



Université Chouaib Doukkali
École Nationale des Sciences Appliquées d'El Jadida
Département Télécommunications, Réseaux et Informatique



RAPPORT DE PROJET

Filière : 2ITE

Niveau : 3^{ème} Année

Toxic Comments Classification



Réalisé Par :

ANEJJAR Ihssane

ELHALLA Zineb

Supervisé par :

Pr. EL MOUTAOUAKIL Khalid

Année Universitaire : 2025/2026

Résumé

Ce rapport présente la conception, le développement et l'évaluation d'un système de classification multi-label de commentaires toxiques utilisant des modèles de Deep Learning. Deux approches principales ont été comparées : un **CNN** (Convolutional Neural Network) et un **BERT** (Bidirectional Encoder Representations from Transformers).

Le modèle CNN exploite des embeddings *FastText* et des filtres convolutionnels multi-échelles pour capturer les motifs locaux du texte, offrant une solution rapide et efficace avec un temps d'entraînement réduit et une performance robuste sur les six catégories de toxicité. En parallèle, le modèle BERT, basé sur un encodage contextuel bidirectionnel via Transformer, permet une meilleure compréhension des dépendances longues et des nuances sémantiques, atteignant une performance légèrement supérieure au prix d'un coût computationnel plus élevé.

Le rapport détaille également les bibliothèques et outils utilisés, incluant PyTorch, TensorFlow, Hugging Face Transformers, NLTK, SpaCy et Streamlit pour le déploiement. L'application déployée permet de tester les modèles en temps réel et d'afficher les prédictions pour chaque catégorie de toxicité.

Enfin, le rapport discute des perspectives d'amélioration, notamment l'optimisation des performances, l'intégration de données supplémentaires et le déploiement cloud pour rendre le service accessible à grande échelle.

Table des matières

Table des matières

Liste des figures

Liste des tableaux 1

Introduction générale 2

1 Contexte et Problématique 3

1.1 Cadre de l'étude 3

1.2 Enjeux et importance 3

1.3 Motivations de la recherche 3

1.4 Problématique centrale 3

1.5 Objectifs du projet 4

2 Jeu de Données et Analyse Exploratoire 5

2.1 Présentation du corpus 5

2.2 Caractéristiques structurelles 5

2.3 Analyse exploratoire approfondie 5

2.3.1 Distribution des classes 5

2.3.2 Stratégie d'équilibrage 6

2.4 Analyse multi-labels 6

2.5 Analyse corrélative 7

3 Ingénierie des Caractéristiques 8

3.1 Typologie des features 8

3.2 Caractéristiques structurelles 8

3.3 Représentations sémantiques 9

3.4 Atténuation des biais 10

4 Modélisation et Architectures 11

4.1 Architecture CNN 11

4.2 Architecture BERT 12

5 Résultats et Analyse Comparative 14

5.1 Évaluation quantitative 14

5.2 Analyse qualitative 14

5.3 Implications pratiques 14

6 Bibliothèques, Outils et Déploiement	15
6.1 Bibliothèques et frameworks utilisés	15
6.2 Outils et environnement	15
6.3 Déploiement du projet	15
7 Conclusion et Perspectives	17
7.1 Synthèse des contributions	17
7.2 Perspectives de recherche	17
7.3 Recommandations	17

Liste des figures

Figure 2.1 :	Histogramme de Distribution des classes	6
Figure 2.2 :	Histogramme d'analyse multi-tag	6
Figure 2.3 :	Matrice de Cramer	7
Figure 3.1 :	Mots Uniques	8
Figure 3.2 :	Top words per class(unigrams)	9
Figure 3.3 :	Top words per class(bigrams)	10
Figure 4.1 :	Architecture du pipeline CNN pour la classification de toxicité	11
Figure 4.2 :	Roc-AUC Metric pour CNN	12
Figure 4.3 :	Roc-AUC Metric pour BERT	12
Figure 4.4 :	Architecture du pipeline CNN pour la classification de toxicité	13
Figure 6.1 :	Déploiement avec Streamlit: Comment	16
Figure 6.2 :	Déploiement avec Streamlit: results	16

Liste des tableaux

Tableau 5.1 : Comparaison des performances CNN vs BERT 14

Introduction Générale

Avec l'essor des plateformes de communication en ligne, la modération de contenu est devenue un enjeu majeur. Les commentaires publiés sur les réseaux sociaux, forums ou sites collaboratifs peuvent contenir des propos offensants ou discriminatoires, affectant l'expérience des utilisateurs et la réputation des plateformes. La détection automatique de ces contenus constitue un défi complexe, notamment en raison de la variété des expressions linguistiques et de la nature multi-label des toxicités.

Ce projet vise à développer et évaluer des modèles de Deep Learning pour la classification multi-label de commentaires toxiques. Deux approches principales ont été étudiées : un CNN (Convolutional Neural Network) pour sa rapidité et son efficacité dans l'extraction de motifs locaux, et un modèle BERT (Bidirectional Encoder Representations from Transformers) pour sa capacité à capturer le contexte et les dépendances sémantiques à longue distance. Les modèles sont évalués sur des métriques pertinentes telles que l'exactitude et le ROC-AUC, afin d'analyser le compromis entre performance, compréhension contextuelle et coût computationnel.

