

## RAPPORT ECRIT FINAL

*Comment développer et optimiser un modèle de classification de texte, en particulier en utilisant le finetuning de BERT, pour la classification des émotions sur la base de données dair-ai/emotion, tout en assurant l'explicabilité des décisions du modèle ?*



COLLO Brian, THIRUALAGAN Thinnujan

ESME

Encadrant, Tuteur de stage : KHEDER Mohamed

## Sommaire : (PREMIERE PARTIE : ETAT DE L'ART)

Introduction

Méthodes classiques pour la classification de texte

Avancées avec le Deep Learning

Revolution avec les modèles bases sur l'attention (BERT)

L'Analyse des Emotions et la Classification des Sentiments avec BERT et DistilBERT

Fine tuning de BERT

Métriques d'évaluation classification de texte

Explicabilité/XAI

Datasets populaires

Enjeux actuels et défis

Applications modernes

Projets Existants

Perspectives pour la classification de texte et analyse de sentiment

Conclusion

Sommaire : (DEUXIEME PARTIE : DEVELOPPEMENT DU PROJET)

Introduction

Remarques du Jury et Améliorations Effectuées

Aspects de la Conception

Développement Réalisé

Analyse des Résultats Obtenus

Conclusion

## Introduction

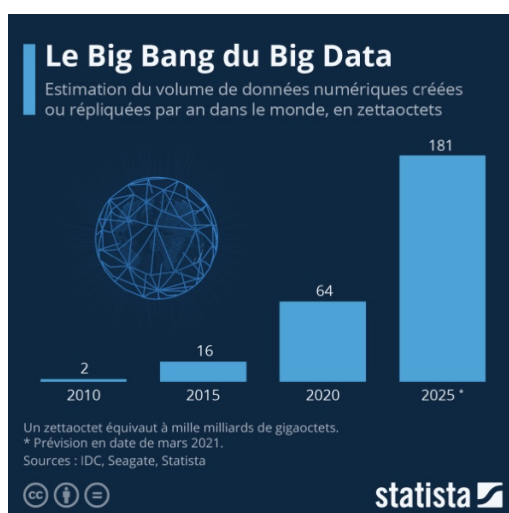
La classification de texte est une tâche fondamentale du traitement automatique du langage naturel (TALN), qui consiste à assigner une étiquette prédéfinie à un texte donné<sup>i</sup> (Manning et al., 2008). Dans le cadre de l'analyse des sentiments, cette étiquette correspond à une émotion telle que la joie ou la colère<sup>ii</sup> (Cambria et al., 2017).

L'utilité de ce procédé est d'extraire des informations significatives de larges volumes de données textuelles non structurées, une capacité cruciale dans de nombreux domaines modernes.

La classification des émotions est utilisable dans de très nombreux domaines comme le commerce et marketing (analyse des avis clients pour améliorer les produits et services)<sup>iii</sup> (Liu, 2012), les réseaux sociaux (analyse des tendances émotionnelles sur une actualité) ou encore la santé mentale (analyse d'émotions négatives dans des textes, messages ou posts sur les réseaux sociaux)<sup>iv</sup> (Calvo et D'Mello, 2010).

Grâce à des modèles comme BERT, il est désormais possible d'améliorer grandement les performances des modèles de classification, en capturant des subtilités linguistiques complexes<sup>v</sup> (Devlin et al., 2019).

Dans un contexte mondial où le nombre de données textuelles augmente exponentiellement, il est crucial que la capacité à analyser rapidement et précisément ces données évolue également. Identifier les émotions sert à mieux comprendre les comportements pour mieux adapter les stratégies commerciales et même sociétales, afin de répondre aux besoins des utilisateurs de manière proactive.



Selon les prévisions, le volume mondial de données devrait dépasser 180 zettaoctets d'ici 2025, avec une croissance annuelle moyenne de près de 40 % sur cinq ans.<sup>vi</sup>

Cependant, malgré les avancées technologiques, plusieurs défis persistent :

- **Les émotions sont subjectives et contextuelles**, rendant leur classification complexe<sup>vii</sup> (Schuller et al., 2011).
- **Les jeux de données**, comme dair-ai/emotion, nécessitent des ajustements spécifiques pour capturer les nuances des émotions exprimées dans des contextes variés.

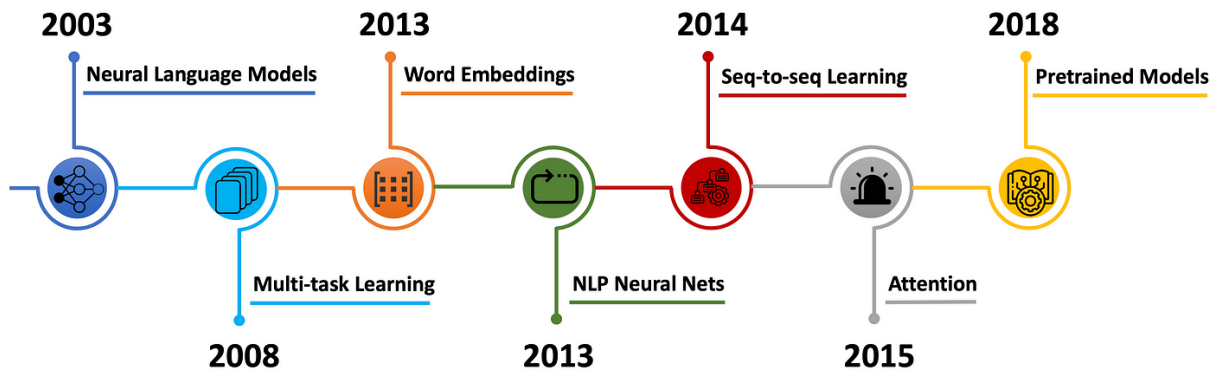
Les modèles de classification de texte doivent relever plusieurs défis pour être efficaces :

- **Représentation du texte** : Les modèles doivent comprendre des nuances linguistiques, comme l'ironie ou les subtilités des émotions exprimées dans des phrases courtes, notamment dans des tweets<sup>viii</sup> (Pang et Lee, 2008).
- **Explicabilité** : Les approches traditionnelles de Deep Learning, bien qu'efficaces, manquent souvent d'interprétabilité. Cela limite leur utilisation dans des applications où la transparence est essentielle (par exemple, la santé mentale ou les services juridiques)<sup>ix</sup> (Doshi-Velez et Kim, 2017).
- **Adaptabilité des modèles** : Les données utilisées pour l'entraînement sont souvent spécifiques à un contexte. Leur capacité à généraliser à d'autres cas d'utilisation reste une limite importante<sup>x</sup> (Ruder et al., 2019).

Cet État de l'art se propose d'explorer les méthodes modernes de classification de texte, en mettant un accent particulier sur :

- L'utilisation de **BERT, plus particulièrement de DistilBert et de son fine-tuning** pour améliorer les performances des modèles sur la classification des émotions.
- L'explicabilité des décisions des modèles grâce à XAI, permettant d'identifier les caractéristiques textuelles influençant les prédictions.
- Les défis et opportunités liés à la base de données dair-ai/emotion, qui contient des tweets classifiés en six émotions distinctes.

Le lecteur en apprendra aussi plus sur les méthodes qui ont précédé BERT. Quelques méthodes classiques pour la classification de texte seront développées et expliquées comme le TF-IDF puis le Random Forest. Le RNN et le LSTM seront développés dans une partie qui traitera des avancées avec le Deep Learning.



Cette frise chronologique montre l'évolution dans le temps des différentes méthodes, nous traiterons de NLM avec les n-grams, Word embedding avec Word2Vec, Seq-to-Seq Learning avec les LSTM et des pretrained models avec BERT et GPT. <sup>xi</sup>

L'état de l'art explorera aussi différentes Métriques d'évaluation pour la classification de texte, le domaine de l'explicabilité avec XAI, quelques datasets populaires, mes enjeux et défis de la classification de texte et de sentiment, quelques applications modernes et les perspectives de méthodes et de technologie pour la classification. Le tout pour donner du contexte et de la profondeur à l'utilisation de BERT, qui est le sujet central de notre état de l'art.

## Méthodes classiques pour la classification de texte

Première étape : la préparation des données

L'apprentissage supervisé repose sur l'utilisation d'un ensemble de données annotées pour entraîner un modèle capable de prédire des étiquettes pour de nouvelles données. Dans le cas de la classification de texte, il est essentiel de convertir les données textuelles en représentations numériques exploitables par les algorithmes.

Deux approches populaires sont :

- TF-IDF : méthode classique pour représenter des textes sous forme de vecteurs, où chaque dimension correspond à un mot ou un terme.
  - ➔ TF (Term Frequency) : mesure la fréquence d'un mot dans le document
  - ➔ IDF (Inverse Document Frequency) : réduit l'importance de certains mots communs (par exemple : le, et, de)

Les deux notions sont combinées par une simple multiplication et donnent un score pour chaque terme. En combinant ces deux mesures, TF-IDF permet de mettre en avant les termes les plus représentatifs d'un document par rapport au corpus global (Ramos, 2003).

