Détection de contenu toxique

Détecter différents types de commentaires toxiques en ligne (menaces, obscénité, insultes, ...).

Auteur : RENAUDIN Remi, RABEMANANJARA Joëlla, EMZIANE Abdel-malek, TERCHANI Loucas

15 Janvier 2025

Table des matières

1 Introduction

1.1 Contexte et Motivation

Les discussions en ligne sont devenues un moyen essentiel d'échanger des idées et d'aborder des sujets importants. Cependant, ces espaces sont souvent confrontés à des comportements négatifs tels que les commentaires toxiques, les menaces, les insultes, et la haine basée sur l'identité. Ces interactions nuisibles découragent la participation et entravent la recherche d'opinions divergentes.

Les plateformes en ligne ont du mal à modérer efficacement ces comportements, ce qui pousse certaines communautés à restreindre ou à fermer complètement leurs sections de commentaires. Une meilleure gestion de la toxicité en ligne pourrait permettre de créer des environnements de discussion plus respectueux et productifs.

Dans ce contexte, l'équipe Conversation AI, soutenue par Jigsaw et Google, a développé des modèles pour détecter les comportements toxiques via l'API Perspective. Bien que ces modèles soient déjà largement utilisés, ils présentent encore des lacunes importantes, notamment en termes de précision et de personnalisation selon les types de toxicité.

1.2 Objectifs

Ce projet a pour but de développer un modèle capable de :

- Détecter plusieurs types de toxicité, notamment les menaces, les obscénités, les insultes et la haine basée sur l'identité.
- Améliorer la précision par rapport aux modèles existants en réduisant les erreurs.
- Permettre la personnalisation des modèles, afin que les utilisateurs puissent choisir les types de toxicité à détecter en fonction de leurs besoins spécifiques.
- Fournir un pipeline automatisé qui peut être facilement intégré dans des environnements de modération en ligne.

1.3 Répartition du Travail

Le projet a été réalisé collaborativement sur Google Colab, permettant à tous les membres de contribuer au code et de suivre l'avancement en temps réel. Voici les principales contributions de chaque membre :

- **Joëlla Rabemananjara** : Responsable du modèle basé sur TF-IDF, incluant le nettoyage des données et la mise en œuvre de la régression logistique.
- **Loucas Terchani** : Responsable de la partie GloVe, avec l'entraînement et l'optimisation des réseaux de neurones LSTM.
- **Abdel-Malek Emziane** : Initialement assigné à l'implémentation de BERT, mais en raison de limitations matérielles sous Google Colab, il s'est concentré sur l'optimisation des paramètres pour les modèles TF-IDF et GloVe.

— **Rémi Renaudin** : Responsable de la mise en commun des modèles, incluant la combinaison des prédictions, l'analyse des résultats et leur présentation finale.

2 Méthodologie

2.1 Démarches Générales

La méthodologie suivie dans ce projet repose sur plusieurs étapes clés visant à garantir un traitement rigoureux des données et une conception efficace des modèles :

Étape	**Description**
Entrée	Données brutes : commentaires en ligne.
Préprocessing	— Nettoyage des textes : suppression des URL, ponctuation, chiffres et conversion en minuscules. — Représentation numérique : — TF-IDF pour une matrice sparse. — Embeddings GloVe pour une représentation dense.
Modélisation	— Entraînement de modèles pour chaque type de toxicité : — Régression logistique avec TF-IDF. — LSTM avec embeddings GloVe.
Ensemble	— Combinaison des prédictions des modèles TF-IDF et GloVe. — Moyennage des probabilités pour améliorer la robustesse.
Sortie	— Prédictions pour chaque classe de toxicité : — Toxic, Severe Toxic, Obscene, Threat, Insult, Identity Hate.

TABLE 2.1 – Pipeline fonctionnel du système de détection de toxicité.

2.2 Dataset et Problématique

Le projet s'appuie sur le dataset de la compétition **Jigsaw Toxic Comment Classification Challenge**, qui donne une base annotée pour différents types de toxicité (ex toxic, severe toxic, obscene). Ce dataset est multi-label, et sa distribution déséquilibrée.

3 Implémentation

3.1 Environnement de Travail

Le projet a été implémenté en Python sur Google Colab, tirant parti des GPU pour accélérer les calculs. Les bibliothèques principales incluent :

- **Pandas et NumPy** : Manipulation et analyse des données.
- **Scikit-learn** : Extraction TF-IDF et modèles de machine learning.
- **TensorFlow/Keras** : Construction de réseaux de neurones avec embeddings.
- **Matplotlib/Seaborn** : Visualisation.

