Parliamentary questions ministries

Lucian Istrati

Proposed approaches

The classes used from the dataset (most frequent 10%) were: 'Ministre de la Santé', 'Ministre de l Environnement', ' Ministre du Développement durable et des Infrastructures', ' Ministre de la Justice', 'Ministre des Finances', "Ministre de l'Education nationale", 'Ministre de la Sécurité sociale', "Ministre de l'Intérieur", 'Premier Ministre' .

From the data point of view I tried:

- Just textual content
- Textual content combined with other categorical features

From the model point of view I tried:

- Classical machine learning models: several sklearn models and xgboost
- Deep learning approaches: CamembertModel, FasttextModel

Classical Models

The tried models were: BernoulliNB, MultinomialNB, Perceptron, AdaBoostClassifier, RandomForestClassifier, DecisionTreeClassifier, XGBClassifier (with a CountVectorizer vectorization approach)

Several preprocessing techniques were tried (spacy “fr_core_news_sm” pipeline was used for all):

- Keeping and removing the French stopwords from this pipeline;
- Adding the part-of-speech of the word to the actual content of the word;
- Adding the dependency of the word (from the Dependency Parse Tree of the document) to the actual content of the word;

The categorical features I tried separately without the subject feature were: qp_type, answer_type and has_answer.

Deep Models

A Camembert model with Camembert Tokenizer (from the transformers library) was tried with a batch size of 8 and trained with the Pytorch API and an accuracy of 71% was obtained.

The Fasttext model obtained a 67% accuracy with the train_supervised() function after tuning the hyperparameters of the fasttext model with the autotuneValidationFile option.

Production deployment

- Inference should be done in a cloud environment in order for this project to be scalable;
- Possible candidates for this integration are platforms like Amazon Web Services or Microsoft Azure;
- In order to integrate it in cloud we would need to create an Anaconda Environment that contains all the necessary dependencies for this project with the compatible versions, to do that we could use Docker as it is platform independent with a given set of commands to run for installing and preparing everything such that the pipeline will be compliant for deployment.

Further works

- other word embeddings for French such as FlauBERT;
- other vectorization approaches such as tfidfvectorizer at char/word level;
- count vectorizer at character level;
- lexicon based approach with words that tend to be used quite frequently in questions addressed to certain ministries;
- other preprocessing techniques for the text that are more aggressive with stop-words.

Allocated time

The allocated time for this project was around 7 hours in total.