Project background

For this project a Swedish BERT model was fine-tuned on the task of classifying the political party of speakers in the Swedish parliament. There were two tests done, one on political speeches from 1994-2021 and one on the same data truncated to only contain speeches between 2010-2021. The reason why the test was run twice was to see if more recent political speech was easier to predict, inspired by a paper (Sapiro-Ghelier 2018) that seemed to indicate this.

The main motivation behind this project is based on the idea that a person belonging to a particular political party and representing that party in the parliament can be seen as an easily identifiable proxy for that person's general political bias.

One example of why recognizing political biases is of interest is to avoid receiving biased information from news sources that are subtly biased so that the reader is unknowingly receiving his information from biased news sources (Gangula et al 2019).

Previous work using NLP to analyse the political bias of tweets and facebook posts is quite common, there has also been work done on identifying the political bias of a speaker in other domains like news articles and other sources. (David et.al 2016, Gobeck et.al 2011, Gangula et al 2019). This project was meant to be done in a similar vein, but on political speech in the swedish parliament.

BERT specifically has been used for detection of policy preferences of members in the British parliament (Abercrombie et al 2019) and other topics like detecting populist content in text (Ulinskaite & Pukelis 2021).

The different parties in parliament were during most of the time of the data collection split into two opposite political blocks and since 2010 a large independent party (SD). The blocks have changed over time and the parties themselves have changed as well. But the general political leanings of the parties have mostly stayed the same, for example, the social democrats(S) are more left leaning than the moderates(M) and vice versa.

It should be noted that in general it is difficult to do a proper objective analysis of the different parties' political bias. There are a variety of different ways to categorize political bias, there is the left-right spectrum, GAL-TAN etc and even if a decision is made for which scale to use, actually classifying the parties according to that scale is not necessarily easy without being subjective. It seems reasonable that there are general observations to be made however.

If the model is able to pick up on general political biases of the speaker it would be seen as preferable if, when the model predicted the wrong party, it would predict a party that is similar in bias to the actual party. If the model predicts the wrong party and the predicted party is on the opposite end of the political spectrum as the actual party it would seem as if the model has not picked up on the bias of the speaker but rather something else.

Data resources

The collected data was downloaded from the Swedish parliament (https://data.riksdagen.se/data/anforanden/) which is a collection of over 300,000 speeches stretching back to 1994.

All of the speeches are held in the parliament by members of the parliament (349 at any given time) or ministers. Each speech is either an introductory speech or a reply to another speech.

The speeches are annotated with various information like name of speaker, party of speaker, year the speech was held etc. Only the actual text of the speech and the party of the speaker was used.

The data is in several formats but the data needed for this project was in xml format. A function was made to parse the xml files to get the information wanted and put it in a CSV file.

Methods

Preprocessing

There were some issues with the data for this particular task. The speeches are labeled with an abbreviation of the name of the speaker's party but some parties have changed names over the years, KDS (kristdemokratiska samlingspartiet) changed name to KD(kristdemokraterna) and FP(Folkpartiet) to L(Liberalerna). Party names were lowercased and speeches with no party label were removed.

Also, some of the speakers did not belong to parties. To remedy this every row of the CSV file that did not belong to a party was removed, and the party labels of the parties that had changed names of the years were unified into one single label for each party.

