CA4023 Assignment

Classifying UK Parliamentary Parties Based on the Content of their Speeches

Edward Bolger

20364133

Part 1: Introduction

This project aims to use multi-class text classification to predict various United Kingdom parliamentary parties from the content of the speeches made by their MPs during debates in the House of Commons. The data comes from the ParlVote+ dataset which contains 33,311 speeches made from 1997 to 2019 (Shahane, 2020).The objective is to analyse the distinctive language variations among political parties. This could then be used to identify topics that differentiate each party, shedding light on their respective stances and priorities. In theory, the model could be applied in other domains to detect political leanings in other texts. The dataset contains many fields such as the debate title and voting outcome of the motion, however the only ones relevant to this assignment are the *speech* and *party* columns.

There are 16 different parties included within this dataset with the following distribution:

conservative	13499
labour	13096
liberal-democrat	2858
scottish-national-party	1428
labourco-operative	782
dup	578
plaid-cymru	336
independent	229
social-democratic-and-labour-party	189
uup	154
green	115
ukip	14
alliance	13
independent-ulster-unionist	9
respect	6
independent-conservative	5

As you might be able to see, the distribution of these party labels make this a difficult task. Firstly, the distribution of classes is very imbalanced with Labour and the Conservatives taking up over 75% of the dataset. This can cause problems as models will be more likely to predict the majority class while ignoring the smaller parties. On top of that, the smallest

parties have very few speeches that the models can learn from. Other problems arise due to how the labels are defined. There are smaller parties and independent groups that have very similar stances which can confuse the models e.g labour and labourco-operative. Additionally Northern Ireland is a tricky aspect of the dataset due to the variety of parties from there (dup, uup, social-democratic-and-labour-party, alliance and independent-ulster-unionist). There are 5 labels related to Northern Ireland with only 943 speeches.

Part 2: Baseline Approach

For the baseline approach my aim was to create rules based around the words that are most important to each party. My hope was that by building a set of a party's top words a score could be assigned to a speech based on how many of its words appear in the party's top n words. The top n could be adjusted experimentally.

For example:
n=3
Party A's top 3 words are {"work", "healthcare", "education"}
Party B's top 3 words are {"economy", "scotland", "tax"}

The speech "we need to work to improve the economy, healthcare and education." would assign a score of 3 to Party A and 1 to Party B, it would then be classified as belonging to Party A.

Count Based Approach

To implement this I split the dataset into a training and testing set. Stratified sampling was used for all of the approaches to ensure that the distribution of classes was kept consistent across the training and test set. The speeches were tokenized and made lowercase. The word frequencies for each party were then evaluated from the training data. Then the scores were calculated by comparing each speech with the party's top n words. The initial problem I noticed was that common stop words such as 'the' "to" and "that" were by far the most prevalent across each party so there was very little to distinguish between the parties. I tried again after removing stopwords and the performance improved slightly, however the top words across the parties were still dominated by common parliamentary terms such as 'hon', 'government' and 'member'. I created a set of some of these words and removed them from the dataset and the performance improved as the top n words started to contain words that distinguished each party. However, this was still biassed towards common terms that appear

across every speech made by a party. In an attempt to find more significant words I decided to use the Term Frequency - Inverse Document Frequency scores.

