# Projet NLP: Détection de fake news


Réalisé par :
PAQUES Anatole, ADNINE Aya, DUCOULOMBIER Quentin, JUMEL Paul

ING3 IA B

Année: 2024-2025

**Table des matières :**

# 1. Introduction

Dans un monde où l'information circule à une vitesse sans précédent, la capacité de distinguer le vrai du faux est devenue une nécessité incontournable. Les fake news, amplifiées par les réseaux sociaux et les plateformes numériques, représentent une menace majeure pour la société, influençant les opinions publiques, les élections et les comportements individuels. Face à ce défi, les technologies d'intelligence artificielle (IA) et d'apprentissage automatique offrent des outils prometteurs pour analyser et détecter ces contenus trompeurs.

Ce projet s'inscrit dans cette démarche, en explorant différentes méthodologies et approches pour la détection automatique des fake news et l'analyse de la prise de position dans des articles. À travers deux ensembles de données distincts, nous avons évalué diverses techniques, allant des algorithmes traditionnels, comme les forêts aléatoires (Random Forest), à des modèles d'apprentissage profond avancés tels que BERT, RoBERTa et XLNet.

L'objectif de ce travail est de comparer l'efficacité de ces modèles sur des tâches spécifiques tout en identifiant les limites des solutions proposées. Ce projet académique vise également à mettre en pratique des concepts étudiés en cours et à développer une meilleure compréhension des défis liés à l'application de l'IA dans le domaine de la désinformation.

# 2. Méthodologies de travail

Pour mener à bien ce projet, nous avons adopté une approche structurée en deux phases principales, chacune centrée sur un jeu de données distinct.

Dans un premier temps, nous avons travaillé sur le Fake and Real News Dataset, un ensemble de données conçu pour classifier les articles comme étant vrais ou faux. Afin d'établir une base solide, nous avons implémenté un algorithme de Random Forest. Cette étape a été réalisée rapidement, car elle visait principalement à fournir un modèle de référence simple pour évaluer la faisabilité et la performance initiale de la tâche de classification.

Ensuite, nous avons orienté nos efforts vers le Stance Detection Dataset, un ensemble de données plus complexe qui requiert d'identifier les relations (*agree*, *disagree*, *discuss*, *unrelated*) entre des titres et des corps d'articles. Ici, chaque membre de l'équipe a exploré et implémenté un modèle spécifique, notamment BERT, RoBERTa et XLNet, afin de comparer leurs performances sur cette tâche.

Enfin, nous avons testé des méthodes avancées telles que LoRA (Low-Rank Adaptation) et Few-Shot Learning. Bien que prometteuses, ces approches n'ont pas produit de résultats significatifs dans notre contexte.

## 3. Dataset 1 : Fake and Real News Dataset

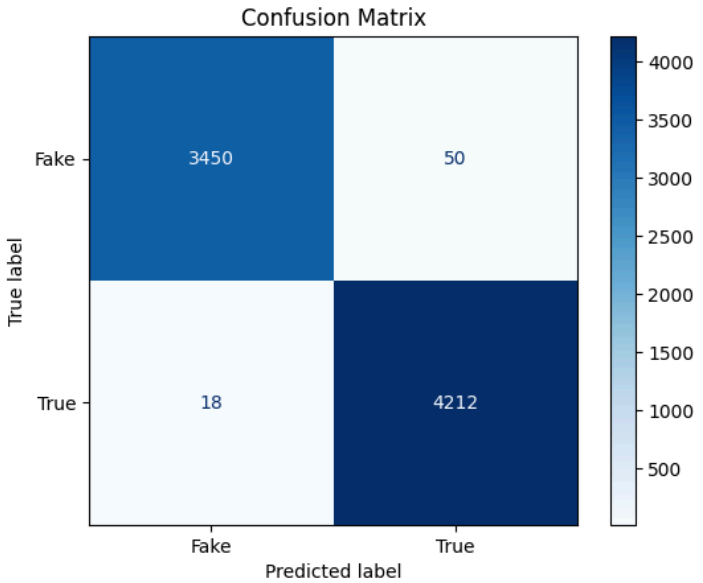
Le dataset utilisé dans cette première partie est séparé en deux fichiers: un fichier Fake.csv contenant 23502 Fake news et un fichier True.csv contenant 21417 Real news. Les deux sont composés de quatre colonnes:

1. ***Title*** correspondant au titre de l'article
2. ***Text*** correspondant au corps de l'article
3. ***Subject*** correspondant au sujet de l'article
4. ***Date*** correspondant de la date de publication de l'article

Après avoir fusionné les données et ajouté une colonne ***label*** (0 pour "Fake" et 1 pour "True") pour constituer le dataset final, nous avons procédé à plusieurs étapes de traitement des données.