3.2 Processus Technique

3.2.1 Préparation des Données

- **Nettoyage** : Suppression des URL, ponctuation et mise en minuscules.
- **Représentation** :
  - **TF-IDF** : Vocabulaire limité à 10 000 termes les plus fréquents.
  - **Embeddings GloVe** : Tokenisation des commentaires et construction d'une matrice d'embeddings.

3.2.2 Conception des Modèles

1. **TF-IDF + Régression Logistique** : Entraînement de modèles séparés pour chaque classe avec optimisation d'hyperparamètres (e.g., `C`, `solver`).
2. **LSTM avec Embeddings GloVe** : Construction d'un réseau de neurones avec :
   - Une couche `Embedding` initialisée avec les vecteurs GloVe.
   - Deux couches LSTM bidirectionnelles.
   - Une couche de sortie `sigmoid` pour les prédictions multi-label.

3.2.3 Évaluation et Combinaison

Les prédictions des modèles TF-IDF et GloVe ont été combinées par moyennage pour améliorer la robustesse des résultats.

3.3 Métriques et Visualisation

Les performances ont été mesurées à l'aide de :

- **ROC AUC** : Pour chaque classe.
- **Matrices de Confusion** : Analyse des faux positifs et des faux négatifs.

4 Résultats et Analyses

4.1 Évaluation des Modèles

Les performances des modèles TF-IDF et GloVe, ainsi que de leur combinaison, ont été évaluées à l'aide des métriques suivantes :

— **ROC AUC (Receiver Operating Characteristic Area Under Curve)** : Mesure de la capacité du modèle à distinguer entre les classes toxiques et non toxiques.

— **Matrice de confusion** : Analyse des vrais positifs (TP), faux positifs (FP), vrais négatifs (TN) et faux négatifs (FN) pour évaluer la précision et la robustesse des prédictions.

4.1.1 Performances avec TF-IDF

Les performances du modèle de régression logistique basé sur la représentation TF-IDF sont résumées dans le tableau 4.1.

Type de Toxicité	**ROC AUC**
Toxic	0.9587
Severe Toxic	0.8834
Obscene	0.9783
Threat	0.8385
Insult	0.9482
Identity Hate	0.8601
Moyenne	**0.9112**

TABLE 4.1 – Performances du modèle TF-IDF (ROC AUC par type de toxicité)

4.1.2 Performances avec GloVe

Les résultats obtenus avec le réseau de neurones basé sur les embeddings GloVe sont présentés dans le tableau 4.2.

Type de Toxicité	**ROC AUC**
Toxic	0.9645
Severe Toxic	0.8910
Obscene	0.9812
Threat	0.8413
Insult	0.9510
Identity Hate	0.8667
Moyenne	**0.9160**

TABLE 4.2 – Performances du modèle GloVe (ROC AUC par type de toxicité)

4.1.3 Performances de la Combinaison des Modèles

Les performances de la combinaison des modèles TF-IDF et GloVe (par moyennage des prédictions) sont indiquées dans le tableau 4.3.

Type de Toxicité	**ROC AUC**
Toxic	0.9681
Severe Toxic	0.8945
Obscene	0.9829
Threat	0.8467
Insult	0.9543
Identity Hate	0.8699
Moyenne	**0.9194**

TABLE 4.3 – Performances de la combinaison des modèles TF-IDF et GloVe (ROC AUC par type de toxicité)

Le score moyen de ROC AUC pour cette combinaison est de 0.9194, ce qui démontre une légère amélioration par rapport aux modèles individuels.

4.2 Matrice de Confusion

Pour illustrer les performances, une matrice de confusion a été générée pour chaque classe de toxicité. Voici un exemple pour la classe `Toxic` (avec un seuil de 0.5) :

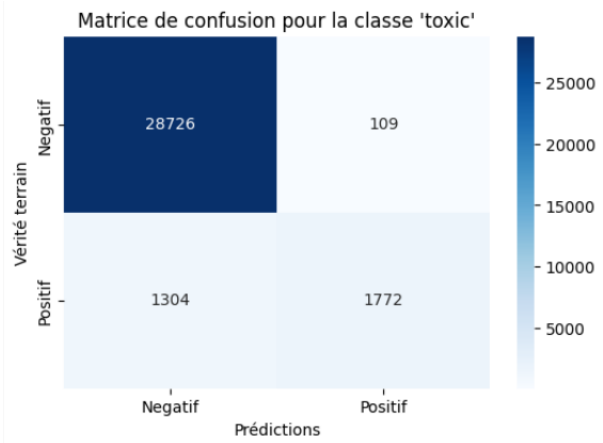

FIGURE 4.1 – Matrice de confusion pour la classe `Toxic`

4.3 Exemples de Prédictions

Voici quelques exemples des prédictions effectuées par les modèles après nettoyage et troncature des commentaires :