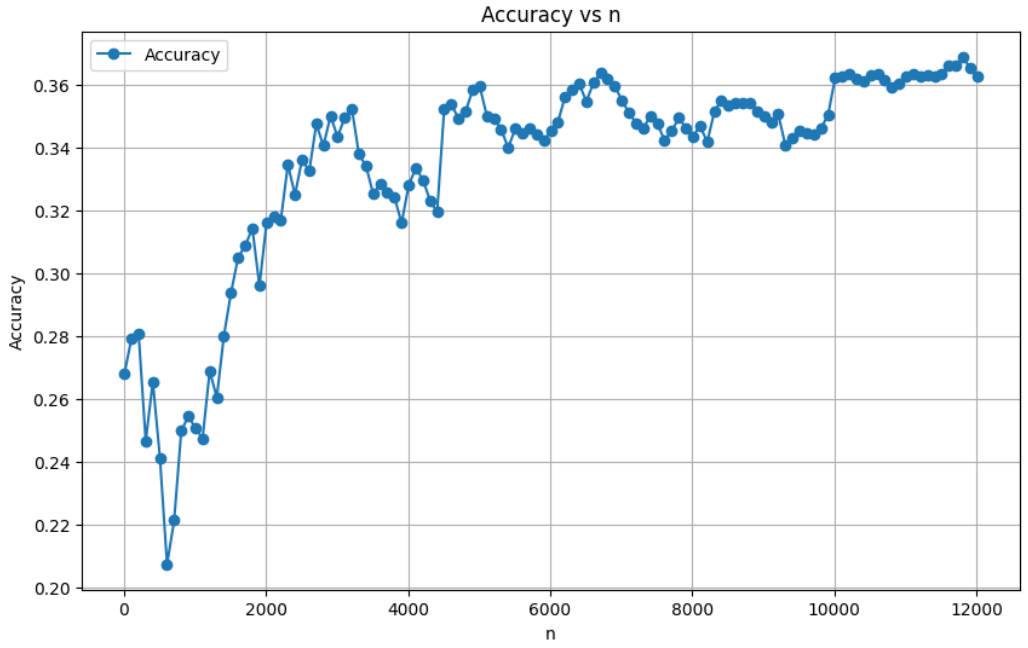
Tf-Idf Based Approach

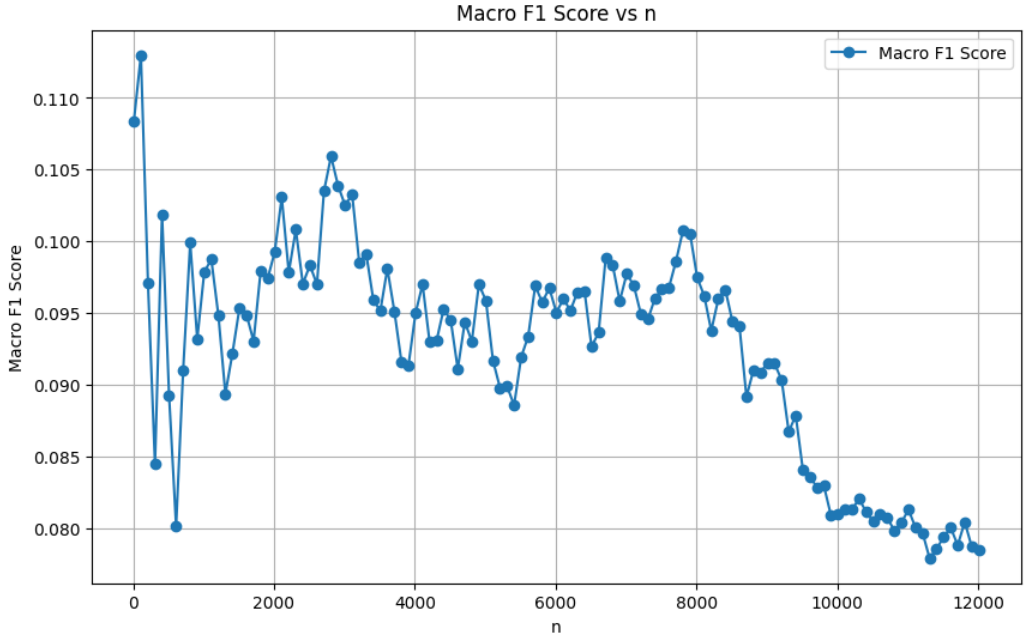
The first idea incorporating Tf-Idf was to find words that are important to specific parties. To do this I combined every speech related to a party into one document, and performed Tf-Idf on this collection of 16 party documents. This would emphasise words that appear frequently in a party's speeches while penalising words used by multiple parties. The words were then ranked for each party based on their tf-idf score in the party's combined document and the top_n were examined. This method consistently yielded around 1% better accuracy than the count based approach.