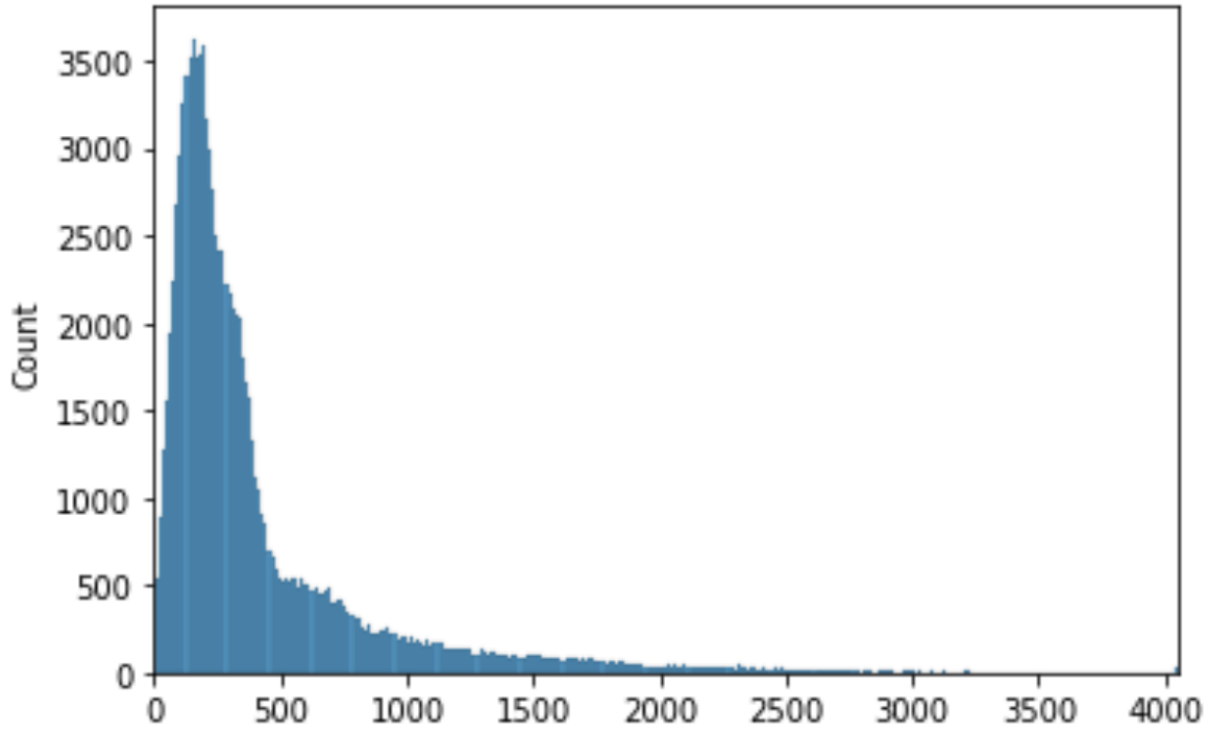
Another issue with the data was the uneven distribution of how many times each party had spoken in the parliament. The social democrats who had been the biggest party in Sweden for many of the years this data was recorded, had over 100,000 speeches while a small party that was only in parliament for 4 years only had about 1000 speeches.

This issue was partly fixed when only using data from 2010 since there have been no new parties in the parliament except for SD since then, so the data distribution is more even, some truncation was used to further even it out.

A simple solution that resulted in a bit of data loss was to first remove the party with only 1000 speeches. Then to reduce the maximum number of allowed speeches per party to about double that of the now smallest number of speeches in order to even the distribution.

The speeches also varied quite a lot in length, a few were over 2500 words long while most were around 250 words.

(Image was taken from sample of the text with individual sequence lengths)



The data was split up into 80% training and 20% testing. The data was also stratified according to parties so that training and testing both contain a roughly equal proportion of each party. The training and testing split is then batched using pytorch dataloaders, where they are also shuffled. Some different batch sizes were tried but a batch size of 16 was chosen.

In order to keep the training and testing data separate and to avoid having to run the entire program again the data was saved in files that can later be loaded.

The general idea is to input the preprocessed data in Swedish BERT, then create another model that fine tunes it by linear layers ending in 8 classes representing the different parties.

Bert

There is a multilingual BERT that could be used on swedish text, but since 2020 there is also a swedish BERT(kb-bert) trained on a large amount of swedish text (Malmsten et al 2020) that outperforms the multilingual BERT, so kb-bert was chosen. Kb-bert is of type bert-base with 12 transformers and 12 bidirectional attention heads in contrast with bert-large with 24 transformers and 16 bidirectional attention heads.

