News Topic Classification Using Natural Language Processing

Zhang, Mengyu | zhang.mengyu2@northeastern.edu
Patel, Yesha | patel.yes@northeastern.edu
Singh, Utkarsh | singh.utka@northeastern.edu

ABSTRACT

With the rapid increase of generation of new articles every single day fueled by the increase of presence of social media platforms – the abundance of news articles is overwhelming. For example, The Washington Post publishes around 1,200 pieces of content per day, including stories, graphics, videos, and blog posts.

Sometimes such an amount becomes overwhelming for the people who would like to sift through a particle kind of article on the day. Our project aims to ease this for the daily news reader by dividing the articles into mainly 4 categories – World, Sports, Business and Sci/Tech by developing a Natural Language Processing Model. In the following paper we have run various Machine Learning models to evaluate the accuracy on which model is more adept at classification of news articles into the 4 broad categories.

INTRODUCTION

With the consumer demand increasing on segregation of topics so as to help them sift through articles on a much faster basis – the necessity for having a classifier is of the utmost importance. The aim of our project is to determine which model is more adept at giving a higher accuracy in classification considering two different scenarios – one where the dataset is lemmatized and the other where we use stemming. We considered the 2 different possibilities for pre-processing since the root word generation can affect the way in which the NLP models behave. For our analysis of classification, we implemented and compared 4 models – Multinomial Naïve Bayes (M.N.B.), Bidirectional Encoder Representation from Transformers (B.E.R.T.), Long – Short Term Memory (L.S.T.M.) models and Hidden Markov Model (H.M.M.). The corpus is taken from the Kaggle repository under the name – "AG News Classification Dataset" – which contains news articles gathered by academic news research engine ComeToMyHead which has been operational since July 2004. The corpus consists of 127600 samples, with 3 attributes corresponding to class index (1 to 4, each number specifying for 4 main categories – World, Sports, Business and Sci/Tech), title of the article and Description.

BACKGROUND

In the paper "Multi-Language Spam/Phishing Classification by Email Body Text: Toward Automated Security Incident Investigation" the authors successfully classified emails into phishing, spam and ham emails with accuracy was 90.07% ± 3.17%.

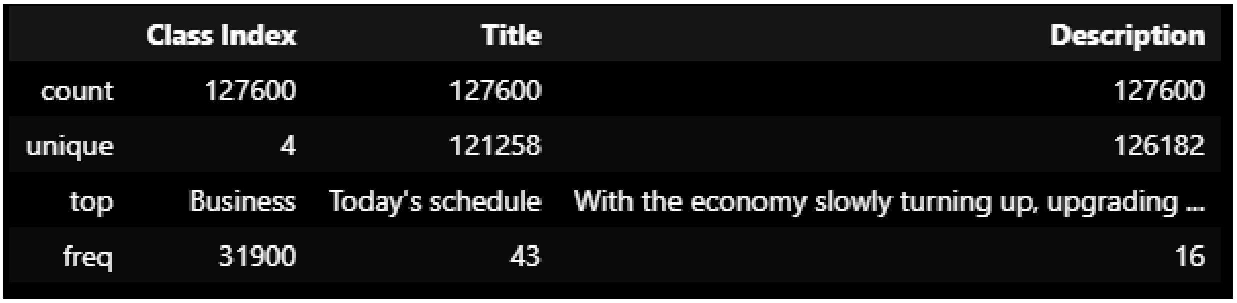
In the paper "Multi-Label Classification of E-Commerce Customer Reviews via Machine Learning" the authors have successfully implemented multilabel classification with the help of Random Forest (RF), Support Vector Classification (SVC), Naive Bayes (NB), Multi-label k-Nearest Neighbor (Ml-kNN), One-versus-Rest Logistic Regression (OvsR-LR) getting favorable results using basic machine learning models. Their Approach got a Micro Precision 0.9157.

In the paper “An Ensemble Machine Learning Approach for Fake News Detection and Classification Using a Soft Voting Classifier | European Journal of Electrical Engineering and Computer Science” used a soft voting classifier to aggregate four machine learning algorithms, namely, Naive Bayes, SVM and Logistic Regression for the classification of news articles as fake or real. Their proposed ensemble approach produced accuracy of 93% and precision - 94%.

APPROACH

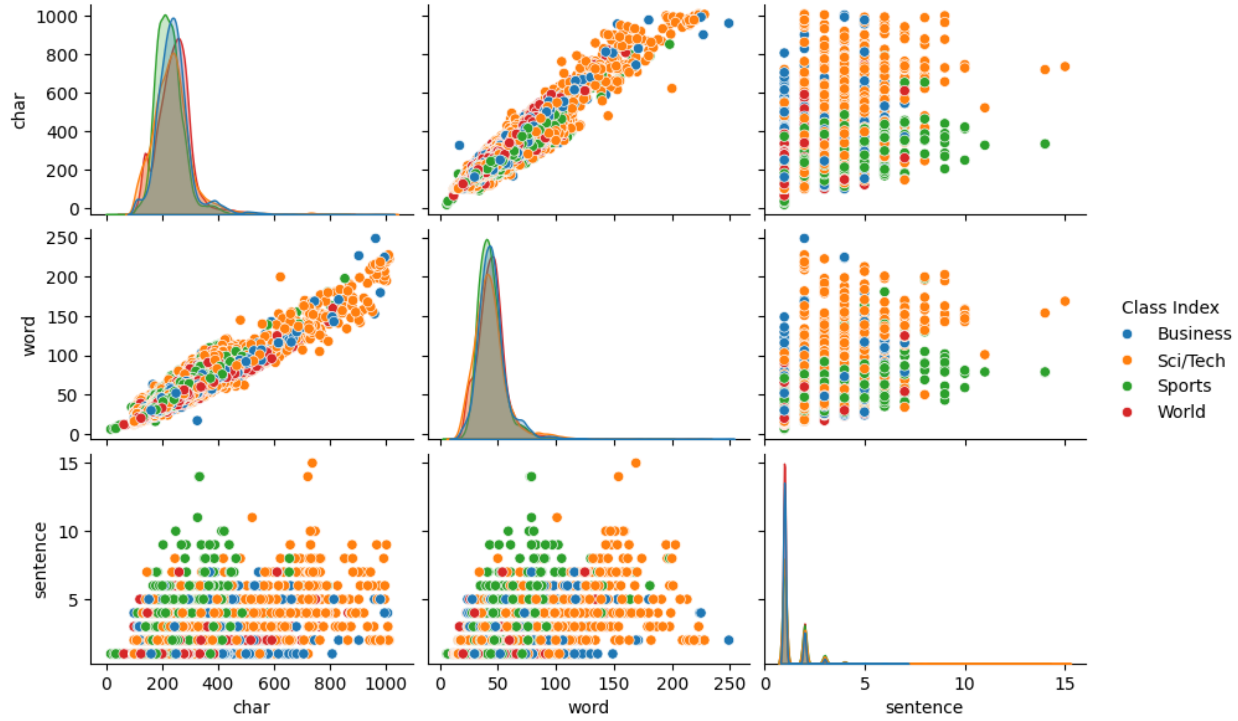
Exploratory Data Analysis

Data Description:

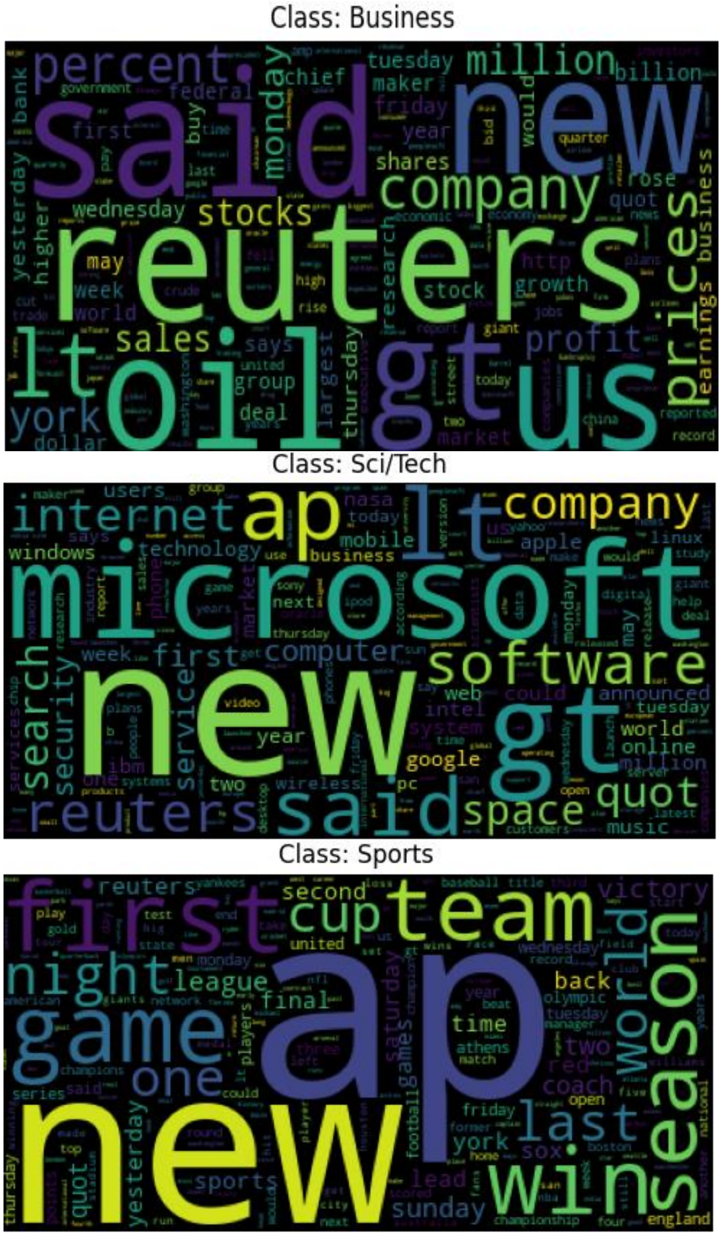


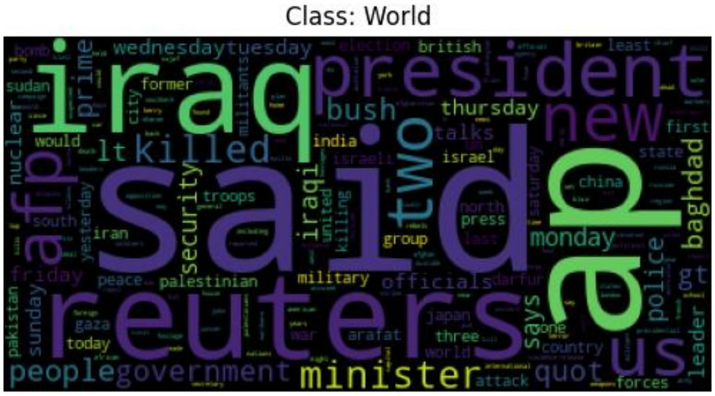
	Class Index	Title	Description
count	127600	127600	127600
unique	4	121258	126182
top	Business	Today's schedule	With the economy slowly turning up, upgrading ...
freq	31900	43	16

Distribution of Number of Characters, Words and Sentences



Popular Words in Each Class





Data Preprocessing and Cleaning

The Corpus initially consisted of two files train and test where the train file consisted of 120000 observations along with 7600 observations in the test file. Both files were combined and checked for Null Values – 0 null values were found. The Merged corpus was then split randomly into train and test sets in the ratio of 70:30 respectively. This led to 89320 records in the train set and 38280 records in the test set. The occurrences of the target classes in the corpus were 25% each – implying a perfectly balanced corpus. Duplicate values were also checked for the News Articles (combination of title and description attribute) – 0 duplicate values were found.

3 different parameters were visualized as well - number of characters in a message, the Number of words in a message, the Number of sentences in a message for each particular class as seen in Figure .

On further Exploratory Analysis of our Data – we got to know the most used words for each Target Class as mentioned in word cloud Figure

Then moving ahead towards the train and test set a new column was introduced with the merged values if the Title and Description column. The new column was preprocessed by lowercasing, removing HTML tags, punctuations, numbers and tokenization. Post preprocessing the corpus was then divided into two main categories: one where the corpus was lemmatized and the other where stemming was applied. Word embedding techniques like CountVectorizer and Term Frequency - Inverse Document Frequency vectorizer (TFIDF Vectorizer) were used which essentially help us convert our corpus to vectors for our models to interpret.

MODEL IMPLEMENTATION

Multinomial Naïve Bayes

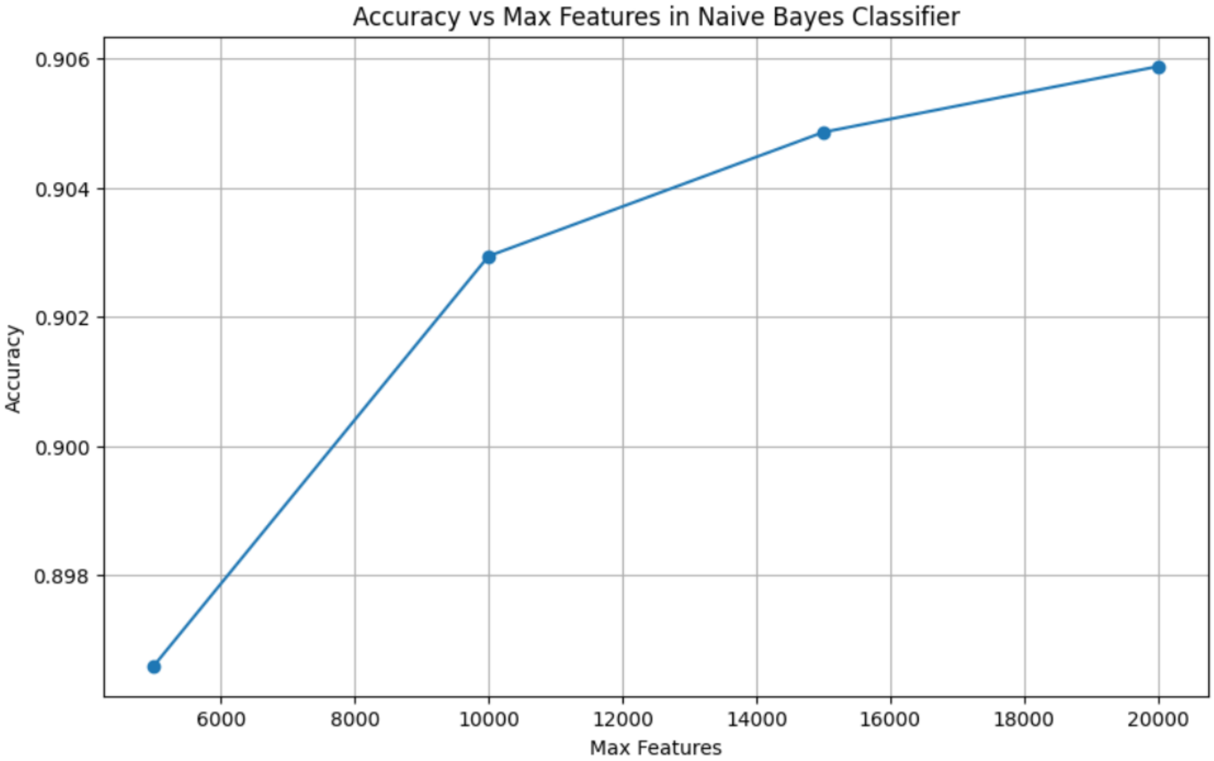
A systematic evaluation was conducted to determine the optimal number of features for Naive Bayes classifier performance in text categorization using the AG News dataset. Multinomial Naive Bayes was chosen as the algorithm due to its popularity, efficiency, and simplicity in text classification tasks.

