UNIVERSITÉ PARIS CITÉ

UFR MATHÉMATIQUES ET INFORMATIQUE

Comparaison de plusieurs approches de classification pour l'analyse des données sociales

Master 1 Données, Connaissances et Intelligence

Quentin WENDLING – Nicolas FRANCESCHINI

Encadré par Valentina Dragos

Année universitaire 2024 – 2025

Table des matières

1. Introduction

L’analyse des données sociales, en particulier celles issues des réseaux sociaux, suscite un intérêt croissant dans de nombreux domaines scientifiques, institutionnels et industriels. En effet, la masse d’informations générée quotidiennement par les utilisateurs permet non seulement de mieux comprendre les dynamiques sociales contemporaines, mais aussi d’identifier des comportements collectifs, de détecter des signaux faibles, voire d’anticiper certaines situations critiques.

Cette capacité à capter en temps réel l’opinion publique ou l’émergence de phénomènes sociaux rend ces données particulièrement précieuses pour des applications telles que la gestion de crise (désinformation, mobilisation, violences), le marketing digital (analyse de sentiment, ciblage comportemental) ou encore la modération de contenu en ligne, notamment dans le cadre de la lutte contre les discours haineux, sexistes, racistes ou extrémistes.

C’est dans cette dynamique que s’inscrit notre projet, qui vise à analyser et classifier automatiquement des contenus à caractère problématique (discours extrémistes, sexistes, haineux, etc.) issus de différentes sources textuelles : réseaux sociaux (comme Twitter), forums, plateformes de discussion ou encore extraits d’articles. L’objectif principal de notre étude est de comparer l’efficacité de différentes approches de classification textuelle, classiques et modernes, afin de déterminer lesquelles sont les plus pertinentes pour traiter le français, une langue souvent sous-représentée dans les outils de traitement automatique du langage.

2. Etat de l’art des méthodes de classification textuelle

La classification textuelle constitue un champ de recherche particulièrement actif en traitement automatique du langage naturel, en raison de ses nombreuses applications concrètes dans des domaines variés. Elle permet, entre autres, d’automatiser la modération de contenus en ligne, de filtrer les courriers électroniques indésirables (spams), ou encore d’organiser et classer efficacement de grands volumes de documents. Cette polyvalence en fait un outil précieux tant pour les entreprises que pour les institutions publiques, les médias ou les chercheurs.

En conséquence, un grand nombre de méthodes de classification ont été développées au fil des années, allant des approches statistiques traditionnelles aux modèles d'apprentissage profond les plus récents. Toutefois, il est important de souligner que la plupart de ces méthodes sont conçues et optimisées pour traiter des textes en anglais, langue largement dominante dans les publications scientifiques et les jeux de données disponibles à grande échelle.

Ainsi, malgré la richesse et la diversité linguistique du web, la langue française reste relativement peu représentée dans les corpus d'entraînement et les benchmarks internationaux. Elle est rarement utilisée comme langue cible dans les grands modèles ou les études de référence. Cette situation limite la performance des méthodes de classification sur des textes en français, car ces outils ne tiennent pas compte des spécificités lexicales, grammaticales ou culturelles de la langue.

2.1 Modèles adaptés au français : CamemBERT

Malgré la prédominance des modèles anglophones dans ce domaine, certaines méthodes ont été spécifiquement conçues ou adaptées pour le français. Parmi les plus notables figure CamemBERT, un modèle de langage préentraîné de type Transformer, qui s’appuie sur l’architecture de RoBERTa (lui-même une version optimisée du célèbre modèle BERT).

À la différence de ses prédécesseurs majoritairement entraînés sur des données en anglais, CamemBERT a été entraîné exclusivement sur un large corpus de textes francophones, extrait de OSCAR (Open Super-large Crawled ALMAnaCH coRpus). Il s’agit d’un corpus multilingue collecté à partir du web, qui a été nettoyé, filtré et organisé par langue. Dans le cas de CamemBERT, seuls les textes en français ont été conservés, ce qui lui permet de mieux saisir les subtilités grammaticales, syntaxiques et sémantiques de la langue française.

