Université de Caen Normandie

Projet
Master 1 Informatique

Identification de contenus injurieux, racistes ou homophobes

Étudiants

Théo RAULT (22102889)
Pierre SOCHON (22107881)

Encadrants

Tanguy GERNOT
Emmanuel GIGUET

12 mai 2025

Table des matières

1 Introduction

Sur Internet et particulièrement sur les réseaux sociaux, les propos injurieux, racistes ou homophobes se multiplient, souvent grâce à l'anonymat offert par les plateformes. Cette prolifération nuit gravement aux communautés en ligne, rendant nécessaire l'instauration de mécanismes de modération efficaces et rapides.

L'identification automatique de contenus haineux constitue donc un enjeu crucial, tant pour la protection des utilisateurs que pour le maintien d'espaces d'échanges sains.

Cependant, cette tâche soulève plusieurs défis :

— Variété des formes d'expression haineuse (directe, implicite, sarcastique),

— Diversité linguistique (expressions en français, anglais, dialectes...),

— Disponibilité limitée de jeux de données adaptés, notamment en français,

— Nécessité d'algorithmes capables d'appréhender les subtilités du langage naturel.

Le projet présenté ici s'inscrit dans ce contexte : il vise à développer et évaluer des prototypes d'identification automatique de contenus injurieux, racistes et homophobes, en se concentrant principalement sur le texte français.

Pour cela, plusieurs axes ont été explorés :

— La constitution de jeux de données adaptés, en combinant des corpus existants en français et en anglais ;

— Le fine-tuning de modèles de langage pré-entraînés (CamemBERT, FlauBERT, XLM-RoBERTa) sur la tâche de classification multi-label ;

— Une réflexion sur l'utilisation potentielle du prompt engineering sur de grands modèles pour améliorer la compréhension contextuelle fine ;

— Une évaluation comparative des performances sur différents jeux de données.

Enfin, le projet s'inscrit dans une perspective d'application en temps réel : l'objectif étant de concevoir un système théoriquement capable de modérer en direct des plateformes telles que les chats Twitch ou les commentaires sur les réseaux sociaux.

2 État de l'art (Veille scientifique)

2.1 Introduction à la détection de contenus haineux

Dans le cadre du traitement automatique du langage naturel (NLP), l'apparition des architectures de type Transformer (Vaswani et al., 2017) a permis des avancées majeures dans la compréhension automatique de texte. Des modèles comme BERT (2018) et ses variantes francophones telles que CamemBERT et FlauBERT sont désormais standards pour des tâches telles que la classification multi-label.

La détection automatique de discours haineux est un domaine de recherche en plein essor, motivé par la montée des contenus problématiques sur les plateformes en ligne. Cependant, ce champ reste complexe, en raison :

de la diversité des définitions du discours haineux,

de la grande variété des cibles (origine, religion, genre, orientation sexuelle, handicap, etc.),

et des différences linguistiques et culturelles selon les corpus.

De nombreux travaux se sont penchés sur la caractérisation, la création de jeux de données et l'amélioration des performances de détection.

2.2 Analyses critiques des jeu de données existants

2.2.1 "Toxic, Hateful, Offensive or Abusive ? What Are We Really Classifying ?" (2020)

Dans cet article, les auteurs analysent plusieurs jeux de données utilisés pour entraîner des modèles de détection de discours haineux, notamment :

Hate Speech and Offensive Language Dataset (Jeux de données utilisés), Automated Hate Speech Detection Dataset (Jeux de données utilisés), ainsi que quatre autres corpus similaires.

Leur étude montre que les définitions du "discours haineux" varient significativement d'un jeu de donnée à l'autre, ce qui rend les comparaisons directes difficiles. Ils mettent également en évidence que la plupart des jeux de données présentent des déséquilibres de classes importants et des annotations parfois subjectives.

Conclusion clé : Le manque d'uniformité dans la construction des jeux de données a un impact direct sur les performances et la généralisabilité des modèles.

2.3 Travaux de recherche approfondis (Thèses)

2.3.1 Thèse de Marzieh Mozafari (2021)

Cette thèse explore en profondeur la détection du discours de haine sur les réseaux sociaux, en s'appuyant sur :

Des approches de Machine Learning supervisé (SVM, Random Forest),

Le fine-tuning de modèles de langage (BERT, RoBERTa),
L'analyse des biais présents dans les jeu de données.
Points clés :
Mise en évidence de la difficulté à généraliser sur des plateformes différentes (Twitter, YouTube, Facebook).
Proposition de techniques d'augmentation de données pour pallier le manque de corpus multilingues de qualité.

2.3.2 Thèse de Tulika Bose (2023)

Dans cette thèse, l'auteure s'intéresse à la détection de discours haineux multilingue et multi-cible :
Utilisation de représentations multilingues pour améliorer les performances sur des textes non anglophones.
Étude spécifique sur la capacité des modèles à détecter plusieurs cibles simultanément (ex : haine envers la religion et le genre).
Points clés :
Les modèles multilingues pré-entraînés (XLM-RoBERTa, mBERT) surpassent les modèles monolingues dans les contextes mixtes.
La tâche multi-label est plus réaliste mais aussi plus difficile que la classification binaire classique.

2.4 Études sociolinguistiques sur les discours haineux

2.4.1 "Tweets injurieux et haineux contre les journalistes et les « merdias »" (2018)

Cet article analyse un corpus de tweets insultants et haineux visant des journalistes pendant l'élection présidentielle française de 2017. Il montre que les attaques sont souvent indirectes, utilisant l'ironie, le sarcasme ou des stéréotypes implicites.
Enseignement pour notre projet :
La complexité linguistique du discours haineux nécessite des modèles capables d'analyser non seulement la sémantique mais aussi le sous-texte (ironie, insinuations).

2.5 Approches multilingues du discours haineux

2.5.1 "Multilingual and Multitarget Hate Speech Detection in Tweets" (2020)

Les auteurs proposent un modèle multitâche capable de détecter :
Le discours haineux dans plusieurs langues,
La ou les cibles spécifiques du discours.
Ils démontrent que l'apprentissage multitâche améliore la performance globale et la robustesse, même dans des langues sous-représentées.

2.5.2 "Multilingual and Multi-Aspect Hate Speech Analysis" (2019)

Dans ce travail, l'analyse porte à la fois sur :

— Le contenu du discours,

— La cible,

— L'aspect émotionnel associé (colère, mépris, etc.).

Cette approche enrichit la classification classique en fournissant une analyse multi-aspect, ce qui est particulièrement pertinent pour les discours complexes.

2.6 Synthèse

Les travaux de l'état de l'art mettent en lumière plusieurs points clés :

— L'importance de disposer de jeux de données bien définis et équilibrés,

— La pertinence des approches multilingues dans un contexte globalisé,

— La nécessité de s'adapter aux spécificités linguistiques et culturelles pour détecter correctement les discours haineux,

— L'intérêt croissant pour des tâches de classification multi-label et multi-aspect, plus représentatives de la réalité.

Ces constats ont orienté nos choix méthodologiques, tant dans la construction des jeux de données que dans le choix des modèles expérimentés dans ce projet.

3 Constitution des jeux de données

3.1 Jeux de données utilisés

Pour construire notre base d'apprentissage, nous avons utilisé plusieurs jeux de données publics disponibles sur des plateformes de recherche ou via des initiatives open-source. Cependant, nous avons constaté que peu de jeux de données existaient en français, et encore moins adaptés à des tâches de classification multi-label ciblant précisément des catégories comme l'homophobie, le racisme ou l'incitation à la violence.

Les principaux jeux de données exploités sont les suivants :

— **CONAN Dataset** (en français et en anglais) : corpus de narratifs contre le discours de haine, multilingue.

— **MLMA Hate Speech Dataset** (anglais et français) : corpus annoté en plusieurs catégories de haine (race, religion, orientation sexuelle).

— **Hate Speech and Offensive Language Dataset** : corpus anglophone historique, souvent utilisé dans les benchmarks.

— **FTR-dataset** (français) : corpus de tweets francophones portant sur le racisme.

— **ETHOS Hate Speech Dataset** : corpus anglais annoté en multi-label.

— **White Supremacist Forum Dataset** : corpus de forums haineux anglophones.

Ces jeux de données, bien que diversifiés, présentaient des différences en termes de structure et d'annotations. Un important travail de normalisation a été nécessaire pour les rendre exploitables dans un cadre multi-label homogène.

3.2 Création des jeux de données finaux

À partir des sources collectées, nous avons construit quatre jeux de données finaux :

— **Français uniquement** : fusion des jeux de données francophones (CONAN_fr, FTR, MLMA_fr).

— **Anglais uniquement** : fusion des jeux de données anglophones (Davidson, Ethos, Vicomtech, CONAN_en, MultiTarget CONAN).

— **Français + Anglais mélangés** : union des corpus français et anglais sans traduction.

Chaque exemple est annoté selon un schéma multi-label couvrant 11 catégories, telles que `injure_insulte`, `racial`, `religieux`, `lgbtq`, `menace`, etc. Vous pouvez retrouvé plus d'explication sur chacune de ces catégories en **Annexe**

3.3 Prétraitement et nettoyage

Les étapes de traitement des données incluent :

— **Normalisation des colonnes** : alignement des colonnes texte et des labels sur un schéma commun.

— **Suppression des doublons** : élimination des entrées redondantes après fusion des jeu de données.

— **Filtrage des textes vides ou trop courts** : suppression des entrées avec des textes inférieurs à un seuil minimal de mots.

— **Uniformisation des labels** : harmonisation des annotations pour garantir la compatibilité multi-label (présence de 0.5 pour les catégories non renseignées).

— **Découpage en ensembles** : division en train (70%), validation (10%) et test (20%) de manière stratifiée lorsque possible.

Ce processus a permis de garantir une base de données cohérente et adaptée à la tâche multi-étiquettes. C'est seulement après nettoyage que les jeux de données finaux ont été divisés en ensembles d'entraînement, de validation et de test selon une répartition classique (70% / 10% / 20%). La séparation des jeux d'entraînement, validation et test a été réalisée par stratification, garantissant une répartition homogène des différentes catégories de labels à travers les ensembles. Cela permet d'éviter des biais d'entraînement ou de validation, assurant ainsi une évaluation fiable des modèles. La taille finale des jeux de données, ainsi que la répartition des exemples par catégorie, seront détaillées dans la section des **Résultats expérimentaux**.

4 Approches expérimentales

4.1 Fine-tuning de modèles pré-entraînés

Nous avons choisi d'adopter une approche basée sur le fine-tuning de modèles de langage pré-entraînés pour adapter des représentations générales à notre tâche spécifique de classification multi-label de contenus injurieux.

Trois modèles ont été sélectionnés pour cette expérimentation :

— **CamemBERT** : modèle francophone basé sur l'architecture RoBERTa, pré-entraîné sur le corpus OSCAR (français).

— **FlauBERT** : modèle de langue française pré-entraîné sur des sources diversifiées (Wikipedia, CommonCrawl, livres, etc.).

— **XLM-RoBERTa** : modèle multilingue robuste couvrant 100 langues, basé sur l'architecture RoBERTa.

Ces modèles ont été choisis pour évaluer l'impact de l'origine linguistique du pré-entraînement (monolingue vs multilingue) sur la détection de discours haineux en français.

Le fine-tuning a été réalisé selon une procédure standard adaptée pour une tâche de classification multi-label :

— Utilisation de la fonction de perte **BCEWithLogitsLoss** adaptée à la classification binaire indépendante par label.

— Modification du head de classification pour produire 11 scores correspondant aux 11 labels.

Un soin particulier a été apporté à l'équilibrage entre sous-apprentissage et surapprentissage, notamment par la surveillance des courbes de loss et de F1-score sur l'ensemble de validation à chaque époque.

4.1.1 Optimisations matérielles

L'entraînement a été réalisé sur un matériel personnel disposant d'une carte graphique NVIDIA RTX 4060 Ti (8 Go VRAM). Afin d'optimiser la consommation mémoire et accélérer les calculs, nous avons activé :

— **Précision mixte (fp16)** : permettant de réduire la taille des calculs sans perte notable de précision.

— **Gradient Accumulation** : utilisé pour certains modèles lourds (notamment FlauBERT) pour simuler un batch size plus élevé tout en respectant les contraintes de mémoire GPU.

La parallélisation du chargement des données a également été activée (`dataloader_num_workers=4`) pour limiter les goulets d'étranglement CPU lors de l'entraînement.

4.2 Exploration théorique du prompt engineering sur les LLM

Outre le fine-tuning traditionnel, nous avons exploré théoriquement l'utilisation du **prompt engineering** appliqué aux grands modèles de langage (LLMs) comme Deepseek, Granite, ou encore Mistral-small.

L'idée sous-jacente est que ces modèles, ayant été pré-entraînés sur des corpus extrêmement vastes et variés, sont capables de comprendre des nuances complexes, telles que l'ironie, le sarcasme ou les expressions codées typiques des discours haineux.

Cependant, pour des raisons matérielles (coût, temps d'inférence), cette approche n'a pas pu être testée sur l'ensemble des jeu de données et se concentrera donc sur le jeu de donnée Français uniquement.