Two techniques were employed to process the data: stemming and lemmatization. TF-IDF was applied to both techniques to convert the data into vectors for further processing. The experiment iterated through a predefined set of max_features values, ranging from 5000 to 20000, for both stemming and lemmatization separately. For each unique max_features value, a separate Naive Bayes classifier was trained on the corresponding training set. The trained models were then used to predict categories on the test set, derived from the test subset of the dataset.

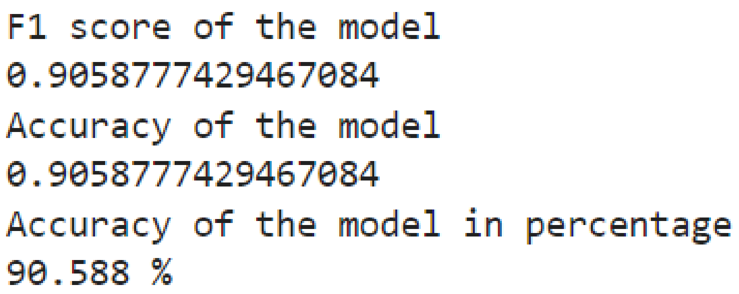
The accuracy of stemming and lemmatization was visualized with different numbers of max_features. The optimal Naive Bayes model for both stemming and lemmatization was obtained with a max_features value of 20000, resulting in an accuracy score exceeding 90%. And the Naïve Bayes model which uses the Lemmatization techniques perform slightly better than the one using the Stem method.

The findings from this exploration provide valuable insights into feature selection for Naive Bayes classifiers in the realm of text classification, emphasizing the significance of tuning the max_features parameter to improve model accuracy.

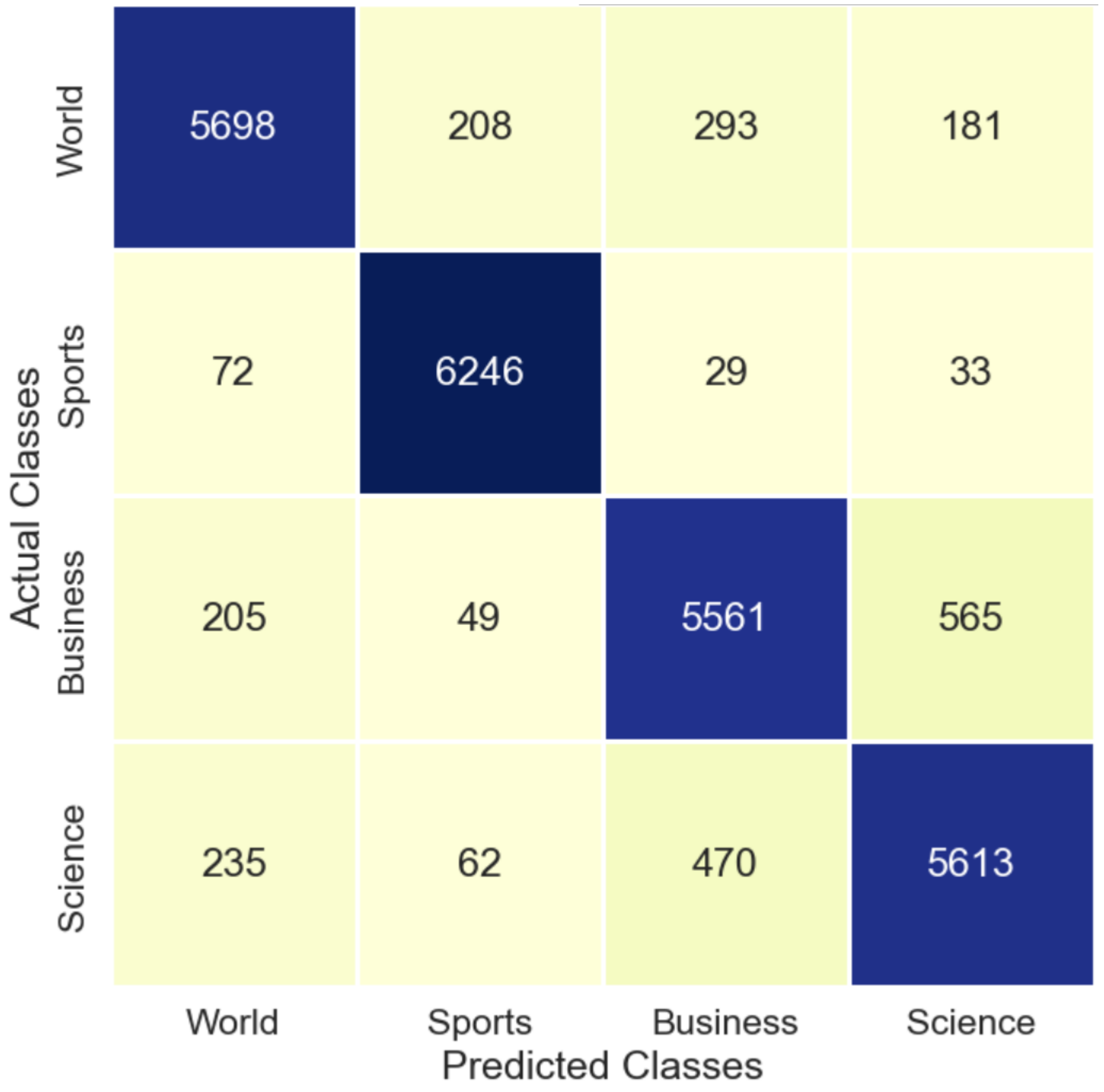
Using Lemmatization



```
Best max_features: 20000 with accuracy: 0.9058777429467084
```