Outre sa puissance de traitement, héritée de l’architecture RoBERTa, CamemBERT bénéficie également d’une forte notoriété dans la communauté scientifique et technologique francophone. Il est open-source et bien documenté, ce qui représente un atout non négligeable : de nombreuses ressources, tutoriels, exemples de code et solutions aux problèmes fréquents sont facilement accessibles en ligne, facilitant ainsi sa prise en main et son intégration dans des projets.

Par ailleurs, CamemBERT est un modèle flexible et polyvalent, capable d'être fine-tuné pour des tâches spécifiques comme la classification de texte, l'analyse de sentiments, la reconnaissance d'entités nommées ou encore la détection de discours haineux. Cette adaptabilité repose sur la possibilité d'ajuster ses paramètres et de l'entraîner sur des jeux de données ciblés, ce qui en fait un outil particulièrement pertinent pour le développement d'applications en français dans divers domaines.

2.2 Détection de la haine en ligne : AngryBERT

Dans le prolongement de cette logique d'adaptation linguistique et contextuelle, des modèles dérivés de CamemBERT ont été développés pour répondre à des besoins plus spécifiques. C'est notamment le cas d'AngryBERT, un modèle entraîné pour la détection des discours haineux, agressifs ou violents dans les textes francophones, en particulier sur les réseaux sociaux.

AngryBERT reprend l'architecture et les connaissances linguistiques de CamemBERT, mais il est fine-tuné sur des corpus annotés contenant des messages hostiles, issus principalement de plateformes telles que Twitter. Cela lui permet de reconnaître plus efficacement les propos haineux ou toxiques, en tenant compte des particularités du langage courant sur internet, comme l'usage d'argot, les fautes d'orthographe, les abréviations ou encore le sarcasme.

Ce modèle s'avère particulièrement utile dans des contextes où la modération automatisée du contenu est nécessaire, comme dans les forums, les plateformes de streaming ou les services de messagerie. Grâce à cette spécialisation, AngryBERT surpasse les modèles généralistes dans la détection de la haine ou de l'agressivité, domaines dans lesquels un modèle standard comme CamemBERT, bien qu'efficace, peut montrer des limites.

Tout comme son prédécesseur, AngryBERT est également open-source et accessible via des bibliothèques telles que Hugging Face Transformers, ce qui permet de l'intégrer facilement dans des pipelines de traitement de texte en français. De plus, sa spécialisation en fait un modèle de référence pour les travaux de recherche ou les projets industriels liés à la lutte contre les discours haineux, contribuant ainsi à rendre les espaces numériques plus sûrs.

2.3 Autres méthodes

Il est également possible d'utiliser des algorithmes classiques de classification supervisée pour des tâches de classification textuelle, même si ces méthodes n'ont pas été initialement conçues spécifiquement pour traiter des données textuelles. Ces algorithmes, tels que les machines à vecteurs de support (SVM), les forêts aléatoires (Random Forest) ou encore XGBoost, ont la particularité d'être généralistes et adaptables à divers types de données, y compris des représentations vectorielles issues de textes.

Pour appliquer ces modèles à des données textuelles, une étape préalable indispensable consiste à transformer le texte en une représentation numérique, souvent sous forme de vecteurs. Cette transformation peut prendre la forme de modèles simples BOW (bag-of-words) ou des représentations pondérées comme le TF-IDF (Term

Frequency-Inverse Document Frequency). Ces représentations permettent de convertir les documents en vecteurs numériques exploitables par les algorithmes classiques.

Bien que ces méthodes ne capturent pas directement la structure syntaxique ou le contexte sémantique complexe des textes, elles peuvent néanmoins fournir de très bonnes performances, notamment sur des corpus de taille modérée ou des tâches de classification relativement simples. Par exemple, les SVM sont réputés pour leur capacité à gérer des espaces à très haute dimension, ce qui est souvent le cas avec des vecteurs textuels. Les forêts aléatoires offrent quant à elles une robustesse face au bruit et aux données non linéaires tandis qu' XGBoost est particulièrement apprécié pour sa rapidité d'exécution et sa capacité à obtenir d'excellents résultats grâce à l'optimisation de fonctions de perte complexes et à la régularisation, ce qui le rend très efficace sur des jeux de données variés.