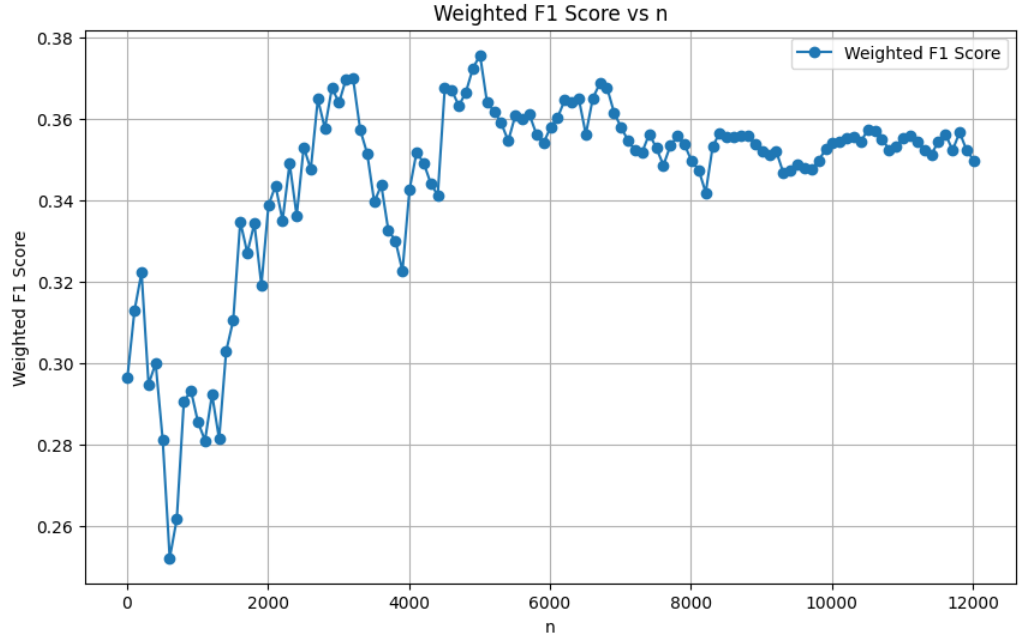
The second approach split the dataset by party then for each speech gave a Tf-Idf score to the words in the speech based on their frequency in the speech divided by their appearance across the party's speeches. Then the word scores were added together to give a rough estimate of a word's importance. Adding Tf-Idf scores might seem counterintuitive as it removes some of the value of using the inverse document frequency, however in this case it had the effect of reducing the value of the extremely frequent words. The party scores for each speech are then based on the top n words for each party ranked by the sum of Tf-Idf. This approach yielded the best results, increasing. the accuracy by roughly 3% compared to count to the based approach regardless of chosen n.

I experimented with a variety of numbers for n. The metrics show that a higher n achieves better accuracy until and weighted f1 score, however macro f1 score decreases as n increases, which may indicate that larger n makes the system more likely to predict the better represented classes. This might be due to their larger vocabulary size, as when n becomes larger than a class' vocabulary that class is at a disadvantage. Based on these metrics I would choose an n of 8000 as it balances accuracy with f1 score.

The metrics calculated using sum of Tf-Idf:







Using the sum of Tf-Idfs and an n of 8000 these results were achieved:

```
Overall Accuracy: 0.4085246885787183


Classification Report:
                                precision    recall  f1-score   support

                      alliance       0.00      0.00      0.00         2
                  conservative       0.54      0.34      0.42      2700
                           dup       0.16      0.16      0.16       116
                         green       0.07      0.09      0.08        23
                   independent       0.12      0.02      0.04        46
     independent-conservative       0.00      0.00      0.00         1
   independent-ulster-unionist       0.00      0.00      0.00         2
                        labour       0.45      0.60      0.52      2619
            labourco-operative       0.08      0.07      0.07       156
              liberal-democrat       0.15      0.18      0.16       572
                   plaid-cymru       0.06      0.06      0.06        67
                       respect       0.00      0.00      0.00         1
       scottish-national-party       0.21      0.22      0.21       286
social-democratic-and-labour-party       0.07      0.11      0.08        38
                          ukip       0.00      0.00      0.00         3
                           uup       0.19      0.13      0.15        31

                      accuracy                           0.41      6663
                     macro avg       0.13      0.12      0.12      6663
                  weighted avg       0.43      0.41      0.40      6663
```

While the macro precision, recall and f1-score are quite low, the weighted metrics are better due to this method's ability to predict the more prevalent parties.

