Classification de résumés d'articles scientifiques à partir de la classification des revues

Léo Gaillard, Lucas Anki, Pascal Cuxac

CLASSIFICATION DE RÉSUMÉS D'ARTICLES SCIENTIFIQUES À PARTIR DE LA CLASSIFICATION DES REVUES

Léo Gaillard, Lucas Anki & Pascal Cuxac

Inist-CNRS (UAR 76), 2 rue Jean-Zay, 54500 Vandœuvre-lès-Nancy, France
[prénom].[nom]@inist.fr

Contexte

Objectif : Construire un modèle de classification d'articles scientifiques en fonction de leur résumé. La classification utilisée est la classification en domaines scientifiques Science-Metrix[1].

Spécificité : Pour les données annotées, le label ne correspond pas à la classification de l'article, mais de la revue dont il est issu : les données ne sont pas correctement labellisées pour notre objectif.

Ressources utilisées : Istex, Science-Metrix[1] Faiss[2], Fasttext[3], xgboost[4].

Construction des données

Nous prenons 2,7 millions d'articles scientifiques (issus d'Istex), en sélectionnant ceux ayant un résumé pas trop court (plus de 100 caractères) et en anglais. Nous avons forcément des classes très déséquilibrées : nous avons retiré 11 classes sur les 174 du troisième niveau de la classification.

ISTEX

Initiative d'excellence en Information Scientifique et Technique

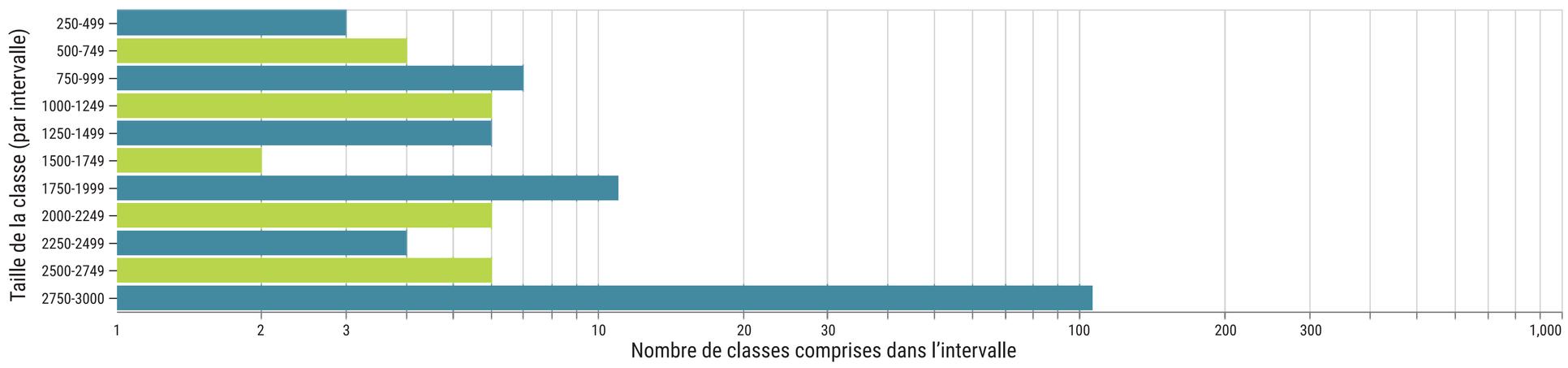
En chiffres

- **28 millions** de publications scientifiques
- **51** langues
- **47** corpus éditeurs
- **700** ans de publications

Son contenu

- Textes intégraux aux formats d'origine et standardisés (XML TEI)
- Textes nettoyés
- Métadonnées
- Enrichissements : entités nommées (Unitex), termes (Teeft), structuration du pdf (Grobid), domaines scientifiques (Nb, Multicat)