En résumé, même si ces algorithmes ne sont pas spécialement conçus pour le traitement du langage naturel, ils restent des solutions fiables et largement utilisées pour des applications de classification textuelle, en particulier lorsqu'une simplicité d'implémentation, une rapidité d'exécution ou une interprétabilité sont recherchées.

3. Méthodologie

Afin de détecter automatiquement les discours haineux, sexistes ou discriminatoires sur les réseaux sociaux, nous avons combiné des techniques d'analyse de texte et d'apprentissage supervisé. Nos expérimentations se sont appuyées principalement sur des données brutes issues de Twitter, Reddit, …, ainsi que sur le corpus enrichi Emotyc, contenant des annotations émotionnelles et linguistiques.

Nous avons exploré deux approches principales, la première repose sur des modèles classiques utilisant des représentations vectorielles simples des textes, tandis que la seconde s'appuie sur des modèles de type transformer, en particulier CamemBERT, pré entraîné spécifiquement pour la langue française.

3.1 Prétraitement des données

La majorité des corpus que nous avons utilisés étaient déjà propres et ne nécessitaient pas de nettoyage. Nous avons vérifié l'absence de valeurs manquantes (Nan) dans les fichiers, ce qui a confirmé que les données étaient bien structurées.

Cependant, pour deux fichiers spécifiques, un traitement particulier a été nécessaire, le premier nécessitait une séparation des blocs de texte à l'aide d'une chaîne délimitante spécifique afin d'extraire correctement les phrases et leurs caractéristiques. Le second contenait les émotions associées à chaque phrase, mais ne comprenait pas les labels de classification (haineux / non haineux). Pour ce cas, nous avons effectué une jointure avec un autre fichier contenant les étiquettes correspondantes, afin de rendre le corpus exploitable pour l'apprentissage supervisé.

Une fois les données préparées, les textes ont été transformés en vecteurs numériques à l'aide de deux méthodes. La première, CountVectorizer, repose sur le modèle Bag-of-Words. Elle consiste à compter la fréquence d'apparition de chaque mot dans un texte. Chaque message est ainsi représenté par un vecteur de fréquences, où chaque position correspond à un mot du vocabulaire. Cette méthode est simple mais efficace pour détecter la présence de termes caractéristiques. La seconde, Tf/IDF Vectorizer (Term Frequency-Inverse Document Frequency), reprend le même principe mais en y ajoutant une notion d'importance. Elle tient en compte à la fois la fréquence d'un mot dans un texte et sa rareté dans l'ensemble du corpus. Ce qui signifie que plus un mot est spécifique à un texte donné, plus il aura de poids dans la représentation finale. Cela permet de mettre en avant les mots réellement significatifs tout en atténuant l'impact des mots très fréquents et peu informatifs.

Ces représentations vectorielles ont servi d'entrée aux modèles de classification testés par la suite.

3.2 Méthodes classiques

Après la vectorisation des textes, nous avons entraîné plusieurs modèles classiques d'apprentissage supervisé pour effectuer la classification binaire (discours haineux / non

haineux). Ces modèles sont bien connus pour leur efficacité sur des représentations textuelles simples comme celles fournies par CountVectorizer ou bien TfidfVectorizer.

Les modèles que nous avons utilisés sont SVM (Support Vector Machine) qui est un algorithme de classification robuste, particulièrement adapté aux espaces de grandes dimensions comme ceux générés par les représentations textuelles. Ce modèle cherche à trouver une frontière optimale entre les deux classes en maximisant la marge entre les points les plus proches. Le suivant modèle que nous avons utilisé est le Random Forest Classifier, ce modèle repose sur un ensemble d'arbre de décision. Il permet de gérer les relations non linéaires entre variables et réduit le risque de surapprentissage (overfitting) grâce à la combinaison des résultats de plusieurs arbres. Enfin le dernier modèle "classique" que nous avons utilisé est le XGBoost (Extreme Gradient Boosting), il s'agit d'un modèle de boosting très performant qui construit progressivement des arbres pour corriger les erreurs des précédents. Il est particulièrement efficace pour les données structurées et les jeux de données déséquilibrés. Chacun de ces modèles ont été entraînés à partir des vecteurs obtenus via Bag-of-Words et Tf/IDF, puis évalués à l'aide de métriques classiques telles que la précision, le rappel, la F-mesure (F1-score) et l'accuracy. Ces résultats nous ont permis d'avoir une première base de comparaison avant de passer à une approches plus avancées comme les modèles de type transformer.

