Apprentissage artificiel : Classification des partis politiques

CHENG Weixuan
Université Sorbonne Nouvelle
M2

GUI Kexin
Université Paris Nanterre
M2

HUANG Yidi
Université Paris Nanterre
M2

Abstract

L'étude aborde la classification des interventions parlementaires européennes, comparant la régression logistique, SVM et Random Forest. Après vectorisation TF-IDF, les modèles sont évalués sur des corpus unilingues et multilingues. Les résultats révèlent une efficacité comparative, avec des nuances dans la performance par méthode, indiquant l'importance d'un réglage fin des hyperparamètres pour optimiser la classification.

1 Introduction

Le projet s'inscrit dans le cadre de la cinquième édition du Défi Fouille de Textes (DEFT),[1] qui se concentre sur la fouille d'opinion et repose sur l'exploitation d'un corpus multilingue comprenant des discours issus de débats parlementaires européens, avec des interventions en français, anglais et italien. La tâche consiste à identifier le parti politique d'appartenance d'un parlementaire à partir de ses interventions.

Les applications de cette tâche de fouille de textes englobent l'analyse et le suivi de l'image publique des politiciens, offrant ainsi un éclairage sur la manière dont les opinions exprimées dans les débats parlementaires peuvent influencer la perception générale d'un projet politique.

2 Préparation des Données

2.1 Présentation du corpus

Le corpus d'analyse vient de DEFT09 , il contient les opinions politiques dans les débats du Parlement européen et un ensemble d'articles issus des journaux Le Monde (France), The Financial Times (Royaume-Uni) et Il Sole 24 Ore (Italie).

Vue que le corpus est au format XML, il faut d'abord prétraiter des données pour extraire les contenus des balises nécessaires dans les fichiers csv. Nous avons donc utilisé la bibliothèque xml.etree.ElementTree dans python pour extraire les textes et la partie concernée, dont les balises <texte> et <PARTI> dans le fichier xml. Les partis extraits peuvent être considérés comme l'étiquette du classifieur.

2.2 Prétraitement du corpus

Le prétraitement du corpus dans notre étude est une étape fondamentale, garantissant la qualité et la pertinence des données traitées. Nous avons opté pour la bibliothèque SpaCy pour sa robustesse, son efficacité et sa capacité à gérer plusieurs langues, afin de préparer notre corpus multilingue. Ce processus de prétraitement implique plusieurs étapes clés : nettoyage, tokenisation et élimination des Stopwords.

Pendant le nettoyage, tous les symboles spéciaux, la ponctuation et les espaces superflus sont retirés pour obtenir un texte plus épuré et uniforme et les stopwords sont également supprimés. Cette élimination rend le texte plus concis et réduit la spécificité lexicale, ce qui est crucial pour la vectorisation ultérieure. Le texte est également converti en minuscule pour uniformiser davantage le corpus, facilitant ainsi sa classification.

2.3 Vectorisation

Nous avons opté pour la méthode TF-IDF (Term Frequency-Inverse Document Frequency) pour la vectorisation des données textuelles. Cette technique évalue non seulement la fréquence des mots au sein d'un document individuel (Term Frequency) mais aussi leur fréquence relative dans l'ensemble du corpus (Inverse Document Frequency). Elle attribue plus d'importance aux termes qui sont fréquents dans un document spécifique mais rares dans l'ensemble du corpus, permettant ainsi de distinguer les mots significatifs qui caractérisent chaque document.

En utilisant TF-IDF, nous avons transformé

[1] https://deft.lisn.upsaclay.fr/

nos données textuelles nettoyées en vecteurs numériques. Chaque intervention est représentée par un ensemble de caractéristiques numériques qui capturent l'essence de son contenu textuel, facilitant ainsi l'identification précise de l'appartenance de chaque intervention.

3 Modélisation

Nous avons utilisé trois modèles pour classifier les partis politiques : Régression logistique (RL), Random Forest et SVM. Pour chaque modèle de classification, l'ensemble de données a été divisé en un ensemble d'apprentissage ('train set') et un ensemble de test ('test set').