**TF-IDF**

TF-IDF is a measure of originality of a word by comparing the number of times a word appears in a doc with the number of docs the word appears in.

$$\text{TF-IDF} = \text{TF}(t, d) \times \text{IDF}(t)$$

Term frequency

Number of times term  $t$  appears in a doc,  $d$

Inverse document frequency

$\log \frac{1 + n}{1 + \text{df}(d, t)}$

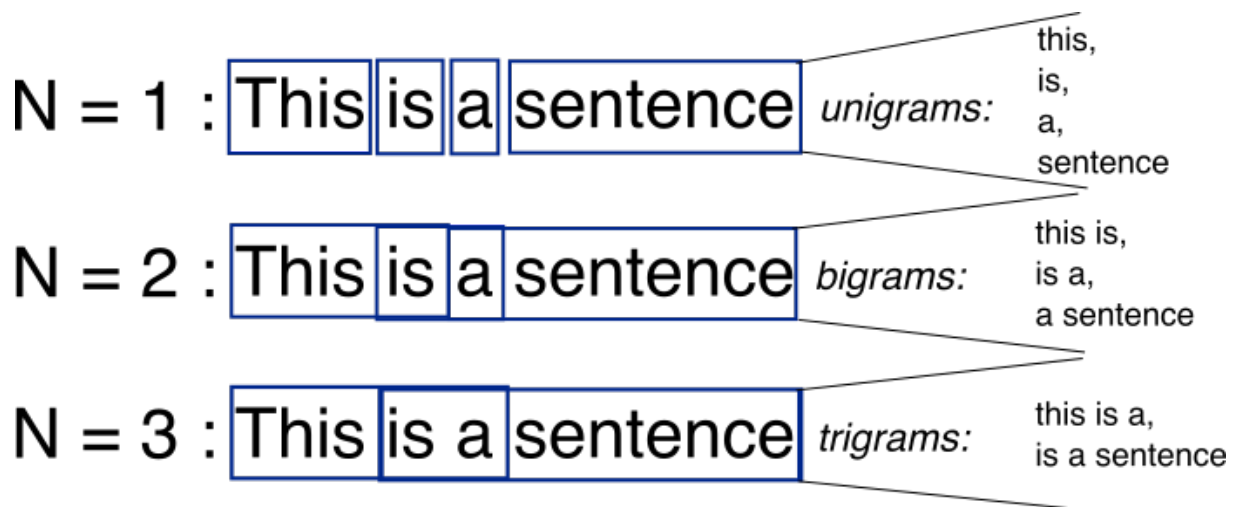
$n$  ← # of documents

$\text{df}(d, t)$  ← Document frequency of the term  $t$

« Le TF-IDF est une mesure d'originalité d'un mot »<sup>xii</sup>. Sur l'image, on peut voir la formule très simple qui relie la fréquence du terme (nombre entier) à sa fréquence inverse (aussi nombre entier). La formule est une simple multiplication.

- N-grams : capturent les séquences de n mots consécutifs dans un document
  - ➔ Unigrams (n=1) : Analyse des mots isolés
  - ➔ Bigrams (n=2), Trigrams (n=3) : Capturent les relations entre mots voisins, améliorant la capacité à comprendre les expressions et les contextes locaux (Jurafsky & Martin, 2023).

Ces représentations permettent de mieux capturer la structure du langage, notamment dans des tâches comme l'analyse des sentiments.



Le schéma montre un découpage par n-grams très simple pour  $n = 1$  ou  $n = 2$  ou  $n = 3$ <sup>xiii</sup>.

À partir d'un corpus, calcul des probabilités qu'un mot apparaisse après un autre (probabilité que « This » suive « a », probabilité que « a » suive « sentence »), puis en utilisant ces probabilités prédiction du mot le plus probable qui suit une séquence donnée.

Application concrète : clavier prédictif (sur téléphone, ordinateur) pour proposer les prochains mots. Autre application concrète : suggestion du prochain mot dans les moteurs de recherche (en fonction des groupes de mots fréquemment cherchés ensemble)



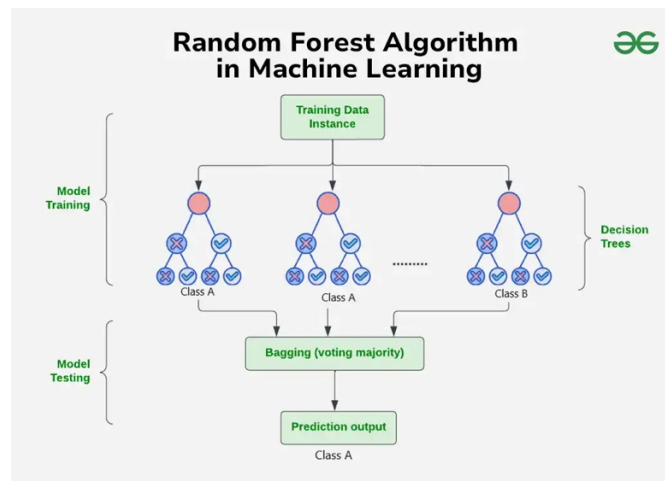
## Deuxième étape : Application des algorithmes

Une fois les textes représentés sous forme de vecteurs, les algorithmes supervisés peuvent être utilisés pour la classification. Parmi les plus courants :

- Naive Bayes : algorithme probabiliste simple mais puissant, particulièrement efficace pour les tâches de classification de texte comme la détection de spams ou l'analyse de sentiments (McCallum & Nigam, 1998). L'algorithme repose sur le théorème de Bayes et suppose l'indépendance conditionnelle des mots (limitation car elle simplifie excessivement les relations entre les mots dans le texte ce qui pourra occasionner des prédictions ou résultats sous-optimaux). La simplicité de l'algorithme en fait un choix idéal pour des tâches où la rapidité et l'efficacité priment.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

- Random Forests : méthode d'ensemble basée sur des arbres de décision (Breiman, 2001). Construction de plusieurs arbres de décision sur des sous-échantillons aléatoires des données, puis combine les prédictions. Random Forests est robuste face au surapprentissage et peut traiter efficacement des données de haute dimension comme générées par TF-IDF ou N-grams. L'algorithme possède néanmoins de nombreux mauvais aspects : temps de calcul long (due à la construction des nombreux arbres de décision), consommation mémoire élevée, perte d'interprétabilité et risque de surapprentissage.



Le schéma nous représente en image l'algorithme du random Forest et ces différentes étapes.<sup>xiv</sup>

Si les méthodes classiques, comme celles basées sur TF-IDF ou n-grams, offrent une bonne performance pour des tâches simples de classification de texte, elles présentent toutefois des limitations importantes. En particulier, elles ne capturent ni le contexte global ni les relations sémantiques profondes entre les mots d'un texte.

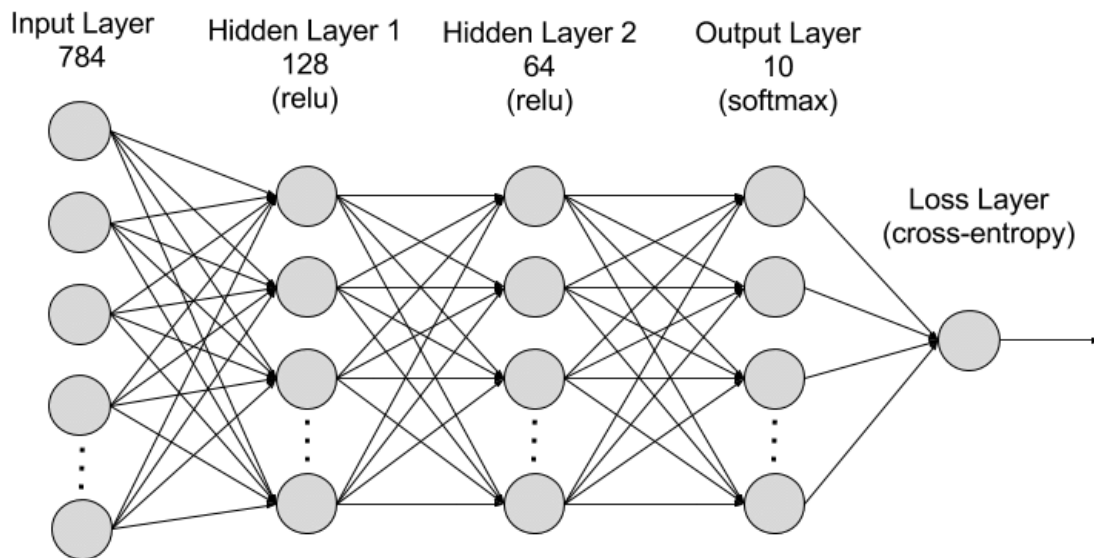
C'est ici que des modèles avancés comme **BERT** apportent une véritable révolution. En utilisant une architecture basée sur les transformateurs, BERT permet :

- De capturer les dépendances entre mots dans les deux directions (avant et arrière), offrant une compréhension contextuelle bien supérieure.
- De s'affranchir des représentations fixes comme TF-IDF, en générant des représentations dynamiques des mots adaptées au contexte spécifique du texte.

Ces avancées, que nous explorerons plus en détail dans la suite de ce document, permettent de surmonter les limites des méthodes classiques et d'atteindre des performances sans précédent dans la classification de texte.