3.3 Modèle CamemBERT

En complément des modèles classiques, nous avons utilisé CamemBERT, un modèle de type transformer spécialement conçu pour le traitement de la langue française. Il s'agit d'une variante francophone du modèle BERT (Bidirectional Endercoder Representations from Transformers) qui est préentraînée sur un large corpus de textes en français. Contrairement aux autres méthodes classiques qui se reposent sur des représentations simplifiées de textes (comme par exemple des vecteurs de fréquences), CamemBERT permet de capturer la structure et le contexte linguistique de chaque mot en tenant en compte l'environnement de celui-ci dans la phrase. Cela en fait donc un modèle particulièrement performant pour des tâches complexes comme la détection de discours haineux où le sens dépend souvent du ton, des nuances ou de la combinaison de plusieurs mots.

3.2.1 Mise en oeuvre

Afin d'adapter CamemBERT à notre tâche de classification, nous avons utilisé la version "TFCamemBertForSequenceClassification" proposée par la bibliothèque transformers de Hugging Face. Les textes ont d'abord été tokenisés à l'aide de CamembertTokenizer, puis convertis en tenseurs (input_ids et attention_mask) nécessaires à l'entraînement du modèle.

Les données ont ensuite été divisées en ensemble d'entraînement et de test, puis transformées en objets "tf.data.Dataset", permettant un traitement par lots (batchs). Le modèle a été compilé avec l'optimiseur Adam, une fonction de perte adaptée à la classification multi-classe (SparseCategorcalCrossentropy) et une métrique d'évaluation basée sur l'accuracy.

3.2.2 Entraînement

Tous les modèles ont été exécutés sur Google Colab, en utilisant un compte classique avec des ressources limitées. En ce qui concerne CamemBERT, l'entraînement s'est révélé particulièrement coûteux en temps (il fallait compter environ 5 heures pour 1 à 3 époques selon la taille du corpus).

En raison de ces contraintes techniques, nous avons donc choisi de limiter le nombre d'époques afin de rester dans les capacités offertes par Colab. Malgré cette limitation, le modèle a montré de bonnes performances dès les premières itérations, notamment grâce à son préentraînement sur un large corpus de textes en français.

3.3 Enrichissement avec les caractéristiques Emotyc

Afin d'améliorer la précision de nos modèles de classification, nous avons enrichi les corpus bruts (haineux, sexiste, reddit…) à l'aide de caractéristiques émotionnelles issues du corpus Emotyc. Celui-ci contient des annotations permettant d'identifier certains traits expressifs dans les textes, tels que la peur, l'embarras, la colère, etc.

Ces informations émotionnelles offrent un éclairage complémentaire sur le contenu des messages, en apportant une dimension qualitative qui dépasse la simple analyse lexicale des modèles.

3.4 Traitement et fusion des données

L'intégration de ces annotations a nécessité un traitement particulier. Les phrases et leurs traits émotionnels étaient regroupés dans des blocs séparés par une chaîne spécifique. Nous avons dû extraire chaque phrase et leurs attributs associés. De plus, les phrases annotées ne comportaient pas directement de label de classification (par exemple haineux / non haineux). Il a donc effectué une jointure avec un autre fichier (du corpus brut) contenant ces étiquettes pour reconstruire un corpus cohérent et exploitable.