3.1 Régression Logistique

La régression logistique (RL) est un modèle de classification statistique. Elle est une extension de la régression linéaire, mais adaptée aux situations où la variable dépendante est catégorielle. Ce modèle modélise la probabilité qu'une entrée x appartienne à une classe spécifique P(Y=1), et la décision est prise en comparant cette probabilité à un seuil. Efficace pour la classification binaire, la régression logistique s'étend également aux problèmes multiclasses via des approches telles que la régression logistique multinomiale et la méthode One-vs-Rest (OVR).

Dans notre tâche multiclasses de classification de parti politique, l'estimation précise des paramètres est cruciale pour assurer l'efficacité du modèle. Pour calibrer au mieux le modèle de régression logistique, il est essentiel d'ajuster finement l'hyperparamètre C, d'évaluer l'impact des pénalités L1 et L2 sur les performances, de sélectionner un solveur le plus adapté, etc.

Pour déterminer la valeur optimale de la régularisation C, nous avons tracé des courbes d'apprentissage sur une gamme étendue de valeurs C et nous avons évalué une valeur de C=100, ce qui suggère une faible régularisation, favorisant la complexité du modèle dans le but d'améliorer la F1-score (Figure 1). De plus, en pratique, le choix du solveur a également un impact significatif sur la convergence du modèle. Dans cette étude, le solveur "sag", connu pour sa vitesse dans les grands jeux de données de multiclasses a été privilégié.

3.2 Random Forest

RandomForest est un modèle qui fusionne les techniques de bagging (bootstrap aggregating) et de RSM (random subspace method) en utilisant des arbres de décision et la vote majoritaire. Il génère plusieurs échantillons aléatoires à partir du corpus, pour chacun desquels un sous-ensemble de caractéristiques est sélectionné, et un arbre de décision aléatoire est entraîné en tant que classifieur. Cette approche permet une agrégation des prédictions issues de multiples sous-modèles, conduisant ainsi à des résultats plus fiables et généralisables.

Les hyperparamètres les plus cruciaux dans RandomForest sont n_estimators et max_depth. Le premier détermine le nombre d'arbres de décision indépendants utilisés pour la prédiction. Il implique la diversité des arbres, mais aussi le coût computationnel. max_depth contrôle la profondeur maximale de chaque arbre de décision, influençant la complexité du modèle et sa propension au surapprentissage.

Pour l'optimisation des hyperparamètres via le méta-apprentissage, la méthode choisie est la Grid Search, qui explore de manière séquentielle différentes combinaisons d'hyperparamètres pour trouver la configuration optimale du modèle.

L'hyperparamètre n_estimators est testé en fixant max_depth. L'intervalle de test est de 50 à 100 avec un pas de 10, révélant une augmentation du f-score avec l'augmentation de n_estimators, mais la courbe de Figure 1 se stabilise progressivement.

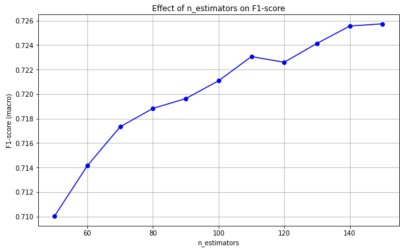

Figure 1: Effect of n_estimators on F1-score

`max_depth` est également testé dans cette plage. Figure 2 a montré une amélioration initiale de la performance avec l'augmentation de max_depth, mais au-delà de 150, le modèle présente des signes de surapprentissage. Ainsi, pour éviter le surapprentissage et réduire le temps d'entraînement du modèle, il est fixé comme : n_estimators=110, max_depth=90.

3.3 SVM

Le SVM, ou machine à vecteurs de support, est un modèle visant à trouver un hyperplan dans un espace de dimensions supérieures pour séparer efficacement les classes de données. L'objectif est

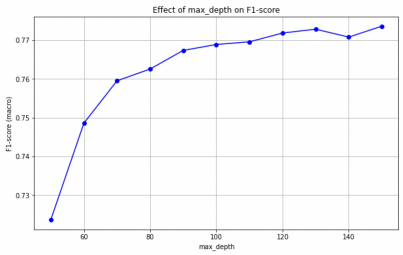

Figure 2: Effect of max_depth on F1-score

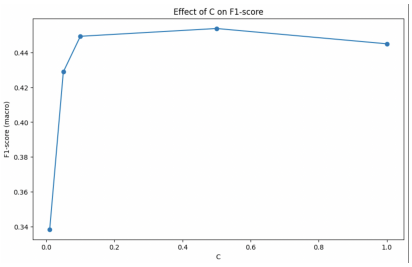

Figure 3: Effect of C on F1-score [0,1]