## Avancées avec le Deep Learning

Les réseaux de neurones récurrents (RNN) traitent les données séquentiellement en conservant une mémoire des états précédents grâce à des boucles internes dans leurs couches. À chaque étape, l'état précédent est combiné à l'entrée actuelle pour produire une sortie et un nouvel état caché.



Cette image nous représente un schéma d'un RNN type. La couche d'entrée reçoit les **données initiales** que le réseau va traiter, captation de l'information brute (on pourrait comparer cette couche aux yeux). Les hidden layers font tout le travail complexe d'analyse. Chaque couche cachée (hidden layer) reçoit les données de la couche précédente. On appelle ces couches « cachées » car elles ne sont pas visibles pour l'utilisateur. Ce qui se passe à l'intérieur est une série de calculs mathématiques (on pourrait apparenter ces couches à notre cerveau qui réfléchit). La couche de sortie sert à produire et afficher le résultat final (on pourrait apparenter cette couche à la bouche, qui servira pour transmettre la réponse ou réflexion faite juste avant). Source de l'image<sup>xv</sup>.

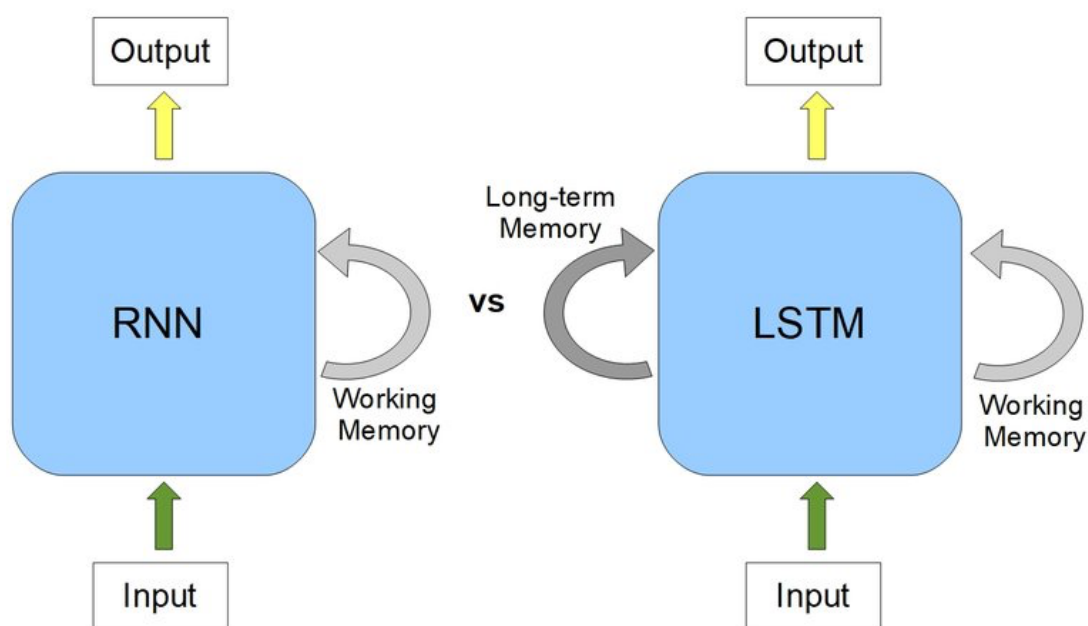
Les réseaux de neurones récurrents (RNN) sont une classe de modèles adaptés au traitement de données séquentielles, tel que le texte. Les RNN introduisent des connexions récurrentes qui permettent de conserver une "mémoire" des états précédents pour capturer les dépendances temporelles dans les données<sup>xvi</sup> (Elman, 1990). Cependant, leur structure souffre de limitations, notamment le problème de vanishing gradients (Le problème des **vanishing gradients** survient lorsque, dans les réseaux neuronaux profonds, les gradients deviennent extrêmement petits lors de la rétropropagation, empêchant les couches initiales d'apprendre efficacement), qui rend difficile l'apprentissage des dépendances à long terme

dans les séquences. Cela les rend moins adaptés à des tâches complexes nécessitant une compréhension globale du contexte, comme la classification d'émotions dans des textes. Ces défis ont ouvert la voie à des architectures plus performantes, notamment celles utilisées dans BERT.

En effet, BERT surmonte les limitations des RNN en adoptant une architecture Transformer qui permet de traiter l'ensemble de la séquence textuelle simultanément, au lieu de la traiter de manière séquentielle. Ce changement structurel permet de capturer des relations à longue portée entre les mots, une capacité essentielle pour modéliser des émotions complexes dans un texte.

Malgré leurs limites, les RNN ont été largement utilisés dans des tâches comme la reconnaissance de la parole<sup>xvii</sup> (Graves et al., 2013) ou la traduction automatique<sup>xviii</sup> (Sutskever et al., 2014). Ces modèles ont marqué un tournant dans le traitement du langage naturel (NLP) en raison de leur capacité à traiter efficacement les relations séquentielles au sein des données textuelles.

Pour répondre aux limitations des RNN, Hochreiter et Schmidhuber<sup>xix</sup> (1997) ont introduit les réseaux à mémoire à long terme (LSTM). Grâce à des mécanismes de portes (entrée, sortie, oubli), les LSTM permettent de conserver ou d'oublier des informations spécifiques, facilitant ainsi l'apprentissage des dépendances sur des séquences plus longues. Cette innovation a permis d'améliorer considérablement la performance sur des tâches comme la classification de sentiment<sup>xx</sup> (Tang et al., 2015).

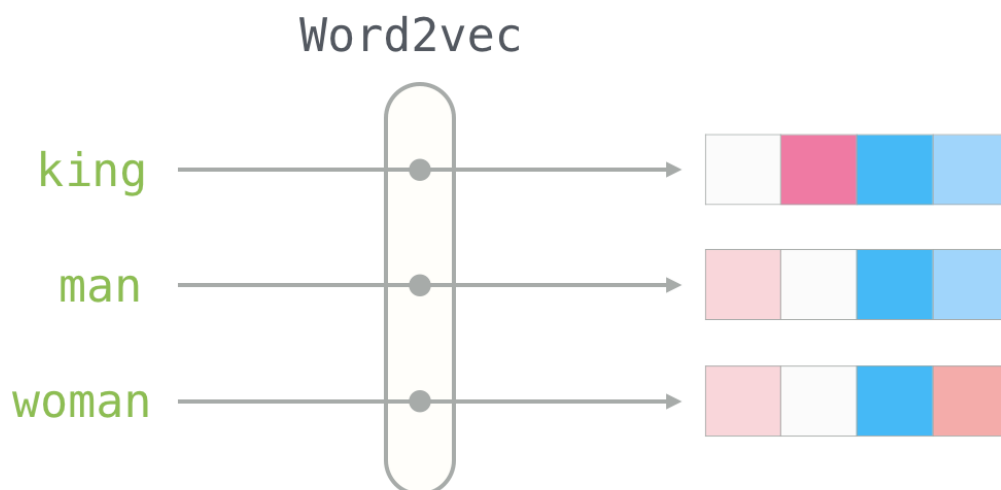


Les LSTM sont une version améliorée des RNN, puisqu'ils peuvent gérer des dépendances longues. Contrairement à un RNN classique, il peut **se souvenir d'informations importantes** sur une longue période et **oublier celles qui ne sont pas utiles**. Voici pourquoi sur le schéma<sup>xxi</sup>, on peut remarquer l'ajout de la notion de long-term memory.

Dans le cadre de la classification des émotions, les LSTM offrent un avantage en capturant les changements subtils dans le ton ou le contexte à travers une phrase. Cependant, bien qu'efficaces, les LSTM restent limités par leur approche séquentielle, qui empêche le traitement parallèle des données.

C'est là que BERT prend l'avantage. Contrairement aux LSTM, BERT utilise l'attention bidirectionnelle, permettant de considérer simultanément le contexte à gauche et à droite d'un mot. Cette capacité rend BERT particulièrement adapté à des tâches nécessitant une compréhension fine des émotions dans un texte, où chaque mot peut dépendre de multiples niveaux de contexte.

L'introduction de Word2Vec<sup>xxii</sup> (Mikolov et al., 2013) a marqué un tournant en NLP en proposant des représentations continues des mots dans un espace vectoriel. Ces Word embeddings capturent des relations sémantiques et syntaxiques, facilitant ainsi les tâches de classification de texte. Par exemple, dans un contexte de classification d'émotions, Word2Vec permet de rapprocher des mots émotionnels similaires, tels que "joyeux" et "heureux", dans l'espace vectoriel.



Le schéma ci-contre représente bien les liens qui peuvent être fait entre certains mots : on retrouve une similarité/association entre le mot man et king (un « roi » ne peut-être qu'un « homme ») ainsi qu'une autre similarité/association entre man et woman (même si ces deux termes peuvent sembler diamétralement opposés, les deux représentent un humain, une

personne) et ces similarités se traduisent ici par un schéma/pattern de couleur semblable que l'on pourra retrouver sur ces associations de mots. Image<sup>xxiii</sup>.

Cependant, Word2Vec reste limité à des représentations statiques, où chaque mot a un unique vecteur, quel que soit le contexte dans lequel il apparaît. Cette limitation pose un problème pour des tâches comme la classification des émotions, où le sens d'un mot peut varier selon le contexte.