Kb-bert (https://huggingface.co/KB/bert-base-swedish-cased) is trained on about 200m sentences from a variety of swedish text sources taken from 1940-2012. There are also a couple of different fine-tuned kb-bert available for more specific purposes like NER (named entity recognition). These fine-tuned models were not used, the fine-tuning was done in a separate model made from scratch.

Using the huggingface library (https://huggingface.co/) each sentence that was kept from the CSV file was tokenized and a CLS token was added to the front of each sequence and a SEP token added to each end of speech.

Regardless of the length of each sequence they were truncated to the maximum sentence length allowed by bert (512 tokens) with padding for the shorter sequences and a removal of any word after a length of 512. This caused some loss of data, a possible solution could have been to split the sequences up before they were truncated and thus use all of the data.

The tokens from each sentence then get token ID and are used as input into kb-bert.

This type of BERT model outputs embeddings for each individual token and for the CLS token (the first token of each sentence). The CLS token can function as a general embedding representing the entire sequence of words for that sentence instead of any one individual word and is thus well suited for this sequence classification task.

Fine tuning model
Another model was then created consisting of three linear layers with relu and dropout in between the layers to prevent overfitting. Finally the last linear layer ends in 8 classes with a sigmoid layer applied to each layer in order to normalize the scores between 0 and 1.

This model takes the embedding from kb-bert while the parameters in kb bert are frozen to prevent training them.

Kb-bert outputs a 768 size vector for the CLS token for each sentence. Each CLS token is used as input into this newly created model and used to predict the party of each speech.

The true labels are one hot encoded and binary cross entropy loss is calculated between each prediction and label. Adam is the optimizer with a learning rate of 0,001.

Results

The final result for using all of the data is 48 % accuracy on this 8 class classification problem. If the data were completely evenly distributed and the model simply guessed for each class the accuracy would be around 12.5%, since the data is not exactly evenly distributed the accuracy would probably be slightly higher since one class (SD) is lower than the others. Regardless 47.5% clearly shows that the model was able to distinguish between the parties.

The result on the data from 2010 was better at 52.5% accuracy while using less data than before, indicating that a lot of the previous data seemed to confuse the model.

Both of these results can be compared with a binary classification task that was done in the US congressional debates (Simoes & Castanos 2020) which used a lot less data (4062 sentences) but got up to around 67% accuracy.

Analysis

The confusion matrix seems to indicate that what party got confused with what other party is primarily based on the different themes talked about by the parties rather than the political bias.

In a sense, the theme one talks about might also indicate a political bias, but the result indicates that parties that are on the opposite end of the political spectrum from each other frequently got confused with each other although their political bias is generally considered to be opposite of each other. For example: MP often gets confused with SD and vice versa although they are on the opposite end of many issues, the main one being immigration.

If there is a speech containing immigration, parties with a higher political stake in that subject might presumably be more active and thus a proportion of their speeches could be higher of a certain theme which the model could then base its predictions on.

This problem could perhaps be remedied by some type of sentiment analysis on the different topics.

All of the data

Predicted / Actual	(c,)	(fp,)	(kd,)	(m,)	(mp,)	(s,)	(sd,)	(v,)
(c,)	2057	378	601	449	550	789	27	815
(fp,)	214	1892	131	497	350	347	70	298
(kd,)	877	284	2912	455	607	386	2	516
(m,)	494	798	333	2115	549	646	85	685
(mp,)	136	288	81	187	1512	364	34	294
(s,)	162	150	70	170	348	1260	37	300
(sd,)	25	171	1	63	40	86	1907	45
(v,)	231	236	67	260	240	319	34	1243

When using all of the data, the model had a very high accuracy on the Swedish democrats (SD) specifically. The SD are in a sense a political outlier in the swedish parliament in that they have not been in the parliament as long as the other parties (since 2010), they are more traditionally conservative and anti-immigration than the other parties, and they are not part of any of the two political coalitions in the parliament

This high accuracy might have indicated something interesting related to the Swedish democratic party if it were not for the fact that the accuracy is much more evenly distributed over the different classes when only using data from 2010.
It seems as if the reason that the model predicted SD so well compared to the other classes was based on either the topics they talked about (usually immigration) or something related to more modern speech, since all data from the SD in the parliament is from 2010.