5 Entraînement des modèles

5.1 Objectif de l'entraînement

L'objectif de cette phase est d'adapter des modèles de langage pré-entraînés (CamemBERT, FlauBERT, XLM-RoBERTa) à la tâche spécifique d'identification multi-label de contenus injurieux, racistes et homophobes en français. Nous avons pour cela appliqué un processus de fine-tuning supervisé, en utilisant nos jeux de données multi-étiquettes conçus à partir de diverses sources francophones et anglophones.

5.2 Paramètres d'entraînement

L'entraînement des modèles a été réalisé en utilisant les hyperparamètres et paramètres suivants :

Paramètre	Valeur	Justification
Learning rate	2e-5	Standard pour le fine-tuning de modèles BERT-like
Batch size (train/val)	8 / 8	Optimisé pour l'utilisation mémoire tout en maintenant une stabilité de l'entraînement
Nombre d'époques	12	Permet d'atteindre la convergence tout en surveillant le surapprentissage
Weight decay	0.01	Pour éviter l'overfitting via régularisation L2
Optimiseur	AdamW	Optimiseur adapté aux modèles Transformers pour une convergence stable
Précision mixte (fp16)	Activée	Accélération de l'entraînement et réduction de la consommation VRAM
Stratégie d'évaluation	Par époque	Permet d'évaluer la performance après chaque époque
Stratégie de sauvegarde	Par époque	Sauvegarde le modèle après chaque époque pour sécuriser les performances intermédiaires
Modèle sauvegardé	Meilleur modèle (F1)	Chargement automatique du meilleur modèle à la fin de l'entraînement
Nombre de workers	4	Accélération du chargement des données en utilisant le parallélisme
Limite de sauvegarde	1 modèle	Économie d'espace disque en conservant uniquement le meilleur modèle
Log	Toutes les 10 étapes	Suivi précis de l'évolution de la perte pendant l'entraînement

TABLE 1 – Paramètres utilisés pour l'entraînement des modèles.

Pour optimiser les performances, nous avons activé l'option `load_best_model_at_end=True` afin de conserver automatiquement le modèle ayant obtenu le meilleur F1-score sur le jeu de validation. De plus, la métrique `f1` est utilisée comme critère pour sélectionner le modèle optimal.

Les logs sont enregistrés toutes les 10 étapes afin de suivre l'évolution de la perte durant l'entraînement. Enfin, nous utilisons l'option `save_total_limit=1` pour limiter la sauvegarde à un seul modèle, évitant ainsi un encombrement du disque.

5.3 Architecture et adaptation du modèle

Afin d'adapter les modèles à notre tâche spécifique, nous avons apporté les modifications suivantes :

Nous avons conservé la couche de classification traditionnelle (couche linéaire) en sortie du modèle pré-entraîné. Cependant, pour traiter la nature multi-label de notre tâche, nous utilisons la fonction de perte *BCEWithLogitsLoss* (Binary Cross-Entropy avec sigmoïde).

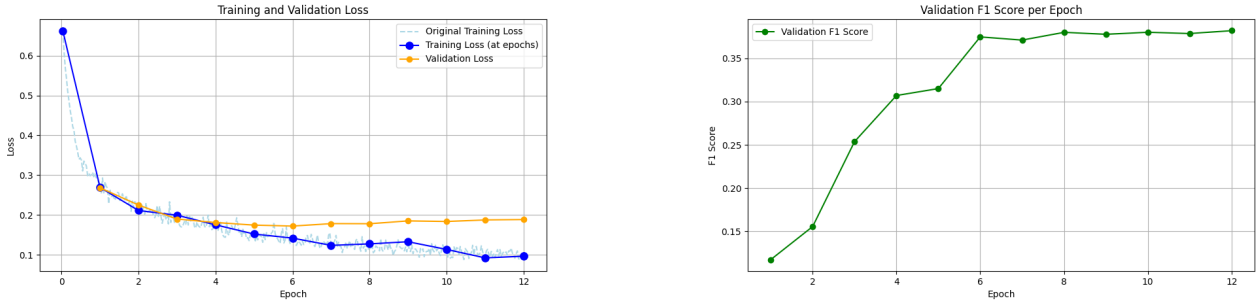
Contrairement à une classification multi-classes classique utilisant une fonction *softmax*, la *BCEWithLogitsLoss* permet d'appliquer une sigmoïde sur chaque logit de manière indépendante, ce qui est essentiel pour notre problématique où plusieurs étiquettes peuvent être présentes simultanément. Cette approche nous permet d'obtenir des probabilités par classe sans les contraindre à se sommer à 1, ce qui est crucial pour une classification multi-label.

5.4 Suivi de l'entraînement

Nous présentons ci-dessous les résultats obtenus sur chaque jeu de donnée pour chacun des trois modèles expérimentés : CamemBERT, FlauBERT, et XLM-RoBERTa. Pour chaque modèle, nous analysons l'évolution de la perte et du F1-score au cours de l'entraînement ainsi que des principales observations.

5.4.1 Entraînement sur jeu de données Français

CamemBERT L'entraînement du modèle CamemBERT sur le jeu de données français montre une amélioration progressive des performances au fil des époques. La courbe de perte (Figure 1) montre une diminution significative de la perte d'entraînement dès les premières époques, atteignant une valeur stable autour de l'époque 8. La perte de validation suit une tendance similaire mais reste légèrement plus élevée, ce qui peut indiquer une certaine généralisation. En ce qui concerne le F1-score, nous observons une augmentation régulière jusqu'à l'époque 6, suivie d'une stabilisation autour de 0.38, suggérant que le modèle atteint un plateau de performance sur ce jeu de données.



(a) Loss Train/Validation

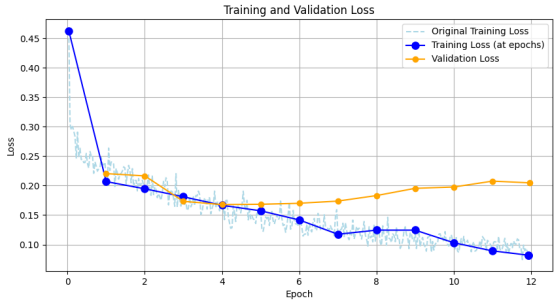
(b) F1-Score Validation

FIGURE 1 – Évolution de la perte et du F1-score pour CamemBERT durant l'entraînement.

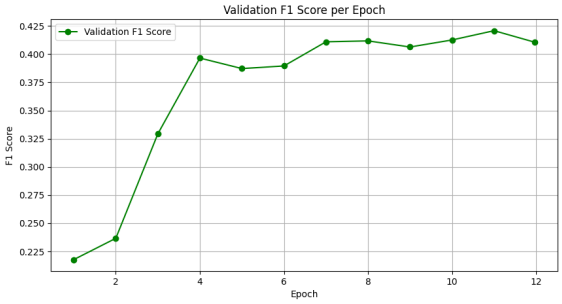
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.2776	0.2682	0.1173	0.372
2	0.2146	0.2248	0.1555	0.3975
3	0.1922	0.1898	0.2537	0.4767
4	0.1772	0.1811	0.307	0.5163
5	0.1563	0.1746	0.3151	0.5318
6	0.1352	0.1722	0.3748	0.5375
7	0.1554	0.1783	0.3711	0.5375
8	0.1128	0.1781	0.3801	0.5389
9	0.1077	0.1852	0.3779	0.5318
10	0.1106	0.1837	0.3801	0.5446
11	0.0968	0.1876	0.3787	0.5375
12	0.0979	0.1885	0.382	0.5347

TABLE 2 – Métriques d'entraînement et d'évaluation par époque

FlauBERT L'entraînement du modèle FlauBERT sur le jeu de données français montre une amélioration notable des performances durant les premières époques. La courbe de perte (Figure 2) diminue fortement au début, puis atteint une stabilité après environ 6 époques. Cependant, la perte de validation tend à augmenter légèrement à partir de l'époque 8, ce qui pourrait indiquer un début de surapprentissage. Concernant le F1-score, nous remarquons une progression rapide lors des premières époques, atteignant un plateau autour de l'époque 7, avec une légère fluctuation par la suite, ce qui indique une stabilité relative des performances.



(a) Loss Train/Validation



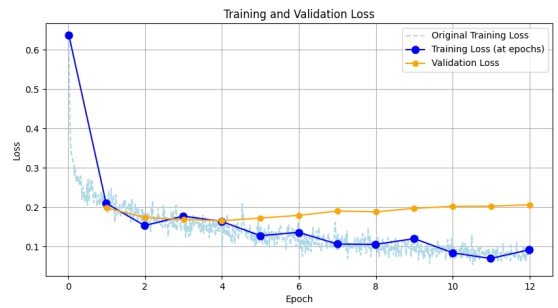
(b) F1-Score Validation

FIGURE 2 – Évolution de la perte et du F1-score pour FlauBERT durant l'entraînement.

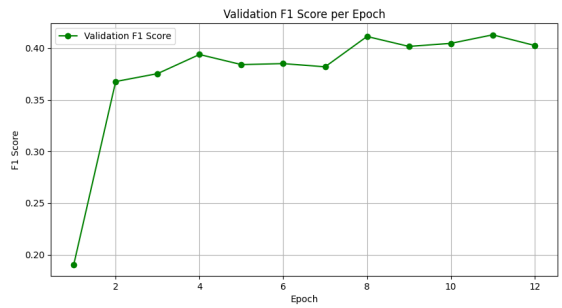
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.2396	0.2205	0.2178	0.2631
2	0.2008	0.2166	0.2366	0.4187
3	0.1843	0.1734	0.3293	0.4851
4	0.1684	0.1679	0.3965	0.4979
5	0.1499	0.1684	0.3872	0.5262
6	0.1329	0.1701	0.3895	0.5417
7	0.1592	0.1738	0.4109	0.546
8	0.1059	0.1832	0.4118	0.5389
9	0.1055	0.1955	0.4063	0.5276
10	0.1073	0.1977	0.4126	0.5375
11	0.098	0.2077	0.4208	0.529
12	0.0819	0.2048	0.4106	0.5304

TABLE 3 – Métriques d'entraînement et d'évaluation par époque

XLM-RoBERTa L'entraînement du modèle XLM-RoBERTa sur le jeu de données français montre une amélioration rapide du F1-score durant les premières époques, atteignant un plateau à partir de l'époque 4 avec une légère variabilité par la suite. La perte d'entraînement diminue de manière significative jusqu'à l'époque 6, tandis que la perte de validation reste relativement stable après une baisse initiale, ce qui pourrait indiquer une bonne généralisation mais avec un léger surapprentissage dans les dernières époques. Le F1-score reste globalement supérieur à 0.4 à partir de l'époque 8, suggérant une stabilité des performances.



(a) Loss Train/Validation



(b) F1-Score Validation

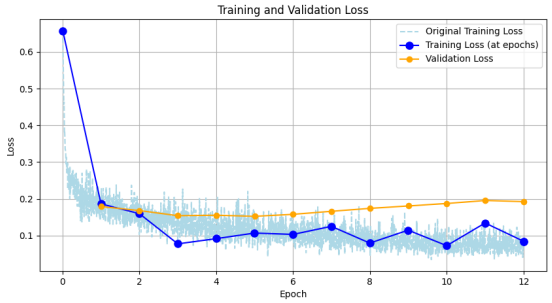
FIGURE 3 – Évolution de la perte et du F1-score pour XLM-RoBERTa durant l'entraînement.

Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.2101	0.1981	0.1905	0.4342
2	0.1532	0.1739	0.3677	0.5219
3	0.1772	0.1683	0.3754	0.5205
4	0.1638	0.1655	0.3939	0.5361
5	0.1275	0.1723	0.3841	0.5502
6	0.136	0.1792	0.385	0.5304
7	0.1063	0.1902	0.382	0.5446
8	0.1053	0.1881	0.4113	0.5502
9	0.1201	0.197	0.4017	0.5431
10	0.0817	0.2021	0.4047	0.5276
11	0.0695	0.2026	0.4129	0.5262
12	0.0916	0.206	0.4027	0.529

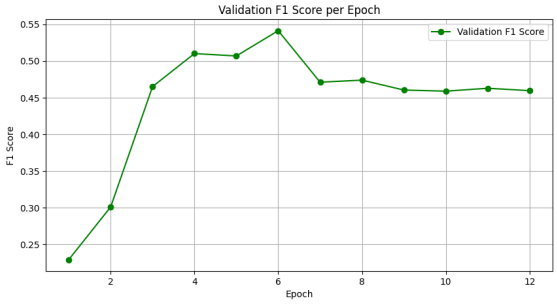
TABLE 4 – Métriques d'entraînement et d'évaluation par époque

5.4.2 Entraînement sur jeu de données Anglais

CamemBERT L'entraînement du modèle CamemBERT sur le jeu de données anglais montre une amélioration rapide du F1-score durant les premières époques, atteignant un pic autour de l'époque 6 avec un score de 0.54. Cependant, à partir de l'époque 7, on observe une légère diminution et stabilisation autour de 0.46, ce qui pourrait indiquer un surapprentissage léger. La perte d'entraînement diminue rapidement dans les premières époques et atteint un plateau autour de l'époque 4. La perte de validation, quant à elle, montre une tendance à l'augmentation après l'époque 6, renforçant l'hypothèse d'un surapprentissage.