BERT dépasse cette limitation en générant des représentations dynamiques des mots, adaptées à leur contexte. Par exemple, le mot "froid" peut signifier une température basse ou un manque d'émotion, selon la phrase. Grâce à sa capacité contextuelle, BERT est capable de désambiguïser ces significations, offrant ainsi des représentations plus précises pour la classification des émotions.

En conclusion de cette partie, on pourrait dire que les RNN, LSTM et Word2Vec ont joué un rôle fondamental dans le développement du Deep Learning pour le traitement du langage naturel et que bien qu'ils aient introduit des avancées significatives, leurs limitations en matière de traitement du contexte global et de parallélisation ont ouvert la voie à des modèles plus sophistiqués comme BERT. En combinant une compréhension fine du contexte et une capacité à traiter les séquences globalement, BERT représente une avancée majeure pour des tâches comme la classification d'émotions, en répondant aux besoins spécifiques de cette problématique.

## **Revolution avec les modèles basés sur l'attention - BERT**

Les modèles basés sur l'attention, proposés par Vaswani et al.<sup>xxiv</sup> (2017), ont révolutionné le traitement du langage naturel avec leur architecture Transformer. Ce modèle élimine les mécanismes récurrents et convolutionnels en introduisant l'attention autorégressive (*self-attention*), permettant de capturer les dépendances globales dans les séquences tout en améliorant la parallélisation durant l'entraînement. Cette architecture a permis des progrès significatifs dans des tâches comme la traduction automatique.

Proposé par Devlin et al.<sup>xxv</sup> (2018), BERT (*Bidirectional Encoder Representations from Transformers*) repose sur un encodeur bidirectionnel, ce qui lui permet de comprendre le contexte d'un mot à partir de ses voisins gauches et droits. Cette approche a marqué un tournant dans le traitement du langage naturel grâce à deux innovations majeures : le masquage de mots (*Masked Language Modeling*) et la prédiction de la phrase suivante (*Next Sentence Prediction*).

BERT est un très bon exemple pour parler des Transformers en général puisqu'il utilise la partie « Encoder » de l'architecture Transformers et mets aussi en lumière les mécanismes clés de ceux-ci : le « Self-Attention » : la capacité à se concentrer sur différentes parties d'une phrase pour comprendre le contexte global ainsi que la notion de bidirectionnalité, caractéristique distinctive des Transformers.

BERT est comme un lecteur attentif, il va d'abord lire toute la phrase pour ensuite essayer d'en comprendre le sujet, contrairement aux RNN ou Word2vec qui lisent les mots les uns après les autres. Pour prendre un autre exemple pour expliquer le fonctionnement de BERT, on peut le comparer à un élève qui sait résoudre deux exercices : trouver le mot manquant (je vais manger du - ? - au cinéma ; mot à trouver : popcorn) et savoir reconnaître si deux phrases sont reliés (je suis allé au cinéma ; j'ai vu le film Transformers -> oui ces deux phrases sont reliés)

Bert est comme un super traducteur qui convertit les phrases en vecteurs mathématiques. Ces vecteurs capturent la signification des mots ainsi que le contexte dans lequel les mots apparaissent. (Chien, chiots et animal seront proches dans l'espace mathématique ; chien et avion seront éloignés puisqu'ils n'ont pas grand-chose à voir entre eux)

Pourquoi BERT est si révolutionnaire ? Puisqu'il arrive à analyser le contexte de manière bidirectionnelle et est assez polyvalent pour être utilisé dans de nombreuses tâches : classification de texte, réponse à des questions, traduction, etc...

DistilBERT, introduit par Sanh et al.<sup>xxvi</sup> (2019), est une version optimisée de BERT obtenue via la distillation de connaissances. Ce modèle réduit le nombre de paramètres de moitié tout en maintenant des performances similaires à celles de BERT. Il est particulièrement utile dans les environnements où les ressources computationnelles sont limitées.

Pour donner un autre exemple avec des images : c'est un peu comme si BERT était un vieux professeur, qui a passé et passe beaucoup de temps à étudier un sujet pour donner une réponse parfaite (réponse complète mais lente) alors que DistilBERT est un tout nouveau professeur, entraîné par BERT, qui a appris l'essentiel de ses connaissances et qui peut donner une réponse beaucoup plus rapide mais en manquant quelques subtilités.

GPT (pour *Generative Pre-trained Transformer*), proposé par OpenAI, adopte une approche autorégressive pour se spécialiser dans la génération de texte. GPT-2 et GPT-3 démontrent des capacités impressionnantes dans la rédaction et le dialogue interactif, bien qu'ils ne soient pas bidirectionnels, ce qui limite leur compréhension contextuelle.<sup>xxvii</sup>

GPT est un modèle génératif, c'est-à-dire qu'il génère du texte, et il essaye de le faire au maximum comme un humain le ferait. Il sait générer, compléter ou reformuler des phrases en se donnant sur un contexte fourni (la plupart du temps en lui fournissant une requête). On pourrait le comparer à un auteur intelligent : on lui donne un thème pour une histoire, avec ou sans quelques consignes sur le style ou l'intention, puis l'auteur nous développe une histoire avec ça. Le gros avantage de GPT est qu'il est pré-entraîné sur une quantité astronomique de textes : livres, articles, etc... Un peu comme un étudiant qui a consulté énormément de livres et qui sait maintenant écrire des dissertations, répondre à des questions et même imiter un style d'écriture en particulier. GPT a quand même quelques différences avec BERT, contrairement à BERT, GPT ne regarde que les mots précédents pour prédire le mot suivant, il écrit ses phrases de manière séquentielle, sans retour en arrière. Néanmoins, il peut aussi faire preuve de plus de créativité que BERT et est comparable à un écrivain créatif qui est capable d'improviser en temps réel.

Le finetuning de modèles comme BERT, T5 ou FlanT5 consiste à adapter un modèle pré-entraîné pour une tâche spécifique, comme l'analyse de sentiment. T5 et FlanT5 adoptent une approche multitâche en reformulant chaque tâche comme un problème de traduction textuelle.<sup>xxviii</sup>

En conclusion, on peut dire que les Transformers ont surpassé les modèles précédents (RNN, LSTM) grâce à leur :

- **Bidirectionnalité complète** : BERT permet une compréhension simultanée des contextes gauche et droit.
- **Gestion du contexte global** : La self-attention capture efficacement les relations globales dans les séquences.
- **Scalabilité** : Leur architecture parallèle permet de traiter de larges corpus.<sup>xxix</sup>

### Analyse Comparative des Techniques de Traitement du Langage Naturel : Étude de NLTK, spaCy, BERT et DistilBERT sur des Jeux de Données de Requêtes Clients

[https://arc.cct.ie/cgi/viewcontent.cgi?article=1000&context=msc\\_da](https://arc.cct.ie/cgi/viewcontent.cgi?article=1000&context=msc_da)

Cette étude approfondie évalue et compare les performances de quatre outils et modèles de traitement du langage naturel (NLP) : NLTK, spaCy, BERT et DistilBERT, en se concentrant sur l'analyse des sentiments dans des requêtes et retours clients. L'objectif principal est d'évaluer l'efficacité et la précision de ces différentes approches pour comprendre et catégoriser les sentiments exprimés par les clients.



Les résultats de l'étude révèlent que BERT et DistilBERT présentent des similitudes dans leurs performances, surpassant les approches plus traditionnelles comme NLTK et spaCy. Notamment, BERT et DistilBERT ont tendance à classer les requêtes comme étant principalement neutres, ce qui suggère une capacité accrue à gérer une variété de sentiments exprimés par les clients. Cette tendance pourrait indiquer une meilleure compréhension contextuelle des nuances du langage naturel par ces modèles basés sur l'attention.

arc.cct.ie

L'étude met également en lumière les avantages de DistilBERT en termes d'efficacité. En tant que version allégée de BERT, DistilBERT conserve une grande partie des capacités de compréhension du langage tout en nécessitant moins de ressources computationnelles. Cette caractéristique le rend particulièrement adapté aux environnements où les ressources sont limitées, permettant une intégration plus facile dans des applications en temps réel ou sur des dispositifs avec une puissance de calcul restreinte.

fr.wikipedia.org

En conclusion, cette étude démontre que les modèles basés sur l'attention, tels que BERT et DistilBERT, représentent une avancée significative par rapport aux technologies NLP traditionnelles. Ils offrent une meilleure précision et une compréhension plus fine des sentiments exprimés dans les requêtes clients, tout en étant plus efficaces en termes de ressources, surtout dans le cas de DistilBERT. Cette révolution dans le traitement du langage naturel ouvre la voie à des applications plus performantes et accessibles dans divers domaines nécessitant une analyse précise du langage humain.

## **L'Analyse des Emotions et la Classification des Sentiments avec BERT et DistilBERT**

### **Décodage des Émotions : Analyse des Sentiments avec DistilBERT**

<https://medium.com/@adityajethani/decoding-emotions-sentiment-analysis-with-distilbert-f7096da29274>

Ce projet explore l'adaptation de **DistilBERT**, une version allégée de BERT, pour la classification des émotions. L'objectif principal est de démontrer comment fine-tuner DistilBERT pour catégoriser des textes en différentes émotions, en mettant l'accent sur les étapes de prétraitement, la tokenisation et l'entraînement du modèle sur un jeu de données annoté. Le projet souligne la capacité de DistilBERT à maintenir des performances élevées

tout en réduisant les ressources computationnelles requises, rendant ainsi l'analyse des sentiments plus accessible et efficace. Les résultats obtenus démontrent que DistilBERT peut rivaliser avec des modèles plus lourds en termes de précision, tout en offrant une efficacité accrue en production.