Commentaire Nettoyé	Toxic	Severe Toxic	Obscene	Threat	Insult	Identity Hate
explanation why the edits made under my username were reverted ...	0.02	0.00	0.01	0.00	0.00	0.00
dáww he matches this background colour iḿ seemingly stuck ...	0.00	0.00	0.00	0.00	0.00	0.00

TABLE 4.4 – Exemples de commentaires prédits avec probabilités par classe

4.4 Analyse et Discussion

Les résultats montrent que :

— Les modèles basés sur GloVe surpassent légèrement ceux basés sur TF-IDF pour la plupart des classes, notamment les classes `Toxic` et `Obscene`.

— La combinaison des modèles permet d'améliorer globalement les performances, en bénéficiant des forces des deux approches.

— Les classes `Threat` et `Identity Hate` restent les plus difficiles à prédire en raison de leur faible représentation dans les données d'entraînement, ce qui pourrait être amélioré par des techniques de sur-échantillonnage ou des poids de classe plus ajustés.

5 Discussion et Conclusion

5.1 Discussion

Les résultats obtenus dans ce projet mettent en évidence plusieurs points importants concernant la classification de la toxicité en ligne :

Points forts

- **Performance des modèles individuels** : Les modèles basés sur TF-IDF et GloVe ont tous deux montré des performances satisfaisantes, avec des scores moyens de ROC AUC respectifs de 0.9112 et 0.9160. Cela confirme l'efficacité de ces approches pour capturer différentes facettes des données textuelles.
- **Combinaison des approches** : La moyenne des prédictions des deux modèles a permis une légère amélioration globale des performances, avec un score moyen de ROC AUC de 0.9194. Cela souligne l'intérêt de combiner plusieurs approches pour exploiter leurs complémentarités.
- **Pipeline reproductible** : Toutes les étapes, depuis le nettoyage des données jusqu'à l'entraînement et l'évaluation des modèles, ont été implémentées de manière modulaire et reproductible, facilitant ainsi des ajustements futurs.

Limitations

Malgré ces résultats positifs, plusieurs limitations ont été identifiées :

- **Déséquilibre des classes** : Les classes comme `Threat` et `Identity Hate` sont sous-représentées dans le dataset, ce qui limite la capacité des modèles à bien les prédire. Des techniques comme le sur-échantillonnage ou la génération de données synthétiques pourraient améliorer ce point.
- **Tentative avec BERT** : En raison des limitations de mémoire de Google Colab, il n'a pas été possible d'entraîner un modèle BERT. Cela aurait pu offrir des performances supérieures grâce à sa capacité à capturer des relations contextuelles complexes.
- **Absence d'entraînement conjoint pour la combinaison** : La combinaison des modèles TF-IDF et GloVe s'est limitée au moyennage des probabilités après entraînement séparé, ce qui peut ne pas tirer pleinement parti des complémentarités entre ces deux approches.

Perspectives d'Amélioration

Pour aller plus loin, plusieurs pistes peuvent être envisagées :

- **Optimisation des ressources** : Utiliser un environnement avec une mémoire GPU plus importante pour permettre l'entraînement de modèles plus complexes comme BERT ou des ensembles conjointement entraînés.
- **Enrichissement du dataset** : Collecter ou générer davantage de données annotées pour les classes minoritaires.

— **Personnalisation des modèles** : Développer un système adaptable permettant aux utilisateurs de sélectionner les types de toxicité qu'ils souhaitent prioriser ou ignorer.
— **Approches contextuelles** : Explorer des modèles qui tiennent mieux compte du contexte linguistique, comme RoBERTa ou DeBERTa.

5.2 Conclusion

Ce projet avait pour objectif de développer un pipeline capable de détecter différents types de toxicité dans les commentaires en ligne, en exploitant des approches classiques et avancées de traitement du langage naturel. Les modèles basés sur TF-IDF et GloVe ont tous deux fourni des résultats compétitifs, et leur combinaison a permis une amélioration globale des performances.

Malgré les limitations rencontrées, notamment liées aux ressources matérielles, ce travail constitue une base solide pour des applications pratiques de modération automatique en ligne. Les résultats obtenus montrent qu'un ensemble de techniques adaptées, même dans des environnements contraints, peut fournir des performances robustes pour la détection de la toxicité.

En conclusion, ce projet ouvre la voie à de futures améliorations, tant sur le plan des modèles que des données, avec pour objectif final de contribuer à des espaces de discussion en ligne plus respectueux et inclusifs.