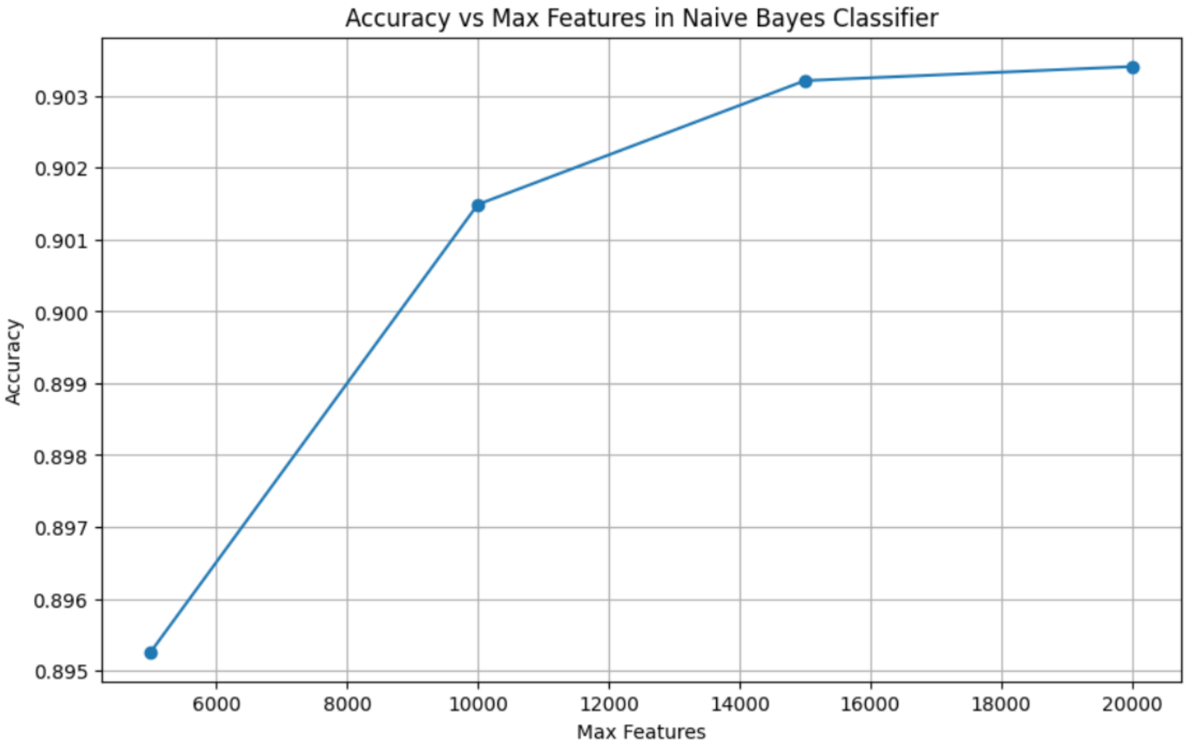


```
F1 score of the model
0.9058777429467084
Accuracy of the model
0.9058777429467084
Accuracy of the model in percentage
90.588 %
```

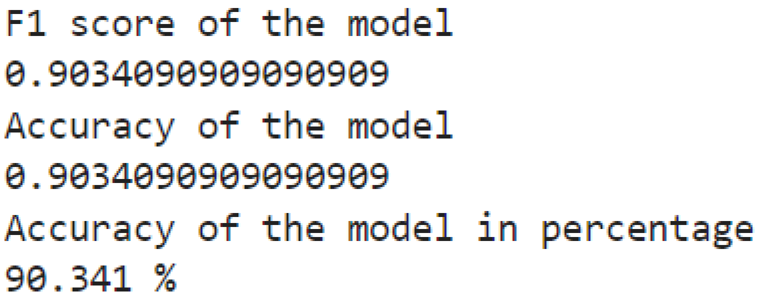


	World	Sports	Business	Science
World	5698	208	293	181
Sports	72	6246	29	33
Business	205	49	5561	565
Science	235	62	470	5613

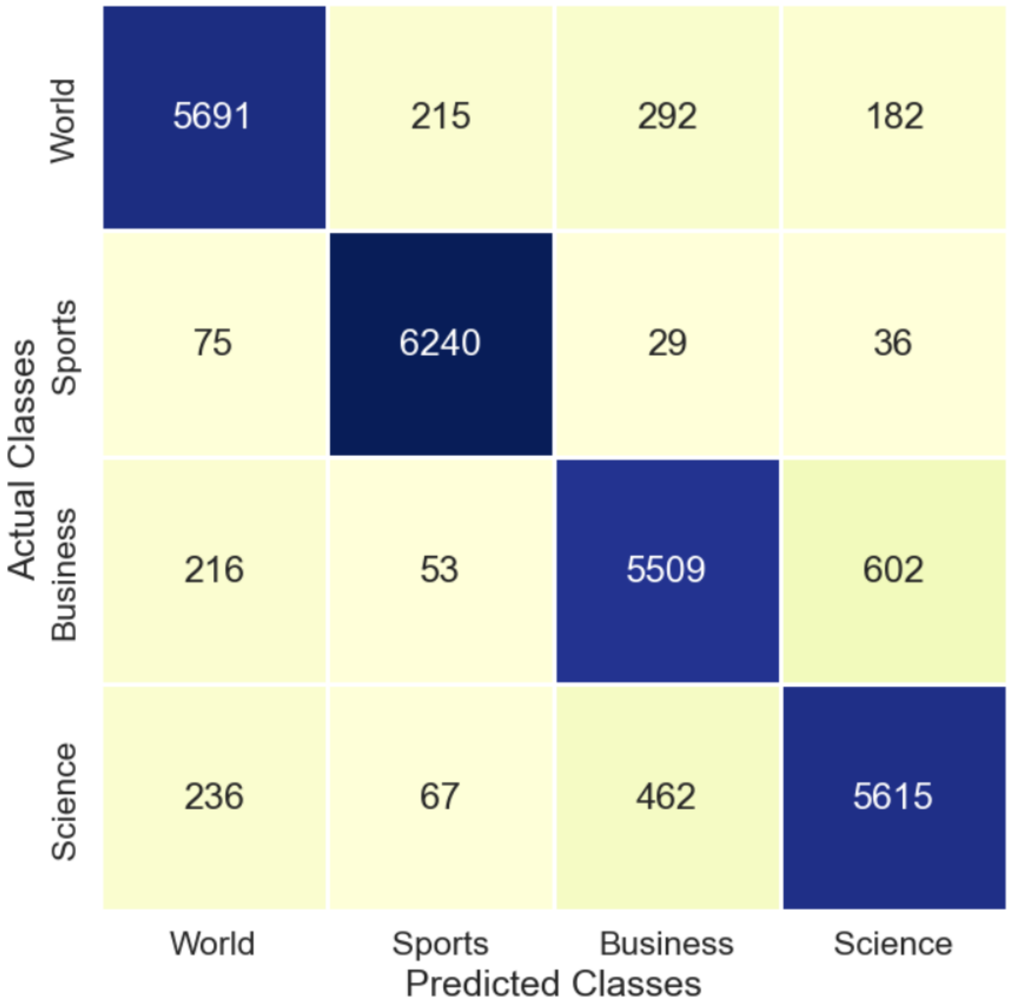
Using Stemming



```
Best max_features: 20000 with accuracy: 0.9034090909090909
```



```
F1 score of the model
0.9034090909090909
Accuracy of the model
0.9034090909090909
Accuracy of the model in percentage
90.341 %
```



	World	Sports	Business	Science
World	5691	215	292	182
Sports	75	6240	29	36
Business	216	53	5509	602
Science	236	67	462	5615

LONG SHORT-TERM MEMORY NETWORKS

Complementing the Naive Bayes classification experiments, an investigation into the capabilities of Long Short-Term Memory (LSTM) networks for the AG News dataset classification task was undertaken. LSTMs, a type of Recurrent Neural Network (RNN), excel at processing sequential data, making them well-suited for text classification challenges where context and word order are crucial.