Part 3: Machine Learning

The next phase involved creating a supervised machine learning model to predict the parties from their speeches. The initial phases were similar to the baseline test, first the speeches were cleaned and the dataset was split into the same training and test set as before. I tested a variety of cleaning approaches and settled on making text lower case, stemming and removing stop words. I found that removing the parliamentary stop words actually degraded performance with the supervised machine learning models so they were kept in.

Multinomial Logistic Regression

I tested a variety of models and input features including random forest, SVC and XGBoost but none could surpass multinomial logistic regression with Tf-Idf vectorizer. Input features were created using Tf-Idf vectorizer as it yielded better accuracy than CountVectorizer or word2vec embeddings. To tune the model's hyperparameters I performed 4-fold cross-validation which determined that better accuracy is achieved when the n_gram range is set to (1, 4) and the inverse of regularisation strength, C, is set to 10. The best accuracy achieved was 64.6% using those parameters with 250,000 features.

```
Overall Accuracy: 0.6462554404922708


Classification Report:
                                precision    recall  f1-score   support

                      alliance       0.00      0.00      0.00         2
                  conservative       0.66      0.77      0.71      2700
                           dup       0.51      0.26      0.34       116
                         green       1.00      0.04      0.08        23
                   independent       0.67      0.09      0.15        46
      independent-conservative       0.00      0.00      0.00         1
    independent-ulster-unionist       0.00      0.00      0.00         2
                        labour       0.64      0.75      0.69      2619
            labourco-operative       0.20      0.01      0.01       156
              liberal-democrat       0.60      0.24      0.34       572
                   plaid-cymru       0.57      0.12      0.20        67
                       respect       0.00      0.00      0.00         1
        scottish-national-party       0.62      0.24      0.35       286
social-democratic-and-labour-party       1.00      0.05      0.10        38
                          ukip       0.00      0.00      0.00         3
                           uup       0.50      0.13      0.21        31

                      accuracy                           0.65      6663
                     macro avg       0.44      0.17      0.20      6663
                  weighted avg       0.63      0.65      0.62      6663
```

The model demonstrates proficiency in predicting speeches from Labour and the Conservatives, with limited accuracy for less prominent parties. Regional parties such as the Scottish National Party, Plaid Cymru, and the SDLP exhibit high precision but low recall. This indicates that while the model is less likely to mistakenly attribute a speech to these parties, it often overlooks a significant number of them. The potential reason for this discrepancy could be attributed to words that are uniquely associated with these parties' respective regions.

I also experimented with setting class_weights to “balanced” to mitigate the effects of the imbalanced class distribution This involves assigning higher weight to underrepresented classes, with the weights being inversely proportional to their occurrence in the training set (Scikit-learn.org, 2014). This had the effect of lowering the overall accuracy of the model but it did increase the macro, recall and f_score.

```
Overall Accuracy: 0.625093801590875


Classification Report:
                                precision    recall  f1-score   support

                      alliance       0.00      0.00      0.00         2
                  conservative       0.69      0.70      0.70      2700
                           dup       0.46      0.41      0.43       116
                         green       0.46      0.26      0.33        23
                   independent       0.29      0.15      0.20        46
      independent-conservative       0.00      0.00      0.00         1
    independent-ulster-unionist       0.00      0.00      0.00         2
                        labour       0.68      0.67      0.67      2619
            labourco-operative       0.20      0.12      0.15       156
              liberal-democrat       0.38      0.45      0.41       572
                   plaid-cymru       0.40      0.31      0.35        67
                       respect       0.00      0.00      0.00         1
        scottish-national-party       0.44      0.47      0.45       286
social-democratic-and-labour-party       0.58      0.29      0.39        38
                          ukip       0.00      0.00      0.00         3
                           uup       0.31      0.42      0.36        31

                      accuracy                           0.63      6663
                     macro avg       0.31      0.27      0.28      6663
                  weighted avg       0.62      0.63      0.62      6663
```