(a) Loss Train/Validation



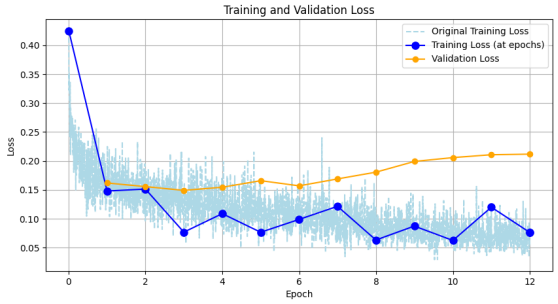
(b) F1-Score Validation

FIGURE 4 – Évolution de la perte et du F1-score pour CamemBERT durant l'entraînement.

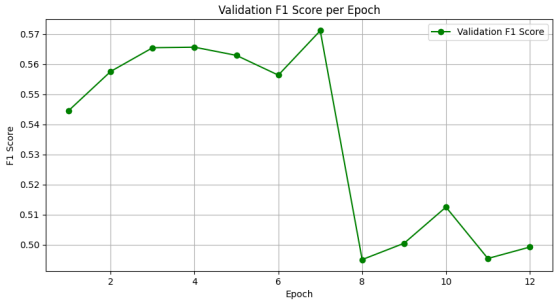
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1863	0.179	0.2293	0.7349
2	0.1597	0.1681	0.3011	0.769
3	0.0775	0.1541	0.4652	0.801
4	0.0917	0.1553	0.5102	0.8177
5	0.1069	0.1524	0.5069	0.8167
6	0.1032	0.1579	0.5414	0.806
7	0.1252	0.1661	0.4712	0.8079
8	0.0796	0.1741	0.474	0.8127
9	0.1147	0.1806	0.4605	0.8046
10	0.0886	0.1874	0.4589	0.8075
11	0.1343	0.1954	0.4629	0.8069
12	0.0845	0.1925	0.4597	0.8039

TABLE 5 – Métriques d'entraînement et d'évaluation par époque

FlauBERT L'entraînement du modèle FlauBERT sur le jeu de données anglais montre une amélioration rapide du F1-score durant les premières époques, atteignant un pic à l'époque 7 avec un score de 0.5711. Cependant, après cette époque, le F1-score chute brusquement et reste instable, oscillant autour de 0.50. Cette dégradation des performances après l'époque 7 pourrait indiquer un problème de surapprentissage ou de dégradation du modèle. La perte d'entraînement diminue de manière significative au début et atteint un minimum autour de l'époque 6, mais la perte de validation commence à augmenter à partir de l'époque 7, ce qui confirme l'hypothèse de surapprentissage.



(a) Loss Train/Validation



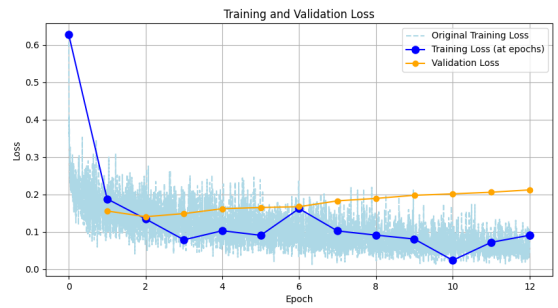
(b) F1-Score Validation

FIGURE 5 – Évolution de la perte et du F1-score pour FlauBERT durant l'entraînement.

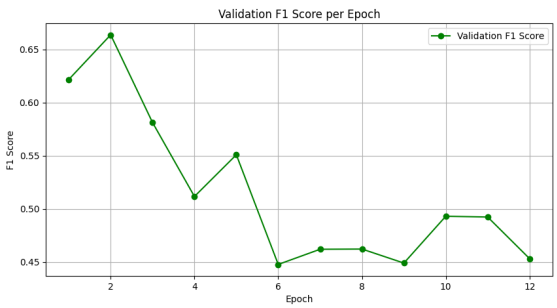
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.148	0.1618	0.5446	0.8016
2	0.1513	0.1556	0.5576	0.8165
3	0.0768	0.1491	0.5655	0.8129
4	0.1088	0.1545	0.5656	0.8173
5	0.0766	0.1658	0.5629	0.8221
6	0.0988	0.1569	0.5564	0.8062
7	0.1215	0.1689	0.5711	0.8067
8	0.0632	0.1806	0.495	0.801
9	0.0873	0.1991	0.5005	0.8012
10	0.09	0.2058	0.5125	0.7972
11	0.12	0.2108	0.4954	0.7974
12	0.053	0.2118	0.4992	0.7968

TABLE 6 – Métriques d'entraînement et d'évaluation par époque

XLM-RoBERTa L'entraînement du modèle XLM-RoBERTa sur le jeu de données anglais montre une amélioration rapide du F1-score lors des deux premières époques, atteignant un pic de 0.6637 à l'époque 2. Cependant, à partir de l'époque 3, les performances se détériorent, le F1-score chute progressivement et oscille autour de 0.45 à partir de l'époque 6, indiquant un possible surapprentissage. La perte d'entraînement diminue significativement lors des premières époques mais montre des fluctuations notables après l'époque 6, tandis que la perte de validation augmente légèrement, ce qui suggère également un problème de généralisation.



(a) Loss Train/Validation



(b) F1-Score Validation

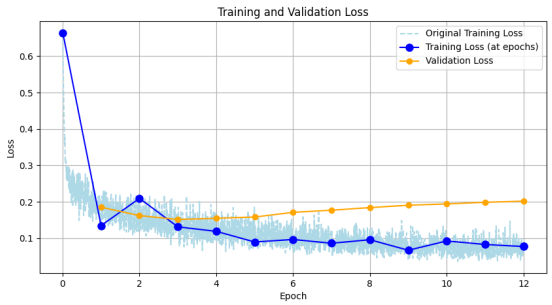
FIGURE 6 – Évolution de la perte et du F1-score pour XLM-RoBERTa durant l'entraînement.

Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1877	0.1554	0.6217	0.815
2	0.1342	0.1401	0.6637	0.8288
3	0.0782	0.1483	0.5812	0.8347
4	0.1025	0.1614	0.5116	0.8215
5	0.0902	0.1646	0.5509	0.82
6	0.1616	0.1668	0.4477	0.8027
7	0.1019	0.1825	0.462	0.796
8	0.0908	0.189	0.4622	0.8144
9	0.0803	0.1974	0.449	0.7954
10	0.0631	0.2012	0.493	0.8048
11	0.0714	0.2055	0.4923	0.8067
12	0.0904	0.2118	0.4532	0.8029

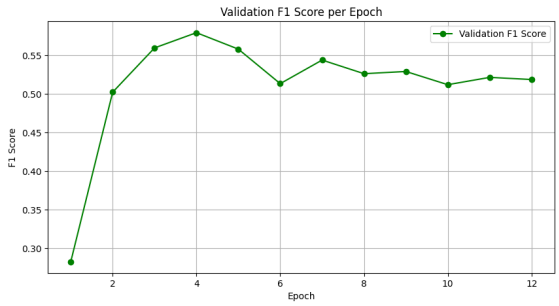
TABLE 7 – Métriques d'entraînement et d'évaluation par époque

5.4.3 Entraînement sur jeu de données Français|Anglais

CamemBERT L'entraînement du modèle CamemBERT sur le jeu de données combiné français et anglais montre une amélioration notable du F1-score durant les premières époques, avec un pic à l'époque 4 autour de 0.58. Par la suite, le modèle montre une légère instabilité, avec des variations du F1-score autour de 0.52, ce qui pourrait indiquer un compromis entre la performance sur les deux langues. La perte d'entraînement diminue fortement au début puis se stabilise autour de l'époque 6, tandis que la perte de validation reste stable avec une légère tendance à la hausse après l'époque 6, suggérant un possible surapprentissage léger en fin d'entraînement.



(a) Loss Train/Validation



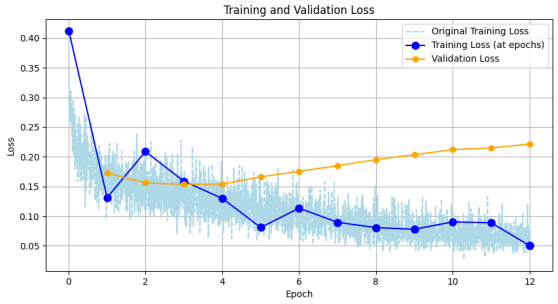
(b) F1-Score Validation

FIGURE 7 – Évolution de la perte et du F1-score pour CamemBERT durant l'entraînement.

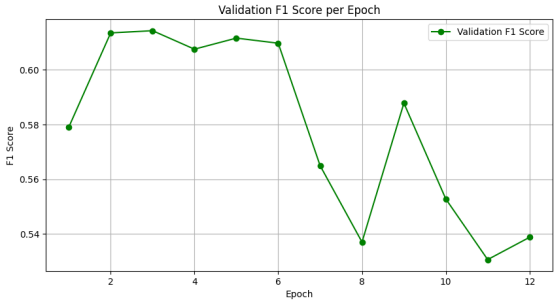
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1694	0.1854	0.2827	0.7023
2	0.1198	0.1623	0.5024	0.755
3	0.1331	0.1516	0.5598	0.7705
4	0.1242	0.155	0.5795	0.7712
5	0.1164	0.1584	0.5582	0.7763
6	0.0893	0.1713	0.5134	0.7658
7	0.1007	0.1771	0.544	0.7729
8	0.1239	0.1843	0.5264	0.7681
9	0.0432	0.1907	0.5292	0.7747
10	0.082	0.1944	0.5121	0.7643
11	0.0595	0.1988	0.5216	0.7709
12	0.0572	0.202	0.5188	0.7703

TABLE 8 – Métriques d'entraînement et d'évaluation par époque

FlauBERT L'entraînement du modèle FlauBERT sur le jeu de données français+anglais montre une amélioration notable du F1-score lors des premières époques, atteignant un pic autour de l'époque 3 avec un score de 0.6143. Cependant, à partir de l'époque 7, le modèle montre une dégradation des performances avec une baisse du F1-score autour de 0.53 à la fin de l'entraînement, ce qui peut indiquer un surapprentissage ou une difficulté à généraliser sur les deux langues simultanément. La perte d'entraînement diminue fortement au début puis se stabilise autour de l'époque 6, tandis que la perte de validation augmente à partir de l'époque 7, renforçant l'hypothèse de surapprentissage.



(a) Loss Train/Validation



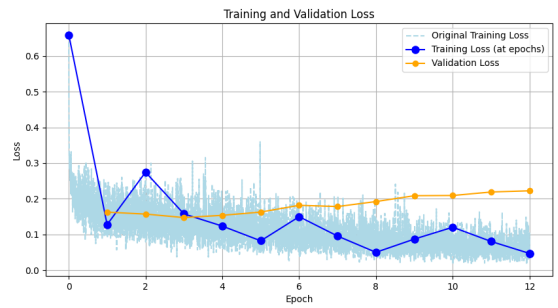
(b) F1-Score Validation

FIGURE 8 – Évolution de la perte et du F1-score pour FlauBERT durant l'entraînement.

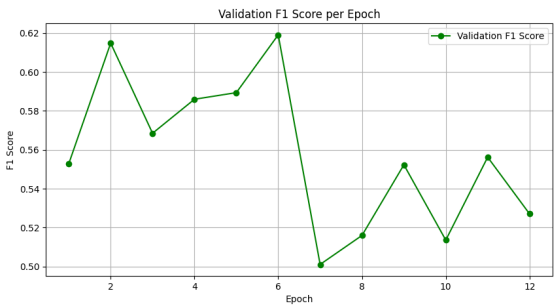
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1456	0.1725	0.5791	0.7599
2	0.1058	0.1559	0.6135	0.7681
3	0.1412	0.1536	0.6143	0.7723
4	0.1109	0.1539	0.6076	0.7776
5	0.1235	0.1662	0.6116	0.7721
6	0.0834	0.1755	0.6098	0.7685
7	0.1011	0.1851	0.565	0.7574
8	0.1099	0.1952	0.537	0.7616
9	0.0606	0.2035	0.5879	0.7656
10	0.0615	0.2121	0.5528	0.7636
11	0.0579	0.2149	0.5306	0.7568
12	0.0465	0.2213	0.5388	0.7585

TABLE 9 – Métriques d'entraînement et d'évaluation par époque

XLM-RoBERTa L'entraînement du modèle XLM-RoBERTa sur le jeu de données français+anglais montre une amélioration rapide du F1-score lors des premières époques, avec un pic à l'époque 6 autour de 0.619. Toutefois, après cette époque, le modèle montre une variabilité importante des performances, avec des baisses significatives du F1-score autour de l'époque 8 et des oscillations entre 0.51 et 0.56 lors des dernières époques. La perte d'entraînement diminue fortement au début et reste relativement basse, mais la perte de validation augmente légèrement après l'époque 6, ce qui suggère un surapprentissage dans les dernières étapes d'entraînement.