En premier le traitement des textes, qui a impliqué plusieurs étapes essentielles, comme la conversion en minuscules, la suppression des caractères spéciaux, la tokenisation et la lemmatisation, tout en excluant les stopwords. Ces textes ont ensuite été transformés en représentations numériques à l'aide de la méthode TF-IDF, qui capture l'importance des mots et phrases dans un corpus.

Nous avons entraîné un modèle **Random Forest Classifier** pour distinguer les nouvelles vraies des fausses. Avec une précision globale de **99.12%** sur les données de test, le modèle a montré une excellente performance, classant correctement la majorité des exemples, comme l'illustre la matrice de confusion ci-dessous. Les résultats confirment l'efficacité de cette approche pour une tâche relativement simple.



# 4. Dataset 2 : Stance Detection Dataset

Le **Stance Detection Dataset** a été utilisé pour la compétition Fake News Challenge (FNC-1) en 2017. Son objectif principal était de déterminer la relation entre un titre et le corps d'un article parmi les catégories **agree**, **disagree**, **discuss**, ou **unrelated**.

## Objectif

L'objectif était de construire des modèles capables de :

1. Identifier si un titre soutient, contredit, discute ou est sans lien avec le contenu d'un article.
2. Fournir une analyse fine pour améliorer la détection de fausses informations (fake news).

## Évaluation des scores

Les soumissions étaient évaluées via un script dédié, selon les règles suivantes :

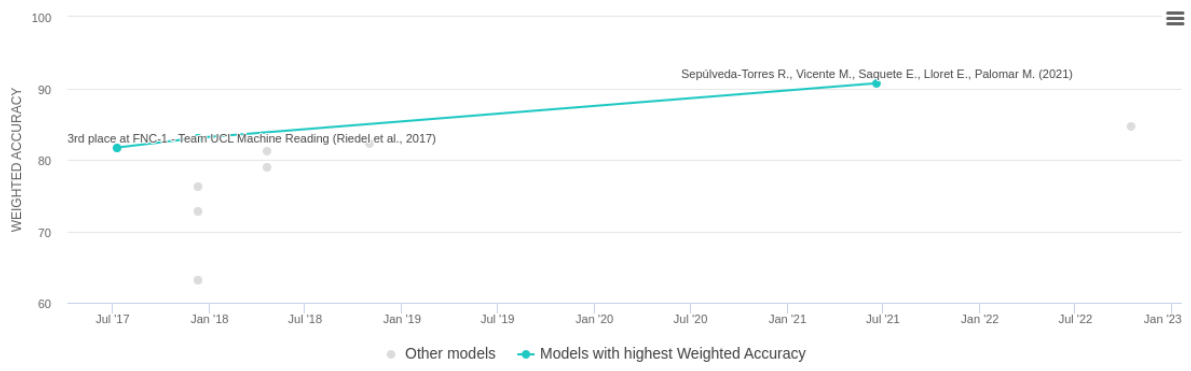
- **+0.25** pour chaque prédiction correcte de la classe **unrelated**.
- **+0.25** pour chaque prédiction correcte d'une classe **related** (agree, disagree, discuss).
- **+0.75** supplémentaires pour chaque prédiction exacte parmi les classes **agree**, **disagree**, ou **discuss**.

Ainsi, le score total récompensait particulièrement la précision dans les prédictions **related** (classe la plus importante).

## Résultats

Lors de la compétition de 2017, les gagnants ont obtenu un score de **9556.50**. Ce score a été calculé sur la base de la somme des points pour toutes les prédictions sur l'ensemble de test. Cependant, ces performances ont été atteintes avant l'arrivée des modèles modernes tels avec des transformers tel que **BERT**, ou des LLM, qui ont depuis révolutionné le domaine du NLP.

Les trois équipes gagnantes ont publié des papier détaillant leurs méthodologies.



## 4.1 BERT et RoBERTa

### 4.1.a Architecture

**BERT (Bidirectional Encoder Representations from Transformers)** repose sur l'architecture des transformers, introduite en 2017 par le papier attention is all you need. Ce qui distingue BERT des autres modèles est sa nature bidirectionnelle : il tient compte du contexte des mots à la fois à gauche et à droite dans une phrase.

Le pré-entraînement de BERT repose sur deux tâches principales :

1. **Masked Language Modeling (MLM)** : une partie des tokens est masquée et le modèle apprend à les prédire en tenant compte du contexte.
2. **Next Sentence Prediction (NSP)** : le modèle apprend à prédire si deux phrases se suivent dans un texte.

