How to classify science paper.
Machine Learning for Natural Language Processing 2022

Cecile Boulangeot
cecile.boulangeot@ensae.fr

Julie Sixou
julie.sixou@ensae.fr

Abstract

This project aims to facilitate the taxonomy of research articles by automating the classification of papers according to the following categories: 'Computer Science', 'Physics', 'Mathematics', 'Statistics', 'Quantitative Biology' and 'Quantitative Finance'. The originality of this dataset, obtained on kaggle, is its strong imbalance as well as the amount of information available, we provide titles and abstracts of the papers.

In this project, we will first perform a naive model that computes a score for each article based on the differences in word occurrence between the study topics. We will then try to beat this score using machine learning and BERT deeplearning models. We will pay particular attention to managing the imbalance between classes.

1 Problem Framing

The data consists of 15972 articles and the following features :

- **TITLE** the title of the article
- **ABSTRACT** its abstract
- **label** the class of the article already vectorized from 0 to 6.

The data is clearly unbalanced with the most present class containing 6539 elements (40%). while the last only contains 161 (1%).

2 Experiments Protocol

2.1 Pre-processing

Being a research article, the dataset is very clean. The preprocessing was mainly focused on formatting (punctuation, capitalization), removing numbers as well as stopword and tokenization. The dataset has been separated in train/test database in order to compare the different models between them.

Unfortunately, for the sake of efficiency, abstracts have not been used in the creation of machine learning algorithms.

2.2 Techniques and models and training

2.2.1 naive model

Following descriptive statistics we noticed that the occurrence of the words used differed according to the categories considered.
Our first idea was to build the following algorithm:

Algorithm 1: Occurence Classification

Result: Classification of a corpus
initialization;
for *each class* **do**
  Calculate of the proportion of each words. If no word set proportion at 0.
end
algorithm;
for *each class* **do**
  Calculate the score of the given text by summing the proportion associate of each words in the text.
end
Get the label associate with the maximum score.

2.2.2 Machine learning

The score obtained allows us to have a basis to evaluate the following machine learning models. The first model considered was a classical SVM following a vectorization made with a CountVectorizer without trying to manage the imbalance of the classes. We then tried to correct this problem using optuna, a library that allows to optimize the hyperparameters by trying to maximize the f1 macro score - the latter considers that all classes have an equal importance. Finally, the last model

studied is a BERT deeplearning model for which we changed the cost function to take into account the imbalance of the classes.

3 Results

Table 1: Quantitative evaluation Linear SVM

Label	naive	SVM	BERT
Computer Science	0.70	0.76	0.73
Physics	0.79	0.80	0.84
Mathematics	0.70	0.70	0.72
Statistics	0.00	0.23	0.35
Quantitative Biology	0.07	0.25	0.48
Quantitative Finance	0.14	0.22	0.44

Our first model gives us a fairly good baseline for evaluating the usefulness of statistical learning. As expected we can see that the main classes are relatively well predicted. However, this method has a lot of trouble to find the less present classes. Moreover we can also note that *computer science* and *statistics* are often confused which lets us think that these two classes have a very close vocabulary.
This underrepresented class problem was also found with our different SVMs. The differences in the results between the models without finetuning and the hyperparameters aiming to correct the class imbalance were not obvious. However, it should be noted that the models learn better than with the naive method. Finally, BERT with our weighted score according to the class imbalance gives us the best results even if as we can see on the confusion matrix the precision remains limited for the 3 minority classes, notably because of an important confusion between the statistical classes and the computer science .

However, even if the gain is significant, it is important to note that Bert is computationaly more costly than the naive or the SVM model.

4 Discussion/Conclusion

The main avenue of research would be to work on the proximity between the computer science and statistics classes. This could be done in particular by using the abstracts that have been left out.

This raises the question of computational power. If we had more resources we could have worked more in depth on the abstracts which would certainly have allowed us to better discriminate between articles.

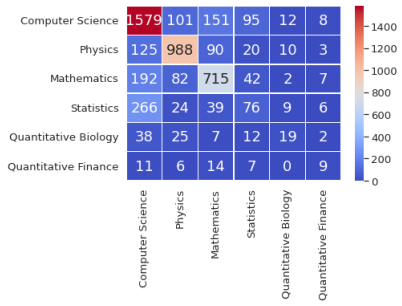

Figure 1: Classification outcomes with SVM

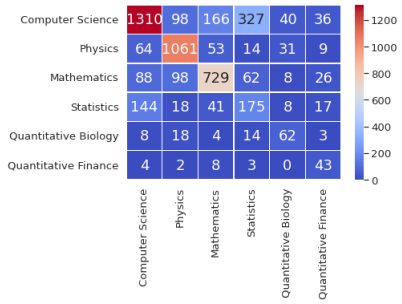

Figure 2: Classification outcomes with SV

However, this experiment is very promising and we have managed to highlight a real gradation of results between the models. BERT seems to be the most efficient model to solve this classification problem.

5 Links

github :
https://github.com/Cecileboul/NLP-classification-science-article.git
colab notebook :
https://colab.research.google.com/drive/11Cz-ApMFY9rU1U9piz4TnCcR5n-St-mx?usp=sharing