In the preprocessing stage, text tokenization and sequence padding were employed to prepare the stemmed text data for neural network processing. A tokenizer was configured to consider the top 5,000 words from the dataset. With complicated model like LSTM, large number of Tokens requires too much computational power, so we choose to proceed with 5000.

Echoing the Naive Bayes classification experiments, the LSTM's performance was benchmarked against the Naive Bayes classifiers to assess its effectiveness in handling text data with intricate dependencies. Preliminary results indicated that the LSTM was adept at capturing nuanced patterns in sequence data, resulting in a robust model with promising accuracy metrics. The precise performance figures, alongside those of Naive Bayes, provided a comprehensive understanding of the strengths and limitations of both approaches in the context of the AG News dataset.

In comparison to the Naive Bayes classifiers, the LSTM demonstrated superior performance, with an accuracy of 91% with Lemmatization texts outperformed the Naïve Bayes classifier. The findings from this exploration underscore the importance of selecting the appropriate algorithm for the task at hand. While Naive Bayes classifiers offer simplicity and efficiency, LSTM networks provide superior performance in handling complex sequential data, making them a valuable tool for text classification tasks.

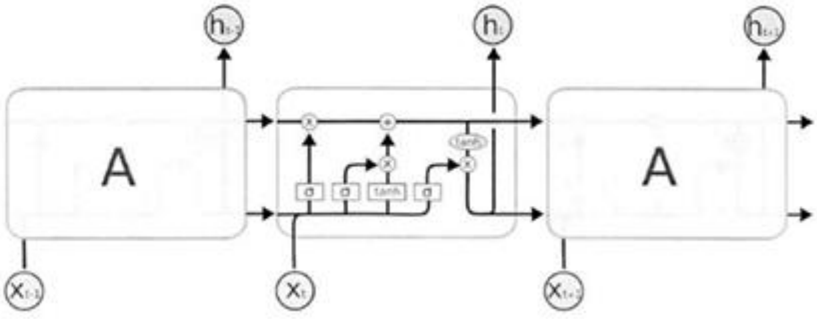
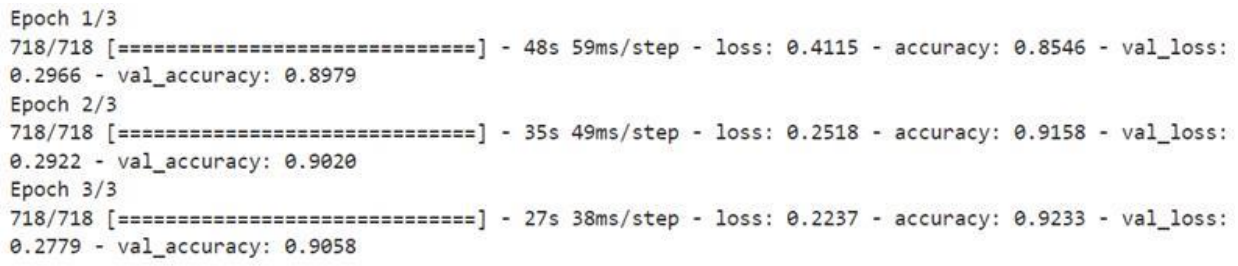
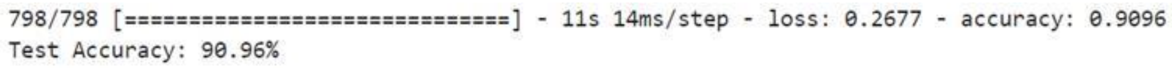


Fig. 2. The structure of LSTM

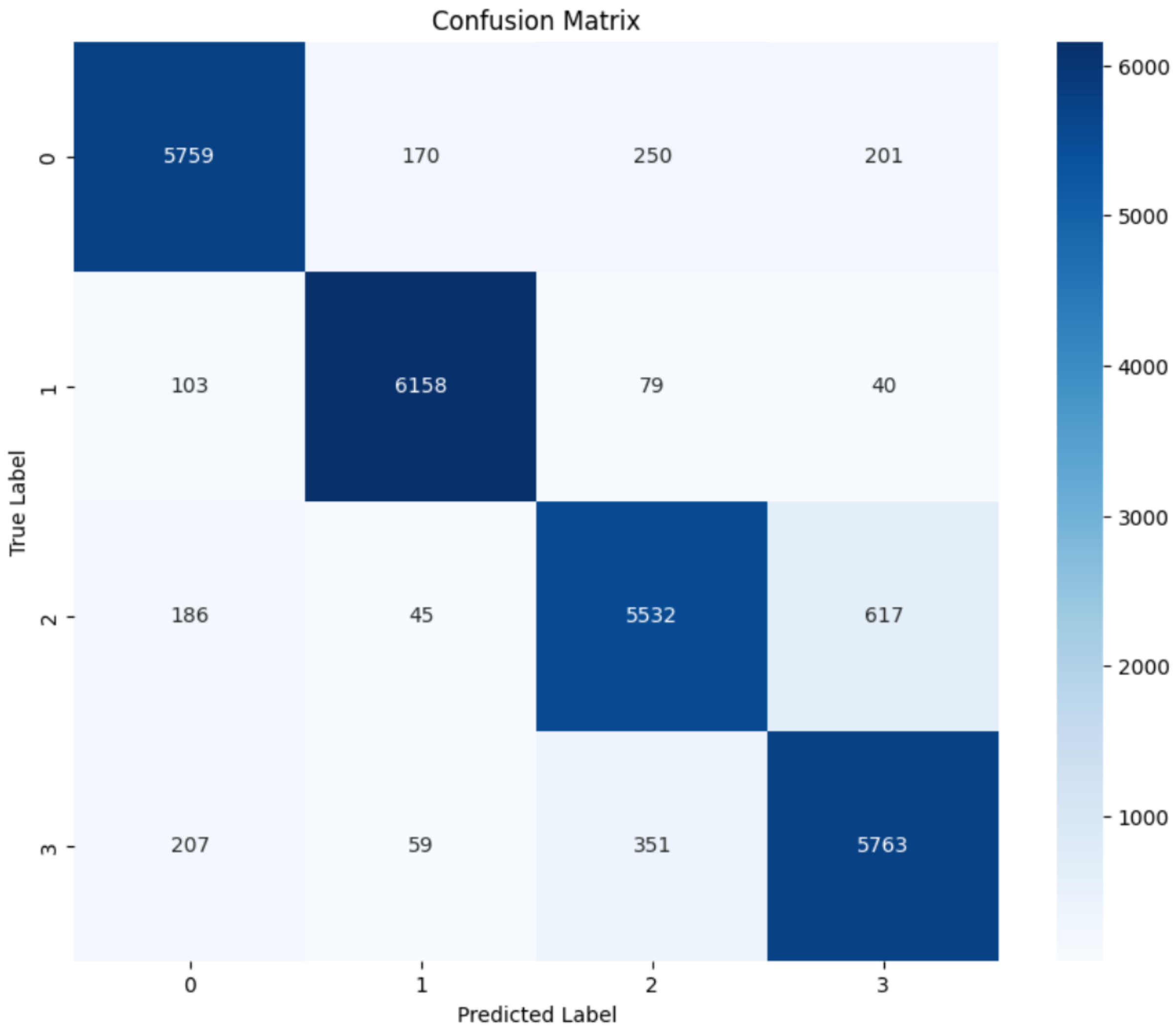
Using Stemming



```
Epoch 1/3
718/718 [==============================] - 48s 59ms/step - loss: 0.4115 - accuracy: 0.8546 - val_loss:
0.2966 - val_accuracy: 0.8979
Epoch 2/3
718/718 [==============================] - 35s 49ms/step - loss: 0.2518 - accuracy: 0.9158 - val_loss:
0.2922 - val_accuracy: 0.9020
Epoch 3/3
718/718 [==============================] - 27s 38ms/step - loss: 0.2237 - accuracy: 0.9233 - val_loss:
0.2779 - val_accuracy: 0.9058
```