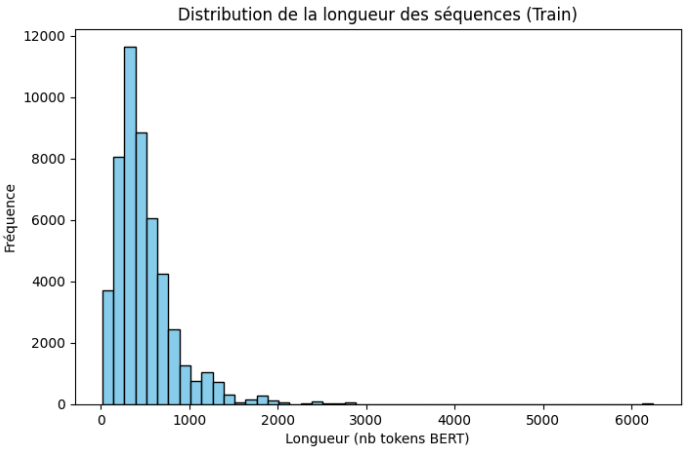
**RoBERTa (A Robustly Optimized BERT Pretraining Approach)** est une amélioration directe de BERT, introduite par Facebook AI. Les principales différences incluent :

1. **Optimisation des hyperparamètres** : RoBERTa est entraîné avec des batches plus grands et des séquences plus longues.
2. **Suppression de NSP (Next Sentence Prediction)** : considéré comme inutile.
3. **Plus de données et d'itérations** : RoBERTa utilise un volume de données pré-entraînement plus important.

RoBERTa est globalement plus performant que Bert mais prend plus de temps d'entraînement.
Dans notre cas, nous utilisons **`bert-base-uncased`** et **`roberta-base`**, des modèles pré-entraînés, adaptés à la tâche de classification.

### 4.1.b Méthodologie

1. **Préparation des données** :
   - Les titres (`Headline`) et corps d'articles (`articleBody`) sont combinés en une seule séquence textuelle, séparée par le token `[SEP]` (pour BERT) ou `</s>` (pour RoBERTa).
   - Les labels (`agree`, `disagree`, etc.) sont convertis en indices numériques via un mapping (`label2id`).
   - **Gestion de MAX_LENGTH** : Pour éviter des problèmes de mémoire GPU, nous avons limité la longueur maximale des séquences à un niveau couvrant environ 85 % des données.

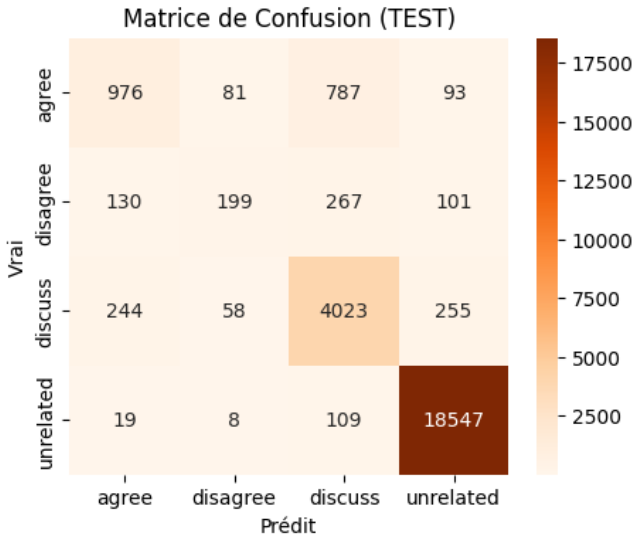


2. **Encodage** :
    - Les textes combinés sont tokenisés avec **BertTokenizer** (pour BERT) ou **RobertaTokenizer** (pour RoBERTa) et encodés en vecteurs d'entrée.
3. **Entraînement** :
    - Nous utilisons **TFBertForSequenceClassification** et **TFRobertaForSequenceClassification** pour fine-tuner les modèles sur les données d'entraînement avec une perte par entropie croissante (SparseCategoricalCrossentropy).
    - Optimisation via Adam (étape de 2e-5) et batch size de 8.
    - BERT est entraîné sur 3 époques, tandis que RoBERTa est limité à 2 époques en raison de contraintes de temps GPU.
4. **Évaluation** :
    - Les performances sont mesurées à l'aide de précision, rappel, F1-score et sur les metrics du dataset.