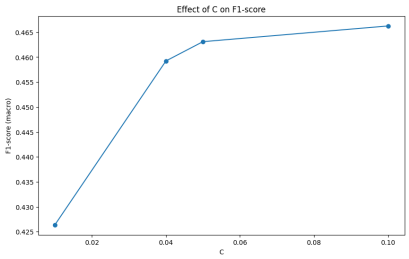

Figure 4: Effect of C on F1-score [0,0.1]

de maximiser la marge entre cet hyperplan et les vecteurs de support, qui sont les points les plus proches de chaque classe. Le SVM utilise des fonctions de noyau pour transformer l'espace des données en un espace où la séparation linéaire est plus aisée. Cette technique est particulièrement utile dans les problèmes de classification et de régression dans des espaces de grandes dimensions.

Le paramètre clé du SVM est le C, un paramètre de régularisation qui contrôle la pénalité attribuée aux erreurs de classification. Un C faible implique une tolérance plus élevée aux erreurs de classification, potentiellement augmentant la capacité de généralisation du modèle mais réduisant sa précision sur les données d'entraînement.

L'optimisation des hyperparamètres de notre modèle SVM a été réalisée via une méthode de Grid Search. Ce processus a commencé par tester un corpus en anglais, pour ensuite appliquer le modèle optimisé à des corpus dans d'autres langues. Durant la phase d'ajustement de C, une exploration initiale dans l'intervalle de 0 à 1 a révélé que la performance optimale se situait autour de 0.1. Cette découverte a été affinée par une recherche plus ciblée dans l'intervalle de 0 à 0.1, confirmant que 0.1 était la valeur la plus efficace pour C.

Par ailleurs, nous avons examiné l'impact du paramètre max_iter, qui détermine le nombre maximal d'itérations de l'algorithme. En testant d'abord un intervalle de 500 à 1000, puis en le réduisant progressivement, il a été observé que la meilleure performance était atteinte dans l'intervalle de 500 à 600. Enfin, l'influence du paramètre tol, qui spécifie la tolérance pour le critère d'arrêt des itérations, a été étudiée. Il a été constaté qu'une tolérance supérieure à 0.01 avait tendance à diminuer les performances du modèle.

4 Résultats et comparaison

La mesure pour l'évaluation des modèles est le F-score, qui met l'accent sur l'équilibre entre la précision et le rappel. Le résultats de trois modèles sont présentés dans Table 1.

4.1 Régression logistique

a). Classification par Langue Individuelle

Dans l'optique d'affiner la performance des modèles de classification, un ajustement des hyperparamètres a été entrepris. Deux approches de classification ont été privilégiées pour analyser leurs performances : la première consistant à classifier les interventions par langue individuelle, et la seconde à traiter un corpus combiné des trois langues. Pour chacune de ces approches, il y a une matrice de confusion et un rapport de classification. Le premier indique le nombre de prédictions correctes et incorrectes pour chaque classe, tandis que le dernier fournit des mesures pour chaque parti politique.

b). Classification de l'Ensemble Combiné

L'application du modèle de RL à l'ensemble combiné des données a permis de relever des insights significatifs sur la performance du modèle. Selon la matrice de confusion (Figure 5), le classi-

	Fr	**En**	**It**	**Multi**
RL	0.77	0.78	0.77	0.77
RF	0.77	0.77	0.77	0.74
SVM	0.60	0.61	0.60	0.60

Table 1: F-score de chaque modèle sur les corpus monolingues et multilingue.

fieur de régression logistique globale a une bonne performance. Les prédictions des cinq classes effectuées sur l'ensemble de test ('test set') sont suffisamment précises, avec un petit nombre d'erreurs.