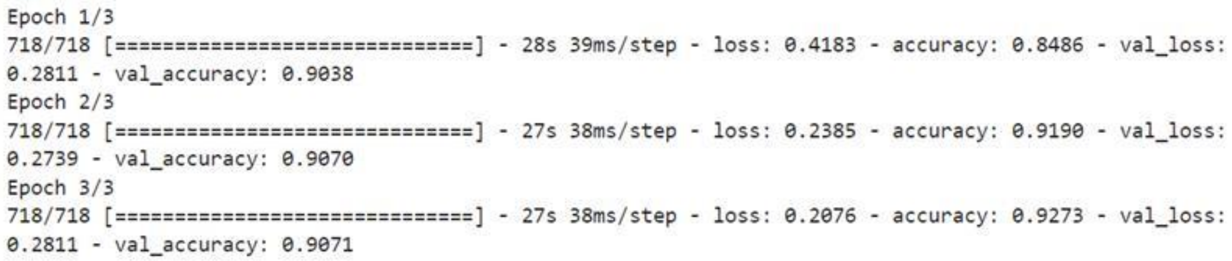


```
798/798 [==============================] - 11s 14ms/step - loss: 0.2677 - accuracy: 0.9096
Test Accuracy: 90.96%
```

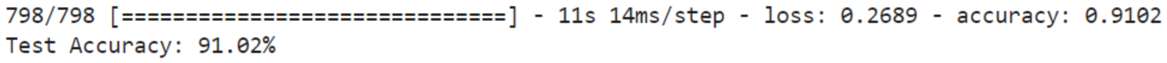


	0	1	2	3
0	5759	170	250	201
1	103	6158	79	40
2	186	45	5532	617
3	207	59	351	5763

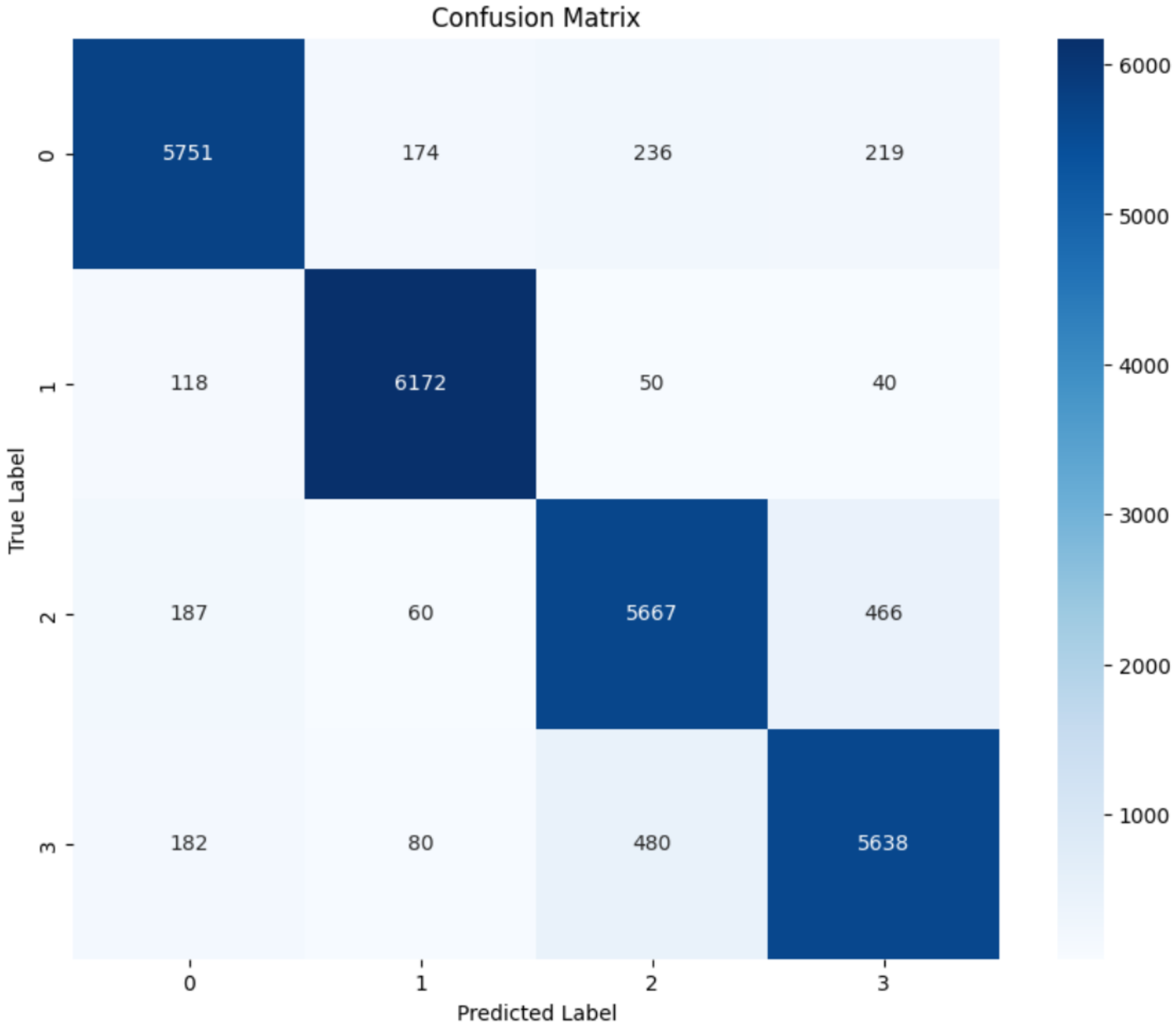
Using Lemmatization



```
Epoch 1/3
718/718 [==============================] - 28s 39ms/step - loss: 0.4183 - accuracy: 0.8486 - val_loss:
0.2811 - val_accuracy: 0.9038
Epoch 2/3
718/718 [==============================] - 27s 38ms/step - loss: 0.2385 - accuracy: 0.9190 - val_loss:
0.2739 - val_accuracy: 0.9070
Epoch 3/3
718/718 [==============================] - 27s 38ms/step - loss: 0.2076 - accuracy: 0.9273 - val_loss:
0.2811 - val_accuracy: 0.9071
```



```
798/798 [==============================] - 11s 14ms/step - loss: 0.2689 - accuracy: 0.9102
Test Accuracy: 91.02%
```