(a) Loss Train/Validation



(b) F1-Score Validation

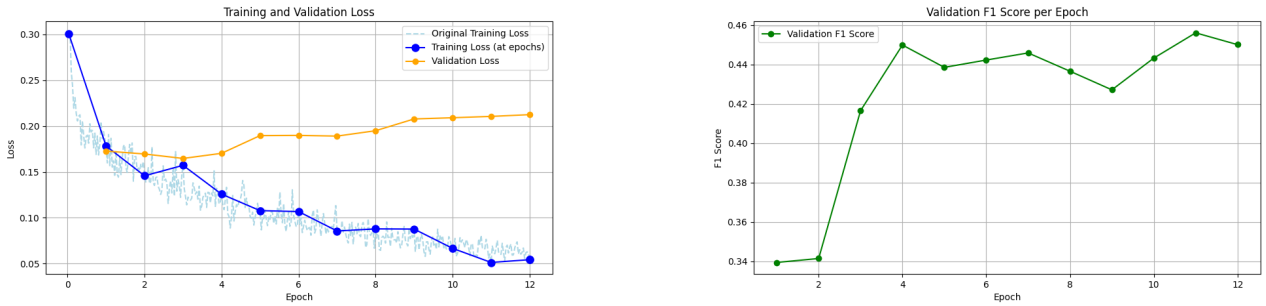
FIGURE 9 – Évolution de la perte et du F1-score pour XLM-RoBERTa durant l'entraînement.

Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1489	0.1623	0.5529	0.7607
2	0.1283	0.1569	0.6148	0.7802
3	0.1392	0.1471	0.5685	0.7721
4	0.1014	0.1534	0.586	0.7827
5	0.0493	0.1625	0.5894	0.7802
6	0.1039	0.1815	0.619	0.7873
7	0.0812	0.178	0.5011	0.7645
8	0.1167	0.192	0.516	0.7723
9	0.0514	0.2084	0.5523	0.7727
10	0.0856	0.2089	0.5136	0.7598
11	0.0865	0.2187	0.5562	0.7627
12	0.1036	0.2221	0.5271	0.7634

TABLE 10 – Métriques d'entraînement et d'évaluation par époque

5.4.4 Entraînement sur jeu de données Anglais→Français

CamemBERT L'entraînement du modèle CamemBERT du jeu de données anglais puis sur le jeu de données français montre une progression modérée des performances. Le F1-score augmente de manière significative lors des premières époques, atteignant un pic autour de l'époque 4 avec un score d'environ 0.45. Après cette phase de montée, le modèle présente des oscillations, avec une légère amélioration vers les dernières époques, suggérant que le transfert d'apprentissage du modèle depuis l'anglais vers le français reste partiellement efficace. La perte d'entraînement diminue régulièrement tandis que la perte de validation montre une tendance à augmenter après l'époque 5, ce qui pourrait indiquer une certaine difficulté de généralisation après le transfert.



(a) Loss Train/Validation

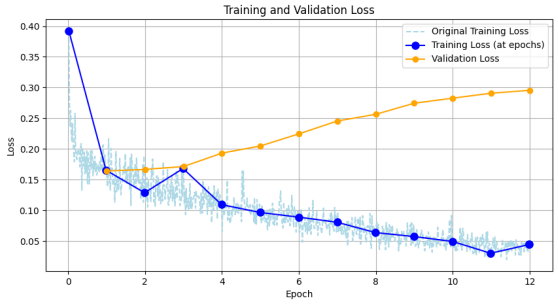
(b) F1-Score Validation

FIGURE 10 – Évolution de la perte et du F1-score pour CamemBERT durant l'entraînement.

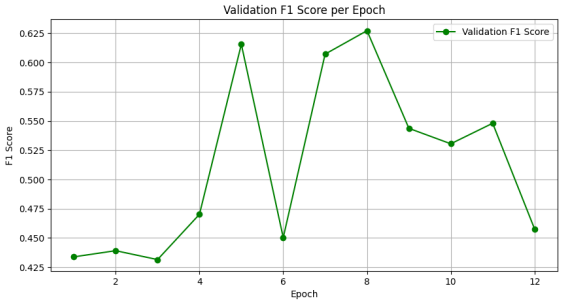
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1859	0.1728	0.3393	0.512
2	0.1612	0.1696	0.3413	0.5318
3	0.1468	0.1647	0.4165	0.5403
4	0.117	0.1703	0.4499	0.5559
5	0.1073	0.1896	0.4385	0.553
6	0.0907	0.1899	0.4422	0.5403
7	0.1133	0.1891	0.4459	0.546
8	0.0668	0.195	0.4366	0.5629
9	0.0786	0.2078	0.4271	0.5375
10	0.0697	0.2091	0.4432	0.546
11	0.0605	0.2106	0.4561	0.5474
12	0.0613	0.2125	0.4501	0.5375

TABLE 11 – Métriques d'entraînement et d'évaluation par époque

FlauBERT L'entraînement du modèle FlauBERT du jeu de données anglais puis sur le jeu de données français montre des variations importantes des performances. Le F1-score atteint un pic remarquable à l'époque 8 avec une valeur de 0.6272, mais il redescend rapidement dans les dernières époques, indiquant une instabilité de la généralisation après le transfert d'apprentissage. La perte d'entraînement diminue régulièrement tandis que la perte de validation augmente significativement après l'époque 6, ce qui suggère un fort surapprentissage lorsque le modèle est réentraîné sur le jeu de données français.



(a) Loss Train/Validation



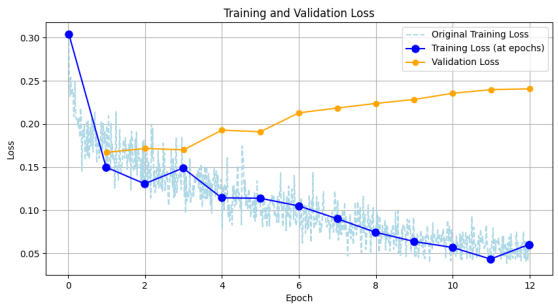
(b) F1-Score Validation

FIGURE 11 – Évolution de la perte et du F1-score pour FlauBERT durant l'entraînement.

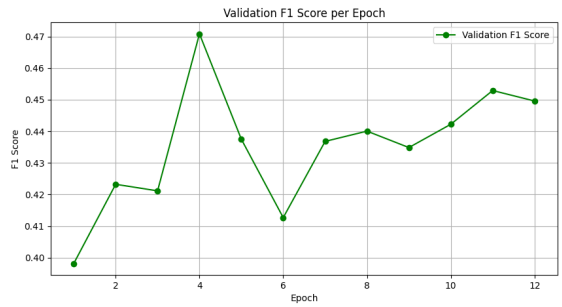
Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1653	0.1641	0.4338	0.5403
2	0.1287	0.1667	0.4391	0.5573
3	0.1681	0.1714	0.4315	0.5347
4	0.1093	0.1932	0.4702	0.5417
5	0.0965	0.2049	0.6156	0.5474
6	0.089	0.2246	0.4503	0.5545
7	0.0806	0.2455	0.6072	0.5375
8	0.0637	0.2564	0.6272	0.5347
9	0.0573	0.2744	0.5437	0.5064
10	0.0571	0.2827	0.5305	0.512
11	0.0301	0.2907	0.5481	0.5149
12	0.0443	0.2955	0.4577	0.5106

TABLE 12 – Métriques d'entraînement et d'évaluation par époque

XLM-RoBERTa L'entraînement du modèle XLM-RoBERTa du jeu de données anglais puis sur le jeu de données français montre une amélioration progressive du F1-score durant les premières époques, avec un pic notable à l'époque 4 atteignant environ 0.47. Après ce pic, le F1-score varie légèrement, oscillant autour de 0.44 dans les dernières époques, ce qui indique une certaine stabilité après un transfert initial. La perte d'entraînement diminue régulièrement au fil des époques, tandis que la perte de validation montre une légère tendance à l'augmentation après l'époque 6, suggérant un risque de surapprentissage sur la langue cible (français).



(a) Loss Train/Validation



afintionF1-Score Validation

FIGURE 12 – Évolution de la perte et du F1-score pour XLM-RoBERTa durant l'entraînement.

Epoch	Training Loss	Validation Loss	F1-Score	Accuracy
1	0.1499	0.1671	0.3982	0.5134
2	0.1307	0.1716	0.4232	0.5163
3	0.1489	0.17	0.4211	0.5276
4	0.1144	0.1929	0.4708	0.4965
5	0.1139	0.191	0.4375	0.5502
6	0.105	0.2129	0.4127	0.5149
7	0.0902	0.2185	0.4368	0.512
8	0.0744	0.2238	0.44	0.5375
9	0.0637	0.2284	0.4348	0.5191
10	0.0668	0.2356	0.4422	0.5149
11	0.0434	0.2398	0.4529	0.5163
12	0.0603	0.2407	0.4496	0.5163

TABLE 13 – Métriques d'entraînement et d'évaluation par époque

5.5 Synthèse de l'entraînement

L'entraînement des modèles CamemBERT, FlauBERT et XLM-RoBERTa a permis d'évaluer leur capacité à détecter des contenus injurieux, racistes et homophobes dans des contextes multilingues (français et anglais).

Globalement, les résultats montrent que les modèles atteignent une performance stable après environ 8 à 10 époques d'entraînement, bien que des signes de surapprentissage apparaissent dans certaines configurations, notamment lorsque les modèles sont entraînés en transfert de langue (anglais vers français). Parmi les trois modèles, XLM-RoBERTa se distingue par une meilleure capacité d'adaptation lors de l'entraînement sur des données multilingues, tandis que FlauBERT montre une plus grande variabilité des performances entre les époques.

Les courbes de perte d'entraînement montrent une diminution rapide au début, puis une stabilisation, tandis que la perte de validation a tendance à augmenter légèrement après la phase de convergence. Cela indique que l'entraînement prolongé au-delà d'une dizaine d'époques peut compromettre la généralisation, en particulier pour des jeux de données combinant des langues différentes.

L'utilisation de la fonction de perte *BCEWithLogitsLoss* s'est révélée pertinente pour la classification multi-label, permettant de traiter chaque classe indépendamment sans contrainte de somme unitaire, ce qui est adapté à la nature du problème abordé.

En termes de stratégies d'entraînement, les résultats indiquent que l'entraînement séparé par langue donne des performances plus stables que l'entraînement combiné ou le transfert de langue, ce qui laisse envisager des pistes d'amélioration, notamment par le biais d'un apprentissage multi-tâche ou d'un équilibrage des données multilingues.

Ainsi, la phase d'entraînement a mis en lumière l'importance de bien équilibrer la durée d'entraînement et la gestion des données multilingues pour améliorer la robustesse et la généralisation des modèles.

6 Utilisation de LLM en Local

6.1 Pourquoi en local ?

Nous avons commencé par utiliser certains types de services en ligne ; Mistral principalement. Cependant, ces services n'offrent pas toujours la possibilité d'utiliser les modèles les plus récents et/ou autant de requêtes que l'on en aurait besoin gratuitement. C'est pour cela que les expérimentations effectuées sur les LLM dans ce projet ont été faites en local sur l'un de nos ordinateurs. Pour finir, afin que l'on puisse comparer de façon identique chaque modèle, tous ont été inférés sur le même GPU : RTX 5080 (16 Go de VRAM).

6.2 Utilisation de Ollama

Pour ces analyses, nous avons utilisé la bibliothèque d'Ollama, nous permettant de tester rapidement et surtout efficacement plusieurs modèles disponibles grâce à leur service. En effet, notre première approche a été d'utiliser Mistral en local grâce à leur bibliothèque mistral_inference puis par la suite celle de huggingFace. Malheureusement, nous avons pu observer des incompatibilités matérielles avec la version de torch requise (incompatibilité de la version CUDA utilisée par la RTX 5080 : CUDA 12.8). Ces raisons nous ont donc orientés vers Ollama qui semble gérer ces problèmes, dont nous n'avions pas de solutions.

6.3 Utilisation de Pydantic

Nous avons passé un certain temps à rechercher des prompts nous permettant de contraindre les LLM à nous fournir des réponses structurées afin d'analyser les réponses données. Lors de nos recherches, nous avons trouvé la bibliothèque Pydantic nous permettant de créer des schémas de réponses très intuitifs. En effet, grâce à cette bibliothèque, il nous suffit de créer des classes python que le modèle devra instancier, nous permettant d'avoir toujours la même structure de réponse.