3.5 Utilisation dans les modèles

Les caractéristiques émotionnelles extraites ont ensuite été intégrées dans les modèles de deux manières différentes, pour les modèles classiques, nous avons ajouté ces informations sous forme de colonnes binaires, indiquant la présence ou non d'un trait émotionnel dans chaque texte. Elles ont été concaténées aux représentations vectorielles générées par TfidVectorizer et CountVectorizer. Pour ce qui est de CamemBERT, nous avons opté pour une approche textuelle. Les traits émotionnels détectés ont été convertis en mots-clés et concaténés directement au texte d'entrée. Cela permet au modèle d'en tenir compte dans sa compréhension globale de la séquence.

4. Etude expérimentale et analyse des résultats

4.1 Corpus brut haineux

Le premier corpus que nous avons utilisé pour évaluer les performances des différents modèles de classification est le corpus “brut haineux”, un ensemble de 600 tweets collectés sur la plateforme X (anciennement Twitter). Bien que ces tweets ne soient pas accompagnés d’annotations des émotions présentes, chacun d’eux a été labellisé manuellement comme "haineux" ou "non haineux", label que l’on va chercher à prévoir avec nos modèles.

Ce corpus est intéressant dans le cadre de notre étude pour plusieurs raisons. Tout d’abord, il est équilibré, c’est-à-dire qu’il contient un nombre à peu près égal de tweets appartenant à chaque classe. Cette caractéristique est cruciale dans le contexte de l’apprentissage supervisé : elle évite les biais de prédiction souvent observés lorsque les classes sont déséquilibrées, où les modèles tendent à sur-prédire la classe majoritaire. Grâce à cette répartition homogène, nous avons pu entraîner et évaluer nos modèles dans des conditions plus stables et fiables, en réduisant les distorsions statistiques liées à la distribution des classes.

En outre, le format court et direct des tweets impose une contrainte linguistique spécifique : les modèles doivent être capables de comprendre un contenu parfois implicite, sarcastique ou codé, exprimé en peu de mots. Cela représente un véritable défi pour les algorithmes de classification textuelle, mais constitue également un bon indicateur de leur robustesse et de leur capacité à traiter des données issues de la vie réelle.

Modèle	CamemBERT	SVM + Tf/IDF	SVM + BOW	RFC + Tf/IDF	RFC + BOW	XGBoost + Tf/IDF	XGBoost + BOW
Accuracy	0.95	0.91	0.89	0.83	0.83	0.80	0.94
Temps exécution	25 mins	~1s	~1s	4s	6s	~1s	~1s

Parmi l’ensemble des modèles testés, CamemBERT a obtenu la meilleure performance avec une précision atteignant 95%. Cette efficacité s’explique par sa capacité à prendre en compte le contexte des mots grâce à son architecture de type transformer. En revanche, ce modèle à un inconvénient, son temps d’exécution qui est relativement élevé par rapport au autre modèles (environ 25 minutes, contre quelques secondes pour les modèles classiques) sur un environnement Colab standard. Cela peut constituer un frein lorsque l’on souhaite peaufiner le modèle ou effectuer plusieurs essais. On remarque que les modèles SVM offrent également de très bon résultats, avec une précision de 91% pour Tf/IDF et 89% pour Bag-of-Words, tout en étant extrêmement rapides à entraîner. A l’inverse, les modèles Random Forest semblent avoir plus de difficultés à bien généraliser, avec une précision de 83% quel que soit le type de vectorisation utilisé. Enfin, XGBoost montre des performances intéressantes, mais uniquement avec Bag-of-Words, atteignant 94% de précision. Avec une

représentation Tf/IDF ses résultats chutent à 80% ce qui suggère qu'il exploite mieux les représentations binaires.

4.2 Corpus brut Reddit toxic

Le second corpus utilisé dans notre étude est le corpus "Reddit Toxic brut", qui contient environ 20 000 textes extraits du célèbre forum communautaire Reddit. Chaque texte est labellisé de manière binaire selon qu'il est considéré comme toxique ou non toxique, ce qui permet d'appliquer des méthodes de classification supervisée sur des données à grande échelle.

