POSOS Data Challenge — Presentation

Hugo Cisneros

March 25th, 2019

Course *apprentissage par réseaux de neurones profonds* — Stéphane Mallat

Context

The task

Classify user-generated drug-related questions into 51 classes without information on the meaning of each class.

Overview

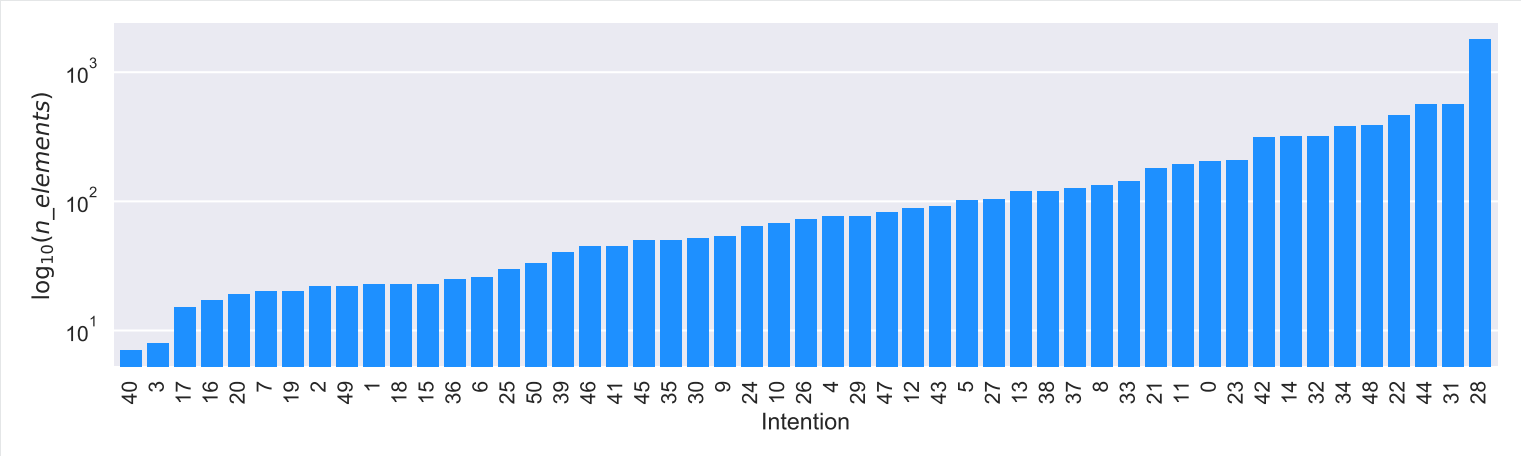
- Dataset analysis & text pre-processing
- Model selection
- Results and analysis

Dataset analysis

Characteristics

- 8028 training sentences, 2035 test sentences
- 9542 words before spell checking $\rightarrow$ 7321 after