6.4 Les prompts

Le souci de structure maintenant mis de côté, l'exploration des différents prompt a été assez brève. En effet, dès que le prompt est assez détaillé et explique en détail ce que l'utilisateur attend, les résultats sont toujours semblables à une exception près : le respect de la structure. Nous avons commencé par donner des batchs de 10 phrases à analyser par prompt. Et à de rares occasions, les modèles oubliaient d'analyser une phrase du batch donné. C'est pour cela que nous nous sommes rabattus sur une autre option que nous avons conservée par la suite : analyser phrase par phrase, autrement dit, un prompt envoyé par phrase analysée. Cette méthode n'a pas entraîné de temps d'analyse supplémentaire notable, ce qui était pourtant ce qui nous faisait le plus peur. Voici donc le prompt utilisé ainsi que la structure demandée en réponse aux LLM :

```python
# Define the schema for the response
class SentenceInfo(BaseModel):
    injure_insulte : bool
    hateSpeech: bool
    racial: bool
    religieux: bool
    genre: bool
    lgbtq: bool
    handicap: bool
    origine_national: bool
    politique: bool
    incitation_violence: bool
    menace: bool
```

FIGURE 13 – Structure de réponse

FIGURE 14 – Prompt

6.5 Les LLM utilisés

Nous avons donc analysé les résultats de plusieurs LLM énoncé ci-dessous :

— Phi4-reasoning :14b (11GB)
— Deepseek-r1 :14b (9GB)
— Qwen3 :14b (5.2GB)
— Granite3.3 :8b (4.9GB)
— Mistral-small :22b-instruct-2409-q4_K_M (13GB)

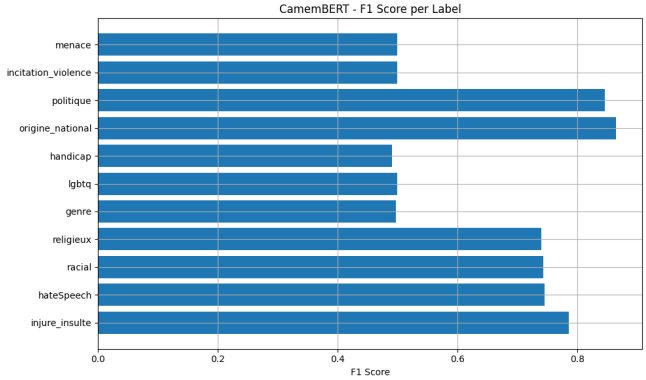
7 Résultats expérimentaux

7.1 MLM

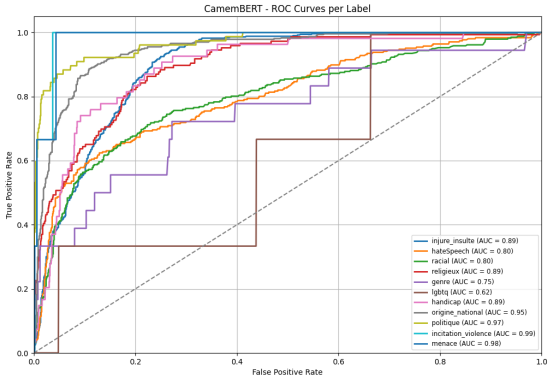
7.1.1 Résultats sur jeu de donnée Français

Jeu de donnée Ratio des labels : Les classes les plus représentées dans le jeu de donnée sont "injure_insulte" (35.7%), "origine_national" (33%), "hateSpeech" (21.6%) et "racial" (23.8%). En revanche, les classes "lgbtq" (0.2%), "incitation_violence" (0.2%), "menace" (0.2%), et "genre" (1.2%) sont significativement sous-représentées.

CamemBERT Les résultats montrent que CamemBERT est particulièrement performant pour les catégories où les exemples sont nombreux et distincts (par exemple, "injure_insulte" et "origine_national"). En revanche, le modèle éprouve des difficultés à bien généraliser sur les catégories rares ou aux frontières sémantiques plus floues (comme "genre" et "lgbtq"). Pour améliorer la détection de ces classes, il serait pertinent d'augmenter le nombre d'exemples pertinents pour compenser le déséquilibre.

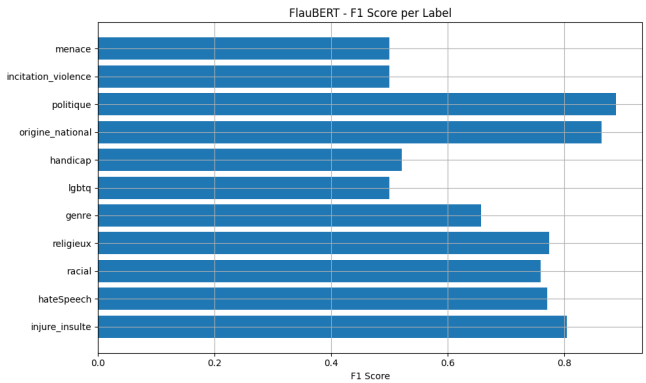


(a) F1-score par catégorie

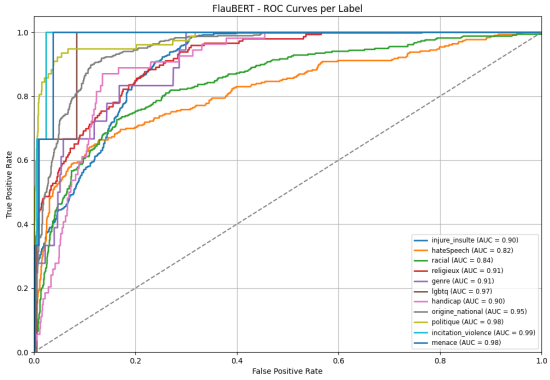


(b) Courbes ROC par catégorie

FlauBERT Les résultats démontre que FlauBERT a une robustesse particulière pour détecter les catégories dominantes, mais sa capacité à identifier les catégories moins fréquentes reste limitée, ce qui souligne l'importance d'un meilleur équilibrage des données pour améliorer la détection des labels rares.

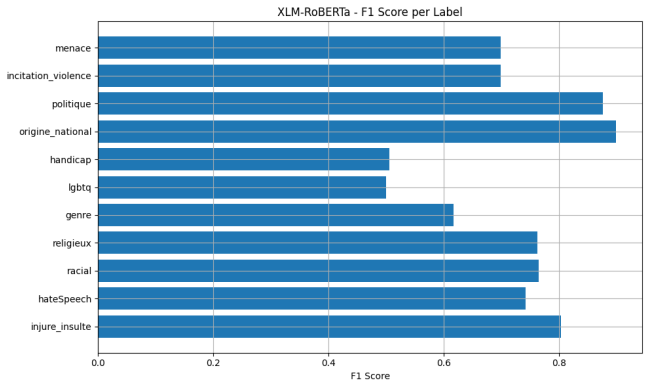


(a) F1-score par catégorie

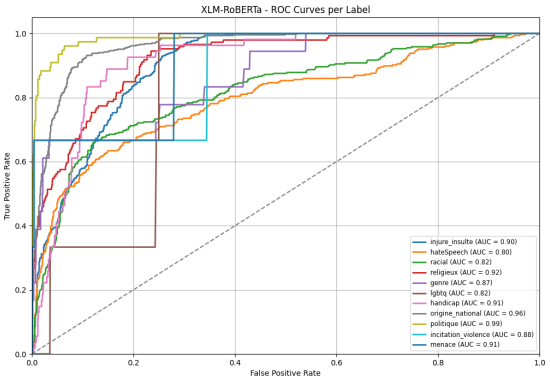


(b) Courbes ROC par catégorie

XLM-RoBERTa Le modèle XLM-RoBERTa présente des résultats globalement satisfaisants, avec un F1-score élevé pour les labels majoritaires tels que *injure_insulte*, *racial*, et *hateSpeech*. Les catégories les plus rares, telles que *lgbtq* et *incitation_violence*, obtiennent des F1-scores plus faibles, ce qui est cohérent avec leur faible présence dans le jeu de données. Concernant les courbes ROC, XLM-RoBERTa obtient des AUC élevés pour la plupart des catégories, notamment *politique* (0.99) et *incitation_violence* (0.88), ce qui témoigne d'une bonne capacité de séparation pour ces classes. Néanmoins, les performances sur les classes rares restent perfectibles, soulignant la difficulté de généralisation sur ces labels minoritaires.



(a) F1-score par catégorie



(b) Courbes ROC par catégorie

Comparaison des modèles CamemBERT affiche un F1-score macro de 0.35, ce qui le place en dessous de FlauBERT (0.42) et XLM-RoBERTa (0.47). En termes de rapidité, XLM-RoBERTa est légèrement plus rapide (37 messages/sec) que CamemBERT (35 messages/sec) et FlauBERT (31 messages/sec).

Modèle	F1-score (macro)	Message/sec	Commentaire
CamemBERT	0.35	35	[Commentaire court pour dire si le modèle est bon]
FlauBERT	0.42	31	[Commentaire court pour dire si le modèle est bon]
XLM-RoBERTa	0.47	37	[Commentaire court pour dire si le modèle est bon]

TABLE 14 – Comparaison finale des performances sur le jeu de test (jeu de données français).

— **CamemBERT** : CamemBERT se distingue par de bonnes performances sur les classes majoritaires comme *injure_ insulte* et *origine_ national*, mais montre des lacunes pour les classes plus rares telles que *lgbtq* et *incitation_ violence*. Cela suggère que ce modèle est efficace lorsque les données sont abondantes et distinctes.

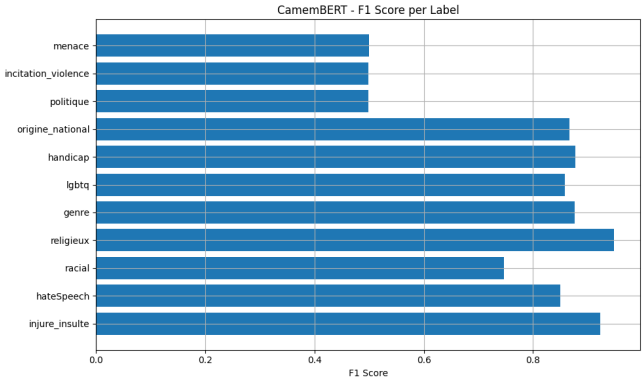
— **FlauBERT** : FlauBERT démontre une robustesse accrue pour la détection des classes fréquentes, tout en rencontrant des difficultés sur les catégories moins représentées, notamment *genre* et *lgbtq*. L'augmentation de la diversité des exemples pour ces catégories pourrait améliorer les performances globales.

— **XLM-RoBERTa** : XLM-RoBERTa présente les meilleures performances globales en termes de F1-score et d'AUC pour la plupart des catégories, notamment *politique* et *incitation_ violence*. Néanmoins, comme les autres modèles, il éprouve des difficultés sur les labels rares. Sa rapidité en fait également un choix favorable pour des tâches en temps réel.

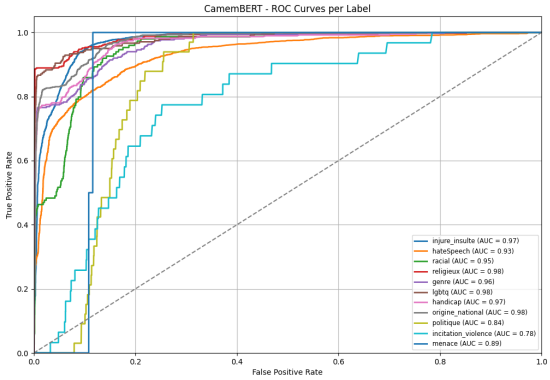
Synthèse : Globalement, XLM-RoBERTa apparaît comme le modèle le plus performant, tant en précision qu'en rapidité. CamemBERT et FlauBERT montrent des performances similaires sur les classes fréquentes, mais FlauBERT surpasse CamemBERT dans la gestion des classes moins représentées. Pour des tâches nécessitant à la fois précision et rapidité, XLM-RoBERTa est le meilleur choix. Cependant, pour améliorer la détection des classes rares, un équilibrage des données reste nécessaire.

7.1.2 Résultats sur jeu de donnée Anglais

CamemBERT CamemBERT démontre une bonne performance sur les données anglaises avec un F1-score macro de 0.55. Le modèle excelle dans la détection des classes fréquentes, comme *injure_insulte* et *origine_national*, mais montre des limites pour les catégories rares telles que *lgbtq* et *incitation_violence*. Les courbes ROC indiquent des AUC élevés pour les classes dominantes, témoignant d'une bonne capacité de séparation.

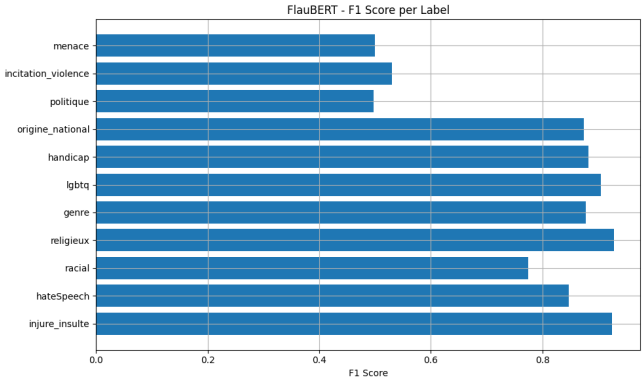


(a) F1-score par catégorie

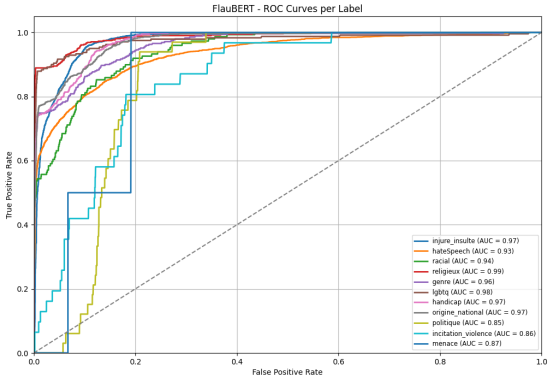


(b) Courbes ROC par catégorie

FlauBERT FlauBERT affiche des performances légèrement supérieures à CamemBERT, avec un F1-score macro de 0.57. Il parvient à mieux capturer les classes intermédiaires et rares, notamment *genre* et *handicap*. L'AUC est également élevé pour les classes majoritaires, mais le modèle reste perfectible sur les catégories peu représentées.

