Classification d'intervention parlementaires du Parlement Européen : une étude comparative des algorithmes de Random Forest et régression logistique

Léna Gaubert, Natacha Miniconi, Qinliang Qi

Abstract

In the context of multinomial classification of multilingual text data, specifically for European Parliament speeches, we present robust multilingual models with rather promising performances and offer leads for further research and trials. **Keywords:** multinomial classification, multilingualism, machine-learning

1 Introduction

L'utilisation d'algorithmes d'apprentissage automatique pour la classification de textes, en particulier dans les domaines du juridique, du médical, ou bien encore des finances, a suscité un intérêt croissant ces dernières années, de part les enjeux que posent les textes écrits dans ces disciplines (lexiques spécifiques aux domaines). En introduisant leur méthode de représentation vectorielle d'extraction de concepts de domaines (domain concept extraction) pour la classification de textes juridiques américains, Chen et al., prouvent que le traitement automatique de données textuelles dans un domaine spécifique nécessite l'optimisation du pré-traitement et de la vectorisation de celles-ci (H. Chen, 2021). L'association de leur nouvelle méthode de vectorisation à l'algorithme de Random Forest leur a permis par ailleurs de surpasser les performances obtenues par le modèle avec TF-IDF mais aussi celles de RNN et text-CNN obtenus avec plongements lexicaux. Dans la continuité de cette démarche, nous proposons d'étudier la tâche 3 proposée au Défi Fouille de Texte en 2009 (DEFT-09) dont l'objectif était de classer des interventions parlementaires tenues entre 1999 et 2004, en anglais, français et italien, selon les 5 partis auxquels elles étaient affiliées. Nous proposons de comparer les performances des algorithmes de Random Forests et régression logistique, d'abord avec vectorisation par TF-IDF. Bien qu'un dilemme subsiste quant à l'implémentation de modèles monolingues ou multilingues dans la gestion de corpus multilingues (G. Manias, 2023), nous avons préféré ici opter pour des modèles multilingues, robuste sur l'ensemble de notre corpus. Opter pour ces deux algorithmes dans le contexte de la classification des discours parlementaires datant de 2009 en français, anglais et italien, nous a semblé pertinent, puisque ceux-ci pallient aux contraintes imposées par les données textuelles (haute dimensionnalité, données éparses et bruitées...) et s'avèrent performants sur la classification de texte (K. Kowsari, 2019) ainsi que d'autres tâches de TAL. L'intégralité de notre travail est disponible sur Github [1].

2 Méthodologie

Le corpus multilingue dont nous disposons fut constitué à partir des débats du Parlement Européen tenues entre 1999 et 2004. Il comprend un total de 32 289 interventions parlementaires affiliés aux 5 partis politiques suivants : les verts/Alliance Libre Européenne (Verts-ALE), la Gauche Unitaire Européenne/gauche verte nordique (GUE-NGL), le Parti Socialiste Européen (PSE), le parti Européen des Libéraux, Démocrates et Réformateurs (ELDR), et le Parti Populaire Européen (PPE-DE). Les interventions ont été finalement mises au format XML, sous trois fichiers, un par langue (anglais, français, italien). Nous avons effectué le même filtrage, ainsi que la même méthode de vectorisation pour chacun des corpus. Une fois l'extraction des discours et des partis associés effectuée depuis les fichiers XML, nous n'avons pas filtré les interventions sur les mots vides. A nos yeux les mots outils grandements utilisés en politique ont une signification particulière, certains partis peuvent preferer un pronom en particulier ce qui pourrait le distinguer des autres partis. Nous avons vectorisé les discours finalement obtenus en appliquant la fonction TF-IDF de scikit-learn (TfidfVectorizer) : d'une part,elle tient compte de l'importance

[1] Github https://github.com/kittog/deft-2009.