Another observation is that V is fairly close on the political spectrum to S and S was the second most common prediction for V, which would seem reasonable if the model picked up on political bias, but then the second most common prediction for V is M which is pretty far from V on the political spectrum. So the model probably did not pick up very well on the political bias of speakers from V.

Data after 2010

Predicted / Actual	(c,)	(fp,)	(kd,)	(m,)	(mp,)	(s,)	(sd,)	(v,)
(c,)	555	28	119	59	30	61	69	86
(fp,)	108	1414	146	211	246	77	285	106
(kd,)	103	32	705	108	59	20	58	97
(m,)	227	306	276	1499	108	88	117	274
(mp,)	202	221	322	79	1425	160	389	189
(s,)	171	38	188	45	153	1106	205	208
(sd,)	92	100	89	31	123	94	876	100
(v,)	196	114	172	221	109	93	124	807

The model predicted slightly better using only the data from 2010-2021 even though the training data was a lot less. Indicating that political speech changes quite a lot over time and the additional data seemed to confuse the model more than it improved it.

The data that stretches back to 1994 presumably contains a different distribution of topics, more manners of speaking and various political changes both in parties themselves and in Swedish politics in general, which made the task of predicting political parties more difficult. The predictions for each party are more evenly balanced when using the recent data.

The model had lower accuracy on SD than previously and higher accuracy on several of the other parties.

Conclusion

It seems like the models predictions are based a lot on what topics speakers from different parties usually talk about in the parliament. It is also possible that the model is picking up on

some combination of political bias and the themes of the speeches but without additional evaluation of what the model is actually basing its predictions on it is difficult to interpret the results further than this.

What is clear however is that Kb-bert managed to find patterns in the data where it was able to predict fairly well on this specific task although it is not clear what those patterns are.

A conclusion to draw fairly confidently is that for kb-bert to be able to do this on the full dataset there have to be some persistent identifiable characteristics of speakers from different parties over time.

Another conclusion is that political speeches over a shorter period of time are easier for the model to predict. Perhaps because the intra-party change within a shorter time frame is smaller than over a longer period of time, so that for example the social democrats from 2010-2020 is more like the same party than from 1994-2020 and is thus easier to predict.

Reference list

Sapiro-Gheiler, E. (2018). "Read My Lips": Using Automatic Text Analysis to Classify Politicians by Party and Ideology. *ArXiv, abs/1809.00741*.

David, Esther & Zhitomirsky-Geffet, Maayan & Koppel, Moshe & Uzan, Hodaya. (2016). Utilizing Facebook pages of the political parties to automatically predict the political orientation of Facebook users. Online Information Review. 40. 610-623. 10.1108/OIR-09-2015-0308.

Golbeck, Jennifer & Hansen, Derek. (2011). Computing Political Preference among Twitter Followers. Social Networks. 36. 1105-1108. 10.1145/1978942.1979106.

Gangula, Rama Rohit & Duggenpudi, Suma & Mamidi, Radhika. (2019). Detecting Political Bias in News Articles Using Headline Attention. 77-84. 10.18653/v1/W19-4809.

Abercrombie, Gavin, Federico Nanni, Riza Theresa Batista-Navarro and Simone Paolo Ponzetto. “Policy Preference Detection in Parliamentary Debate Motions.” CoNLL (2019)

Ulinskaite, J., & Pukelis, L. (2021). Identifying Populist Paragraphs in Text: A machine-learning approach. ArXiv, abs/2106.03161.

Malmsten, Martin & Börjeson, Love & Haffenden, Chris. (2020). Playing with Words at the National Library of Sweden -- Making a Swedish BERT.

Simoes, A. and M. Castanos. “Fine-Tuned BERT for the Detection of Political Ideology.” (2020).