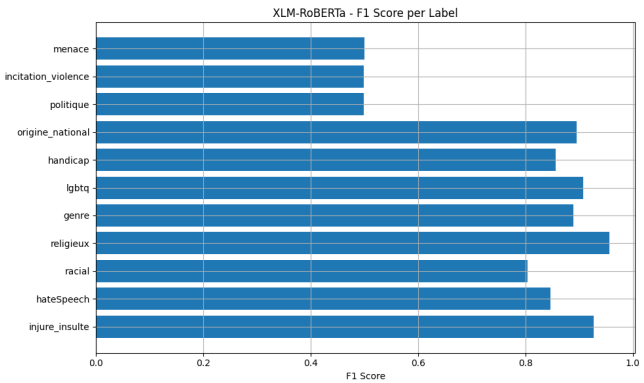


(a) F1-score par catégorie

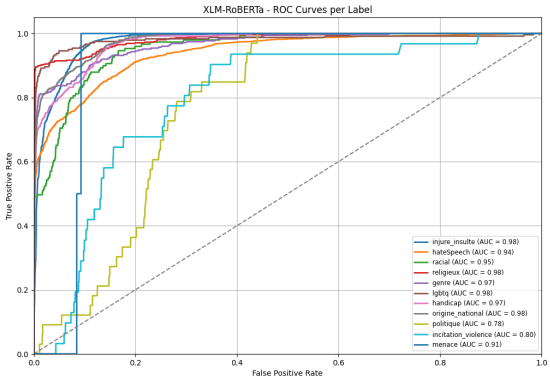


(b) Courbes ROC par catégorie

XLM-RoBERTa XLM-RoBERTa atteint un F1-score macro de 0.575, légèrement supérieur aux deux autres modèles. Sa capacité à capturer les classes majoritaires reste excellente, avec des AUC proches de 1 pour *injure_insulte* et *origine_national*. Cependant, il peine à généraliser sur les classes les plus rares, bien que ses scores restent légèrement supérieurs à ceux de CamemBERT et FlauBERT.



(a) F1-score par catégorie



(b) Courbes ROC par catégorie

Modèle	F1-score (macro)	Message/sec	Commentaire
CamemBERT	0.55	36	Bonne performance globale, mais difficulté sur les classes rares.
FlauBERT	0.57	33	Performance améliorée par rapport à CamemBERT, surtout sur les classes intermédiaires.
XLM-RoBERTa	0.575	39	Modèle le plus performant, rapide et précis sur la majorité des classes.

TABLE 15 – Comparaison finale des performances sur le jeu de test (jeu de données français).

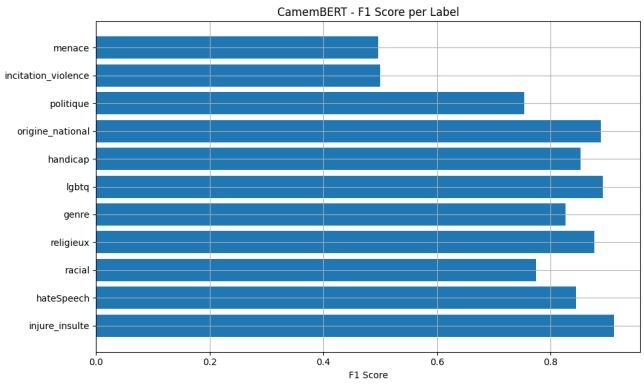
Comparaison des modèles

- **CamemBERT** : Bien adapté aux classes fréquentes, mais perfectible sur les classes rares, notamment lgbtq et incitation_violence.
- **FlauBERT** : Capacité d'adaptation supérieure sur les classes intermédiaires et les labels moins représentés, mais toujours en deçà pour les classes les plus rares.
- **XLM-RoBERTa** : Meilleur compromis entre précision et rapidité, avec de bonnes performances sur les classes fréquentes et une légère amélioration sur les classes rares par rapport aux autres modèles.

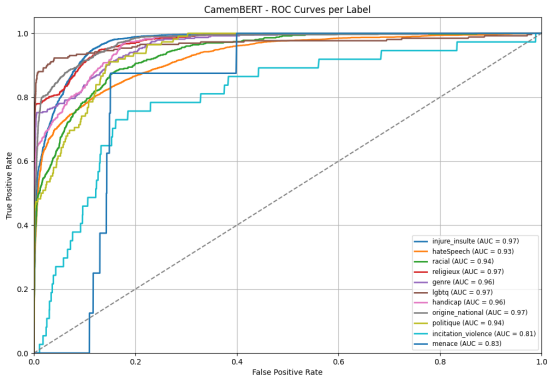
Synthèse : XLM-RoBERTa reste le modèle le plus performant pour le jeu de données anglais, tant en termes de précision que de vitesse. FlauBERT présente une meilleure gestion des classes intermédiaires par rapport à CamemBERT, qui reste moins robuste pour les catégories rares. Pour une utilisation en production, XLM-RoBERTa représente le meilleur choix, bien que l'amélioration des classes rares reste un axe de développement.

7.1.3 Résultats sur jeu de donnée Français|Anglais

CamemBERT CamemBERT montre de bonnes performances pour les données mixtes (français et anglais) avec un F1-score macro de 0.59. Le modèle reste robuste pour les classes majoritaires comme *injure_ insulte* et *origine_ national*, mais peine toujours avec les classes rares, notamment *menace* et *incitation_ violence*. Les courbes ROC montrent des AUC élevés pour les classes fréquentes, indiquant une bonne capacité de séparation.

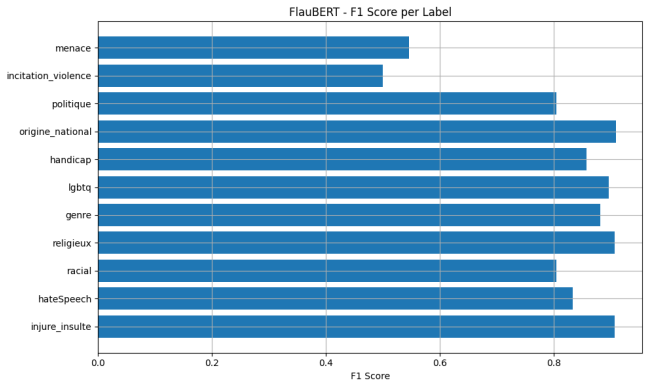


(a) F1-score par catégorie

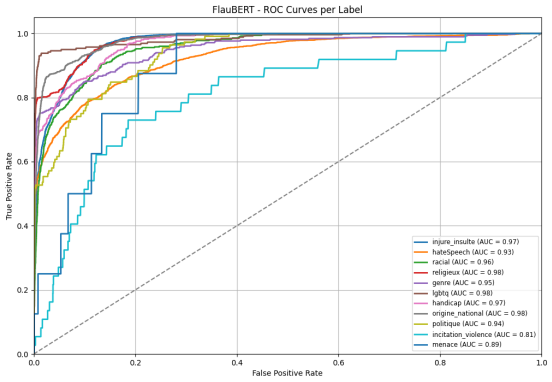


(b) Courbes ROC par catégorie

FlauBERT FlauBERT affiche les meilleures performances parmi les modèles français avec un F1-score macro de 0.625. Il excelle dans la reconnaissance des classes fréquentes tout en offrant une détection améliorée des classes moins fréquentes, notamment *genre* et *handicap*. Son AUC est également satisfaisant pour la plupart des catégories, ce qui souligne sa capacité à bien différencier les classes.

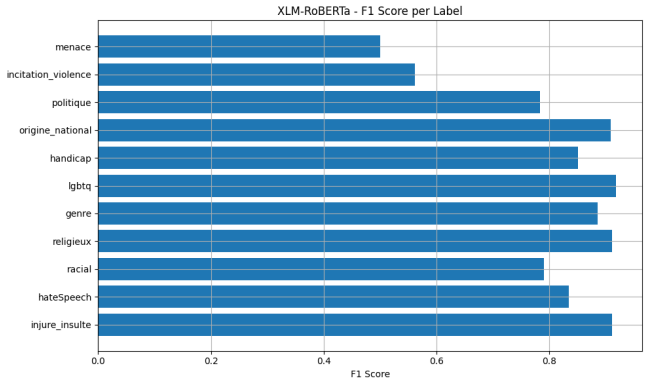


(a) F1-score par catégorie

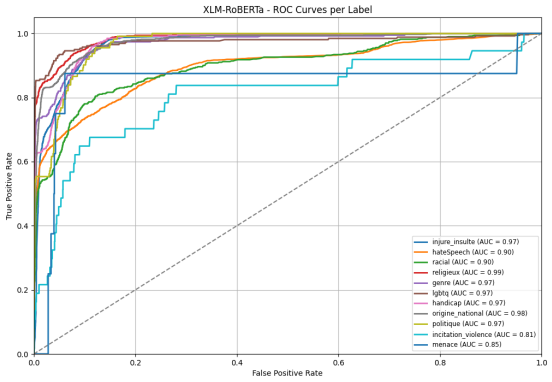


(b) Courbes ROC par catégorie

XLM-RoBERTa XLM-RoBERTa atteint un F1-score macro de 0.626, le plus élevé parmi les trois modèles. Il présente une excellente capacité de détection pour les classes majoritaires et intermédiaires, avec des AUC proches de 1 pour *injure_insulte* et *origine_national*. Toutefois, il reste perfectible pour les classes les plus rares, bien que ses résultats soient légèrement supérieurs à ceux de CamemBERT.



(a) F1-score par catégorie



(b) Courbes ROC par catégorie

Modèle	F1-score (macro)	Message/sec	Commentaire
CamemBERT	0.59	37	Bonne performance globale mais limitée sur les classes rares.
FlauBERT	0.625	34	Excellent compromis entre précision et couverture des classes rares.
XLM-RoBERTa	0.626	37	Meilleur score global, rapidité accrue, mais encore des lacunes sur les classes rares.

TABLE 16 – Comparaison finale des performances sur le jeu de test (jeu de données français).

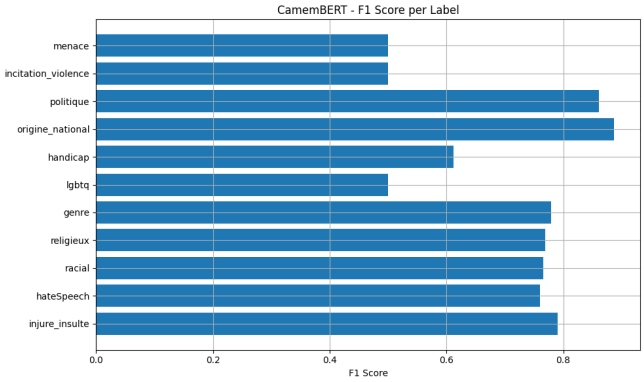
Comparaison des modèles

- **CamemBERT** : Performant pour les classes majoritaires, mais encore limité pour les catégories rares comme incitation_violence.
- **FlauBERT** : Montre une nette amélioration par rapport à CamemBERT dans la gestion des classes moins fréquentes tout en restant performant sur les classes dominantes.
- **XLM-RoBERTa** : Meilleur compromis entre précision et rapidité, avec de bonnes performances sur la plupart des classes malgré quelques faiblesses pour les labels rares.

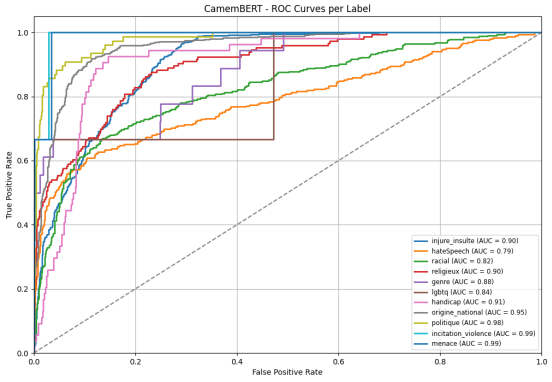
Synthèse : Pour le jeu de données combiné français-anglais, XLM-RoBERTa se distingue comme le modèle le plus performant grâce à son équilibre entre précision et rapidité. FlauBERT se positionne également comme un bon choix pour les classes moins fréquentes, tandis que CamemBERT reste légèrement en retrait pour les catégories rares. Pour des applications en production, XLM-RoBERTa est recommandé, bien que l'enrichissement des classes rares pourrait améliorer les résultats.

7.1.4 Résultats sur jeu de donnée Anglais→Français

CamemBERT CamemBERT montre une performance relativement faible sur le transfert de données anglaises vers françaises, avec un F1-score macro de 0.45. Le modèle semble peiner à généraliser les connaissances acquises en anglais vers le français, en particulier sur les classes rares comme *menace* et *incitation_ violence*. Les courbes ROC révèlent une bonne capacité pour les classes dominantes, mais une moins bonne différenciation pour les catégories moins fréquentes.