Class repartition — training set

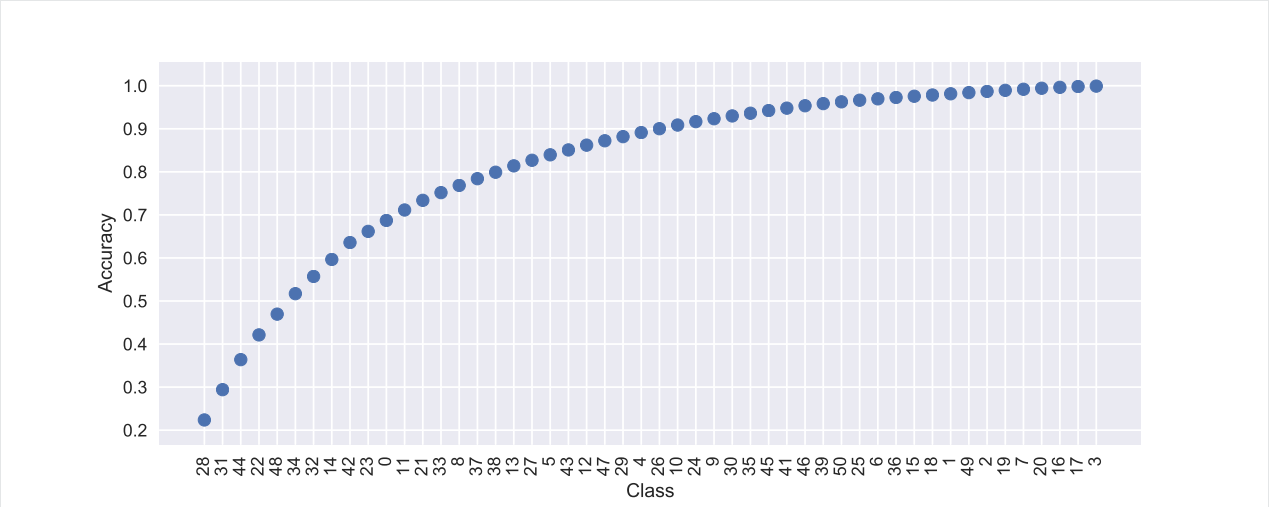


Dataset analysis

Characteristics

- 8028 training sentences, 2035 test sentences
- 9542 words before spell checking $\rightarrow$ 7321 after

Class repartition — training set



$\Rightarrow$ 50% accuracy with the first 5 classes only !

Text pre-processing

Deal with noisy inputs

High variability in text inputs:

- Accentuated or not `très`/`tres`
- Multiple punctuation signs `pilule mini dosée..`
- Abbreviations `ts`, `o`, `aret`
- Apostrophe `qu'y`, `l'`, etc.

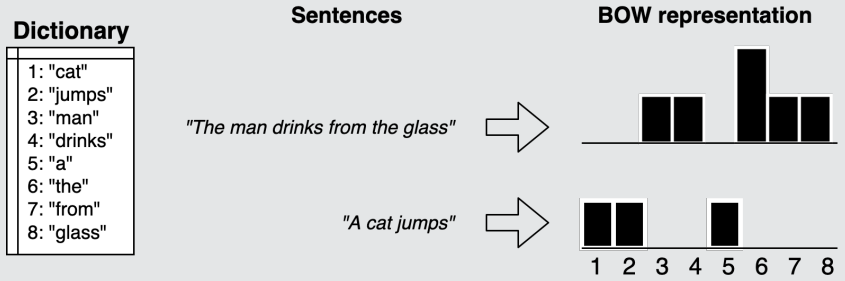
etc.

Deal with spelling errors

Spell checking algorithms with french word count dictionary and drug names scraped from Vidal.

Methods — Linear model

Start simple with linear models



Sentence representations:

- Bag of words
- Tf-Idf: weight words by frequency in sentence compared to corpus frequency

$$\text{tf-idf}(t, d) = f_{t,d} \cdot \log\left(\frac{N}{n_t}\right)$$

Methods — Linear model

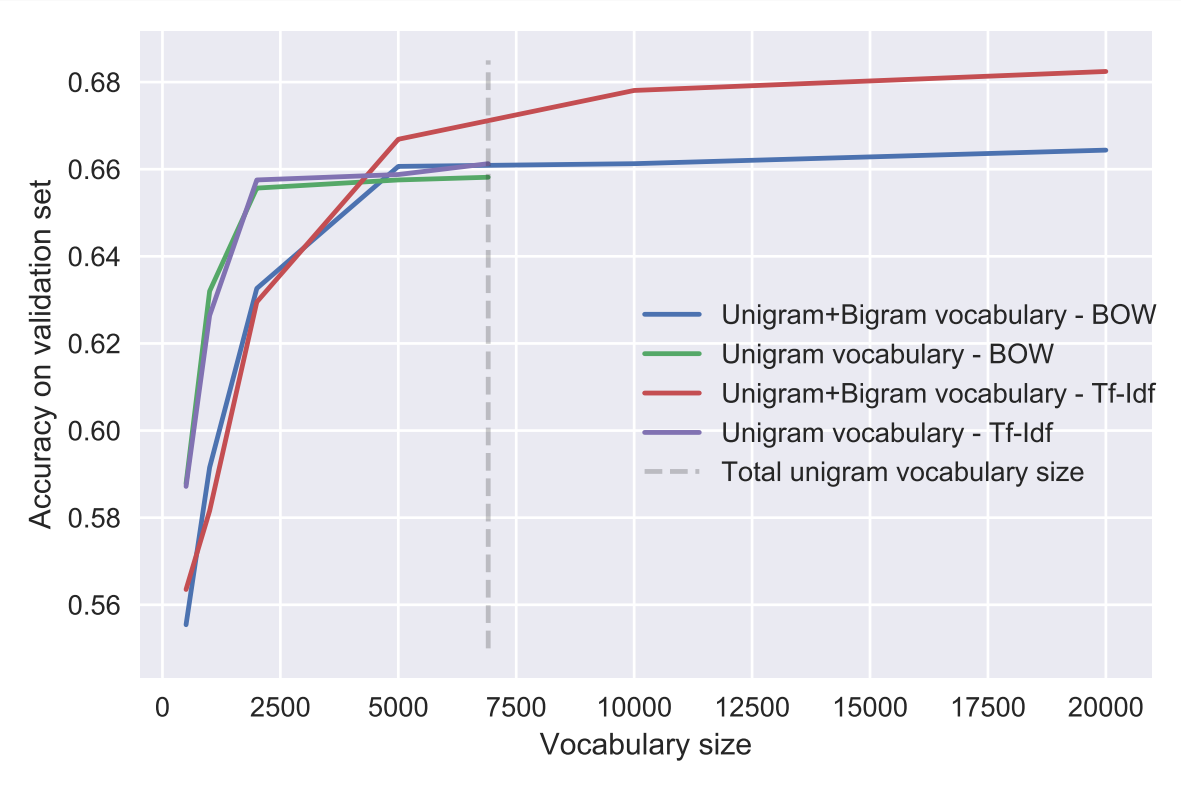
Logistic regression — Mathematical formulation:

$$\min_{w,c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1).$$

Hyper parameter tuning

4-fold cross validation on the training set (not more because some classes have only 5 examples) to choose the parameter C.

Results for linear model



Performance

Although the model is very simple. These are the best performance I got among all models.

Decision trees

Idea

Iteratively make splits that maximize a criterion.

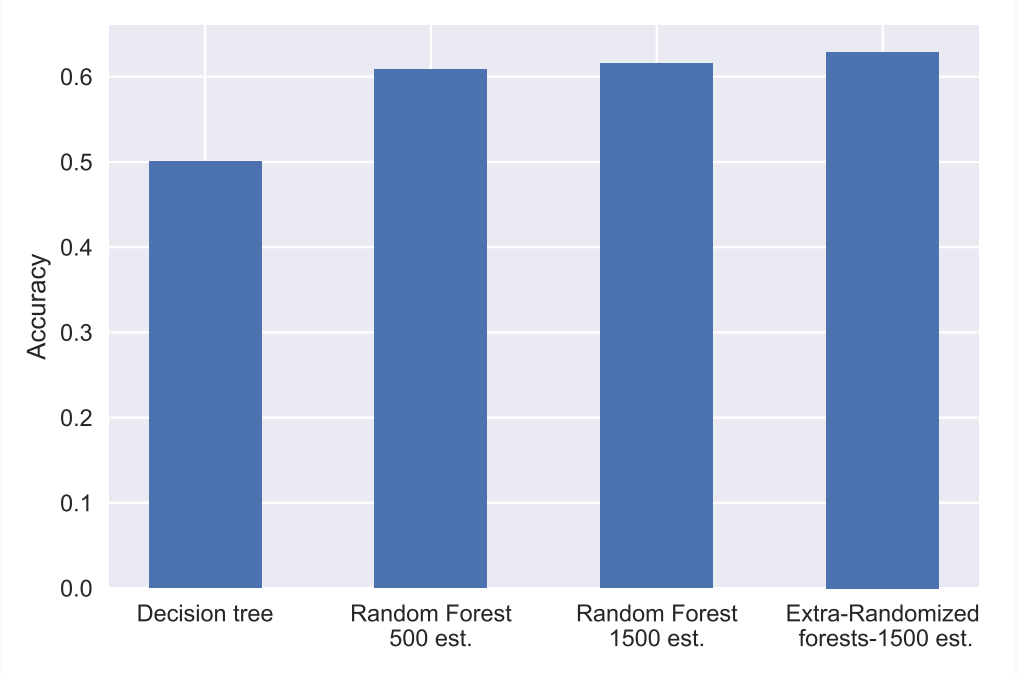
Random forests Build several estimators on random subsets of the features and average all results

Extra randomized forests Splitting thresholds are also drawn at random

The goal of both extension is to make the algorithm more generalizable.

The intuition for using decision tree is that classes could be defined by complex rules such as: "word A and B are present in conjunction with the absence of word C".