Confusion Matrix

	0	1	2	3
0	5751	174	236	219
1	118	6172	50	40
2	187	60	5667	466
3	182	80	480	5638

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

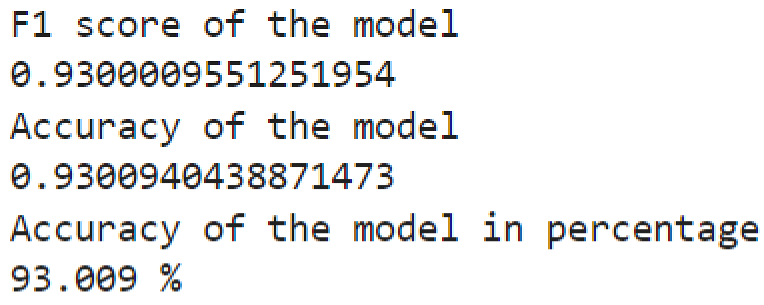
Complementing the Naive Bayes and LSTM classification experiments, an in-depth exploration of the Bidirectional Encoder Representations from Transformers (BERT) model was conducted for text categorization on the AG News dataset.

Our methodology involved preprocessing the AG News dataset's text data to conform to BERT's requirements. This preprocessing step encompassed tokenizing the texts and converting them into a BERT-compatible input format. We employed a pre-trained BERT model, specifically the 'bert-base-uncased' variant and fine-tuned it on our dataset. The fine-tuning process entailed adjusting the model parameters to align with the specifics of our text classification task, enabling the model to learn from the intricacies and patterns within the news dataset.

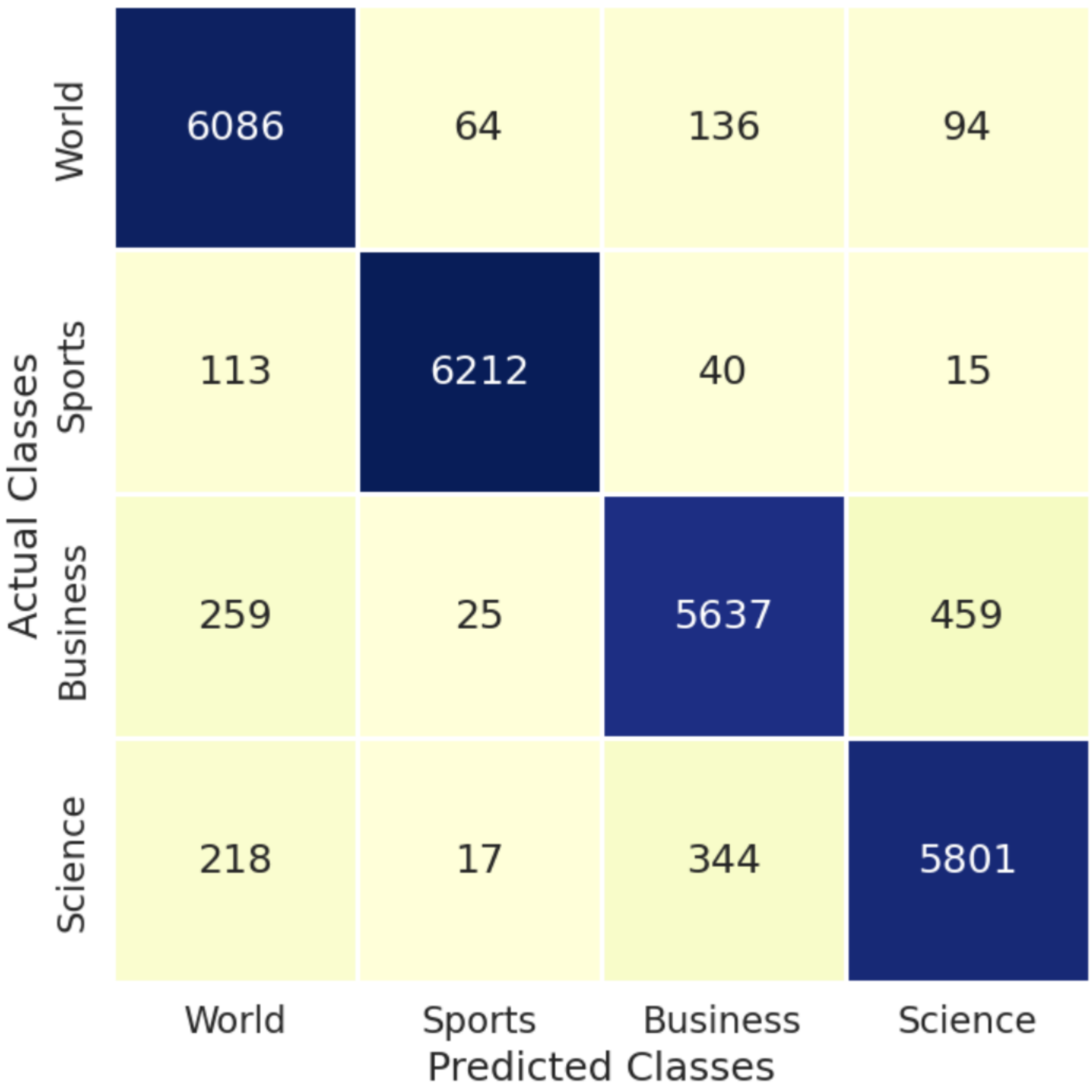
The results obtained from the BERT model were remarkable, showcasing its ability to grasp complex linguistic constructs and effectively classify text with high accuracy. In comparison to the Naive Bayes and LSTM classifiers, BERT surpassed their performance, achieving an accuracy of 93.2%. This superior performance is attributed to BERT's ability to capture long-range dependencies within text sequences, leading to a more nuanced understanding of the context. BERT's contextual awareness is particularly beneficial for tasks where the order of words and context play a critical role in determining the meaning of a sentence or paragraph.

The findings from this exploration underscore the importance of selecting the appropriate algorithm for the task at hand. While Naive Bayes classifiers offer simplicity and efficiency, and LSTM networks provide superior performance in handling complex sequential data, BERT stands out as a powerful tool for text classification tasks that demand high accuracy and in-depth contextual understanding. While Bert taking more time to train and more complicated step to use the model, it is still our best model on the AG news classification task.