### 4.1.c Résultats

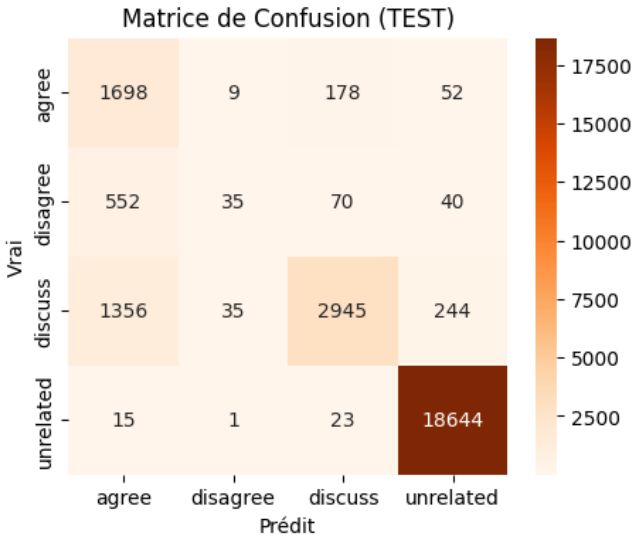
**Test (BERT)**

- **Score TEST** : **10,226.50**.
- **Accuracy** : **91.7%**.




**Test (RoBERTa)**

- **Score TEST** : **9889.00**.
- **Accuracy** : **90.1%**.




## 4.2 XLNet

**Présentation de XLNet**

XLNet est un modèle de langage développé par Google AI, introduit en 2019 comme une amélioration des modèles précédents tels que BERT. Il combine les avantages des modèles autorégressifs et auto-encodeurs en utilisant une approche innovante appelée "modélisation du langage par permutation".

**Performances sur les benchmarks populaires**

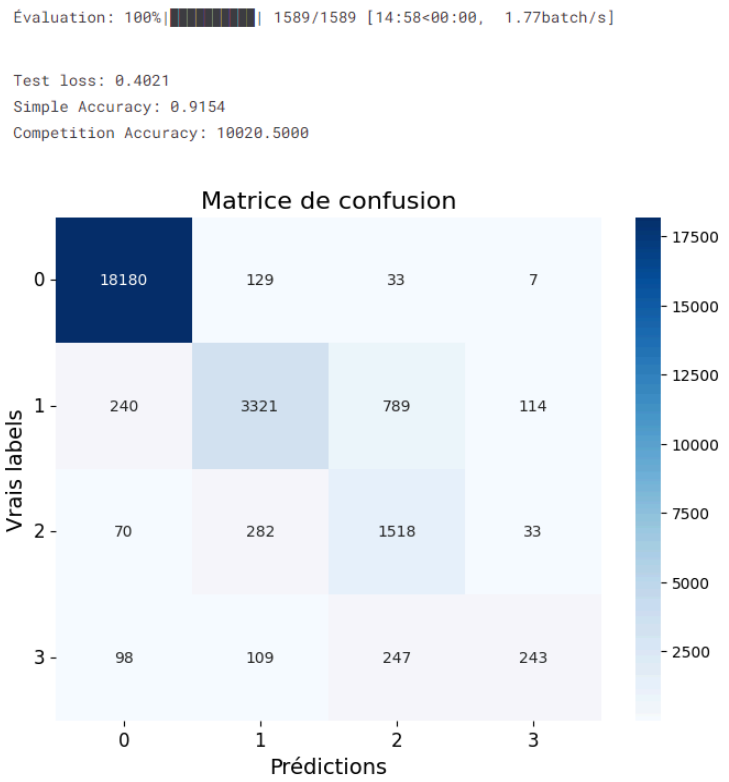
XLNet a démontré des performances exceptionnelles sur divers benchmarks en traitement du langage naturel (NLP). Par exemple, sur le jeu de données **AG News**, utilisé pour la classification de textes, XLNet a atteint des résultats de pointe, surpassant de nombreux modèles concurrents. Actuellement, il est considéré comme le 2ème meilleur modèle de classifications de textes selon Paperswithcode.

**Efficacité de XLNet**

Plusieurs facteurs contribuent à l'efficacité de XLNet :

- **Modélisation par permutation** : En considérant toutes les permutations possibles des séquences de mots, XLNet capture des dépendances bidirectionnelles, offrant une compréhension contextuelle plus riche.
- **Intégration de Transformer-XL** : XLNet incorpore des mécanismes de mémoire à long terme de Transformer-XL, lui permettant de gérer efficacement des contextes plus longs et d'améliorer la cohérence des prédictions.
- **Absence de masquage explicite** : Contrairement à BERT, qui masque des tokens pendant l'entraînement, XLNet évite le décalage entre l'entraînement et l'inférence en n'utilisant pas de masquage explicite, ce qui conduit à des représentations linguistiques plus naturelles.