Le rapport présente d'abord le contexte, les enjeux et la problématique du projet, suivi d'une description du jeu de données et d'une analyse exploratoire incluant la distribution des classes et l'analyse multi-label. L'ingénierie des caractéristiques, la modélisation CNN et BERT, ainsi que les résultats et analyses comparatives sont ensuite détaillés. Enfin, le rapport aborde les bibliothèques et outils utilisés, le déploiement de l'application et se conclut par une synthèse des contributions et des perspectives de recherche.

Ce projet de recherche s'attaque à cette problématique en développant une solution de classification automatique des commentaires toxiques exploitant les avancées récentes en traitement automatique du langage naturel. Notre méthodologie combine une analyse exploratoire rigoureuse avec une modélisation comparative entre architectures CNN et BERT, utilisant le Wikipedia Comment Corpus comprenant 159 571 échantillons d'entraînement et 153 164 de test.

L'analyse préliminaire a révélé un déséquilibre distributionnel significatif nécessitant l'implémentation de techniques d'équilibrage adaptées. L'ingénierie des caractéristiques a intégré des attributs structuraux et sémantiques avancés, tout en identifiant et excluant les variables potentiellement biaisées. Les résultats démontrent que l'architecture CNN atteint une exactitude de 95% avec un temps d'entraînement de 8 minutes, tandis que BERT obtient 96% au prix de 5h30 de calcul, éclairant ainsi les compromis fondamentaux entre performance prédictive et efficacité computationnelle.

Contexte et Problématique

1.1 Cadre de l'étude

La modération des commentaires en ligne constitue un défi sociotechnique majeur dans le paysage numérique actuel. Les plateformes collaboratives ouvertes, à l'instar de Wikipedia, sont particulièrement exposées aux contenus toxiques en raison de leur accessibilité universelle et de l'ampleur des interactions utilisateurs. Cette vulnérabilité intrinsèque nécessite des mécanismes de régulation automatisés pour maintenir l'intégrité des échanges et protéger la communauté des effets néfastes des contenus toxiques.

1.2 Enjeux et importance

Au-delà des manifestations verbales agressives conventionnelles, les commentaires toxiques peuvent induire des séquelles psychologiques durables chez les utilisateurs. Ces impacts incluent des états de burnout avancé, des phénomènes d'isolement social progressif, et dans les situations les plus critiques, des passages à l'acte violents ou suicidaires. La détection proactive de ces contenus s'avère donc essentielle pour préserver la santé mentale collective et assurer la pérennité des espaces d'échange numériques.

1.3 Motivations de la recherche

Notre démarche s'articule autour de trois objectifs fondamentaux : la détection et neutralisation rapide des contenus violents pour maintenir un environnement numérique sain, la limitation de la propagation du cyberharcèlement affectant particulièrement les populations vulnérables, et l'anticipation des comportements extrêmes liés à la pression en ligne par l'identification précoce des signaux de détresse psychologique.

1.4 Problématique centrale

La problématique fondamentale réside dans la conception d'un modèle de traitement automatique du langage naturel capable de classifier des commentaires selon leur niveau de toxicité avec une précision optimale, tout en demeurant scalable et opérationnel en environnement

de production. Ce défi technique implique de trouver l'équilibre optimal entre performance algorithmique et efficacité computationnelle pour des applications à grande échelle.

1.5 Objectifs du projet

Les objectifs spécifiques incluent le développement d'un système de classification multi-labels des commentaires toxiques basé sur le Wikipedia Comment Corpus, la comparaison méthodologique des approches CNN et BERT selon des métriques standardisées, l'optimisation du compromis performance/coût computationnel pour un déploiement industriel, et l'établissement de recommandations pratiques pour l'implémentation en environnement de production.

Jeu de Données et Analyse Exploratoire

2.1 Présentation du corpus

Le jeu de données exploité provient du Wikipedia Comment Corpus, offrant un échantillon représentatif de commentaires réels modérés par la communauté Wikipedia. Ce corpus présente l'avantage de refléter la diversité linguistique et thématique des échanges sur une encyclopédie collaborative, avec des annotations de toxicité validées par des modérateurs humains, garantissant ainsi la qualité et la pertinence des données pour notre étude.

2.2 Caractéristiques structurelles

La base de données comprend 159 571 échantillons pour l'apprentissage et 153 164 pour les tests, assurant une validation robuste des algorithmes développés. Chaque observation est décrite par un identifiant unique, le texte brut du commentaire, et six indicateurs binaires de toxicité couvrant les dimensions principales des contenus néfastes : toxicité générale, toxicité sévère, obscénité, menaces, insultes et haine identitaire.

2.3 Analyse exploratoire approfondie

2.3.1 Distribution des classes

L'analyse distributionnelle révèle un déséquilibre significatif avec 90% des commentaires classés comme non-toxiques. Les catégories toxiques présentent des fréquences décroissantes, allant de 15 294 occurrences pour la toxicité générale à seulement 478 pour les menaces. Cette asymétrie reflète la réalité des plateformes en ligne mais pose des défis importants pour l'apprentissage automatique, nécessitant des stratégies d'équilibrage adaptées.

