Identifying political affiliation from Parliamentary Speeches
Machine Learning for Natural Language Processing 2020

Janine EGUIENTA
ENSAE
janine.eguienta@ensae.fr

Sarah OULKADI
ENSAE
sarah.oulkadi@ensae.fr

Abstract

This work is aimed at predicting the political affiliation of a speaker given his/her speech. We implement Bi-LSTM and HAN models, which are both commonly used for document-level classification purposes. Full and filtered versions of the speeches according to their emotional content are considered. Compared to baseline models, we found out that Bi-LSTM was performing the best, reaching a 68.7% accuracy. The HAN model ranked second and was particularly useful to highlight the relevant information for prediction.

1 Problem Framing

In a speech, political affiliation can be expressed in many subtle ways. Beyond the rhetoric, the lexical field and the length inherent to each political speech, we wish to predict the underlying political affiliation of a speaker.
We are thus dealing with a text classification task, which is one of the most common tasks in NLP.
However, document-level classification is still a research topic today. Recent approaches based on Deep Learning have been developped in order to better reflect document composition. Indeed several challenges exist when working at the document-level : the varying length of the documents, the need to take into account long-term dependencies and the fact that not all the words and sentences convey useful information for the prediction. We thus choose to investigate two types of neural networks models that can deal with these issues : Bi-LSTM (Bidirectional Long short-term memory) and Hierarchical Attention Network (HAN) models.

2 Experiments Protocol

Data

Data was collected from LiPaD[1] which gathers all Canada's parliamentary debates since the 1880's (Rheault and Cochrane, 2019). We selected speeches that were made over the last 6 years and kept only the ones of the 3 most represented political parties : Conservative, Liberal and New Democratic parties. The speeches we use include Statements by Members, Emergency Debates, Routine Proceedings and Government Orders since they are the ones that should convey the more political bias. After removing speeches with less than 50 words and with more than 1,500 words, we ended up with a balanced corpus of 75,844 speeches, which is described in Table 1.
We propose to also study a filtered version of the speeches : for each speech, the 5 sentences with the most emotional content are kept. Emotional scores of sentences were built using `TextBlob`. The corpus was randomly split into training (0.8) and test (0.2) sets. For Bi-LSTM and HAN models, the test set was split in half in order to get a validation set.

Political party	Number of speeches	Average Nb. of sentences	Average Nb. of words
Conservative	28,705	12.2	261.1
Liberal	26,092	10.8	210.2
New Democratic Party	21,047	12.5	263.4

Table 1: Corpus statistics (Averages per speech)

Modelling

We choose to compare Bi-LSTM and HAN models to different baseline models on the two data configurations.

Naive Bayes and SVM classifiers were trained on different text representations : Bag Of Words (BoW) features, TF-IDF features and Doc2Vec embeddings.

2-layer Bi-LSTM: This RNN architecture which overcomes the gradient vanishing problem is

[1] https://www.lipad.ca/data/

known for capturing long term dependencies. We use its bidirectional version to process documents in both directions and retrieve more contextual information. The two layers are made up of 2 Bi-LSTM stacked on top of each other.

HAN model : This model is based on a bidirectional GRU (Gated Recurrent Unit) at the word level followed by an attention model which assigns weights to the most relevant words for the meaning of a sentence. The representation of these informative words is aggregated to form a sentence vector. Using a similar procedure on the sentence vectors allows to generate a document vector that summarizes the content of a given document. This document vector is then used as an input for text classification (Iyyer et al., 2014). By providing a way for capturing the compositional structure of a document while focusing on the most relevant information for prediction, this hierarchical model seemed very appropriate for our classification task.

Implementation

The implementation framework we used for training deep learning models is the TorchText package from Pytorch, which is very convenient for both pre-processing and training. Our code is available online[2]. The implementation of the models was inspired from several online sources[3].

Model training for Bi-LSTM and HAN

The word embedding matrix is initialized with pre-trained GloVe vectors of dimension 100 (after trying different dimensions). The batch size is set to 64 for Bi-LSTM and 32 for HAN (64 creating a problem of memory). Documents of similar lengths (in terms of words for the Bi-LSTM and in terms of sentences for the HAN) are bucketted in a same batch. The dropout probability at each RNN step is fixed to 0.2. For both models, the Adam optimizer and the Cross Entropy Loss are used.

3 Results

Quantitative Evaluation

Model performances are compared in terms of accuracy and weighted version of the F1-score. Results are provided in Table 2.