d'un mot non seulement au sein d'un document spécifique, mais également dans l'ensemble du corpus. D'autre part, elle équilibre la fréquence d'un mot dans le document (Term Frequency) avec sa rareté globale dans le corpus (Inverse Document Frequency). Cette approche accorde une importance accrue aux termes qui apparaissent fréquemment dans un document particulier tout en étant rares dans l'ensemble du corpus, permettant ainsi de capturer des caractéristiques distinctives et informatives. Les corpus train et test ont comme forme finale un fichier csv comprenant le texte tokenizer, le parti, ainsi que son id. Le choix délibéré d'utiliser à la fois Random Forest et la Régression Logistique découle de la volonté d'explorer deux approches distinctes pour la classification de discours parlementaires multilingues. Ces deux algorithmes incarnent des paradigmes différents en termes de complexité et de nature des relations qu'ils peuvent apprendre. D'une part, Random Forest a été choisi pour sa capacité à modéliser des relations complexes et non linéaires. La robustesse de Random Forest face au déséquilibre des classes dans le corpus de discours parlementaires a également motivé son choix, car cette asymétrie est fréquente dans de telles tâches. D'autre part, la Régression Logistique a été sélectionnée pour sa simplicité et son interprétabilité. En tant que modèle linéaire généralisé, la Régression Logistique offre une compréhension claire de la contribution de chaque caractéristique à la prédiction des affiliations politiques. Cette transparence peut être cruciale dans le contexte politique, où l'explication des décisions des modèles est souvent exigée. Les résultats obtenus de ces deux approches permettront une évaluation comparative de leurs performances respectives dans le contexte spécifique de la classification de discours parlementaires multilingues. Cette analyse approfondie déterminera laquelle de ces approches est mieux adaptée à cette tâche particulière, offrant ainsi des perspectives sur la manière dont des modèles plus complexes ou plus simples peuvent répondre aux défis posés par l'analyse de discours politiques dans différentes langues.

3 Résultats

L'implémentation des algorithmes choisis fut réalisée à l'aide des fonctions de scikit-learn. Pour des questions de reproductibilité de nos résultats, nous utilisons la graine aléatoire (*random state*) 42. Les choix des hyperparamètres ont été faits de façon empirique à l'aide de la fonction GridSearch, tout en prenant en compte le déséquilibre présent dans nos données. En effet, sur l'ensemble du corpus train, on compte 20574 pour PPE-DE, 16320 pour PSE, 8064 GUE-NGL, 7128 ELDR. Le corpus test est également déséquilibré : 13713 PPE-DE, 10881 PSE, 5379 GUE-NGL, 4755 Verts-ALE et enfin 4017 ELDR (un graphique de la répartition est disponible sur git). Pour l'entraînement, nous optons pour la stratégie StratifiedKFolds de validation croisée, celle-ci étant adaptée aux jeux de données asymétriques car elle réquilibre les jeux d'apprentissage et de validation, de plus elle permet de mettre en evidence la robustesse ou non de notre model. Nous avons choisi arbitrairement le nombre de 5 folds. C'est ainsi que pour la régression logisitque nous avons opté pour le solver SAGA (A. Defazio, 2014), plutôt que LBGFS (Dong C. Liu, 1989) (solver choisi par défaut par scikit-learn), car il s'avère performant sur les tâches de classification multiclasses sur des grands jeux de données. Nous choisissons un nombre maximal d'itération de l'algorithme de 5000, car en vue de l'importance de notre corpus, cela pourrait s'avérer bénéfique. Par ailleurs, nous choisissons la fonction de coût l2 plutôt que l1 qui est choisie par défaut. En effet, cette dernière ne s'est pas avérée performante : nous notions une baisse de performance conséquente. (Nous passions d'une macro-moyenne pour l'accuracy de 0.61 à 0.30.) Si l2 est plus performant c'est parce qu'elle a tendance à répartir le poids entre toutes les caractéristiques, ce qui peut conduire à des modèles plus stables. Elle est souvent plus robuste face aux valeurs aberrantes que l1, qui est plus linéaire. En effet, nos données contiennent des valeurs aberrantes potentielles, un nombre de token très vari-

Partis	**Précision**	**Rappel**	**F1-score**
ELDR	**0.81**	0.0.63	0.71
GUE-NGL	**0.85**	0.78	**0.81**
PPE-DE	0.74	**0.83**	**0.68**
PSE	0.72	0.75	0.73
VERTS-ALE	**0.79**	0.65	0.72
Accuracy	**0.76**		
macro avg	0.78	0.73	0.75
weighted avg	0.76	0.76	0.76

Table 1: Résumé des résultats de la régression logistique pour l'anglais, le français et l'italien.