Comme son nom l'indique, il s'agit d'un corpus "brut", c'est-à-dire qu'il ne comporte pas d'annotations détaillées, telles que le type de toxicité (haine, harcèlement, insulte, provocation, etc.) ou l'émotion véhiculée par le message. Toutefois, cette limitation est compensée par l'importance du volume de données, qui constitue un atout majeur dans le cadre de l'apprentissage automatique. En effet, l'entraînement de modèles sur des jeux de données de grande taille permet souvent d'améliorer leur capacité de généralisation et leur robustesse face à la variabilité du langage utilisé sur les réseaux sociaux.

Ce corpus nous permet donc d'explorer un aspect complémentaire à celui du corpus "Brut haineux" : tandis que ce dernier était limité en taille mais équilibré, le corpus Reddit présente une grande diversité de données, ce qui nous offre la possibilité de tester l'impact de la taille des données sur les performances des modèles, en comparant notamment les résultats obtenus avec des algorithmes classiques versus des modèles de type Transformer comme CamemBERT.

Modèle	CamemBERT	SVM + Tf/IDF	SVM + BOW	RFC + Tf/IDF	RFC + BOW	XGBoost + Tf/IDF	XGBoost + BOW
Accuracy	0.98	0.98	0.97	0.98	0.98	0.98	0.98
Temps exécution	5h30	7s	7s	40s	26s	6s	~1s

Les résultats obtenus sur ce corpus se sont révélés très satisfaisants. En effet, la précision (accuracy) atteint 98 % pour la majorité des modèles, à l'exception du SVM couplé à la représentation Bag of Words (BOW) qui affiche un score légèrement inférieur à 97 %, soit une différence négligeable en pratique.

Cependant, cette amélioration des performances s'accompagne d'une augmentation notable des temps d'exécution. Si la quantité plus importante de données contribue à de meilleurs résultats, elle entraîne également un temps de traitement plus long. Pour les algorithmes classiques d'apprentissage supervisé, cela reste peu contraignant car les temps restent globalement raisonnables. En revanche, pour des modèles plus lourds comme CamemBERT, l'impact est significatif : il a fallu environ 5h30 d'entraînement pour obtenir des performances équivalentes à celles des autres modèles, soulignant ainsi les limites en termes de coût computationnel de ce type d'architecture.

4.3 Corpus haineux Emotyc

Afin de mesurer l'intérêt des annotations émotionnelles dans la détection automatique de discours haineux, nous avons choisi de comparer les performances de nos différents modèles sur un même ensemble de tweets, mais avec une dimension supplémentaire : l'émotion exprimée dans chaque message. Pour cela, nous avons utilisé le corpus haineux EmoTyc, une version enrichie du corpus "Brut haineux" où des annotations émotionnelles définies grâce à Emotyc sont présentes pour chaque tweet.

Le corpus EmoTyc, issu d'un projet visant à classifier les émotions dans les textes français, contient des annotations émotionnelles telles que la colère, la peur, la joie ou encore la tristesse. Ces émotions sont supposées apporter un contexte affectif utile à la classification, notamment pour affiner la détection des propos potentiellement haineux ou problématiques.

Cependant, contrairement au corpus "Brut haineux", le corpus EmoTyc ne contenait pas initialement les labels "haineux" ou "non haineux". Pour pouvoir exploiter pleinement cette ressource, il a donc été nécessaire de fusionner les deux corpus. Comme expliqué précédemment dans la section "Traitement et fusion des données", cette opération a permis de reconstituer un jeu de données complet, associant à chaque tweet à la fois une étiquette de haine et une annotation émotionnelle. Cette fusion nous a permis d'entraîner et de tester nos modèles dans un cadre enrichi, afin d'évaluer si la prise en compte de l'émotion améliore la précision de la détection des contenus haineux.