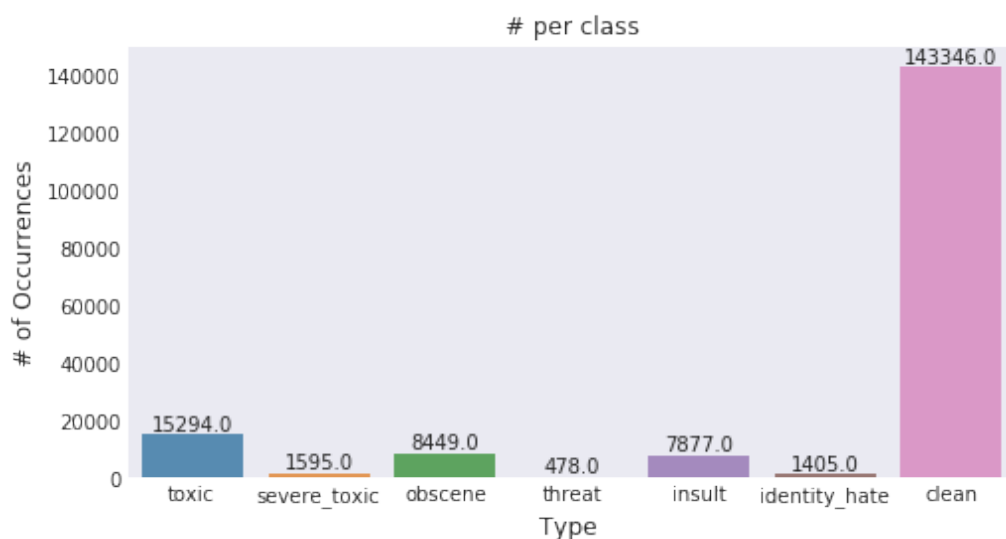


Figure 2.1. Histogramme de Distribution des classes

2.3.2 Stratégie d'équilibrage

Pour pallier le déséquilibre des classes, nous avons la technique SMOTE générant des échantillons synthétiques pour les classes minoritaires. Cette approche présente néanmoins des limitations, notamment un risque de sur-apprentissage avec les réseaux de neurones et une efficacité réduite avec les Transformers due à la nature artificielle des échantillons générés.

2.4 Analyse multi-labels

L'examen des annotations multiples démontre que 90% des commentaires ne portent aucun marqueur de toxicité, confirmant la prédominance des échanges constructifs. La majorité des commentaires toxiques sont associés à un ou deux tags, avec une décroissance progressive de la fréquence. Seulement 0,4% des commentaires cumulent cinq à six tags, représentant des cas exceptionnels d'agressivité multiforme nécessitant une attention particulière.

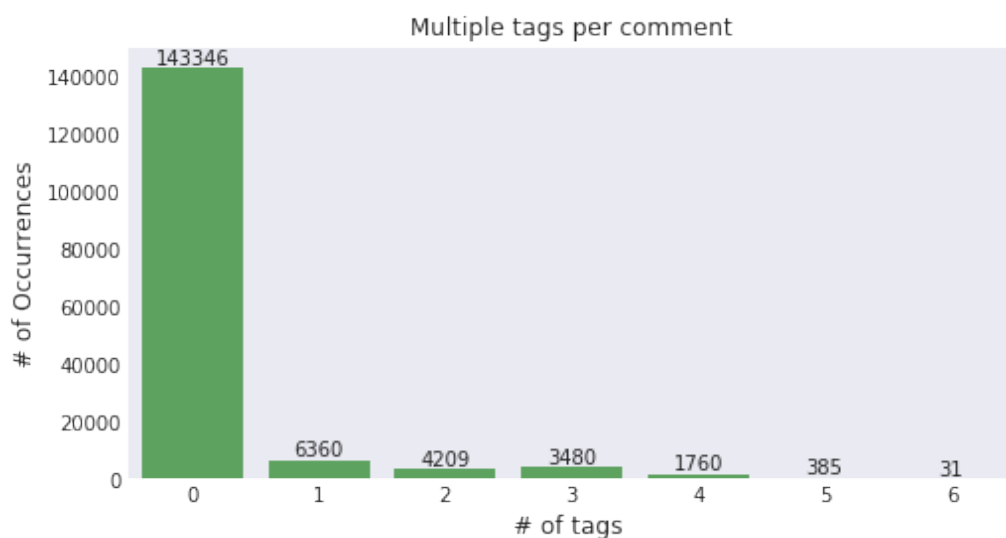


Figure 2.2. Histogramme d'analyse multi-tag

2.5 Analyse corrélative

L'analyse utilisant le coefficient de Cramer's V révèle une association faible à modérée entre les catégories de toxicité, confirmant que bien que liées, ces catégories capturent des dimensions distinctes justifiant leur traitement séparé dans le cadre de notre classification multi-labels.