able entre les textes des partis : dans ce contexte, l2 peut être une meilleure option. Un jeu de données déséquilibré peut entraîner un surajustement du modèle sur la classe majoritaire, négligeant ainsi les classes minoritaires. En choisissant une force de régularisation modérée ($C = 10$), nous encourageons le modèle à éviter des ajustements excessifs aux données d'entraînement, ce qui peut aider à prévenir un surajustement particulièrement risqué dans le cas de classes déséquilibrées.Une tolérance plus stricte peut rendre le processus d'optimisation plus sensible aux petites variations, ce qui pourrait entraîner une convergence plus lente ou même un échec de convergence si la tolérance est trop faible. nous avons donc opter pour $tol = 1e - 3$. Les résultats obtenus pour la classification avec la régression logistique sont très encourageants. D'après les matrices de confusions que nous avons générées, le modèle classe beaucoup mieux les interventions associées au parti PPE-DE : nous obtenons un rappel moyen d'environ 0.8, une précision de 0.74, et une f-mesure de 0.78. A contrario, dans les partis Verts-ALE et ELDR nous avons un rappel plus bas de 0.64 en moyenne. Nous pouvons observer que plus nous avons de données plus le rappel sera grand. Ces résultats déséquilibrés pourraient s'expliquer par le déséquilibre initialement présent dans nos données. Ainsi, de part cette asymétrie, le modèle parvient à distinguer les interventions des différents partis. Les macro-moyennes (d'environ 0.78), quant à elles, révèlent peu de lacunes dans la généralisation du modèle sur l'ensemble des classes. Des ajustements ou des approches plus sophistiquées pourraient être méjoratifs pour améliorer la performance globale du modèle. Nous avons également tenté de paramétré l'hyperparamètre *class weight* à "balanced" : cette option permet de donner à chaque classe un poids proportionnel à la fréquence de la classe dans le corpus. Dans le cas d'un corpus déséquilibré comme celui avec lequel nous travaillons, le choix de cette option nous semblait pertinent. Cependant, les résultats obtenus étaient légèrement moins bon (perte de presque 2% en accuracy). Néanmoins la répartition des textes pour les deux classes minoritaires étaient légèrement meilleure : l'effet balance s'est donc tout de même ressenti.

La paramétrisation de l'algorithme de RandomForest se fait grâce à la fonction GridSearch de scikit-learn. Pour que les résultats de nos deux algorithmes soient comparables, nous avons également opéré ici à une validation croisée (StratifiedKFolds, à $k = 5$), bien que la fonction RandomForest propose déjà l'option OOB (Out of Bag), qui permet d'évaluer les performances du modèle sur le principe de la validation croisée. Finalement, nous conservons les hyperparamètres choisis par défaut par scikit-learn. En effet, nous avons pu nous rendre compte que baisser le nombre d'estimateurs (*n estimators*) à une valeur inférieure à 100 aggravait considérablement nos résultats, il faut donc bien garder un nombre d'estimateurs conséquent. Bien que beaucoup plus long a s'éxecuter que la régression logistique, l'algorithme de Random Forest est légèrement plus performant que la régression logisitique ici avec une accuracy de 0.77, . Ici, ce sont les classes PPE-DE et PSE qui se démarquent avec les mesures de précision, rappel et f-mesure les plus élevées : pour PPE-DE, nous avons une précision de 0.63, un rappel de 0.96 et une f-mesure de 0.76. Les partis les moins représentés présentent des résultats intéressants également : pour les Verts-ALE et ELDR, nous avons une précision de 1 et un rappel et 0.61. Cette mesure de rappel (la plus basse pour cette classification) pourrait s'expliquer par l'asymétrie de nos données : moins le corpus présente de données pour une classe, moins le modèle parvient à détecter les éléments pertinents. L'algorithme de Random Forest se confirme comme un algorithme de choix pour la classification de texte : ce dernier palie davantage aux contraintes imposées par les données textuelles, ce qui peu lui permetre de devancer légérement les résultats obtenus avec la régression logistique.

Partis	**Précision**	**Rappel**	**F1-score**
ELDR	**1**	0.61	0.76
GUE-NGL	**0.97**	0.72	**0.82**
PPE-DE	0.63	**0.96**	**0.76**
PSE	0.85	0.67	0.75
VERTS-ALE	**1**	0.61	0.76
Accuracy	**0.77**		
macro avg	0.89	0.71	0.77
weighted avg	0.82	0.77	0.77

Table 2: Résumé des résultats de random forest pour l'anglais, le français et l'italien.

4 Conclusion

Les résultats obtenus sur nos travaux sur chacun des algorithmes démontrent une légère supériorité de Random Forest avec une précision de 77%,

comparée à celle de la régression logistique qui est de 76%. Cette différence de 1% en faveur de Random Forest suggère que ce modèle ensembliste a mieux adapté sa flexibilitéé à la complexité des relations non linéaires présentes dans les données textuelles. Lorsque nous regardons en détail nous pouvons voir que ce ne sont pas les mêmes partis qui sont mis en valeurs dans ces deux modèles sauf PPE-DE présent de manière aberrante. De plus le temps d'éxecution de Random forest est beaucoup trop long à coté de la Regression logistique.