Pour un modèle entraîné sur 4 epoches nous obtenons 0.915 d’accuracy.



# 5. Méthodes moins précises

## 5.1 LoRa

La méthode **LoRA (Low-Rank Adaptation)** est une technique d'ajustement efficace pour les modèles de langage de grande taille. Elle consiste à geler les poids d'origine du modèle préentraîné et à injecter des matrices de faible rang dans les couches spécifiques. Cette approche réduit significativement les besoins en ressources de calcul et en mémoire, rendant l'entraînement plus rapide et accessible, même sur des infrastructures limitées. LoRA est particulièrement prisée pour les tâches de fine-tuning lorsqu'il est crucial de maintenir des coûts d'entraînement réduits.

Dans le cadre de notre projet, nous avons testé l'entraînement avec la bibliothèque **Unsloth** en appliquant LoRA sur le modèle **Llama 3.1 8B**. Cependant, nous avons rencontré plusieurs contraintes liées au temps d'entraînement. Nous ne disposions pas de suffisamment de matériel pour faire un entraînement complet sur un modèle de ce type.

Afin d'obtenir une évaluation rapide, nous avons tenté un entraînement limité à une seule époque et quelques étapes. Cependant, les résultats obtenus étaient **non significatifs**, nous avons atteint une accuracy de **72%**, ce qui est quasi similaire aux résultats que nous avions eu sur un modèle Llama non fine-tuned.

## 5.2 Few-Shot Learning

Cette implémentation exploite le modèle pré-entraîné **Llama 3.1 8B** via Ollama pour classifier les relations entre des titres et des corps d’articles. L’approche repose sur le few-shot learning, où quelques exemples pertinents sont intégrés dans le prompt pour guider le modèle.

Le modèle Ollama est initialisé comme un serveur, permettant une interaction efficace. Le prompt est construit en combinant quatre exemples de few-shot learning illustrant les relations possibles entre un titre et un corps d’article. Chaque exemple inclut un titre, un extrait de corps et la classification correspondante, accompagnés d’instructions strictes pour limiter les réponses aux seules classes définies. Les corps d’articles sont tronqués à 512 caractères pour éviter de dépasser les limites du modèle.

Les prompts ainsi construits sont envoyés au modèle pour chaque exemple du dataset de test, et les réponses sont récupérées sous forme de texte brut représentant la relation prédite.

Cette approche atteint une accuracy de **72.84%** sur le dataset de compétition. Cette performance est notable pour une tâche de classification complexe utilisant uniquement quelques exemples dans le prompt, sans nécessiter de ré-entraînement du modèle. Un des principaux points forts de cette méthode est sa rapidité et sa flexibilité. Mais, l’accuracy reste très inférieur par rapport au méthode vu précédemment.

## 6. Comparaison des performances

| Modèle | BERT | XLNet | RoBERTa | Llama 3.1 8B |
|---|---|---|---|---|
| **Accuracy** | 0.917 | 0.915 | 0.901 | 0.72 |
| **Score de compétition** | 10226.50 | 10020 | 9889.00 | 4587 |

## 7. Conclusion

Ce projet sur la détection de fake news a permis d'explorer diverses approches pour traiter un problème essentiel dans notre société actuelle. En travaillant sur deux jeux de données distincts, nous avons démontré l'efficacité des modèles traditionnels, comme les forêts aléatoires, pour des tâches simples, avec des résultats très satisfaisants. Cependant, nous avons également mis en lumière les limites de ces approches face à des scénarios plus complexes, nécessitant une compréhension contextuelle plus fine.

L'utilisation de modèles avancés tels que BERT, RoBERTa et XLNet a permis de traiter des tâches plus nuancées, comme la détection de relations entre titres et articles, avec des performances solides. Bien que des méthodes modernes comme LoRA et le few-shot learning aient été testées, elles n'ont pas atteint le même niveau de précision, mais offrent néanmoins des perspectives intéressantes pour des travaux futurs grâce à leur flexibilité et leur rapidité.

En conclusion, ce projet met en évidence l'importance des avancées récentes en traitement du langage naturel dans la lutte contre la désinformation. Les résultats obtenus soulignent non seulement l'efficacité des modèles actuels, mais aussi les défis techniques à relever pour atteindre une robustesse optimale. Ce travail constitue une base solide pour poursuivre la recherche et l'innovation dans ce domaine crucial.

## 8. Liens vers les codes 🔥

Dataset 1 : Random Forest
Dataset 2 :

- BERT
- RoBERTa
- XLNet
- Llama : LoRa
- Few-Shot Learning