Decision trees — Results

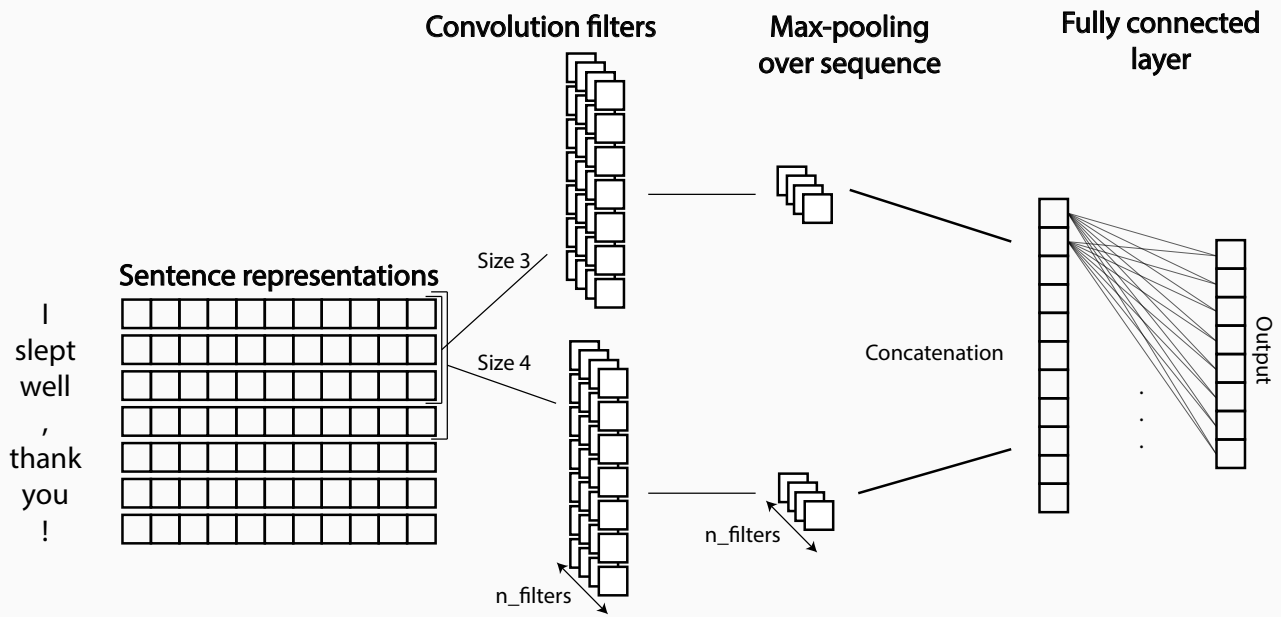


Relatively poor results compared to the logistic regression. And this even when tuning the parameters $\rightarrow$ something is missing to capture what defines a class.

Neural networks

Idea

Use the properties of convolutional networks (translation invariance) on the text. $\Rightarrow$ apply 1D convolutions on sentence inputs.



Architecture inspired from [ZW15] and the challenge's baseline model.

Word embeddings

Idea

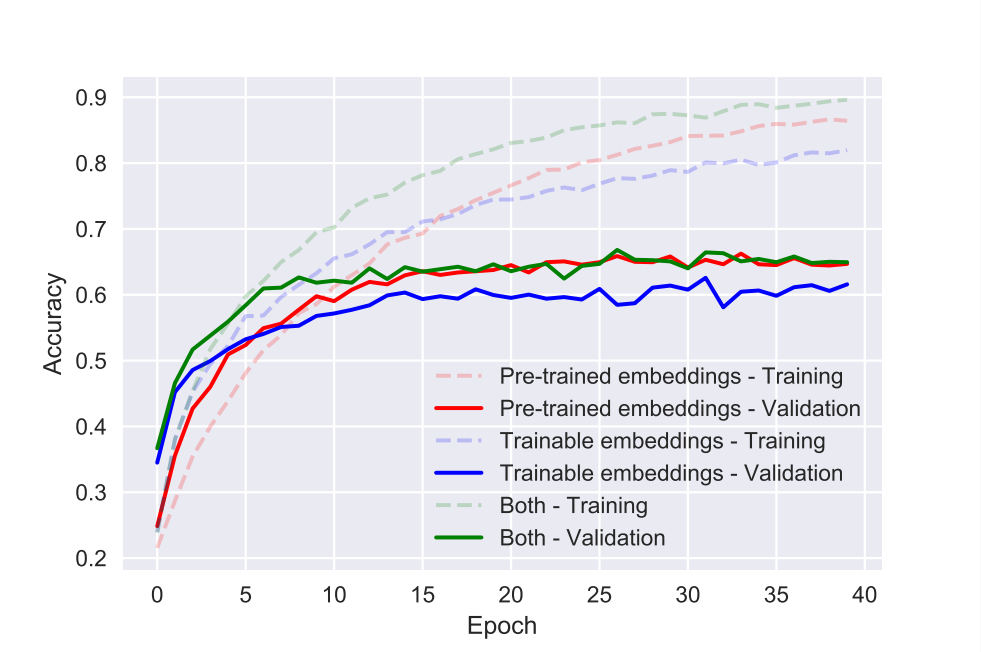
Learn a representation of words in a vector space with interesting structures.