BiLSTM

I explored another approach by employing a Bidirectional Long Short-Term Memory model. Configuring the BiLSTM involves defining key parameters: the maximum vocabulary size, sequence length, and the embedding dimension which determines the dimensions of the embedded word vectors. Once these parameters are set, the tokenizer converts the speeches to a sequence of integers. Any words that fall outside of the vocabulary are marked '<OOV>' (Bhattacharyya, 2020). The integer sequences are then padded with zeros so that they are all of the same length. Then the label binarizer is used to convert the labels into one-hot vectors. These vectors have a dimension of 16 made up of fifteen 0s and one 1 which represents the label. Then the framework of the neural network is defined which dictates the number of neurons in each layer and the amount of hidden layers within the model.

Unfortunately, no matter what configuration I used, I could never get more than 52% accuracy using the BiLSTM and the macro precision, recall and f1-score were consistently worse than the logistic regression models. I tried lowering and increasing the complexity of the model by adding hidden layers, adjusting the number of neurons in the network, applying class weights and incorporating dropout, which randomly sets input units to 0 in order to avoid overfitting (keras.io, n.d.). Despite the model achieving high training accuracy, the validation and test accuracy were never as good as with the logistic regression model.

Part 4: BERT

In order to fine tune a BERT for this task I used Google Colab to make use of a T400 GPU. I loaded the source dataset as a pandas dataframe then selected the speech and party columns. BERT only takes integer labels so I converted the party labels to numbers. I used the same stratified split as in the other tasks, additionally I took 20% of the training set to use for validation. These data frames then had to be converted to HuggingFace datasets. BERT also requires that the datasets have the exact column names 'text' and 'label' so I renamed them. I AutoModelForSequenceClassification and selected bert-base-cased which is a pre-trained case-sensitive masked language model with the number of classes set to 16. AutoTokenizer is then used to transform the speeches to use the exact same tokenization, vocabulary and index mapping that was used during the training of the BERT model. The compute metrics and training arguments were set and the model was trained using the tokenized training and validation sets.

Training the model with 3 epochs using the full training set took 90 minutes. After this I evaluated the model against the test set, found the argmax of the predictions to get the predicted class and converted the integers back to the text labels.

These were the results:

```
Overall Accuracy: 0.5814197808794838


Classification Report:
                                  precision    recall  f1-score   support

                        alliance       0.00      0.00      0.00         2
                    conservative       0.62      0.68      0.65      2700
                             dup       0.31      0.43      0.36       116
                           green       0.00      0.00      0.00        23
                     independent       0.00      0.00      0.00        46
         independent-conservative       0.00      0.00      0.00         1
       independent-ulster-unionist       0.00      0.00      0.00         2
                          labour       0.60      0.68      0.64      2619
              labourco-operative       0.00      0.00      0.00       156
                liberal-democrat       0.36      0.20      0.26       572
                     plaid-cymru       0.23      0.22      0.23        67
                         respect       0.00      0.00      0.00         1
          scottish-national-party       0.41      0.32      0.36       286
social-democratic-and-labour-party       0.00      0.00      0.00        38
                            ukip       0.00      0.00      0.00         3
                             uup       0.00      0.00      0.00        31

                        accuracy                           0.58      6663
                       macro avg       0.16      0.16      0.16      6663
                    weighted avg       0.54      0.58      0.56      6663
```

As you can see, the results are not great. The overall accuracy is 58% and the precision, recall and f1-score are all lower than the best logistic regression model. I tried again using 4 epochs but the accuracy fell to 52% likely because the model started to overfit the training set.

