVM experiments with the combination of different training sizes and vectorization methods are as follows:

```
C       : 10
Gamma   : 1
Kernel  : rbf
```

As seen in **Figure 6**, Tf Idf seems to consistently outperform the N-grams approach for almost all training sizes with the maximum average accuracy score being 92%.





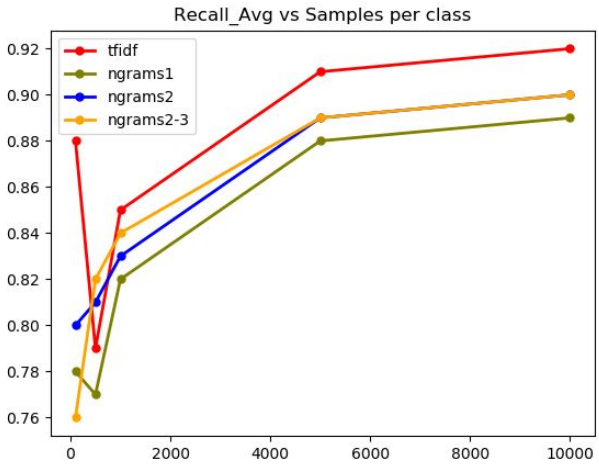


Figure 6. Graphs showing the trend in average Accuracy, Precision, and Recall for different vectorization methods with increasing size of the training set. The X-axis shows the approximate number of training samples per class.

OBSERVATIONS

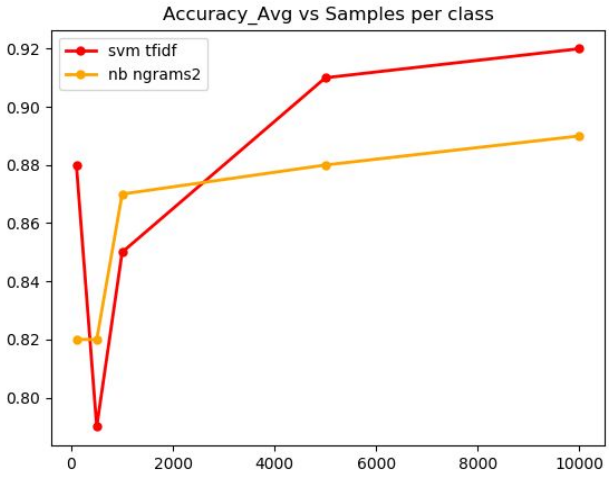
As seen in the MNB case where Tf Idf yields exceptionally high accuracy, precision and average for really small training sets (200 training samples), the same can be evidenced in the SVM classifiers with Tf Idf yielding values higher than 85% for all three metrics on average. This means that for smaller training sets, Tf Idf along with the optimal hyperparameter values helps in creating fairly separable data in a multi-dimensional space resulting in higher performance than that achieved by SVM on N-grams with optimal hyperparameters on smaller training sets.

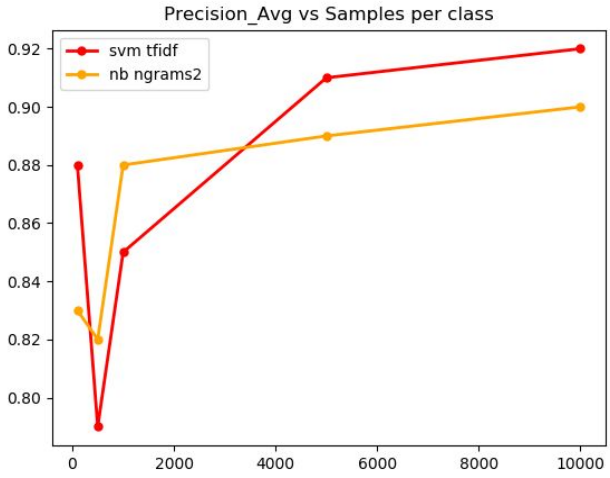
For training sets with samples more than 1000, Tf Idf with optimal hyperparameters still seems to outperform N-grams, however the difference is not very significant in this case. The average accuracies, precisions, and recalls are also in sync with each other for each training set size and vectorization approach. This indicates that the SVM classifier is equally good at predicting both Burma and World categories.

