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

Instructions (to remove):

1. write and present clearly and synthetically your project
2. Add all the references in Appendix (use .bib (author, 2018))
3. 1 page maximum
4. You can add any non-textual content (plots, table, images, schemes...)
5. Should point to your notebook (colab or github/gitlab (e.g. with footnote [1]))

Abstract

Quick Introduction, main point of your work, summary of experiments and results

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 identify/predict/recognize the underlying political affiliation of a speaker in an agnostic way. In order to do this, we propose to implement and compare two sequence labelling models : bi-LSTM and HAN, which are both commonly used for document-level classification purposes. We found out that ...

1 Problem Framing

Our task can be formulated as follows : we have at our disposal political speeches. Each speech is a sequence of words of varying length we want to classify the political affiliation of the speaker. We are thus dealing with a text classification task, which is one of the most common task in NLP
However, document-level classification is still a research topic today (Rhanoui et al., 2019). For a long time, it has consisted in applying machine learning classifiers on some sparse features such as bag-of-words and n-grams. More recently, approaches based on Deep Learning were 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 bring/convey useful information for the prediction. We thus chose to mainly focus our attention on two types of sequence models that overcome these limits/can deal with these issues : LSTM and HAN models, following the approaches of respectively (Rao et al., 2018) and (Iyyer et al., 2014)

- document classification
- long documents
- memory content based
- still studied nowadays
- hierarchical models
- varying length
- détection de l'affiliation politique dans un discours

2 Experiments Protocol

Data (train and evaluation)
Data was collected from LiPaD[2] 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/associated to the 3 most

[1] https://nlp-ensae.github.io/

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

represented political parties : Conservative, Liberal and New Democratic parties in order to create a balanced dataset. 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. These represent 75,844 speeches in total. The corpus of speeches is described in Table 1.
The dataset was randomly split into training (0.7), validation (0.2) and test (0.1) sets. (attention pas de validation pour les modeles baselines).

Political party	Nb. speeches	Avg Nb sentences	Avg speech length (words)
Conservative	28,705	12.2	261.1
Liberal	26,092	10.8	210.2
New Democrat Party	21,047	12.5	263.4

Table 1: Corpus statistics

Models used We propose to compare neural network methods to some common baseline models.
Baseline models We trained Gaussian Naive Bayes and SVM classifiers on different text representations :
- Bag Of Words (BoW) features;
- TF-IDF features;
- Doc2Vec embeddings.

Sequence models
2-layer Bi-LSTM
A form of RNN architecture that takes into account context information of sequences and that overcomes the gradient vanishing problem. In the bidirect architexture, models can look forward and backward to capture features in modeling long texts. It is deemed to better take into account semantic composition say a word about computational resources
HAN model
Hierarchical structure that that mirrors the hierarchical structure of documents. It has two levels of attention mechanisms applied at the word and sentence level, enabling it to attend differentially to more and less important content when constructing the document representation. It uses a GRU (gated recurrent unit) encoder for word and sentence vectors.

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[3]. It is self-contained and results should be reproducible. We got inspired from two main sources to implement our models[4]

- torchtext/pytorch
- cleaning steps
- padding
- bucketting

Model training We herebelow choose to only present the way we trained neural networks models since they are at the core of our study/work. Documents were all pre-processed in the same way : removal of stop-words, numbers and punctuation, lemmatization and transformation to lower case. For the HAN model, speeches were first split into sentences and then each of them was tokenized. In order to construct the vocabulary, tokens appearing less than 5 times were removed and we chose to initialize the word embedding matrix with pre-trained GloVe vectors of dimension 100 after trying different dimensions. We set the size of batches to 64 and we used a bucketting strategy in order for each batch to be composed of documents of same lengths (in terms of words for the LSTM and in terms of sentences for the HAN). We used Adam optimizer to update weights and the Cross Entropy Loss for training the two models. We run 5 epoch for each model. DROPOUT
Préciser séparemment tous les choix de paramètres pour lstm + han et parler de GRU

- tokenization
- bucketting
- padding
- hyperparemeters choices
- nb batch
- already pre trained embeddings glove

[3] ppp
[4] For Torchtext implementation : https://github.com/larry0123du/torchtext-han-example/blob/master/HAN.ipynb?fbclid=IwAR2AH4RMSlQ4Lu913qd0SjSrFajU2GRe1ijwKSI2V7Pztd_8alxQ5Cl0KLw, for the implementation of (Iyyer et al., 2014) (https://github.com/sharkmir1/Hierarchical-Attention-Network) and https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Text-Classification

- description parameters LSTM 2 layers
- description parameters HAN
- losses
- optimizers etc epoch

3 Results

Quantitative Evaluation Performances are compared in terms of accuracy and weighted version of the F1-score. Results are provided in Table 2.
1. compare baseline models between them
Qualitative analysis of sequence models results

- quanti
- quali

4 Discussion/Conclusion

our approach..; for future work, text generation

References

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*.

Guozheng Rao, Weihang Huang, Zhiyong Feng, and Qiong Cong. 2018. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*.

Jacob author. 2018. Some nlp model. *arXiv preprint arXiv:1810.04805*.

Maryem Rhanoui, Mounia Mikram, Siham Yousfi, and Soukaina Barzali. 2019. A cnn-bilstm model for document-level sentiment analysis. *Machine learning and knowledge extraction*.

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