### **Analyse des Sentiments avec BERT à l'aide de Hugging Face**

<https://mostefasiamdi.medium.com/analyse-des-sentiments-avec-bert-a-laide-de-hugging-face-fde3c0993d26>

Ce projet détaille le processus de fine-tuning de **BERT** pour l'analyse des sentiments en utilisant les outils fournis par **Hugging Face**. Il couvre les étapes essentielles telles que le prétraitement du texte, la tokenisation spécifique à BERT et l'entraînement du modèle sur un jeu de données annoté pour la classification des sentiments. L'auteur met en évidence les défis rencontrés lors du fine-tuning et propose des solutions pour optimiser les performances du modèle, offrant ainsi une ressource précieuse pour ceux qui souhaitent appliquer BERT à des tâches similaires. Les résultats obtenus montrent une amélioration significative par rapport aux approches traditionnelles, démontrant l'efficacité de BERT dans la capture des nuances du langage humain.

### **Étude Comparative de NLTK, spaCy, BERT et DistilBERT sur l'Analyse des Sentiments**

[https://arc.cct.ie/cgi/viewcontent.cgi?article=1000&context=msc\\_da](https://arc.cct.ie/cgi/viewcontent.cgi?article=1000&context=msc_da)

Cette étude compare les performances de différents outils et modèles de traitement du langage naturel, notamment **NLTK**, **spaCy**, **BERT** et **DistilBERT**, dans le contexte de l'analyse des sentiments. En utilisant un ensemble de requêtes clients, l'étude évalue l'efficacité de chaque outil en termes de précision, de rappel et de score F1. Les résultats montrent que BERT et DistilBERT surpassent les approches traditionnelles comme NLTK et spaCy, démontrant ainsi l'efficacité des modèles Transformers pour la classification des sentiments. Cette comparaison approfondie fournit des indications précieuses pour le choix des outils adaptés en fonction des besoins spécifiques et des ressources disponibles.

Comparé aux différents projets ici listés, notre projet introduit un fine-tuning unique, de la Data Augmentation sur la base de données Dair-ai/emotion, ainsi qu'un module d'explicabilité avec LIME.

## Fine Tuning de Bert

Le fine-tuning consiste à ajuster les poids d'un modèle pré-entraîné comme BERT pour une tâche spécifique, telle que la classification des émotions ou l'analyse de sentiment. Ce processus permet de tirer parti des représentations contextuelles riches de BERT tout en adaptant le modèle à un domaine ou à un dataset particulier<sup>xxx</sup> (Howard & Ruder, 2018).

Le fine-tuning peut exactement être représenté par la personnalisation : on commence avec un musicien compétent, mais si on veut qu'il devienne un musicien compétent expert en Jazz, on le fait s'entraîner sur des musiques et des instruments de Jazz. Il se personnalise et se spécialise alors encore plus pour un domaine tout en restant « compétent » dans les autres.

Le fine-tuning sert à ajuster les connaissances pour se spécialiser sur une tâche spécifique. Pour un autre exemple avec notre contexte d'analyses de Tweet (de la base de données dair-ai/emotion) : le modèle de base ne connaît pas forcément les nuances spécifiques du langage associés aux tweets, et on va faire en sorte qu'il devienne expert dans ce domaine.

L'un des défis majeurs de l'utilisation de BERT est son manque d'explicabilité en tant que "boîte noire". Pour pallier cette limitation, des méthodes d'interprétation sont souvent appliquées pour expliquer les prédictions du modèle.

Une approche couramment utilisée pour rendre les résultats de classification plus interprétables consiste à mettre en évidence les mots clés ayant contribué à la prédiction, avec des couleurs correspondant à leur influence sur la classification :

- **Mots clés positifs (vert)** : Mots ayant une forte corrélation avec des résultats positifs.
- **Mots clés négatifs (rouge)** : Mots qui orientent la classification vers un résultat négatif.
- **Mots neutres ou ambigus (jaune)** : Mots exerçant une faible influence ou un rôle contextuel.

Par exemple, pour une phrase analysée par BERT, l'utilisation de bibliothèques comme *LIME* (Local Interpretable Model-agnostic Explanations) ou *SHAP* (SHapley Additive exPlanations) permet de visualiser l'impact de chaque mot sur la décision du modèle<sup>xxxi xxxii</sup> (Ribeiro et al., 2016 ; Lundberg & Lee, 2017).

**Texte analysé** : "Ce produit est fantastique, mais la livraison était très lente."

- "fantastique" (vert) : Influence positive forte.
- "très lente" (rouge) : Influence négative forte.
- "livraison" (jaune) : Contexte neutre.

Lors du fine-tuning, BERT peut produire des prédictions avec des scores de probabilité pour chaque classe (positif, négatif, neutre). Ces scores permettent d'interpréter les résultats obtenus :

- **Résultats positifs** : Les phrases contenant des mots fortement positifs, tels que "excellent", "incroyable", ou des superlatifs, sont classées comme positives.
- **Résultats négatifs** : Des termes négatifs, comme "horrible", "mauvais", ou "inadmissible", orientent le modèle vers une classification négative.
- **Résultats neutres** : Les phrases factuelles ou ambiguës, comme "le produit est disponible en trois tailles", ont tendance à être classées comme neutres.

L'analyse contextuelle, telle que l'utilisation de matrices d'attention, permet de visualiser quels mots influencent le plus les décisions de BERT. Des outils comme *Transformers Interpret* de Hugging Face fournissent des visualisations utiles pour identifier ces schémas.<sup>xxxiii</sup>

De très nombreux paramètres peuvent être ajuster selon les cas comme :

- Le Taux d'apprentissage (Learning Rate) qui modifie la vitesse à laquelle le modèle met à jour ses poids pendant l'entraînement, concrètement si celui-ci est trop élevé le modèle risque d'oublier ce qu'il a appris (catastrophic forgetting) mais si le taux est trop bas : l'apprentissage sera beaucoup trop lent.
- La taille du Batch Size qui est le nombre d'exemples utilisé en même temps pour mettre à jour les poids : augmenter la taille demandera plus de mémoire et la diminuer pourra rendre l'apprentissage instable
- Le nombre d'époques : le nombre de fois que le modèle passe sur l'ensemble de données : si on réduit le nombre le modèle risque de ne pas apprendre assez, mais si on l'augmente trop, le modèle aura un risque de surapprentissage.

BERT comparé à DistilBERT est donc plus précis mais plus lourd à entraîner. DistilBERT est plus rapide et léger, moins précis, mais peut donc être dans certains cas beaucoup plus adapté : exemple : quand on veut le faire tourner sur des appareils avec moins de puissance ou voir même si on veut le faire tourner en temps réel. Le fine-tuning est applicable dans ces deux méthodes de la même façon.

Le fine-tuning de BERT illustre parfaitement la puissance de l'adaptation d'un modèle pré-entraîné à des tâches spécifiques. En exploitant ses riches représentations contextuelles, ce processus permet de spécialiser BERT pour des domaines tels que l'analyse de sentiments ou la classification d'émotions, tout en garantissant des performances élevées. Cependant, cette démarche nécessite une attention particulière aux hyperparamètres comme le taux d'apprentissage, la taille du batch, et le nombre d'époques pour éviter les pièges de l'oubli catastrophique ou du surapprentissage.

Enfin, si BERT excelle en précision, son coût en ressources peut s'avérer prohibitif dans certains contextes. DistilBERT offre une alternative légère et rapide, bien que légèrement moins précise. Ces deux approches, bien que différentes, partagent l'essence du fine-tuning et montrent la flexibilité des modèles Transformers pour répondre à une variété de besoins et de contraintes. Cela confirme le rôle central du fine-tuning dans l'adaptation des modèles de langage à des applications modernes, tout en posant des défis techniques et éthiques à relever.

## **Métriques d'évaluation classification de texte**

Les métriques standard, telles que la précision (*precision*), le rappel (*recall*), le F1-score, et la matrice de confusion, sont essentielles pour évaluer la performance des modèles de classification.

- **Précision** : La précision mesure la proportion de prédictions positives correctes par rapport à l'ensemble des prédictions positives. Elle est particulièrement utile lorsque les coûts associés aux fausses prédictions positives sont élevés<sup>xxxiv</sup> (Powers, 2011).  
**C'est en résumé la réponse à la question** : « Parmi tout ce que j'ai dit être vrai, combien l'étaient vraiment ? ». Si on a prédit 10 emails comme spam et que 7 sont vraiment des spams, la précision est de 7/10.
- **Rappel** : Le rappel quantifie la proportion de cas positifs correctement identifiés parmi tous les cas positifs. Cette métrique est essentielle dans des contextes où il est crucial de minimiser les faux négatifs, comme dans les systèmes médicaux<sup>xxxv</sup> (Sokolova et Lapalme, 2009). La réponse à la question : « Parmi tout ce qui est vraiment vrai, combien en ai-je trouvé ? ». Si dans un dossier de 15 mails spams, le modèle n'en a trouvé que 10 (considérés comme spam) alors le recall n'est que de 10/15, 2/3.