The learned word representations are usually closer for similar words and have useful spatial structures that make them ideal for classification purposes.

FastText multi-lingual embeddings [Gra+18]

Pre-trained words embeddings trained with the CBOW model on a dump of the french Wikipedia.

Compare NN models



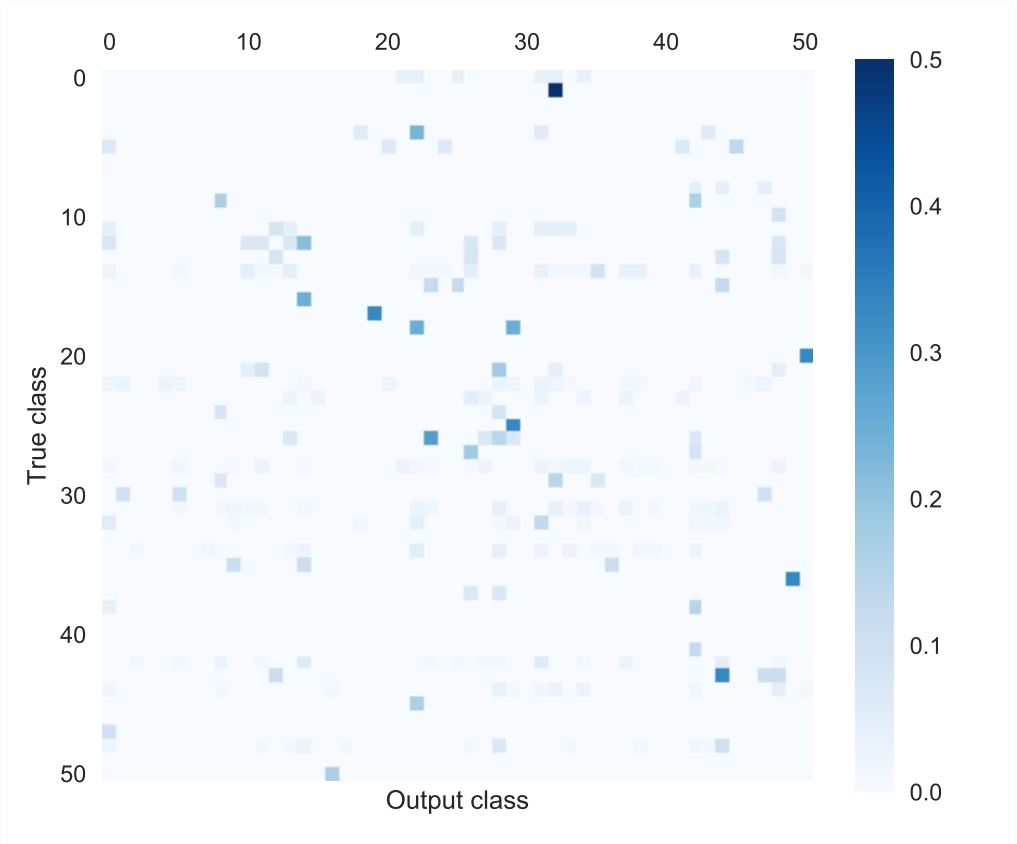
Results

Pre-trained embeddings certainly help with the performance but did not match the logistic regression.