Bien qu'il s'agisse d'un modèle linéaire généralisé, la régression logistique peut parfois avoir des performances limitées dans la capture de la complexité inhérente aux données textuelles. Elle s'ajuste à des relations linéaires, ce qui peut ne pas être suffisant pour modéliser efficacement les relations subtiles et non linéaires présentes dans le langage naturel. Random Forest, en revanche, excelle en tant que modèle ensembliste dans la modélisation de relations complexes. Sa capacité à agréger les prédictions de plusieurs arbres de décision et à gérer les caractéristiques bruyantes des données textuelles lui confère un avantage significatif dans ce contexte.

Nous ne pouvons pas cependant négliger l'asymétrie de notre corpus et son influence sur les résultats présentés. Les partis moins représentés au parlement (Verts-ALE, ELDR) ne peuvent pas donner autant d'interventions à étudier que les partis sur-représentés (PPE-DE, PSE). Peut-être qu'il aurait été plus pertinent de remanier notre corpus afin qu'il soit équilibré : cela aurait cependant demandé de diminuer la quantité de données à passer à nos modèles (pour les partis sur-représentés) qui auraient possiblement montré des performances moins intéressantes (sous-apprentissage).

Bien que les modèles créés soient multilingues et que nous sommes parvenus à démontrer leur robustesse, nous sommes conscientes qu'il est possiblement biaisé. Nos résultats se montrent cependant bien meilleurs par rapport à ceux obtenus par les participants au Défi Fouille de Texte en 2009, qui avait uniquement implémenté un modèle monolingue sur le français. Les résultats obtenus étaient déjà légèrement meilleurs pour les partis PPE-DE et PSE, mais les valeurs de rappel et de précision restaient en dessous de 0.5.

5 Discussion

Au cours de ce projet, de nombreux tests ont été réalisés, bien que tous ne soient pas détaillés ici. Nous avons effectué des prétraitements, notamment en appliquant des filtres sur les stopwords pour chaque langue. Cependant, malgré ces efforts, nous n'avons observé qu'une amélioration marginale des performances. De plus, la lemmatisation des données a été mise en œuvre, mais son exécution était chronophage pour des résultats peu significatifs.

Avec du recul, nous pensons qu'il est judicieux de créer une liste personnalisée de stopwords, en tenant compte de la spécificité thématique de la politique. Dans ce domaine, chaque mot, y compris les pronoms, relève d'une importance cruciale. Ainsi, une liste sur mesure permettrait d'éliminer les termes non pertinents dans le contexte politique.

La structure de nos ensembles d'entraînement et de test a également été le fruit d'une réflexion approfondie. Nous avons opté pour une liste de tokens par texte plutôt qu'une liste de textes non tokenisés, car cela s'est avéré plus efficace. En effet, notre premier essai, qui vectorisait directement les phrases, n'a pas produit de résultats satisfaisants. Les pourcentages de réussite étaient très faibles, que ce soit pour Random Forest ou la Régression Logistique.

Par ailleurs, il serait pertinent d'explorer des approches basées sur des réseaux de neurones tels que les Long Short-Term Memory (LSTM). Ces architectures sont particulièrement adaptées à la modélisation de séquences, ce qui pourrait être bénéfique dans le contexte de l'analyse de textes politiques où la structure et la séquentialité des discours jouent un rôle crucial. Bien que l'exécution de modèles de ce type puisse être plus intensive, les performances potentielles en termes de compréhension contextuelle et de capture des relations à long terme pourraient être significatives.

Une autre piste à explorer serait celle de la nouvelle méthode de vectorisation proposée par Chen et. al, *domain concepts extraction*, qui associée à l'algorithme de Random Forest, démontrait des résultats très prometteurs pour la classification de textes juridiques (H. Chen, 2021). Il serait intéressant d'établir la pertinence de cette méthode de vectorisation pour d'autres domaines, en particulier sa possible application à des interventions parlementaires.

Remerciements

Nous aimerions tout d'abord remercier nos ordinateurs qui ont survécu face à nos modèles surpuissants. Ainsi que les notebooks et les cours de Loïc Grobol.

References

S. Lacoste-Julien A. Defazio, F. Bach. 2014. Saga: a fast incremental gradient method with support for non-strongly convex composite objectives.

Jorge Nocedal Dong C. Liu. 1989. On the limited memory bfgs method for large scale optimization.

A. Kiourtis-C. Symvoulidis D. Kyriazis G. Manias, A. Mavrogiorgou. 2023. Multilingual text categorization and sentiment analysis: a comparative analysis of the utilization of multilingual approaches for classifying twitter data. *Neural Computing and Applications*.

J. Chen-J. Ding H. Chen, Lei Wu. 2021. A comparative study of automated legal text classification using random forests and deep learning.

M. Heidaryasafa-S. Mendu L. Barnes D. Brown K. Kowsari, K. Meimandi. 2019. Text classification algorithms: A survey. *Information*, 10.