- **F1-score** : Le F1-score est la moyenne harmonique de la précision et du rappel, offrant un équilibre entre ces deux métriques. Il est souvent préféré dans des situations où il existe un déséquilibre de classes dans les données<sup>xxxvi</sup> (Chicco et Jurman, 2020).

**Un peu comme une balance que l'on veut équilibrer** : il faut essayer de mettre la même chose des deux côtés.

- **Matrice de Confusion** : Cet outil présente une vue d'ensemble des performances du modèle en classifiant les prédictions en vraies positives, vraies négatives, fausses positives et fausses négatives, offrant une granularité utile pour interpréter les résultats<sup>xxxvii</sup> (Stehman, 1997).

**Un peu comme un tableau qui représenterait les victoires ou erreurs du modèle.**

Dans l'analyse de sentiment, certaines métriques spécifiques sont utilisées pour capturer les nuances émotionnelles :

- **Accuracy** : Bien que couramment utilisée, l'Accuracy peut être trompeuse dans le cas de classes déséquilibrées.
- **Score d'AUC-ROC** : Cette métrique mesure la capacité du modèle à discriminer entre les classes positives et négatives<sup>xxxviii</sup> (Bradley, 1997).
- **Métriques subjectives** : Certains travaux explorent des scores manuels ou des évaluations spécifiques à l'émotion pour capter les subtilités des sentiments exprimés dans les textes<sup>xxxix</sup> (Cambria et al., 2017).

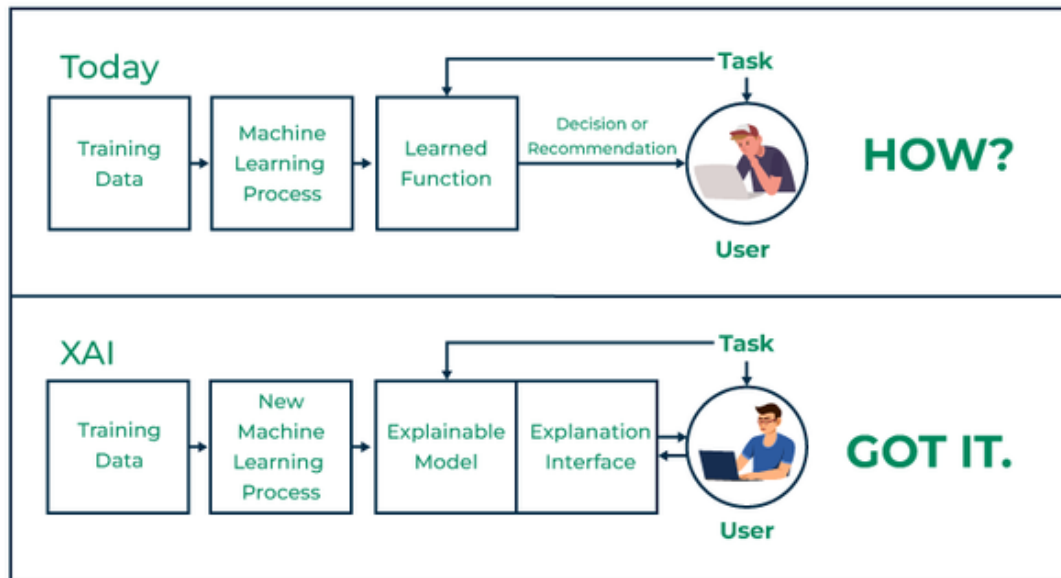
Les métriques actuelles, bien qu'efficaces, ne capturent pas toujours les subtilités du langage naturel, notamment :

- **Sarcasme et Ironie** : Les métriques standard échouent souvent à détecter les formes complexes d'expression comme le sarcasme, où le sens implicite diffère de l'interprétation directe<sup>xl</sup> (Joshi et al., 2017).
- **Tonalité et Multi dimensionnalité** : Les émotions complexes, telles que des sentiments mitigés ou une tonalité ambiguë, sont mal modélisées par des approches binaires ou catégoriques.

Ces limitations soulignent le besoin de nouvelles métriques capables de capturer les nuances émotionnelles dans les textes.

Les métriques d'évaluation jouent un rôle central dans la comparaison des modèles, mais leur efficacité varie selon les contextes. Dans l'analyse de sentiment, où les subtilités linguistiques abondent, une amélioration des métriques actuelles est indispensable pour garantir une évaluation juste et robuste des modèles.

## Explicabilité/XAI



L'explicabilité, c'est la capacité d'un modèle d'IA à expliquer ses décisions de manière que ce soit compréhensible pour un humain.

Le problème vient du fait que beaucoup de modèles (comme les réseaux de neurones ou les Transformers comme BERT et DistilBERT) sont considérés comme des boîtes noires, c'est-à-dire qu'on sait qu'ils fonctionnent bien mais on ne sait pas toujours pourquoi ils prennent certaines décisions.

Exemple : un modèle a classé un mail comme « spam », et on voudrait savoir quelles caractéristiques du message l'ont poussé à faire ce choix.

XAI est une branche de l'IA qui cherche à rendre les décisions des modèles plus transparentes et interprétable. Il utilise différentes techniques pour cela : la méthode dite « globale » : pour reprendre notre exemple de tout à l'heure : l'email est en général classé comme spam s'il contient les mots « gagner », « gratuit » ou « argent ».

XAI peut aussi utiliser des méthodes locales pour expliquer vraiment spécifiquement pourquoi sur Cette décision, le modèle a choisi de classer cet email comme spam : parce que l'email contenait de nombreuses fois les mots « gratuit » et « argent » en plus du fait qu'il ait été envoyé d'une adresse inconnue. Image <sup>xli</sup>



Des exemples d'outils populaires de XAI sont LIME et SHAP :

**LIME (Local Interpretable Model-agnostic Explanations) :**

- Explique les décisions d'un modèle en simplifiant son fonctionnement pour une prédiction donnée.
- Analogie : C'est comme demander à un professeur d'expliquer une question spécifique, pas tout le cours.
- 

**SHAP (SHapley Additive exPlanations) :**

- Donne un score à chaque caractéristique pour montrer son impact sur une décision.
- Analogie : C'est comme mesurer combien chaque joueur a contribué à la victoire dans un match.

L'explicabilité est importante dans un contexte où l'on a besoin de comprendre (comment marche le modèle) avant de (lui) faire confiance.

D'un point de vue éthique aussi, il est important de justifier les décisions pour identifier les biais qu'un modèle pourrait avoir. Et enfin, tout simplement, il est utile de savoir où le modèle se trompe pour savoir comment l'améliorer.

**Comprendre les Modèles d'Apprentissage Automatique avec LIME**

<https://ishwaryasriraman.medium.com/from-opaque-to-transparent-understanding-machine-learning-models-with-lime-3f2a2d147642>

Ce projet présente une introduction détaillée à **LIME** (Local Interpretable Model-agnostic Explanations), une bibliothèque Python conçue pour expliquer les prédictions de tout modèle d'apprentissage automatique. L'article aborde les défis liés à l'interprétabilité des modèles complexes, souvent qualifiés de "boîtes noires", et explique comment LIME peut aider à rendre ces modèles plus transparents. En générant des explications locales pour des prédictions individuelles, LIME permet aux utilisateurs de comprendre les facteurs influençant les décisions du modèle. Bien que cet article ne se concentre pas spécifiquement sur le jeu de données **dair-ai/emotion**, les concepts et techniques présentés peuvent être appliqués à l'analyse des émotions en utilisant ce dataset.

Bien que l'article mentionné n'utilise pas directement le jeu de données **dair-ai/emotion**, il fournit une base solide sur l'utilisation de LIME pour interpréter les modèles d'apprentissage automatique.

## **Datasets populaires**

Les Datasets jouent un rôle central dans le développement et l'évaluation des modèles de traitement du langage naturel (NLP). Ils fournissent les données nécessaires pour entraîner, valider et tester les modèles, tout en influençant leur performance et leur généralisation.

Un dataset est simplement une collection organisée de données utilisée pour entraîner et tester des modèles d'IA. C'est un peu comme une grande bibliothèque où chaque livre représente un exemple de donnée. Chaque livre est tout de même bien structuré de la même manière pour que le modèle puisse apprendre facilement. La structure d'un dataset est organisée sous forme de tableau, avec des lignes et des colonnes. Chaque ligne est une donnée individuelle et chaque colonne communique une information spécifique sur les données (caractéristiques ou attributs). On peut aussi noter que les Datasets peuvent contenir du texte, des chiffres mais aussi pour certains cas des images et du contenu audio.

Parmi les nombreux Datasets disponibles, certains se distinguent par leur popularité et leur adoption dans des recherches académiques et industrielles.