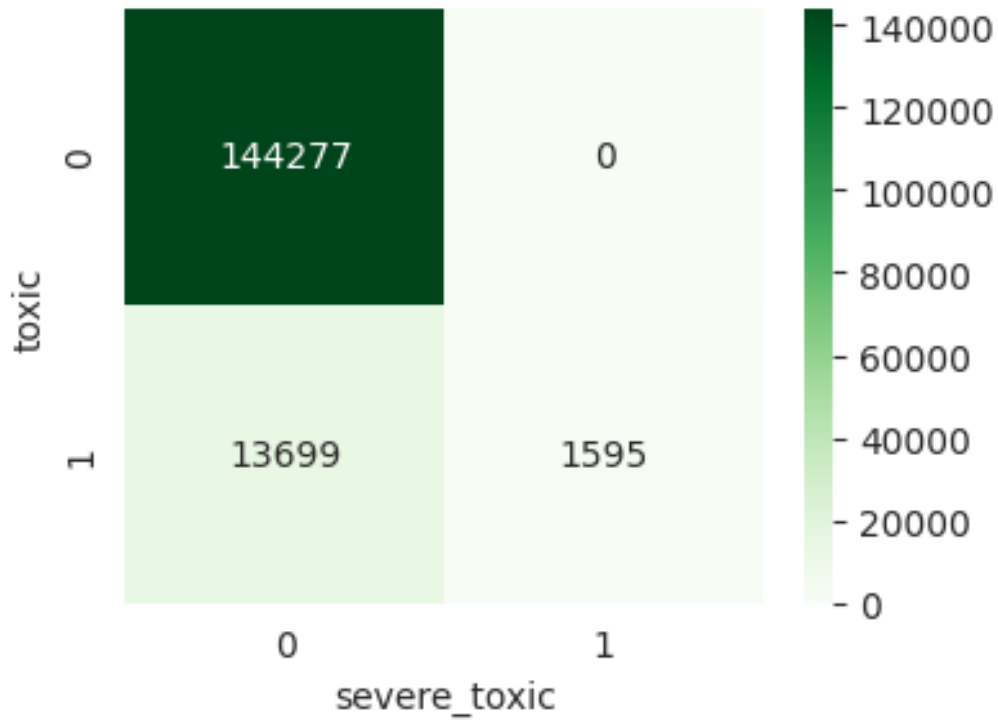


Figure 2.3. Matrice de Cramer

Ingénierie des Caractéristiques

3.1 Typologie des features

Nous avons développé une taxonomie tripartite des caractéristiques intégrant des attributs structurels indirects, des représentations sémantiques directes, et l'identification des variables potentiellement biaisées. Cette approche multi-facettes permet de capturer simultanément les dimensions structurelles, stylistiques et sémantiques des commentaires.

3.2 Caractéristiques structurelles

L'extraction des features indirectes vise à capturer des propriétés stylistiques et structurelles potentiellement indicatrices de toxicité. L'analyse révèle que les commentaires toxiques présentent généralement une longueur réduite, une répétitivité accrue et une diversité lexicale moindre. La pauvreté vocabulaire émerge comme un indicateur robuste de toxicité potentielle, tandis que les échanges constructifs montrent une plus grande variété lexicale.

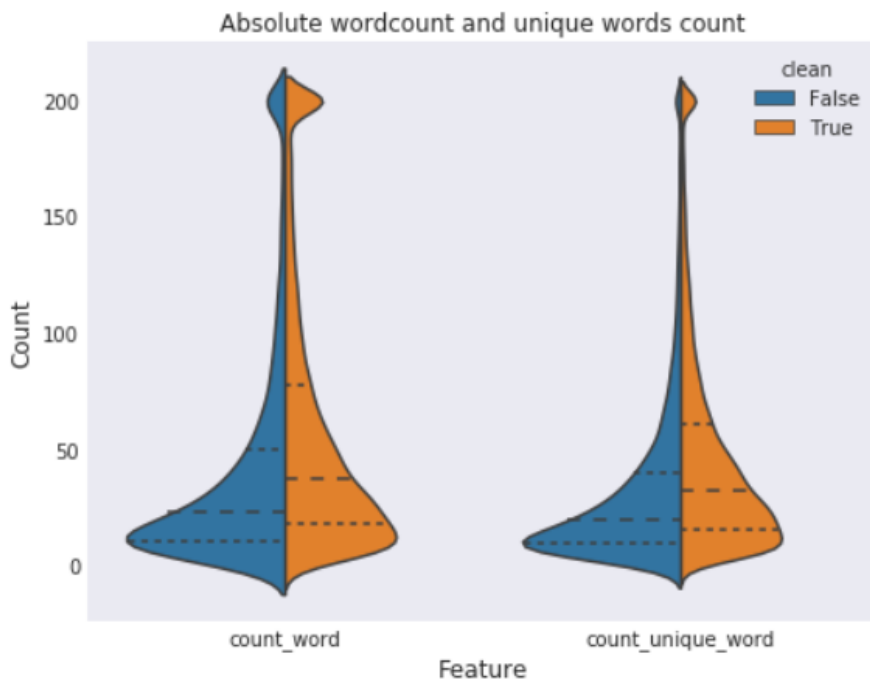


Figure 3.1. Mots Uniques

3.3 Représentations sémantiques

Pour capturer la substance sémantique des commentaires, nous avons abordé plusieurs approches complémentaires incluant TF-IDF avec unigrammes et bigrammes pour l'importance relative des termes, et trois types d'embeddings : Word2Vec pour la prédiction contextuelle, GloVe pour les statistiques globales de co-occurrence, et FastText pour l'information morphologique via les n-grammes de caractères.