COMPARING SVM AND MNB

Based on the individual performances of the classifiers as shown in previous sections, we decided to compare the MNB N-grams vectorization with optimal hyperparameters against SVM Tf Idf vectorization with optimal hyperparameters. For smaller training sets, SVM Tf Idf with optimal hyperparameters performs better than its counterpart. This is largely because of the finding mentioned earlier about the exceptionally high performance of Tf Idf on very small training sets as evidenced in this classification problem. Overall, SVM outperforms MNB with a huge margin. This can be attributed to the fact that

MNB makes a huge and naive assumption about the independence of variables in calculating the probability of a sample belonging to a particular class. This provides significant time and computational advantage to MNB over other classifiers at the cost of accuracy to some extent. SVM is devoid of such huge assumptions because of which it is computationally extremely expensive. Doing a grid search for optimal hyperparameters for SVM with all training samples took approximately 3 hours on Georgia Tech PACE cluster with 12 nodes and 4 cores per node. However, SVMs yielded consistent and higher performance scores than the MNB classifier.





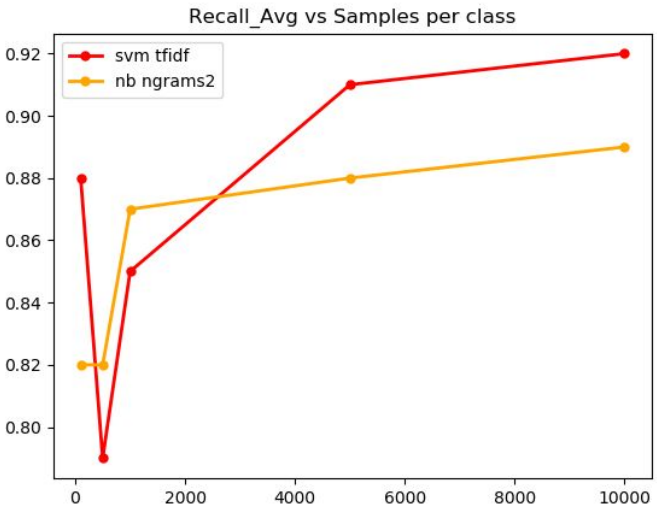


Figure 7. Graphs comparing the trend in average Accuracy, Precision, and Recall for different vectorization methods with increasing size of the training set between SVM and MNB classifiers. The X-axis shows the approximate number of training samples per class.

FUTURE WORK

The main goal of this project is to analyze the performances of traditional supervised learning models on the Burmese Parts of Speech data and the compiled Burmese BBC news articles data. This analysis will be used as a reference point to compare against the performance of the modern Burmese BERT model developed by our lab. The research goal for this comparison between traditional ML algorithms with novel BERT models is to identify the advantages of using Burmese BERT models for various NLP classification tasks like hate-speech detection. We anticipate that there is a crossover point when the novel Burmese BERT model starts to outperform classical ML algorithms on various NLP problems with consistently increasing sizes of the training sets.

CONCLUSION

In this report, we dived deep into the design decisions and practices used to create each classifier for both problems of PoS tagging and BBC news articles classification. We identified the importance of context and the inability of ANNs to learn context in the PoS tagging task. We presented the variations in the performance of SVMs and MNBs on varying training set sizes and vectorization methods. We discovered and justified the exceptionally high performance on unseen data gained while using the Tf Idf approach on both SVM and MNB classifiers trained on very small training sets. Lastly, we also compared SVM with MNB to gain an idea of the advantages and the disadvantages of using one over the other.

REFERENCES:

1. Ding, Chenchen, et al. “A Burmese (Myanmar) Treebank.” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, no. 3, 2020, pp. 1–13., doi:10.1145/3373268.

2. Neunerdt, Melanie, et al. “Part-Of-Speech Tagging for Social Media Texts.” *Language Processing and Knowledge in the Web Lecture Notes in Computer Science*, 2013, pp. 139–150., doi:10.1007/978-3-642-40722-2_15.

3. https://scikit-learn.org/stable/modules/generated/sklearn.featureextraction.DictVectorizer.html

4. Xavier Glorot, Antoine Bordes, Yoshua Bengio ;Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR 15:315-323, 2011.

5. Nwankpa, Chigozie, et al. "Activation functions: Comparison of trends in practice and research for deep learning." *arXiv preprint arXiv:1811.03378* (2018).

6. https://becominghuman.ai/part-of-speech-tagging-tutorial-with-the-keras-deep-learning-library-d7f93fa05537

7. https://github.com/google/corpuscrawler

8. https://github.com/MyanmarOnlineAdvertising/myanmar_language_tools

9. https://github.com/HtooSayWah/Cyberbullying-System

10. Ding, Chenchen, et al. “Word Segmentation for Burmese (Myanmar).” *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 15, no. 4, 2016, pp. 1–10., doi:10.1145/2846095. 11)

11. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

12. https://link.springer.com/chapter/10.1007/978-3-540-30549-1_43

13. https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

14. Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. Deep learning. MIT press, 2016.

15. Aizawa, Akiko. "An information-theoretic perspective of tf–idf measures." Information Processing & Management 39.1 (2003): 45-65.