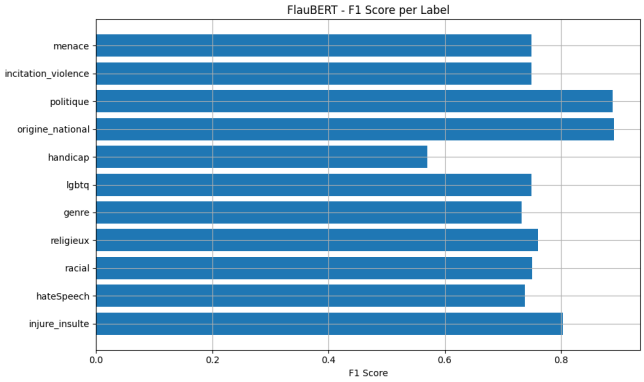


(a) F1-score par catégorie

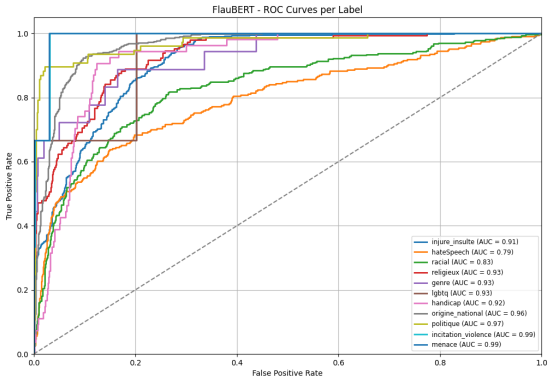


(b) Courbes ROC par catégorie

FlauBERT FlauBERT affiche de loin les meilleures performances dans ce contexte de transfert avec un F1-score macro de 0.58. Sa capacité à s'adapter aux données françaises tout en intégrant les connaissances issues de l'anglais est notable, notamment pour les classes intermédiaires et rares telles que *handicap* et *genre*. Les courbes ROC montrent également une bonne séparation pour la plupart des classes.



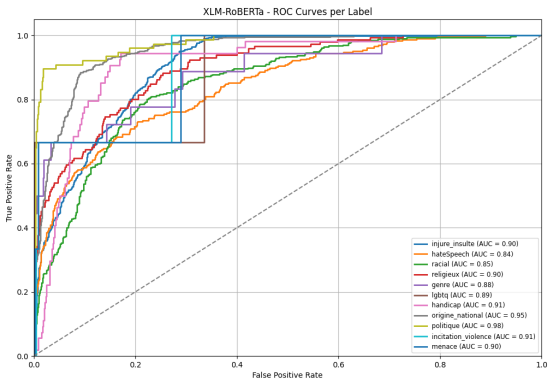
(a) F1-score par catégorie



(b) Courbes ROC par catégorie

XLM-RoBERTa XLM-RoBERTa atteint un F1-score macro de 0.52, montrant une performance intermédiaire par rapport aux deux autres modèles. Bien qu'il surpasse CamemBERT dans la généralisation des données anglaises vers le français, il reste moins performant que FlauBERT, en particulier pour les classes à faible fréquence comme *lgbtq*.

(a) F1-score par catégorie



(b) Courbes ROC par catégorie

Modèle	F1-score (macro)	Message/sec	Commentaire
CamemBERT	0.45	34	Faible performance de généralisation entre langues, notamment pour les classes rares.
FlauBERT	0.58	31	Excellente capacité d'adaptation inter-langues, surtout pour les classes intermédiaires et rares.
XLM-RoBERTa	0.52	37	Performance intermédiaire, meilleure que CamemBERT pour le transfert inter-langues, mais inférieure à FlauBERT.

TABLE 17 – Comparaison finale des performances sur le jeu de test (jeu de données français).

Comparaison des modèles

- **CamemBERT** : Bien adapté aux classes majoritaires, mais montre de sérieuses limites pour transférer les connaissances linguistiques entre l'anglais et le français.
- **FlauBERT** : Montre une très bonne capacité de généralisation des données anglaises vers françaises, surtout pour les classes rares.
- **XLM-RoBERTa** : Une performance intermédiaire, avec une bonne vitesse mais une précision qui reste inférieure à FlauBERT dans ce contexte de transfert.

Synthèse : Pour le transfert des données anglaises vers françaises, FlauBERT est le modèle le plus performant grâce à sa meilleure gestion des classes rares et intermédiaires. XLM-RoBERTa montre une bonne rapidité et un compromis acceptable, tandis que CamemBERT reste limité sur ce type de tâche.

7.1.5 Comparaison finale des modèles et jeux de données

Synthèse finale : Après analyse des performances sur les différents jeux de données (Français, Anglais, Français|Anglais, Anglais→Français), il apparaît que XLM-RoBERTa se distingue globalement comme le modèle le plus performant. Il combine à la fois rapidité d'exécution et précision sur les classes majoritaires et intermédiaires, ce qui le rend particulièrement adapté pour des applications nécessitant un traitement en temps réel.

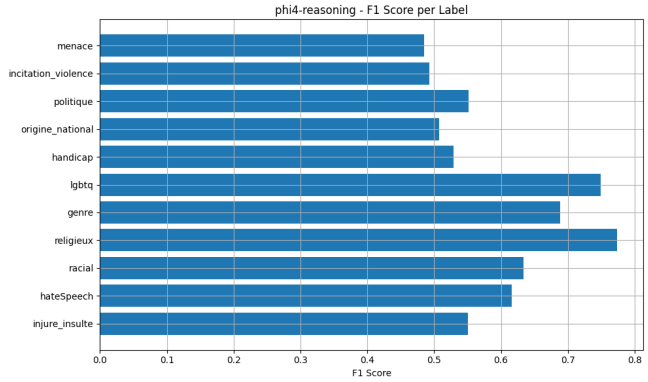
FlauBERT, quant à lui, présente un excellent compromis lorsque les classes rares et intermédiaires sont particulièrement pertinentes. Il surpasse systématiquement CamemBERT dans ces situations et se rapproche même de XLM-RoBERTa pour les tâches impliquant la gestion de données linguistiques mixtes ou le transfert entre langues (Anglais→Français).

CamemBERT, bien que robuste pour les classes dominantes, montre des limitations sur les classes rares et les transferts inter-langues. Son F1-score reste inférieur à celui des deux autres modèles dans la plupart des scénarios, mais sa rapidité d'exécution le rend encore utile dans des contextes où la précision sur les classes minoritaires n'est pas primordiale.

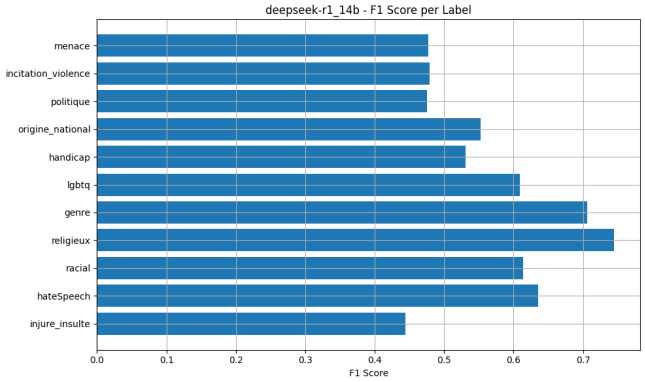
Recommandation : Pour des tâches multilingues ou avec des données mixtes, XLM-RoBERTa est le meilleur choix, notamment grâce à son équilibre entre vitesse et précision. Si l'objectif est de traiter des données en français avec une attention particulière aux classes rares, FlauBERT s'avère être une option robuste. Pour des applications moins sensibles aux classes minoritaires, CamemBERT peut être envisagé pour sa rapidité.

7.2 LLM sur jeu de donnée Français

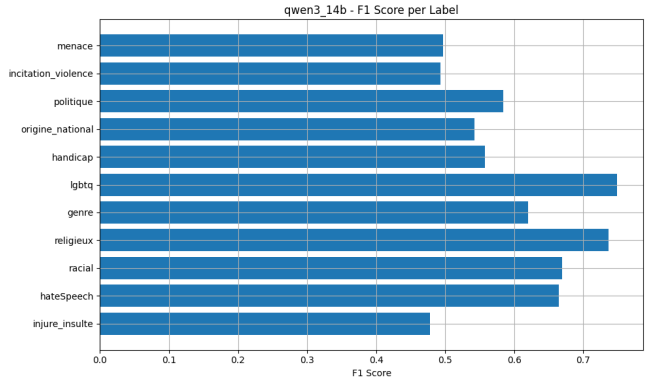
Comme énoncé plus tôt, par soucis de consommation et de temps d'inférence, les résultats ci-dessous ont été calculés seulement sur 1000 phrases du jeu de donnée Français.



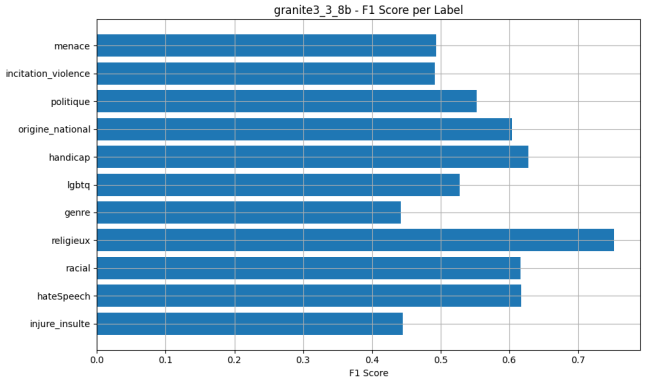
Phi4-reasoning
Temps d'inférence : 26m 16s



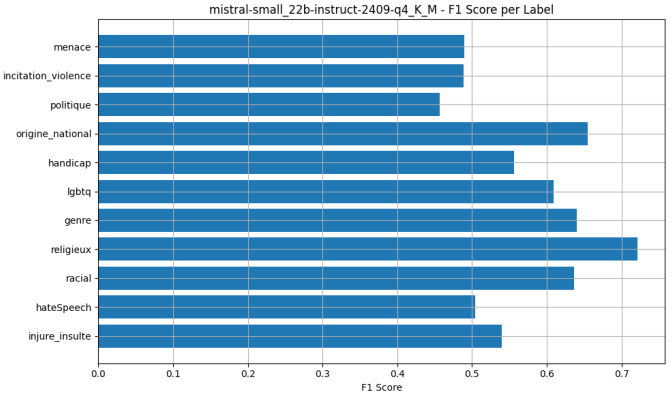
Deepseek-r1
Temps d'inférence : 29m 30s



Qwen3
Temps d'inférence : 20m 12s



Granite3.3
Temps d'inférence : 17m 50s



Mistral-small
temps d'inférence : 48m 48s

7.2.1 Analyse

Bien qu'il faille prendre des pincettes avec ces résultats, on peut constater que les LLM ne sont pas très performants pour jouer artificiellement le rôle de classifieur multi label. De plus, on peut également constater que le nombre de paramètres d'un modèle n'est pas vraiment révélateur de sa performance à classifier (ex : Mistral-small moins performant que Phi4). Pour finir, il est intéressant de remarquer que le temps d'inférence n'est pas non plus révélateur de qualité (une nouvelle fois prenons comme exemple Phi4 et Mistral-small)

8 Limites et axes d'amélioration

8.1 Limites identifiées

Complexité computationnelle (MLM et LLM) Les modèles de langage multi-label (MLM) et les grands modèles de langage (LLM) partagent une complexité computationnelle élevée. Les modèles MLM comme XLM-RoBERTa, FlauBERT et CamemBERT nécessitent un matériel GPU performant pour atteindre des temps d'inférence raisonnables. Cependant, les LLM présentent des exigences encore plus importantes en termes de calcul et de mémoire. Par exemple, Mistral-small (22B) nécessite plus de 13 Go de VRAM et reste relativement lent lors de l'inférence. Cette complexité rend difficile l'utilisation de ces modèles en production pour des tâches nécessitant une modération en temps réel.

Manque de données et déséquilibre des classes Le principal problème rencontré lors de l'entraînement des modèles est le manque de données pour certaines classes, notamment les catégories rares telles que lgbtq et incitation_violence. Ce déséquilibre impacte directement les performances de tous les modèles testés, limitant leur capacité à bien généraliser sur ces classes minoritaires. Pour pallier ce problème, il serait pertinent d'explorer des techniques telles que le sur-échantillonnage ciblé ou la génération de données synthétiques afin d'augmenter le volume d'exemples représentatifs.

Transfert de connaissances entre langues Lors du transfert d'un modèle entraîné sur des données anglaises vers des données françaises, des difficultés notables apparaissent, notamment pour les classes rares. Les performances chutent en raison d'une généralisation limitée, ce qui indique que les modèles ont du mal à intégrer simultanément les spécificités linguistiques des deux langues. Une approche possible pour améliorer ce point consisterait à utiliser des techniques d'entraînement multi-tâche ou à intégrer des modules spécifiques pour chaque langue dans un même modèle.