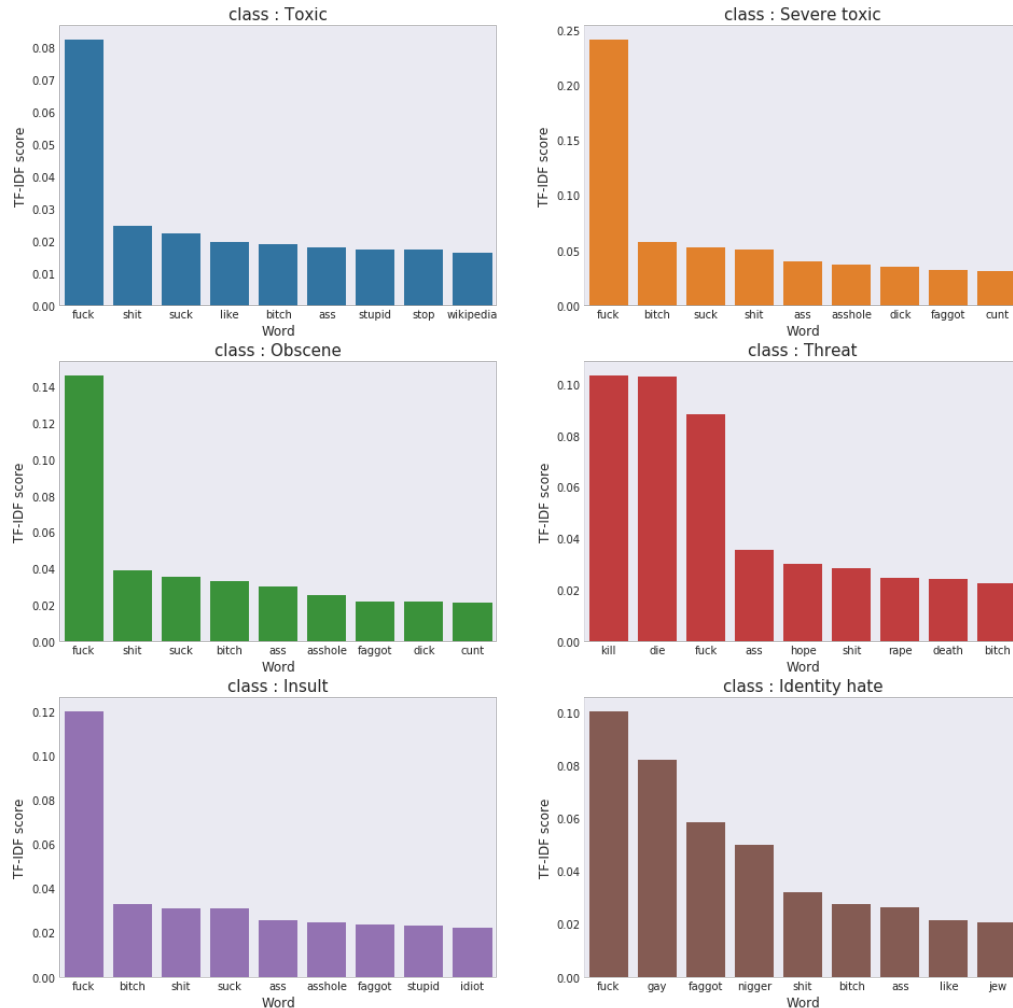


Figure 3.2. Top words per class(unigrams)

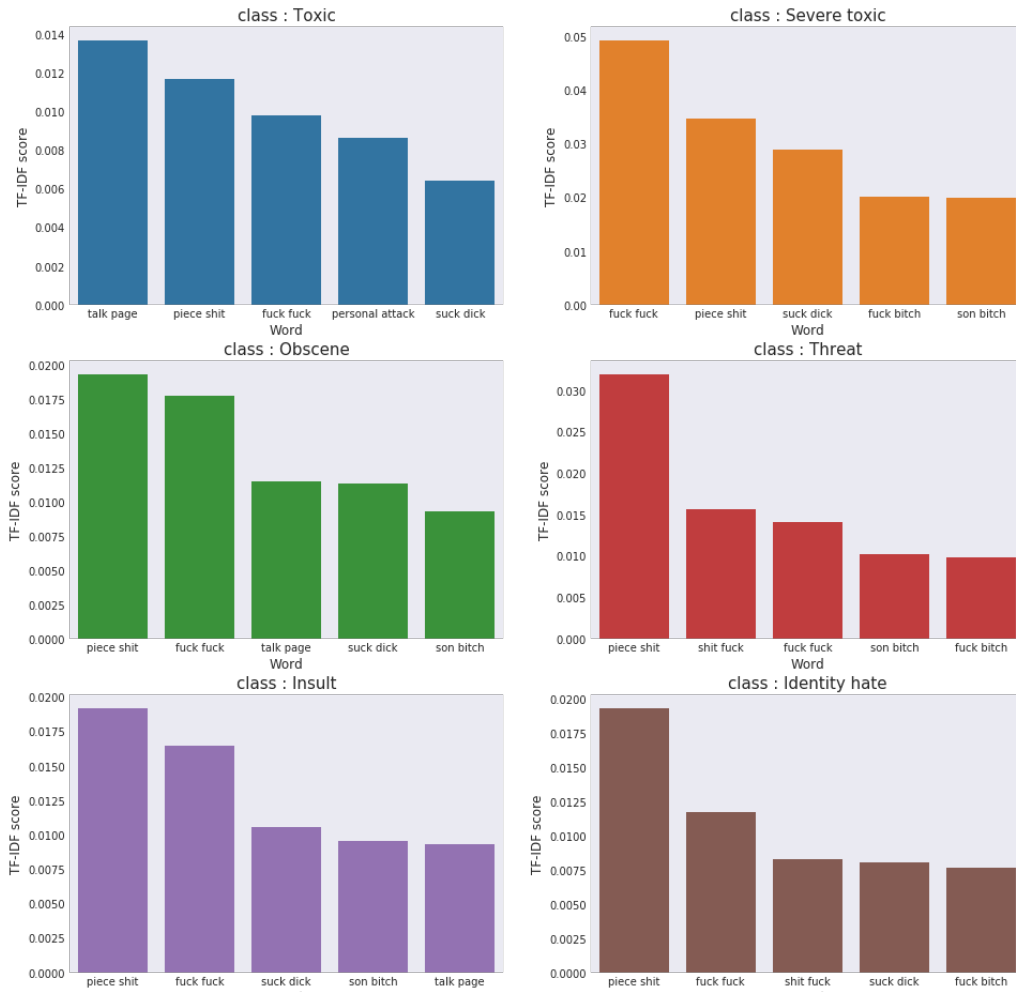


Figure 3.3. Top words per class(bigrams)

3.4 Atténuation des biais

L'identification des "leaky features" permet de l'exclusion des noms d'utilisateurs et adresses IP, variables fortement corrélées à la toxicité mais introduisant des biais importants. Leur inclusion risquait de conduire à un surapprentissage où le modèle associerait certains utilisateurs à la toxicité plutôt que le contenu sémantique des commentaires.