Dataset	Taille	Structure	Tâches Principales
Dair-ai/emotion	16 000 d'exemples	Textes courts, multi-labels	Classification d'émotions
IMDB Reviews	50 000 d'exemples	Phrases annotées	Analyse de sentiment
Twitter Sentiment140	1,6 million d'exemples	Tweets annotés	Analyse de sentiment
Yelp Reviews	Plus de 5 millions	Textes longs, multi-labels	Classification de sentiment, recommandation

Le dataset dair-ai/emotion est disponible sur la plateforme Hugging Face. Cette plateforme est dédiée au traitement du langage naturel et propose des bibliothèques open-source pour accéder facilement à des modèles pré-entraînés et pour les fine-tuner dans différentes tâches. Le dataset se concentre sur la classification des émotions dans des textes courts. Ce dataset est particulièrement adapté aux tâches multi-label, permettant de classer un texte dans plusieurs catégories émotionnelles.

**Taille et Structure** : Avec environ 20 000 exemples, ce dataset est compact mais bien équilibré entre les classes. Les textes sont courts, souvent constitués d'une ou deux phrases, ce qui le rend efficace pour des tâches nécessitant une analyse rapide et précise.

**Adéquation aux tâches** : Idéal pour la classification des émotions, ce dataset inclut des étiquettes telles que "joie", "tristesse", "colère", "surprise", "amour", et "peur". Cette granularité permet de mieux capturer les nuances émotionnelles dans les données textuelles.

**Critiques** : Bien qu'efficace pour la classification émotionnelle, le dataset peut être limité pour des analyses contextuelles complexes, comme le sarcasme ou des émotions combinées. De plus, les textes courts réduisent le contexte disponible pour les modèles.

**Importance pour le projet** : Ce dataset est au cœur de la problématique abordée dans ce projet, offrant une opportunité de tester et d'optimiser le finetuning de modèles basés sur Transformers comme BERT. <sup>xlii</sup>

Le dataset IMDB Reviews, proposé par Maas et al. <sup>xliii</sup> (2011), est l'un des benchmarks les plus populaires pour l'analyse de sentiment. Il contient 50 000 critiques de films annotées comme positives ou négatives. Les données sont équilibrées, avec 25 000 exemples pour l'entraînement et 25 000 pour le test.

- **Taille et Structure** : Les critiques sont de longueur variable, allant de quelques phrases à des paragraphes plus longs, ce qui reflète la diversité linguistique des textes réels.
- **Adéquation aux tâches** : Ce dataset est particulièrement adapté pour évaluer des modèles de classification binaire, mais moins pertinent pour capturer des sentiments plus complexes comme la neutralité ou le sarcasme.
- **Critiques** : Bien que largement utilisé, IMDB Reviews peut entraîner des modèles à sur-apprendre des patrons spécifiques au domaine cinématographique, limitant leur généralisation à d'autres domaines.

**Twitter Sentiment140** : Ce dataset se compose de 1,6 million de tweets annotés automatiquement en positif, négatif ou neutre via des émoticônes. Bien qu'il offre un grand volume de données, l'annotation automatique peut introduire des biais <sup>xliv</sup> (Go et al., 2009).

**Yelp Reviews** : Avec plus de 5 millions de critiques, ce dataset offre un large éventail de données pour des tâches comme la recommandation ou l'analyse de sentiment. Cependant, la longueur des textes peut poser des défis pour les modèles nécessitant des entrées fixes. <sup>xlv</sup>

Les Datasets IMDB Reviews et dair-ai/emotion offrent des ressources riches et diversifiées pour le développement de modèles NLP. Cependant, leur efficacité dépend de leur adéquation à la tâche visée. Pour ce projet, dair-ai/emotion se démarque par sa spécialisation dans la classification d'émotions, offrant une base solide pour expérimenter et optimiser des modèles de classification basés sur Transformers.

Un dataset est un peu comme un livre d'exercices pour enseigner à un enfant. L'enfant s'entraîne avec des exemples (dataset d'entraînement), il fait des tests intermédiaires pour voir où il peut s'améliorer (dataset de validation) et puis pour finalement faire le dernier gros exercice/examen pour prouver qu'il a bien tout compris (dataset de test)

## **Enjeux actuels et défis**

Les défis techniques liés à l'utilisation et au déploiement des modèles de traitement du langage naturel sont nombreux, notamment :

- **Langues sous-représentées** : Les modèles actuels, comme BERT ou GPT, sont souvent optimisés pour des langues largement représentées dans les Datasets, telles que l'anglais. Les langues à ressources limitées restent peu performantes, ce qui pose des défis pour leur adoption mondiale<sup>xlvi</sup> (Conneau et al., 2020).
- **Enjeu technique** : la gourmandise en ressource. BERT est très gros et demande énormément de puissance de calcul pour être entraîné ou utilisé, pour certaines entreprises ces coûts vont donc peut-être se compter en milliers d'euros. Ce rapport de puissance peut souvent ne pas être du tout adapté dépendant de la tâche, un peu comme utiliser un bulldozer pour planter des fleurs, ce qui est un peu trop... D'où l'apparition de modèles plus petits et plus efficaces comme DistilBERT.
- **Domaines spécifiques** : Le fine-tuning intensif est requis pour adapter les modèles pré-entraînés à des domaines spécifiques (médical, juridique). Cette étape peut être coûteuse en termes de données et de calcul, particulièrement pour des corpus spécialisés<sup>xlvi</sup> (Lee et al., 2020).
- **Longueur des textes** : Les Transformers, limités par la taille de leur fenêtre d'attention (typiquement 512 tokens pour BERT), rencontrent des difficultés à traiter des documents longs ou des contextes nécessitant une mémoire étendue. Des variantes comme Longformer ou BigBird tentent de résoudre ce problème<sup>xlvi</sup> (Beltagy et al., 2020).

Les modèles NLP soulèvent plusieurs questions éthiques :

- **Biais des modèles** : Les biais liés au genre, à la race ou à l'origine ethnique sont amplifiés lorsque les modèles sont entraînés sur des Datasets non représentatifs ou biaisés<sup>xlix</sup> (Bolukbasi et al., 2016). Ces biais peuvent entraîner des discriminations ou des stéréotypes dans les applications pratiques. On pourrait comparer ce comportement à un enfant qui grandit dans un « mauvais environnement » et qui sera tenté d'imiter de mauvais comportements. La solution est encore de développer des techniques pour détecter et corriger ces biais via par exemple par des réentraînements sur des données diversifiées.)
- **Transparence et explicabilité** : Les modèles comme BERT et GPT sont souvent décrits comme des boîtes noires, ce qui rend difficile l'explication de leurs décisions. Cela limite leur adoption dans des contextes nécessitant de la transparence, comme le médical ou le juridique<sup>l</sup> (Lipton, 2016). La solution pour rendre ces modèles plus transparents sont les modules XAI.
- **Confidentialité des données** : Les modèles pré-entraînés sur des corpus massifs peuvent involontairement mémoriser des données sensibles, compromettant la vie privée des individus<sup>li</sup> (Carlini et al., 2020).

Malgré leurs performances impressionnantes, les modèles actuels échouent dans certains contextes spécifiques :

- **Sarcasme et ironie** : Les modèles peinent à détecter les expressions subtiles où le sens implicite diffère du texte littéral<sup>lii</sup> (Joshi et al., 2017).
- **Applications en temps réel** : L'optimisation pour des appareils embarqués, tels que les smartphones, reste un défi en raison des contraintes en termes de calcul et de mémoire. Des approches comme DistilBERT ou TinyBERT tentent de répondre à ce besoin<sup>liii</sup> (Sanh et al., 2019 ; Jiao et al., 2020).
- **Applications sensibles** : Dans des domaines comme le médical, les erreurs de classification peuvent entraîner des conséquences graves, rendant cruciale l'intégration de mécanismes de contrôle qualité et d'explicabilité<sup>liv</sup> (Panch et al., 2019).

Les défis techniques, éthiques et pratiques liés aux modèles NLP soulignent la nécessité de recherches continues pour améliorer leur adaptabilité, leur transparence et leur inclusivité. Ces enjeux doivent être pris en compte pour garantir une adoption responsable et efficace des technologies NLP dans des applications réelles.

## **Applications modernes**

L'analyse de sentiment est l'une des applications les plus populaires des modèles NLP modernes. Elle consiste à extraire et classer des émotions ou opinions à partir de textes non structurés, avec des utilisations variées :

- **Surveillance des tendances sur les réseaux sociaux** : Les entreprises et institutions surveillent les discussions publiques pour comprendre les opinions des consommateurs, analyser les réactions à des événements ou anticiper des crises. Par exemple, une étude menée par Pak et Paroubek<sup>lv</sup> (2010) a démontré l'efficacité des modèles NLP pour analyser les opinions exprimées sur Twitter.
- **Satisfaction client** : Les entreprises utilisent l'analyse de sentiment pour évaluer les retours clients à partir de commentaires, avis ou enquêtes. Ces informations permettent d'améliorer leurs produits et services<sup>lvi</sup> (Medhat et al., 2014).
- **Études de cas** : Un exemple marquant est celui de la société Uber, qui utilise des modèles NLP pour analyser en temps réel les commentaires des utilisateurs, identifier les problèmes récurrents, et adapter leurs offres en conséquence.

La classification de textes, facilitée par les modèles comme BERT ou GPT, est utilisée dans divers domaines :

- **Fake News** : La détection des fausses informations est devenue essentielle avec l'essor des réseaux sociaux. Des études montrent que les modèles Transformers surpassent les approches traditionnelles pour identifier des articles mensongers ou biaisés<sup>lvii</sup> (Zhou et al., 2020).