[2]https://github.com/jeguienta/ProjetNLP_ENSAE
[3](Github, b), (Github, c), (Github, a)
[4]Document Bag of Words

Data	Non filtered		Filtered	
Score	Accuracy	F1	Accuracy	F1
BoW + NB	58.0	57.9	36.4	36.4
BoW + SVM	61.3	61.2	36.1	36.1
Tf-Idf + NB	59.0	58.5	38.0	34.4
Tf-Idf + SVM	61.5	61.3	38.2	37.7
DBoW[4]+ NB	36.7	36.8	27.8	12.2
DBoW + SVM	32.7	25.8	27.7	12.1
Bi-LSTM	68.7	68.6	68.1	67.9
HAN	62.6	62.6	59.0	58.8

Table 2: Performance comparison (in%)

The first thing we notice is that Bi-LSTM and HAN models perform better than baseline models. The highest accuracy and F1-score, which are respectively of 68.7% and 68.6%, are reached when using the Bi-LSTM model. We surprinsingly notice that there is very little gain using HAN compared to using a SVM classfier on simpler documents features.
Filtering beforehand the speeches according to their sentence emotional score drastically worsens the performances of baseline models. However, when using Bi-LSTM and HAN models on filtered speeches, resulting accuracy and F1-score are very similar to when keeping complete speeches. Indeed, it would also have been interesting to score sentences of a speech based on their political content instead of only using an emotional score.

Qualitative analysis

In our work, we analysed the attention weights provided by the HAN model. Figure 1 shows an example of a well-classified speech for the Conservative party. The score for each sentence is provided and words are colored from yellow to red according to their weight. The most important words include *terrorist*, *government* and *extremist*. This shows the ability of the HAN model to single out the relevant information in a speech.

4 Discussion/Conclusion

The Bi-LSTM turned out to be the most performant model for our document-level classification task. Performances of the HAN model were surprisingly disappointing but implementing this model allowed us through different visualizations to tag at the most relevant information for classifying a political speech. In order to go further in the analysis of the results, we are thinking of generating speeches from the outputted predictions to be able to compare models between them in a more qualitative way.

Appendix

```
True label: Conservative,
Predicted label: Conservative
Score: 0.984565019607544
Word attention
0.1 Mr Speaker in the dark and dangerous world we live
 in it is important that any responsible government ta
kes steps to keep Canadians safe
0.12 That is why our Conservative government has made i
t a criminal offence to go overseas to engage in terr
orism
0.12 We have also taken steps to be able to strip the
 citizenship of those convicted of terrorist offences
0.11 Despite this the Liberal leader has a different ap
proach
0.11 He says that the Boston bombing was caused by a
feeling of exclusion
0.15 The senior Liberal member for Kingston and the Isl
ands says that Liberals see a light and beauty inside
 every person specifically citing a terrorist who behead
ed three western journalists
0.15 Most shockingly the Liberal leader goes shopping fo
r votes in an extremist mosque in Montreal and has th
e temerity to expect the government to tell him he sh
ould not associate with those who condone the subjugati
on of women
0.14 Our Conservative government will stand up for law
abiding Canadians against radical extremists
```

Figure 1: Visualization of attention weights for the best predicted Conservative speech.
Word weight has been normalized by the sentence weight. We observe that sentences have varying weights. This speech is more related with security issues.

```
True label: New Democratic Party,
Predicted label: Conservative
Score: 0.969073
Word attention
0.2 Madam Speaker my riding is Kootenay Columbia which
is located in British Columbia
0.2 We have had a carbon tax in place there for many
 years
0.19 What we have been hearing today from our friends
in the Conservative Party is that somehow this carbon
tax is going to drive people into poverty
0.19 I wonder if the hon
0.22 member could talk about what the impacts of the f
ederal carbon tax might be on British Columbians and w
hether it is going to drive up prices everywhere and
drive everyone into poverty
```

Figure 2: Visualization of attention weights for a misclassified example.
Word weight has been normalized by the sentence weight. Here, the important words are *carbon* and *tax* which indeed also appear in a lot of correctly classified conservative speeches. The model also considers the word *hon*, which is indeed a stopword, to be important.

References

Github. a. Example of torchtex han.

Github. b. Hierarchical attention network.

Github. c. A pytorch tutorial to text classification.

Mohit Iyyer, Peter Enns, Jordan Boyd-Graber, and Philip Resnik. 2014. Hierarchical attention networks for document classification. *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Ludovic Rheault and Christopher Cochrane. 2019. Word embeddings for the analysis of ideological placement in parliamentary corpora. *Political Analysis*.