Modélisation et Architectures

Cette étude compare les modèles **CNN** et **Transformers (BERT)** afin d'évaluer le compromis entre *efficacité computationnelle* et *compréhension contextuelle* des séquences textuelles.

4.1 Architecture CNN

Le modèle **CNN** (Convolutional Neural Network) déploie une architecture spécialisée pour le traitement séquentiel du texte. Le pipeline commence par une phase de prétraitement incluant le nettoyage et la normalisation des commentaires bruts. Les textes sont ensuite transformés en représentations vectorielles via une couche d'embedding utilisant des plongements lexicaux *FastText* pré-entraînés.

La partie convolutionnelle applique des filtres *Conv1D* de différentes tailles pour capturer des motifs locaux à diverses échelles linguistiques. Les caractéristiques extraites sont ensuite agrégées via des opérations de pooling avant d'être traitées par des couches denses fully-connected pour la classification finale dans les six catégories de toxicité.

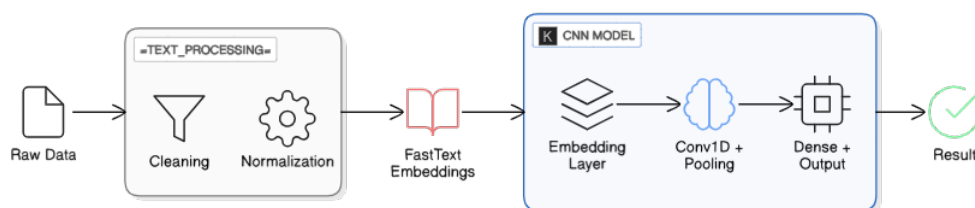


Figure 4.1. Architecture du pipeline CNN pour la classification de toxicité

La figure suivante montre la courbe ROC-AUC du modèle **CNN**. Elle illustre la capacité du modèle à distinguer les différentes classes de toxicité, malgré une architecture plus simple et un temps d'entraînement réduit, ce qui en fait un modèle efficace et rapide pour cette tâche.

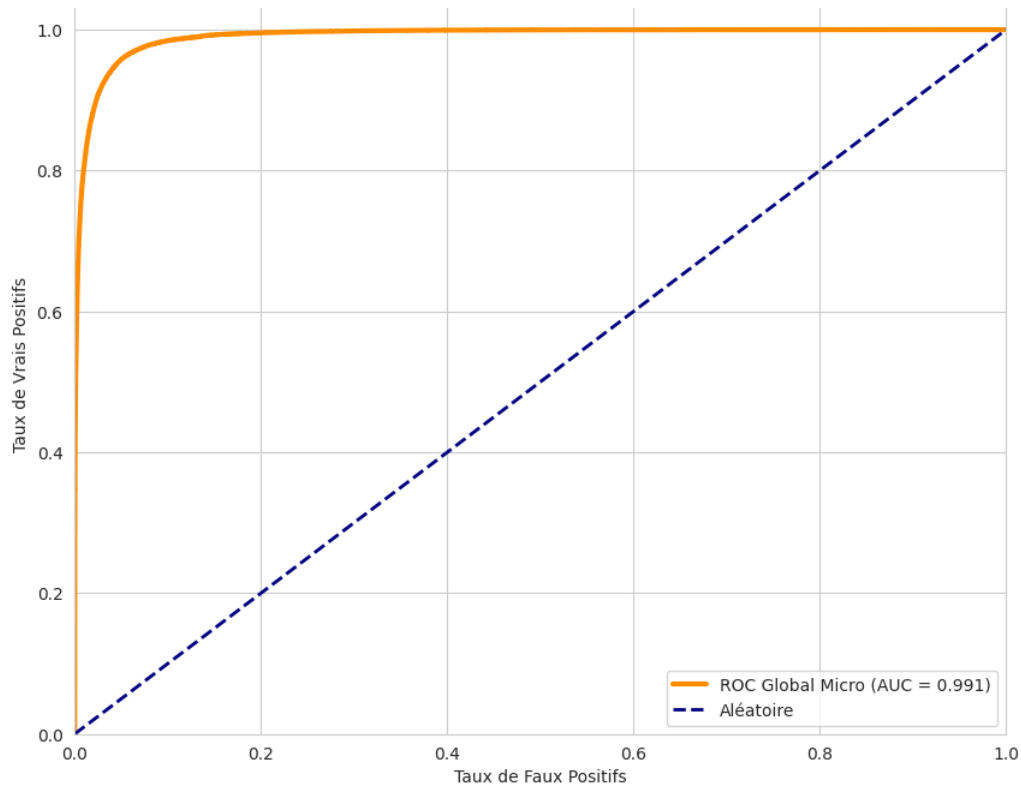


Figure 4.2. Roc-AUC Metric pour CNN

4.2 Architecture BERT

Le modèle **BERT** (Bidirectional Encoder Representations from Transformers) suit une approche de fine-tuning sur un modèle pré-entraîné. Le pipeline intègre une tokenisation spécifique au vocabulaire BERT, suivie d'un encodage contextuel bidirectionnel via les couches Transformer. Cette architecture permet de capturer les dépendances à longue distance et les nuances sémantiques complexes dans les commentaires. La sortie des encodeurs BERT est ensuite acheminée vers une tête de classification adaptative spécialement conçue pour la tâche multi-labels, permettant de prédire simultanément les six dimensions de toxicité avec une compréhension contextuelle approfondie.