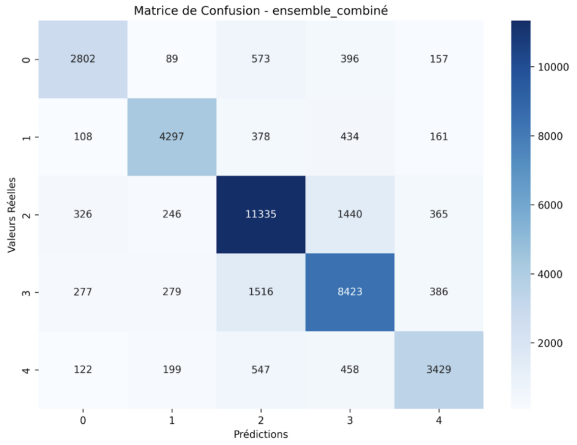

Figure 5: Matrice de confusion pour l'ensemble de données

D'après le rapport de classification, le modèle présente une précision globale de 0.78.

Bien que le modèle de RL affiche une performance respectable, des améliorations peuvent être apportées. Ces variations sont indicatives des défis inhérents à la classification de discours politiques, où chaque parti présente des spécificités langagières qui peuvent soit faciliter soit compliquer leur identification automatique.

4.2 Random Forest

Les résultats des modèles démontrent une performance globale solide, avec les f-score supérieures à 0.7. Cependant, une analyse plus approfondie révèle l'impact du déséquilibre de la répartition des échantillons sur les performances. Les catégories présentant un nombre beaucoup plus faible ou plus élevé d'échantillons affichent des f-scores moins favorables. Ceci s'explique par le fait que la quantité des données peut rendre plus difficile la construction d'un classifieur pour ces catégories.

La détection des partis politiques dans un contexte multilingue (f-score : 0.74) semble plus complexe que dans un contexte monolingue (f-score : 0.77). Cela pourrait être attribué à la diversité linguistique, ainsi qu'à la variation de la distribution des caractéristiques à travers différentes langues.

En plus, il est à noter que le parti politique le mieux détecté est GUE-NGL, ce qui peut être attribué à une répartition équilibrée des échantillons pour cette catégorie, facilitant ainsi l'apprentissage du modèle.

4.3 SVM

Les résultats du modèle indiquent une performance globalement bonne, avec un score F moyen supérieur à 0.6. Cependant, une analyse plus détaillée révèle l'impact du déséquilibre des échantillons sur les performances. Les catégories avec un nombre significativement plus faible ou plus élevé d'échantillons ont montré des scores plus bas. Cela peut s'expliquer par le fait que la rareté des données augmente la difficulté de construire des classificateurs robustes pour ces catégories.

En particulier, nous avons observé que l'application des mêmes paramètres de modèle à des échantillons dans différentes langues produisait des résultats légèrement différents.

Textes en anglais (f-score : 0.606), en français (f-score : 0.599), en italien (f-score : 0.596), multilingues (f-score : 0.603). Bien que les différences numériques ne soient pas grandes, nous pouvons toujours observer des variations, qui peuvent être liées aux différences de caractéristiques des corpus.

5 Conclusion

Globalement, parmi les trois modèles utilisés dans l'expérience, les résultats de LR et RF sont supérieurs à ceux du SVM, ce qui peut s'expliquer par plusieurs raisons. Tout d'abord : les caractéristiques des données. LR et RF se comportent généralement mieux avec des données éparses, qui sont courantes dans les données textuelles.

Ensuite, la complexité et à l'applicabilité des modèles. Le RF peut traiter les relations non linéaires et les interactions complexes entre les caractéristiques, et la LR offre une bonne applicabilité pour certaines tâches de classification textuelle.

Enfin, la question du déséquilibre des catégories dans les tâches textuelles : si un ensemble de données présente ce problème, RF et LR peuvent traiter plus efficacement cette situation, alors que le SVM nécessite des techniques et des ajustements spéciaux pour traiter le déséquilibre des catégories.

Pour aller plus loin, les résultats pourraient être améliorés par le réglage fin des hyperparamètres, l'utilisation de techniques de prétraitement plus avancées, ou l'expérimentation avec d'autres modèles de classification plus sophistiqués ou des approches d'ensemble. La compréhension approfondie des résultats actuels servira de base solide pour de tels efforts d'amélioration.

References

2024a. sklearn.linear_model.decisiontree. Consulté le 1 Jan. 2024.

2024b. sklearn.linear_model.logisticregression. Consulté le 3 Jan. 2024.

2024c. sklearn.linear_model.randomforest. Consulté le 1 Jan. 2024.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Jason Brownlee. 2024. Tune hyperparameters for classification machine learning algorithms. Consulté le 2 jan. 2024.