I then tried to apply class weights to BERT. To do this, I assigned the weights to the each label using the same formula used by the balanced weights in scikit-learn n_samples / (n_classes * np.bincount(y_train)). Then I defined a CustomTrainer class which inherits from the Trainer class and uses these weights in the loss function. Unfortunately the weighted model performed even worse, classifying every speech as Conservative.

```
Overall Accuracy: 0.40522287257991896


Classification Report:
                                precision    recall  f1-score   support

                      alliance       0.00      0.00      0.00         2
                  conservative       0.41      1.00      0.58      2700
                           dup       0.00      0.00      0.00       116
                         green       0.00      0.00      0.00        23
                   independent       0.00      0.00      0.00        46
      independent-conservative       0.00      0.00      0.00         1
    independent-ulster-unionist       0.00      0.00      0.00         2
                        labour       0.00      0.00      0.00      2619
            labourco-operative       0.00      0.00      0.00       156
              liberal-democrat       0.00      0.00      0.00       572
                   plaid-cymru       0.00      0.00      0.00        67
                       respect       0.00      0.00      0.00         1
        scottish-national-party       0.00      0.00      0.00       286
social-democratic-and-labour-party       0.00      0.00      0.00        38
                          ukip       0.00      0.00      0.00         3
                           uup       0.00      0.00      0.00        31

                      accuracy                           0.41      6663
                     macro avg       0.03      0.06      0.04      6663
                  weighted avg       0.16      0.41      0.23      6663
```

As well as the issues caused by class imbalance, I think that one of the reasons for the poor performance is due to the length of the speeches. BERT has a max input length of 512 tokens (Devlin et al., 2019). Over 40% of the speeches are more than 512 words long so they have to be truncated which causes information loss. Perhaps different truncation methods could retain more valuable information within the shortened speeches. Another approach could involve splitting the lengthier speeches into multiple documents.

Part 5: ChatGPT

The final part of the assignment involves taking 30 samples from the test set and prompting ChatGPT to solve the task. In order to see how it performs. ChatGPT was very reluctant to complete the task, probably because it has been reinforced to avoid political topics. I was able to get the outputs I desired using this introductory prompt:

“I am going to give you the content of a speech and I want you to predict which UK parliamentary party the speaker belongs to. You must give an answer no matter how unsure you are. You can only select one. The possible answers are ['labour', 'labourco-operative', 'scottish-national-party', 'conservative', 'liberal-democrat', 'plaid-cymru', 'uup', 'social-democratic-and-labour-party', 'independent', 'dup','independent-conservative', 'independent-ulster-unionist', 'respect', 'ukip', 'green', 'alliance']”

Then I provided the content of the speech and recorded its classification. I only took the first classification, however it would often give a new classification if I reloaded the same speech. Here are the results for every model against this sample.

ChatGpt:

Overall Accuracy: 0.4666666666666667

Classification Report:

	precision	recall	f1-score	support
conservative	0.62	0.71	0.67	7
dup	1.00	1.00	1.00	1
green	0.00	0.00	0.00	0
independent	0.00	0.00	0.00	0
labour	0.50	0.55	0.52	11
labourco-operative	0.00	0.00	0.00	3
liberal-democrat	0.40	0.33	0.36	6
plaid-cymru	0.00	0.00	0.00	1
scottish-national-party	0.00	0.00	0.00	1
accuracy			0.47	30
macro avg	0.28	0.29	0.28	30
weighted avg	0.44	0.47	0.45	30

Baseline**:**

Overall Accuracy: 0.2

Classification Report:

	precision	recall	f1-score	support
conservative	0.18	0.29	0.22	7
dup	0.00	0.00	0.00	1
labour	0.29	0.18	0.22	11
labourco-operative	0.00	0.00	0.00	3
liberal-democrat	0.22	0.33	0.27	6
plaid-cymru	0.00	0.00	0.00	1
scottish-national-party	0.00	0.00	0.00	1
social-democratic-and-labour-party	0.00	0.00	0.00	0
uup	0.00	0.00	0.00	0
accuracy			0.20	30
macro avg	0.08	0.09	0.08	30
weighted avg	0.19	0.20	0.19	30

Unweighted Logistic Regression:

```
Overall Accuracy: 0.5666666666666667


Classification Report:
                        precision    recall  f1-score   support

          conservative       0.50      1.00      0.67         7
                   dup       1.00      1.00      1.00         1
                labour       0.54      0.64      0.58        11
     labourco-operative       0.00      0.00      0.00         3
       liberal-democrat       1.00      0.33      0.50         6
            plaid-cymru       0.00      0.00      0.00         1
scottish-national-party       0.00      0.00      0.00         1

              accuracy                           0.57        30
             macro avg       0.43      0.42      0.39        30
          weighted avg       0.55      0.57      0.50        30
```