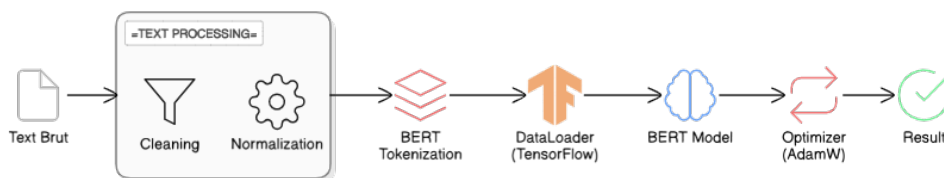


Figure 4.3. Roc-AUC Metric pour BERT

La figure suivante présente la courbe ROC-AUC du modèle **BERT**. Elle illustre la capacité du modèle à distinguer efficacement les différentes classes de toxicité, grâce à son encodage contextuel bidirectionnel et sa tête de classification adaptée.

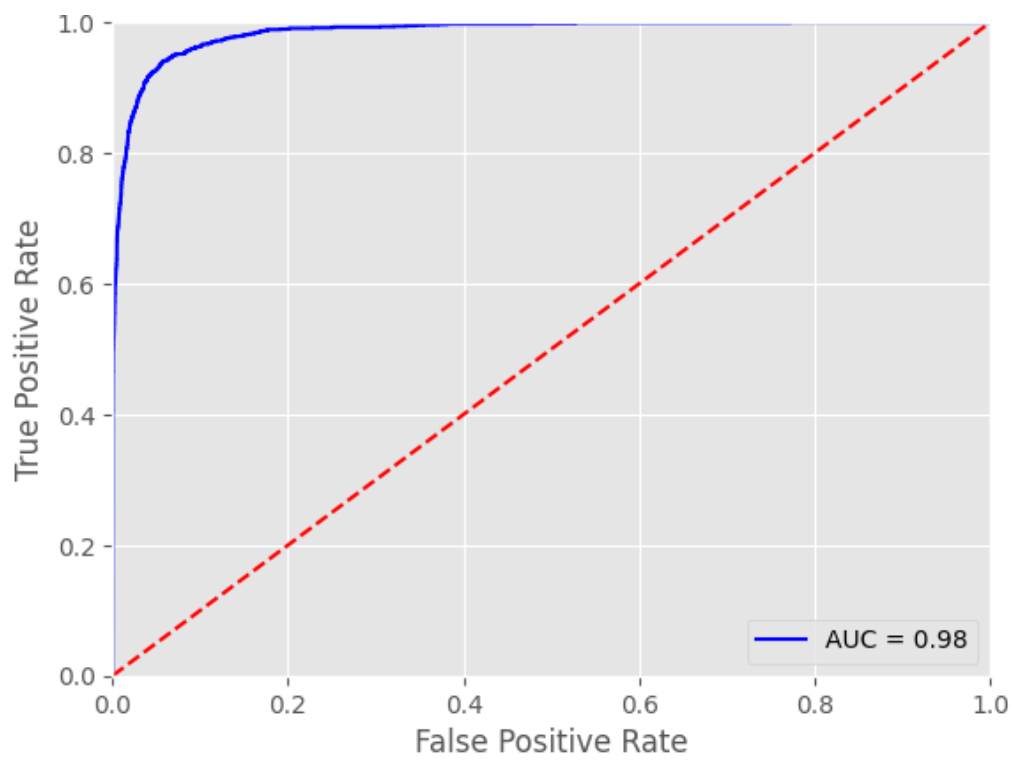


Figure 4.4. Architecture du pipeline CNN pour la classification de toxicité

Résultats et Analyse Comparative

5.1 Évaluation quantitative

L'analyse comparative révèle que le CNN atteint 95% d'exactitude et 99% de ROC AUC en 8 minutes sur CPU, tandis que BERT obtient 96% d'exactitude et 98% de ROC AUC au prix de 5h30 sur GPU T4. Cette différence souligne le compromis fondamental entre performance prédictive et efficience computationnelle dans les applications de classification textuelle.

Modèle	Accuracy	ROC AUC	Training Time	Ressources
CNN	95%	99%	5 min	GPU T4
BERT	96%	98%	5h30	GPU T4

Tableau 5.1. Comparaison des performances CNN vs BERT

5.2 Analyse qualitative

L'examen qualitatif met en évidence des caractéristiques distinctes entre les deux ap-proches. BERT excelle dans la compréhension contextuelle profonde grâce à son mécanisme d'attention, offrant une robustesse supérieure pour la toxicité implicite. Le CNN se distingue par son efficacité pour capturer des motifs locaux et sa légèreté opérationnelle, le rendant plus adapté aux déploiements à grande échelle avec contraintes de ressources.

5.3 Implications pratiques

Pour les applications industrielles nécessitant un traitement en temps réel de volumes importants, le CNN représente le choix optimal grâce à son excellent rapport performance/ef-ficacité. BERT, bien que plus performant dans certains cas, convient mieux aux contextes où la précision maximale est critique et où les ressources computationnelles sont abondantes.