Modèle	CamemBERT	SVM + Tf/IDF	SVM + BOW	RFC + Tf/IDF	RFC + BOW	XGBoost + Tf/IDF	XGBoost + BOW
Accuracy	0.97	0.91	0.93	0.85	0.86	0.86	0.86
Temps exécution	45 mins	~1s	~1s	~1s	~1s	~1s	~1s

Les résultats présentés ci-dessus suggèrent que l'ajout d'annotations émotionnelles ne constitue pas nécessairement un levier d'amélioration pour les modèles de classification, lorsque les performances sont déjà élevées sur un corpus initial. En comparant les scores obtenus avec le corpus haineux EmoTyc à ceux issus du corpus brut haineux, on constate que les résultats restent quasiment identiques, en termes d'accuracy et de métriques globales. Cela tend à montrer que, dans un contexte où les caractéristiques lexicales et syntaxiques suffisent à distinguer efficacement les propos haineux, l'apport d'une analyse émotionnelle peut être redondant, voire inutile.

Cette observation soulève ainsi une question importante quant à la pertinence de l'enrichissement émotionnel dans certains cas : si les signaux textuels sont suffisamment marqués pour permettre une classification fiable, l'intégration de dimensions affectives ne semble pas améliorer la prédiction de manière significative.

Par ailleurs, cet enrichissement n'est pas sans conséquence sur les ressources nécessaires. En particulier, pour un modèle de type CamemBERT, l'ajout des annotations émotionnelles a quasiment doublé le temps d'exécution, sans gain de performance en retour. Cette observation met en évidence un rapport coût-bénéfice défavorable dans ce

contexte précis, et invite à une réflexion sur la pertinence de ces enrichissements en fonction des objectifs et des contraintes de calcul.

4.4 Autres corpus

Enfin, nous avons également testé nos modèles sur d'autres corpus complémentaires, bien que les données qu'ils contiennent et les résultats obtenus se soient révélés moins pertinents ou moins exploitables que ceux issus des corpus principaux présentés précédemment. Malgré leur intérêt plus limité, nous avons jugé utile de les inclure dans notre étude, ne serait-ce que pour souligner certaines faiblesses ou limites rencontrées par nos approches de classification.

Leur inclusion contribue ainsi à mettre en lumière les conditions dans lesquelles nos modèles perdent en performance, ce qui est essentiel pour évaluer leur robustesse et leurs limites d'application. Mentionner ces jeux de données "moins réussis" offre donc une vision plus complète et plus honnête de notre démarche expérimentale.

4.4.1 Corpus sexiste

Ce corpus comprend environ 300 textes, chacun accompagné d'un label binaire "sexiste" ou "non sexiste", ainsi que d'annotations émotionnelles. Il s'agit d'un jeu de données intéressant en théorie, car il permettrait d'analyser la corrélation entre le contenu émotionnel d'un message et sa potentielle connotation sexiste. Toutefois, dans le cadre de notre expérimentation, ces annotations émotionnelles n'ont pas été exploitées.

En effet, ce corpus a été utilisé en amont de notre étude sur l'analyse des émotions, à un stade où nous nous concentrions exclusivement sur la classification binaire du contenu (sexiste ou non) sans tenir compte des annotations. Par conséquent, seule la variable principale "sexiste / non sexiste" a été utilisée pour entraîner et évaluer nos modèles. Ce choix méthodologique reflète notre volonté de procéder par étapes, en commençant par une analyse purement thématique avant d'introduire progressivement des dimensions plus complexes, comme l'émotion.

Modèle	CamemBERT	SVM + Tf/IDF	SVM + BOW	RFC + Tf/IDF	RFC + BOW	XGBoost + Tf/IDF	XGBoost + BOW
Accuracy	0.98	0.80	0.77	0.77	0.70	0.65	0.62
Temps exécution	31 mins	~1s	~1s	~1s	~1s	~1s	~1s

4.4.2 Corpus brut extrémiste

Ce corpus est constitué de 8 textes, chacun annoté par un ou plusieurs labels descriptifs indiquant des thématiques ou éléments potentiellement liés à des discours extrémistes. Les textes sont issus d'une source brute, séparés manuellement à l'aide d'un délimiteur spécifique (~~~~) et continent pour chaque entrée un contenu textuel libre ainsi qu'une

ligne de labels introduite par le caractère *. Ces labels peuvent inclure des catégories telles que “Xenophobe”, “Identitaire”, “nationaliste”, ou d’autres propos radicaux.