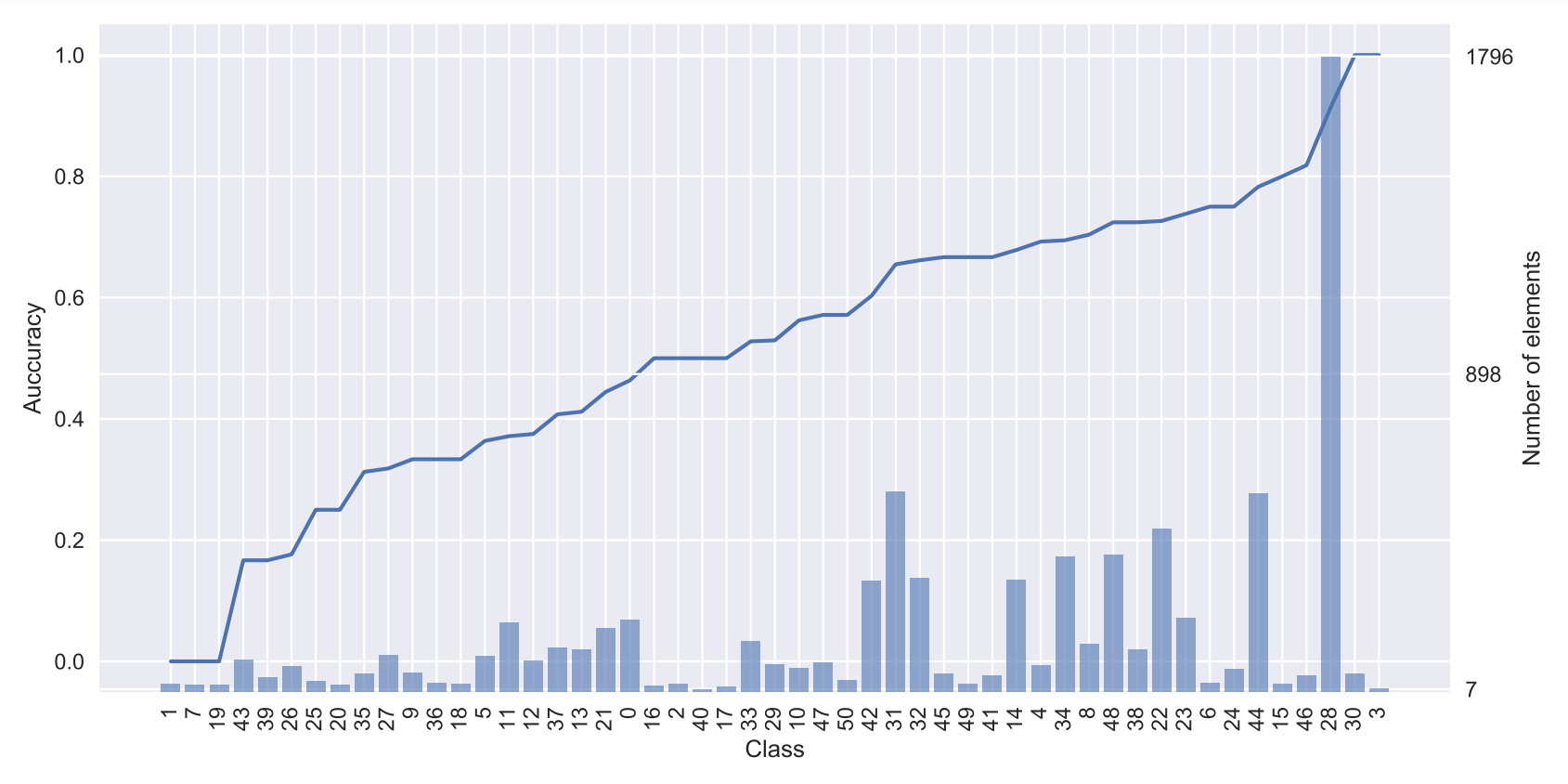
Possible reasons ?

- Word embeddings of too low quality ?
- Pre-processing ?

Error analysis and conclusion



Error analysis and conclusion



Possible improvements

- Better embeddings: use embeddings trained on medical NLP datasets (I couldn't find one in french).
- Use ensemble methods with one ore more of the models presented above (they might have complementary strength).
- With class information (either explicit class names of simple tf-idf on classes) one could get "easy wins" by designing specific rules when possible.

Thank you!

References

Ye Zhang and Byron Wallace. "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification". In: *arXiv:1510.03820 [cs]* (Oct. 13, 2015). arXiv: 1510.03820. URL: http://arxiv.org/abs/1510.03820 (visited on 03/22/2019).

References II

Edouard Grave et al. “Learning Word Vectors for 157 Languages”. In: *arXiv:1802.06893 [cs]* (Feb. 19, 2018). arXiv: 1802.06893. URL: http://arxiv.org/abs/1802.06893 (visited on 01/17/2019).