Bibliothèques, Outils et Déploiement

6.1 Bibliothèques et frameworks utilisés

Pour ce projet, plusieurs bibliothèques Python et frameworks ont été exploités :

- **PyTorch et TensorFlow** : pour l'implémentation des modèles CNN et BERT.
- **Transformers (Hugging Face)** : pour l'utilisation et le fine-tuning de BERT.
- **NLTK et SpaCy** : pour le prétraitement du texte (nettoyage, tokenisation, normalisation).
- **FastText** : embeddings lexicaux pré-entraînés pour le modèle CNN.
- **Scikit-learn** : pour les métriques de performance (ROC-AUC, exactitude, classification report) et la gestion des jeux de données.
- **Pandas et NumPy** : manipulation et traitement des données.
- **Matplotlib et Seaborn** : visualisation des résultats et courbes ROC.

6.2 Outils et environnement

- **Jupyter Notebook / Google Colab** : pour l'expérimentation et l'entraînement des modèles.
- **GPU (T4)** : accélération des calculs pour le fine-tuning de BERT.
- **Streamlit** : pour la création d'une interface web permettant de tester les modèles en temps réel.

6.3 Déploiement du projet

Le déploiement a été pensé pour rendre le modèle accessible via une interface web :

- Création d'une application **Streamlit** pour la saisie de commentaires et l'affichage des prédictions.

- Chargement des modèles pré-entraînés (CNN et BERT) pour effectuer la classification.
- Visualisation des résultats sous forme de scores de probabilité pour chaque catégorie de toxicité.
- Possibilité d'extension vers un déploiement cloud (AWS, Heroku, ou autre) pour rendre l'application accessible en ligne.

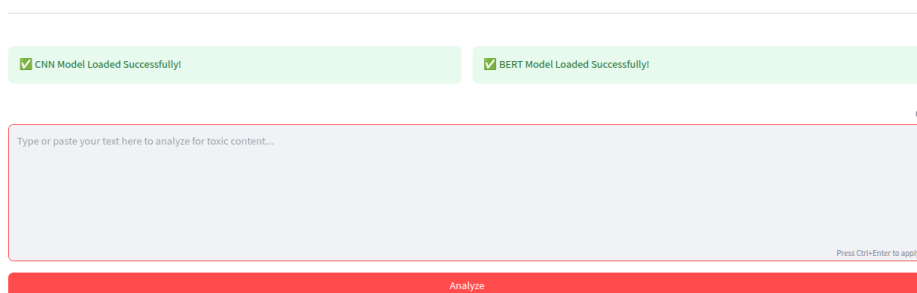


Figure 6.1. Déploiement avec Streamlit: Comment

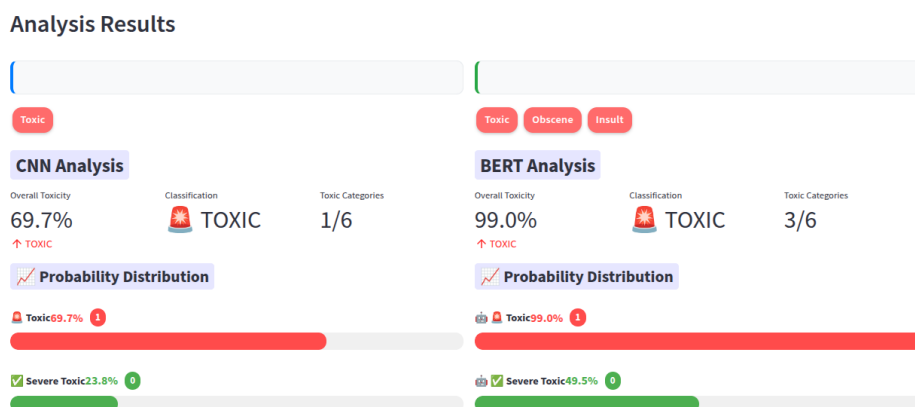


Figure 6.2. Déploiement avec Streamlit: results

Conclusion et Perspectives

7.1 Synthèse des contributions

Ce projet démontre la faisabilité opérationnelle de la classification automatique des commentaires toxiques avec des niveaux de performance élevés. L'architecture CNN émerge comme le compromis optimal entre performance prédictive et efficacité computationnelle, atteignant 95% d'exactitude avec seulement 8 minutes d'entraînement. Bien que BERT présente des performances légèrement supérieures, son coût computationnel prohibitif le rend moins adapté aux déploiements à grande échelle dans des contextes industriels.

7.2 Perspectives de recherche

Plusieurs axes d'amélioration méritent d'être explorés, incluant le développement d'architectures hybrides combinant les efficacités CNN avec la puissance contextuelle des mécanismes d'attention, l'implémentation de techniques d'équilibrage adaptatif plus sophistiquées, l'extension vers des capacités multilingues, et l'optimisation des mécanismes d'inférence pour des environnements de production exigeants.

7.3 Recommandations

Pour les implémentations pratiques, nous recommandons une approche pragmatique privilégiant l'efficacité globale plutôt que la performance brute. Le choix architectural doit considérer les contraintes spécifiques du contexte applicatif, incluant les volumes de données, les ressources disponibles, et les exigences de temps réel. Les résultats de cette étude fournissent un cadre décisionnel éclairé pour guider ces choix techniques dans des scénarios industriels variés.