Limites des approches basées sur les LLM Les LLM montrent des résultats incohérents lorsqu'ils sont appliqués à la tâche de classification multi-label. Bien que ces modèles soient capables de comprendre des nuances complexes du langage, leur manque de structure standardisée dans les réponses rend difficile l'exploitation systématique des prédictions. De plus, le temps d'inférence des LLM est généralement plus élevé que celui des MLM, ce qui rend leur utilisation en modération en temps réel peu pratique. Pour réduire ces limitations, des recherches sur le prompt engineering et sur des architectures hybrides intégrant des modules spécialisés pourraient améliorer la cohérence et la rapidité des résultats.

8.2 Axes d'amélioration

Amélioration des performances sur les classes rares L'un des axes majeurs d'amélioration concerne l'amélioration des performances sur les classes rares (comme *lgbtq* et *incitation_violence*). Pour cela, plusieurs stratégies peuvent être envisagées :

- **Sur-échantillonnage ciblé** : Générer artificiellement des exemples supplémentaires pour les classes sous-représentées.
- **Augmentation de données** : Utiliser des techniques d'augmentation (paraphrase, synonymie) pour accroître la diversité des exemples.
- **Génération synthétique** : Exploiter des modèles génératifs pour créer des données artificielles pertinentes et équilibrées.

Optimisation des modèles pour un usage en temps réel La complexité computationnelle des modèles actuels (notamment XLM-RoBERTa et les LLM) limite leur usage en production. Pour rendre l'application plus viable en temps réel, plusieurs approches peuvent être envisagées :

- **Compression de modèle** : Réduire la taille des modèles via des techniques comme la quantification ou le pruning, tout en maintenant des performances acceptables.
- **Distillation de modèle** : Créer des versions légères des modèles lourds (comme XLM-RoBERTa) via une approche teacher-student.

Meilleure gestion des données multilingues Pour améliorer la généralisation sur des données issues de langues différentes (comme le transfert Anglais→Français), il est crucial de renforcer les capacités multilingues des modèles :

- **Entraînement multitâche** : Exploiter des techniques d'apprentissage multitâche où chaque tâche représente une langue spécifique.
- **Alignement cross-lingual** : Tirer parti de méthodes d'alignement des représentations (comme les embeddings multilingues) pour uniformiser les performances sur plusieurs langues.

Approches hybrides entre MLM et LLM Étant donné que les LLM et MLM présentent des forces complémentaires, une approche hybride pourrait offrir des gains significatifs :

- **Filtrage initial avec des modèles légers** : Utiliser un MLM rapide pour les cas simples et basculer vers un LLM pour les cas plus complexes.
- **Pipeline d'inférence modulaire** : Créer une architecture en cascade où les LLM n'interviennent que sur les prédictions incertaines des MLM.
- **Fine-tuning adapté aux LLM** : Entraîner les LLM avec des prompts calibrés spécifiquement pour la tâche de classification multi-label.

9 Conclusion

9.1 Synthèse des travaux réalisés

Ce projet a porté sur l'identification automatique de contenus injurieux, racistes et homophobes dans un contexte multilingue (français et anglais). Face à la prolifération de discours haineux sur les réseaux sociaux, nous avons cherché à développer des modèles robustes capables de détecter efficacement ces contenus.

Pour ce faire, nous avons procédé en plusieurs étapes :

1. **Constitution des jeux de données** :
   - Fusion de jeux de données multilingues (français et anglais) provenant de différentes sources pour garantir une couverture variée des propos haineux.
   - Création de jeux de données spécifiques : français, anglais, mixte (français|anglais), et traduit (anglais vers français).
   - Nettoyage et normalisation des données pour garantir la cohérence des annotations multi-labels.
2. **Fine-tuning de modèles pré-entraînés (MLM)** :
   - Modèles testés : CamemBERT, FlauBERT, XLM-RoBERTa.
   - Entraînement avec des hyperparamètres optimisés, suivi de l'évolution des performances via les métriques F1-score et courbes ROC.
   - Évaluation comparative sur les différents jeux de données pour identifier les forces et faiblesses de chaque modèle.
3. **Utilisation de modèles de langage large (LLM)** :
   - Tests réalisés sur des modèles LLM tels que Phi4-reasoning, Deepseek-r1, Qwen3, Granite3.3, et Mistral-small.
   - Utilisation de prompts calibrés et d'outils comme Pydantic pour structurer les réponses.
   - Comparaison des performances avec les modèles MLM.
4. **Analyse des résultats** :
   - XLM-RoBERTa s'est révélé le plus performant parmi les MLM, avec un bon équilibre entre rapidité et précision, surtout en contexte multilingue.
   - FlauBERT a bien performé sur les classes rares, particulièrement utile pour les contenus en français.
   - CamemBERT a montré ses limites sur les classes rares et en transfert linguistique.
   - Les LLM, bien que puissants en compréhension contextuelle, se sont montrés inefficaces pour la classification multi-label et l'inférence en temps réel.

9.1.1 Comparaison LLM vs MLM

Critère	MLM	LLM
Performance globale	F1-score jusqu'à 0.626	F1-score autour de 0.55
Rapidité d'inférence	~37 messages/sec	~20 minutes pour 1000 messages
Gestion des classes rares	Modérée, meilleures performances avec FlauBERT	Très variable, dépend du modèle et du prompt
Complexité computationnelle	Modérée (~8 Go VRAM)	Élevée (~13 Go VRAM)
Adaptabilité multilingue	Bonne pour XLM-RoBERTa	Très dépendante du prompt et du modèle
Utilisation en production	Adapté pour la modération en temps réel	Peu adapté en raison de la lenteur et de la variabilité des résultats

TABLE 18 – Comparaison entre les modèles MLM et LLM

Les MLM (notamment XLM-RoBERTa) sont clairement mieux adaptés aux tâches de classification multi-label et à la détection de contenus haineux en temps réel. En revanche, les LLM manquent de structure et de rapidité pour être utilisés efficacement dans ce contexte. Les LLM, malgré leur capacité de compréhension contextuelle, se révèlent peu fiables et trop lents pour des tâches de modération automatique.

9.2 Réponse à la problématique

La problématique posée était de savoir s'il est possible de développer un système automatique capable de détecter efficacement et en temps réel des contenus injurieux, racistes et homophobes sur les réseaux sociaux, en prenant en compte les défis liés à la diversité linguistique et à la variabilité des propos haineux.

Réponse Oui, il est possible de développer un tel système, mais cela nécessite de faire des compromis entre précision et rapidité. Les modèles MLM (en particulier XLM-RoBERTa) apparaissent comme la solution la plus viable pour une modération en temps réel, grâce à leur équilibre entre performance et coût computationnel.

Toutefois, les résultats montrent également que les modèles actuels ont encore des marges de progression, notamment pour :

— La gestion des classes rares (comme *lgbtq* ou *incitation_violence*), où des techniques d'augmentation de données seraient nécessaires.

— Le traitement multilingue, où l'intégration de techniques d'alignement cross-lingual pourrait améliorer la généralisation entre les langues.

— La rapidité d'inférence, en explorant des techniques de compression et de distillation pour rendre les modèles plus légers sans perte de performance.

Pour des applications nécessitant une détection rapide et précise (comme la modération de contenus en ligne), les MLM sont à privilégier. Les LLM, en raison de leur lenteur et de leur instabilité dans la classification multi-label, ne sont pas encore suffisamment fiables pour un déploiement en production.

Perspectives À l'avenir, il serait pertinent d'envisager des solutions hybrides combinant l'efficacité des MLM pour les cas standards avec la capacité contextuelle des LLM pour les cas plus complexes, notamment pour les discours ambigus ou utilisant de l'ironie. De plus, le développement de jeux de données équilibrés et diversifiés serait essentiel pour garantir une meilleure robustesse des modèles face aux classes rares et aux contextes multilingues.

A Annexes

A.1 Répartition du travail

A.1.1 Préparation et Constitution des jeux de données

Ensemble :

— Recherche et collecte des jeux de données multilingues (français et anglais).
— Prétraitement des données (nettoyage, normalisation des colonnes, suppression des doublons).
— Fusion des jeux de données pour créer les corpus finaux (Français, Anglais, Mixte).
— Création des ensembles d'entraînement, de validation et de test (stratification).

A.1.2 Entraînement des Modèles

Théo :

— Fine-tuning des modèles CamemBERT, FlauBERT et XLM-RoBERTa.
— Suivi des métriques d'entraînement (perte, F1-score) et ajustement des hyperparamètres.
— Optimisation matérielle (fp16, gradient accumulation).
— Analyse des courbes d'apprentissage (perte et F1-score) pour chaque modèle.
— Comparaison des performances des modèles sur les jeux de données français, anglais, mixtes et traduits.

Pierre :

— Installation et configuration des modèles LLM en local (Ollama, Pydantic).
— Conception des prompts pour la classification multi-label.
— Fine-tuning ou ajustement des modèles LLM (Phi4, Deepseek-r1, Qwen3, Granite3.3, Mistral-small).
— Analyse des performances (F1-score, temps d'inférence) pour chaque modèle.
— Comparaison des modèles en fonction des temps d'inférence et de la précision.

A.1.3 Analyse des Résultats et Comparaison des Modèles

Théo :

— Synthèse des résultats expérimentaux pour les modèles MLM.
— Comparaison des modèles MLM entre eux (CamemBERT, FlauBERT, XLM-RoBERTa).
— Identification des forces et faiblesses des MLM pour la détection de discours haineux.

Pierre :

— Synthèse des résultats expérimentaux pour les modèles LLM.
— Comparaison des performances des différents LLM (Phi4, Deepseek-r1, Qwen3, Granite3.3, Mistral-small).
— Identification des limitations des LLM dans le contexte de la classification multi-label.

Ensemble :

— Comparaison globale entre LLM et MLM (précision, temps d'inférence, robustesse).
— Réflexion sur l'hybridation possible entre les deux approches.

A.2 Jeux de données utilisés

Voici la liste détaillée des jeux de données exploités pour la constitution de nos corpus d'entraînement, de validation et de test :

— **CONAN Dataset (en français et en anglais)**
Counter Narratives dataset multilingue pour lutter contre les discours de haine.
Source : `https://github.com/marcoguerini/CONAN`
— **MultiTarget CONAN (en anglais)**
Extension du CONAN dataset permettant l'annotation de multiples cibles de discours de haine.
Source : `https://github.com/marcoguerini/CONAN`
— **MLMA Hate Speech Dataset (français et anglais)**
Corpus annoté en multi-label pour différents types de discours haineux (race, religion, orientation sexuelle).
Source : `https://huggingface.co/datasets/nedjmaou/MLMA_hate_speech`

— **Hate Speech and Offensive Language Dataset (Davidson, anglais)**
Corpus historique pour la détection du discours haineux et offensant sur Twitter.
Source : `https://github.com/t-davidson/hate-speech-and-offensive-language`

— **FTR Racism Dataset (français)**
Corpus ciblant spécifiquement les propos racistes sur Twitter en langue française.
Source : `https://github.com/NataliaVanetik/FTR-dataset`

— **ETHOS Hate Speech Dataset (anglais)**
Corpus annoté en binaire et multi-label pour la détection de discours haineux.
Source : `https://github.com/intelligence-csd-auth-gr/Ethos-Hate-Speech-Dataset`

— **White Supremacist Forum Dataset (anglais)**
Corpus issu de forums d'extrême droite, contenant des propos haineux explicites.
Source : `https://github.com/Vicomtech/hate-speech-dataset`

Chaque corpus a été retraité et harmonisé pour s'adapter à notre tâche de classification multi-étiquette, selon les catégories définies dans la section précédente.

A.3 Détail des catégories (labels) utilisés

Les exemples de notre corpus final sont annotés selon 11 catégories distinctes, permettant une classification fine des contenus haineux et injurieux.

— **injure_insulte** : propos insultants, moqueurs ou dégradants envers une personne ou un groupe, sans nécessairement relever du discours de haine.

— **hateSpeech** : discours de haine explicite visant un groupe ou une personne en raison de caractéristiques telles que l'origine, la religion, le genre, etc.

— **racial** : attaque ou stigmatisation basée sur l'origine ethnique ou la "race" perçue d'un individu ou d'un groupe.

— **religieux** : propos visant une religion ou un système de croyances, incluant l'islamophobie, l'antisémitisme, etc.

— **genre** : contenu sexiste, misogyne ou dénigrant basé sur le genre (hommes, femmes, etc.).

— **lgbtq** : propos homophobes, transphobes ou hostiles envers la communauté LGBTQ+.

— **handicap** : discours stigmatisant ou moqueur envers les personnes en situation de handicap.

— **origine_national** : discrimination ou stigmatisation liée à la nationalité ou à l'origine géographique d'un individu ou groupe.

— **politique** : propos haineux ou violents envers des opinions, partis ou affiliations politiques spécifiques.

— **incitation_violence** : appel direct ou indirect à la violence physique ou psychologique contre un individu ou un groupe.

— **menace** : menace explicite ou implicite dirigée contre une personne ou une communauté.

Chaque exemple du corpus peut être associé à une ou plusieurs de ces catégories de manière indépendante, conformément à une approche de classification multi-label.