En raison de sa très petite taille, le corpus nous a permis d’observer la puissance des nos différents modèles, mais ne nous a pas permis d’obtenir de résultats significatifs en termes de performance.

Modèle	CamemBERT	SVM + Tf/IDF	SVM + BOW	RFC + Tf/IDF	RFC + BOW	XGBoost + Tf/IDF	XGBoost + BOW
Accuracy	0.37	0.05	0.05	0.05	0.04	0.03	0.04
Temps exécution	5s	~1s	~1s	~1s	~1s	~1s	~1s

Comme nous pouvons observer ci-dessus, les performances des modèles sont dans l’ensemble très faibles. Nous constatons cependant que CamemBERT parvient à obtenir une précision nettement supérieure aux autres, atteignant 37% contre seulement 3 à 5% pour les modèles classiques. Bien que ces résultats restent insuffisants, ils s’expliquent principalement par le nombre très limité d’exemples disponibles dans ce corpus. Cela montre à quel point la taille et la qualité des données sont déterminantes.

5. Conclusion et perspectives

En conclusion, notre étude comparative met en lumière plusieurs enseignements importants sur les méthodes de classification textuelle appliquées à des contenus haineux, sexistes ou toxiques en langue française. Parmi les différents modèles testés, CamemBERT s'impose comme le plus performant en termes de précision, confirmant l'efficacité des architectures de type Transformer pour ce type de tâche. Toutefois, cette performance a un coût computationnel non négligeable, avec des temps d'exécution bien supérieurs à ceux des modèles classiques, en particulier sur des corpus volumineux. Ce facteur doit être pris en compte lors du choix d'un modèle, notamment en contexte opérationnel ou en traitement de masse.

Par ailleurs, nos expérimentations montrent que les représentations TF-IDF et Bag of Words (BoW) offrent des performances globalement équivalentes sur la plupart des modèles testés. Une exception notable est observée avec XGBoost, pour lequel BoW se révèle plus efficace sur le corpus brut haineux, ce qui laisse supposer que ce type de représentation peut mieux capturer certaines structures lexicales spécifiques à ce modèle.

De plus, nos tests confirment que les modèles supervisés sont beaucoup plus performants sur des corpus riches en données, ce qui renforce l'importance de la taille des jeux d'entraînement dans la qualité des classifications obtenues.

Concernant l'apport des annotations émotionnelles, les résultats indiquent qu'elles n'améliorent pas significativement les performances lorsque les modèles sont déjà efficaces sans elles. Cette observation s'est notamment vérifiée avec le corpus EmoTyc, dont les performances se sont avérées similaires à celles obtenues sur les mêmes données non annotées émotionnellement, tout en doublant le temps d'exécution pour des modèles complexes comme CamemBERT.

Enfin, pour approfondir cette étude, un axe d'amélioration pertinent serait de tester l'effet des annotations émotionnelles sur des corpus où les performances initiales sont faibles. Cela permettrait de déterminer si ces annotations peuvent réellement servir de levier d'amélioration lorsque les données textuelles ne suffisent pas à elles seules à bien discriminer les propos problématiques.

6. Bibliographie

Awal, M. R., Cao, R., Lee, R. K.-W., & Mitrović, S. (s.d.). *AngryBERT: Joint learning target and emotion for hate speech detection*.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., Villemonte de la Clergerie, É., Seddah, D., & Sagot, B. (2020). *CamemBERT: a Tasty French Language Model*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7203–7219). Association for Computational Linguistics.

Dragos, V., & Constable, Y. (2023). *Comparison of classification techniques for extremism detection in French social media*. In *Proceedings of the 26th International Conference on Information Fusion (FUSION)* (pp. 1–6). IEEE.

Hugging Face. *CamemBERT model documentation*. https://huggingface.co/docs/transformers/en/model_doc/camembert

Scikit-learn developers. (2007) *Ensemble methods — Scikit-learn documentation*. https://scikit-learn.org/stable/modules/ensemble.html

TensorFlow. *TensorFlow: An end-to-end open source machine learning platform*. https://www.tensorflow.org

Google Colaboratory. *Colab: Google Research*. https://colab.research.google.com