Use Lemmatization

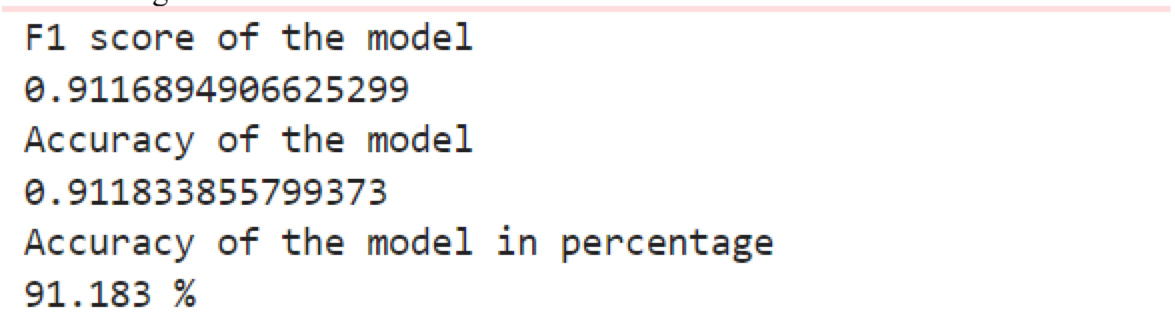


```
F1 score of the model
0.9300009551251954
Accuracy of the model
0.9300940438871473
Accuracy of the model in percentage
93.009 %
```

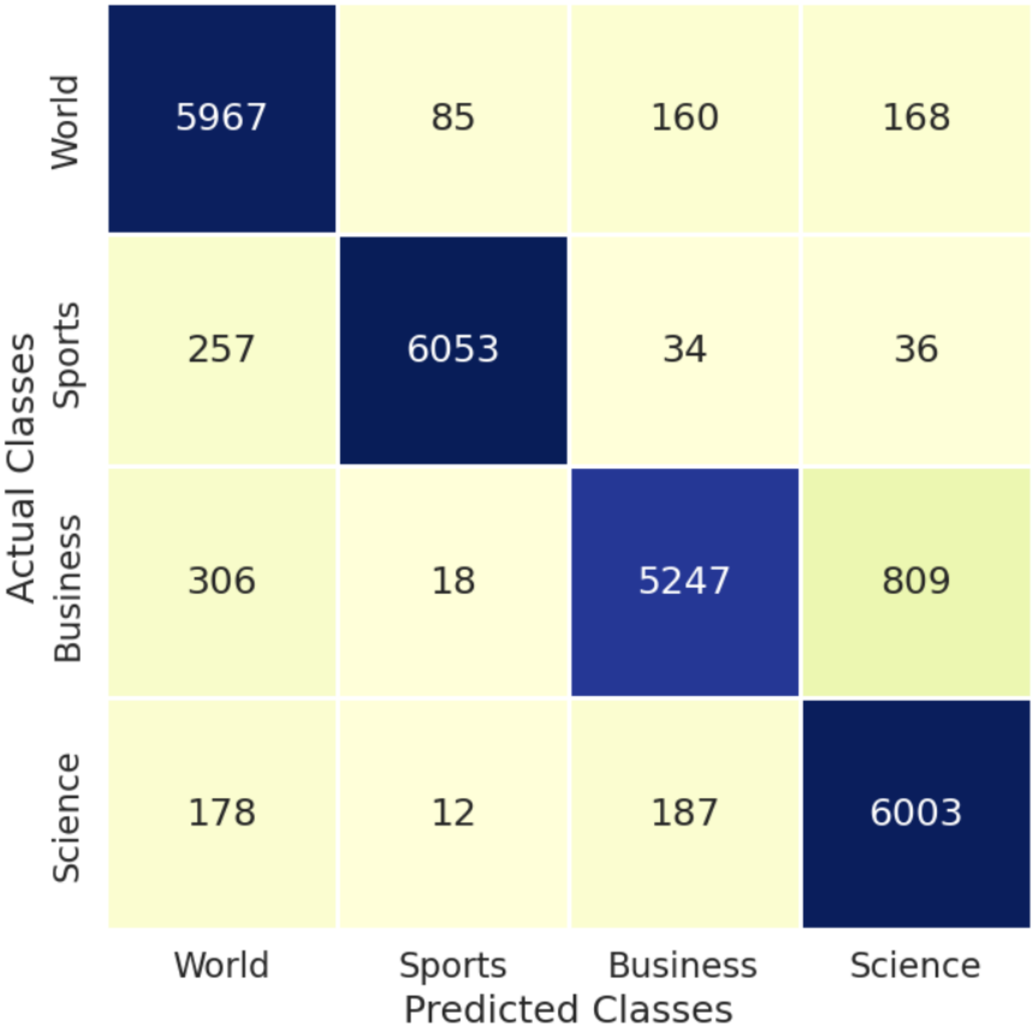


	World	Sports	Business	Science
World	6086	64	136	94
Sports	113	6212	40	15
Business	259	25	5637	459
Science	218	17	344	5801

For Stemming



```
F1 score of the model
0.9116894906625299
Accuracy of the model
0.911833855799373
Accuracy of the model in percentage
91.183 %
```



	World	Sports	Business	Science
World	5967	85	160	168
Sports	257	6053	34	36
Business	306	18	5247	809
Science	178	12	187	6003

Conclusion:

Our project mainly focuses on research of classification models based on Deep Learning and traditional machine learning method. With the various tools we use to preprocess the texts data and tokenize the texts, we developed several models including Naïve Bayes, LSTM and Bert. Out of the three models we trained, the Bert gives the best accuracy for classifying the different categories of news. With over 93% accuracy, we can say that the Bert is the best model for classifying the news category. But there are cons for the Bert model, which is the training time, the Bert model alone takes longer than the Naïve Bayes and LSTM combined.

In the future steps for this project, there are several things we can do: There are many other good models for the NLP classification problem, and we would like to try it out. We will also try to optimize the Bert model which make it more time efficient.

BIBLIOGRAPHY

- Jiang, Qinghua. "Text Classification Method Based on LSTM." *EAI URB-IOT*, 2022, EAI, doi: 10.4108/eai.20-12-2021.2315016. https://doi.org/10.4108/eai.20-12-2021.2315016
- Article: https://www.theatlantic.com/technology/archive/2016/05/how-many-stories-do-newspapers-publish-per-day/483845/
- Zhang, Wei & Gao, Feng. (2011). An Improvement to Naive Bayes for Text Classification. Procedia Engineering. 15. 2160-2164. 10.1016/j.proeng.2011.08.404.
- Yongjun Hu, Jia Ding, Zixin Dou, Huiyou Chang, "Short-Text Classification Detector: A Bert-Based Mental Approach", Computational Intelligence and Neuroscience, vol. 2022, Article ID 8660828, 11 pages, 2022. https://doi.org/10.1155/2022/8660828
- S. Liu, H. Tao and S. Feng, "Text Classification Research Based on Bert Model and Bayesian Network," 2019 Chinese Automation Congress (CAC), Hangzhou, China, 2019, pp. 5842-5846, doi: 10.1109/CAC48633.2019.8996183.