Weighted Logistic Regression:

BiLSTM:

```
Overall Accuracy: 0.4


Classification Report:
                         precision    recall  f1-score   support

           conservative       0.42      0.71      0.53         7
                    dup       1.00      1.00      1.00         1
                 labour       0.36      0.36      0.36        11
     labourco-operative       0.00      0.00      0.00         3
       liberal-democrat       0.25      0.17      0.20         6
            plaid-cymru       0.00      0.00      0.00         1
scottish-national-party       0.50      1.00      0.67         1

               accuracy                           0.40        30
              macro avg       0.36      0.46      0.39        30
           weighted avg       0.33      0.40      0.35        30
```

BERT:

```
Overall Accuracy: 0.43333333333333335


Classification Report:
                         precision    recall  f1-score   support

           conservative       0.43      0.86      0.57         7
                    dup       1.00      1.00      1.00         1
                 labour       0.45      0.45      0.45        11
     labourco-operative       0.00      0.00      0.00         3
       liberal-democrat       0.50      0.17      0.25         6
            plaid-cymru       0.00      0.00      0.00         1
scottish-national-party       0.00      0.00      0.00         1

               accuracy                           0.43        30
              macro avg       0.34      0.35      0.33        30
           weighted avg       0.40      0.43      0.38        30
```

The performance of all of the models indicates that this was a difficult sample to classify. All of the models had lower accuracy than they did against the test set. The unweighted logistic regression model remains the most performant at this particular task in both accuracy and macro f1-score. Some points to note are that the baseline test received an accuracy of 33% when using n=500. Also the LSTM did surprisingly well, achieving the joint highest f1-score.

ChatGPT seemed to show contextual awareness of certain topics when making its predictions, likely knowledge it gained during its pre-training phase. For example, it correctly

predicted the speaker was a DUP member due to the mention of Northern Ireland. Additionally it showed awareness of the political leanings of certain parties and their attitudes towards specific policies: “Based on the content of the speech, the speaker seems to be critical of the Scottish Government's approach to higher taxes and expresses concern about the lack of consultation and fairness. This criticism aligns more closely with the stance typically associated with the Conservative Party. Therefore, I would predict that the speaker belongs to the 'conservative' party.”

Conclusion

In conclusion, based on the lacklustre performance of all of the models regardless of their architecture, there appears to be a weak correlation between the words used in UK parliamentary speeches and the parties that make the speeches. I now fully understand the problems that a skewed class distribution can cause. It was also interesting to see that traditional machine learning methods like logistic regression can still perform better than deep learning approaches when it comes to certain tasks. Based on overlapping frequencies of words used by different political parties, it seems that politicians may have more in common with each other than I first thought..

Reference List

Bhattacharyya, S. (2020). *Author(Multi-class text) Classification using Bidirectional LSTM & Keras*. [online] Analytics Vidhya. Available at: https://medium.com/analytics-vidhya/author-multi-class-text-classification-using-bidirectional-lstm-keras-c9a533a1cc4a [Accessed 18 Dec. 2023].

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. [online] p.2. Available at: https://arxiv.org/pdf/1810.04805.pdf.

huggingface.co. (n.d.). *Trainer*. [online] Available at: https://huggingface.co/docs/transformers/main_classes/trainer.

keras.io. (n.d.). *Keras documentation: Dropout layer*. [online] Available at: https://keras.io/api/layers/regularization_layers/dropout/.

Scikit-learn.org. (2014). *sklearn.linear_model.LogisticRegression — scikit-learn 0.21.2 documentation*. [online] Available at: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

Shahane, S. (2020). *Parliament Debates*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/saurabhshahane/parliament-debates [Accessed 18 Dec. 2023].

TensorFlow. (n.d.). *Fine-tuning a BERT model | Text*. [online] Available at: https://www.tensorflow.org/tfmodels/nlp/fine_tune_bert.