Répartition des tailles des classes des données d'entraînement



Discrimination des données non pertinentes

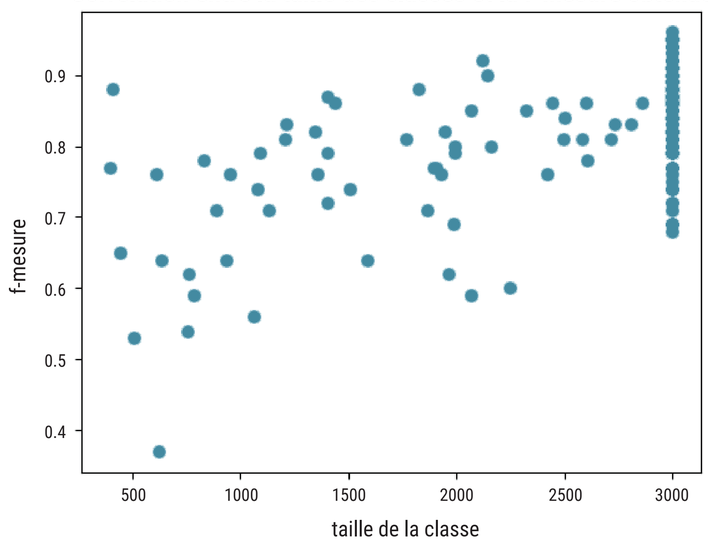
Nous sommes partis du postulat qu'une certaine proportion d'articles scientifiques avait la même classification que la revue de laquelle ils sont issus. Après vectorisation, l'ensemble des documents « correctement » labellisés devraient donc former un noyau assez dense. Les documents « mal » classés quant à eux seraient plus dispersés et éloignés du noyau (entre ce noyau et celui de la classe à laquelle ils doivent vraiment appartenir). Nous souhaitons repérer ces noyaux et sélectionner uniquement ces données pour entraîner notre modèle.

Algorithme

Pour sélectionner des données au sein des noyaux de chacune des classes, nous utilisons un algorithme de K-PPV : on calcule pour chaque document combien d'autres documents ont le même label que lui parmi ses k plus proches voisins. Les documents constituant le noyau seront ceux qui en ont le plus. Pour ce faire, nous utilisons BERT pour l'embedding de nos résumés et faiss[2] qui permet de faire un K-PPV optimisé sur les vecteurs obtenus.

Pour chaque classe c constituée de n_c documents, nous gardons **max(3000, $\lfloor p \times n_c \rfloor$)** (où $\lfloor . \rfloor$ est la fonction partie entière, et $0 \leq p \leq 1$ la proportion de documents à conserver pour constituer le jeu d'entraînement).

Corrélation entre la taille des classes et la f-mesure obtenue



Coefficient de corrélation de Kendall
0.41

p-valeur
$8.37e^{-12}$

L'accuracy gagnée en équilibrant les classes lors de l'entraînement n'est pas significative

Résultats

Pour évaluer cette méthode, nous créons trois jeux de données d'entraînement et de test :

- D1 est constitué en prenant **p = 0.3.**
- D2 est constitué en posant **p = 0.7.** Nous avons plus de données pour les classes de tailles petites et normales mais elles sont plus dispersées.
- D3 est constitué sans prendre en compte le classement des documents obtenus après KPPV et servira de jeu de données de référence lors de la comparaison des accuracies.

Pour fastText nous utilisons leur embedding par défaut pour l'anglais. Pour XGBoost, nous gardons les embeddings BERT utilisés pour les KPPV.

	fastText	XGBoost
D_3	0.58	0.53
D_2	0.76	0.74
D_1	**0.84**	**0.82**

accuracy obtenue en fonction des bibliothèques d'entraînement utilisées et de la proportion prise au moment du kppv

Sélectionner les données d'entraînement les plus pertinentes permet d'accroître significativement l'*accuracy* de notre modèle.

Ressources bibliographiques

[1] Science-Metrix , 500-4428 Boul. Saint Laurent, Montreal, QC H2W 1Z5, Canada (https://science-metrix.com/fr/)

[2] Johnson, J., M. Douze, et H. Jégou (2019). Billion-scale similarity search with GPUs. IEEE Transactions on Big Data 7(3), 535–547.

[3] A. Joulin, E. Grave, P. Bojanowski, T. Mikolov (2016), Bag of Tricks for Efficient Text Classification, https://doi.org/10.48550/arXiv.1607.01759

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

www.istex.fr

Inist

cnrs