- **Discours de haine** : Les plateformes sociales utilisent des modèles NLP pour modérer les contenus inappropriés ou haineux. Cependant, la détection reste un défi en raison de la subjectivité et de la diversité linguistique des discours<sup>lviii</sup> (Schmidt & Wiegand, 2017).
- **Études de cas** : Facebook et Twitter ont intégré des systèmes de modération automatisés basés sur des modèles NLP pour détecter et supprimer les contenus haineux en temps réel.

Les modèles modernes NLP se prêtent également à d'autres usages innovants, illustrant leur impact pratique dans des domaines variés :

- **Santé mentale** : Des systèmes d'analyse de sentiment sont utilisés pour détecter les signes précoces de dépression ou de stress en analysant des journaux personnels ou des messages sur les réseaux sociaux. Une étude de Benton et al.<sup>lix</sup>(2017) a démontré l'utilité de tels outils dans le suivi de la santé mentale.
- **Surveillance économique** : Les institutions financières utilisent des modèles NLP pour analyser les actualités économiques et les rapports financiers, permettant de prédire les fluctuations du marché<sup>lx</sup> (Tetlock, 2007).
- **Étude de cas : Google** : Google utilise des modèles NLP dans ses produits comme Google Assistant pour offrir des interactions plus naturelles et comprendre les intentions des utilisateurs avec précision.

## Projets existants

Voici une liste non-exhaustive de quelques projets existants traitant déjà du domaine de notre projet. Vous les présenter servira à donner quelques exemples annexes de l'utilité des modèles dont on a parlé.

[https://www.tensorflow.org/text/tutorials/text\\_classification\\_rnn?hl=fr](https://www.tensorflow.org/text/tutorials/text_classification_rnn?hl=fr)

Le tutoriel explique comment créer un modèle basé sur un **RNN (LSTM ou GRU)** pour classer des textes en utilisant TensorFlow. Il couvre le traitement des données textuelles, la construction du modèle et l'évaluation des performances.

[https://huggingface.co/docs/transformers/tasks/sequence\\_classification](https://huggingface.co/docs/transformers/tasks/sequence_classification)

Ce guide montre comment utiliser **BERT**, via la bibliothèque Hugging Face, pour des tâches de classification de texte. Il détaille le chargement d'un modèle pré-entraîné, son fine-tuning sur des données spécifiques et son évaluation.

<https://www.kaggle.com/code/imdevskp/imdb-review-classification-lstm-gru-cnn-glove>

Ce notebook Kaggle illustre la classification d'avis de films IMDB en utilisant des modèles basés sur **LSTM**, **GRU**, et **CNN** avec des embeddings GloVe. Il inclut des étapes de prétraitement, de construction des modèles et de comparaison des performances.



## **Perspectives pour la classification de texte et analyse de sentiment**

L'évolution rapide des modèles de traitement du langage naturel (NLP) ouvre de nouvelles perspectives pour la classification de texte et l'analyse de sentiment. Les avancées technologiques et méthodologiques actuelles annoncent des transformations majeures dans la manière dont les données textuelles seront traitées et interprétées à l'avenir.

L'une des directions les plus prometteuses pour l'avenir est l'intégration d'approches multimodales. Ces modèles combinent des données textuelles et visuelles pour capturer des informations contextuelles riches.

- **Exemples d'applications :**
  - Détection de sarcasme dans des mèmes : La combinaison du texte contenu dans une image et des éléments visuels permet d'améliorer la compréhension contextuelle<sup>lxi</sup> (Kaur et al., 2020).
  - Analyse des sentiments dans des posts sociaux : L'utilisation conjointe de la légende textuelle et de l'image associée enrichit l'interprétation émotionnelle.
- **Technologies émergentes :**
  - Modèles multimodaux comme CLIP<sup>lxii</sup> (Radford et al., 2021) ou Flamingo<sup>lxiii</sup> (Alayrac et al., 2022), qui intègrent efficacement des représentations textuelles et visuelles.
  - Réseaux neuraux spécifiques, comme les *Vision-Language Transformers*, capables de fusionner les deux modalités.

Les Transformers continueront d'évoluer pour relever les défis actuels :

- **Réduction des coûts computationnels** : Le développement de variantes plus légères, telles que TinyBERT ou Efficient Transformers, rendra les modèles accessibles sur des appareils embarqués.
- **Adaptation aux langues sous-représentées** : Des initiatives comme BLOOM (BigScience) visent à élargir la couverture linguistique des modèles pré-entraînés<sup>lxiv</sup> (Scao et al., 2022).

L'analyse de sentiment et la classification de texte gagneraient en pertinence grâce à une meilleure prise en compte des facteurs culturels et sociaux :

- **Détection des biais** : Intégrer des mécanismes pour détecter et corriger les biais liés au genre, à la race ou à l'origine sociale dans les prédictions.
- **Contextualisation** : Exploiter des données historiques ou régionales pour fournir une analyse plus fine et adaptée à des contextes spécifiques.<sup>lxv</sup>

Les modèles génératifs, comme GPT-4 ou d'autres architectures émergentes, joueront un rôle croissant dans :

- **L'analyse proactive** : Prédire les émotions ou sentiments associés à des textes générés ou projetés.
- **Les interfaces utilisateur intelligentes** : Assurer une interaction plus naturelle avec les systèmes, où les analyses de sentiments influencent directement les réponses générées.

Les perspectives pour la classification de texte et l'analyse de sentiment se situent à l'intersection de l'innovation technologique et de la prise en compte des besoins humains. Les approches multimodales, les améliorations des Transformers, et l'intégration de contextes culturels marquent des étapes clés vers une compréhension plus profonde et nuancée des données textuelles.

## **12) Conclusion**

La classification de texte, et en particulier l'analyse des émotions, représente un défi majeur dans le domaine du traitement du langage naturel (NLP). À travers cet état de l'art, nous avons exploré comment développer et optimiser un modèle de classification de texte basé sur le fine-tuning de BERT, appliqué à la base de données dair-ai/emotion, tout en assurant l'explicabilité des décisions du modèle.

Dans les **méthodes classiques pour la classification de texte**, les approches telles que le bag-of-words ou TF-IDF ont fourni des bases solides pour représenter les données textuelles. Cependant, leur incapacité à capturer le contexte ou la structure syntaxique complexe a souligné la nécessité de techniques plus avancées.

Les **avancées avec le Deep Learning**, à travers les RNN, LSTM et Word2Vec, ont permis de mieux intégrer les relations contextuelles et sémantiques dans les données textuelles. Ces modèles ont ouvert la voie à des architectures plus puissantes, mais limitées par des problèmes de scalabilité et d'efficacité.

La **révolution avec les modèles basés sur l'attention**, en particulier avec les Transformers comme BERT, a marqué une avancée fondamentale. Ces modèles offrent une compréhension contextuelle riche et une capacité à traiter les dépendances globales, rendant possible le fine-tuning pour des tâches spécifiques comme la classification des émotions. Leur bidirectionnalité et leur capacité à surmonter les limites des approches précédentes en font un choix idéal pour des tâches complexes.

Les **métriques d'évaluation** jouent un rôle clé pour mesurer la performance des modèles. Les métriques classiques telles que la précision, le rappel et le F1-score, combinées à des outils plus spécifiques pour l'analyse de sentiment, permettent d'évaluer et de comparer les modèles. Ces métriques sont essentielles pour valider les résultats obtenus avec dair-ai/emotion, tout en identifiant les limites des modèles dans la capture des nuances émotionnelles.

L'importance des **datasets populaires** comme IMDB Reviews et dair-ai/emotion a été mise en évidence pour leur rôle dans l'entraînement et l'évaluation des modèles. En se concentrant sur un dataset riche et adapté comme dair-ai/emotion, il devient possible de développer des modèles capables de capturer une large gamme d'émotions humaines.

Les **enjeux actuels et défis** techniques et éthiques, tels que les biais de genre ou de race, la transparence, et l'adaptation à des contextes spécifiques, soulignent l'importance de développer des modèles responsables et inclusifs. Ces considérations sont particulièrement cruciales pour une application sensible comme la classification des émotions.

Les **applications modernes** démontrent la pertinence pratique des modèles NLP dans des domaines variés, de la surveillance des tendances sur les réseaux sociaux à la détection des discours de haine. Le fine-tuning de BERT pour dair-ai/emotion peut non seulement améliorer les performances, mais aussi ouvrir de nouvelles possibilités dans des contextes applicatifs réels.

Dans le **fine-tuning de BERT**, l'intégration de méthodes d'explicabilité, comme la mise en évidence des mots clés en couleurs, répond directement à la problématique de transparence et d'interprétation. Ces outils permettent de comprendre les décisions du modèle, renforçant ainsi la confiance des utilisateurs dans les prédictions.

Enfin, les **perspectives** indiquent que des approches multimodales, combinant texte et image, ainsi que des modèles encore plus efficaces et adaptés, transformeront encore davantage le domaine. Ces avancées ouvriront de nouvelles opportunités pour résoudre les défis actuels et répondre à des besoins émergents.

En conclusion, cet état de l'art a démontré que le développement et l'optimisation d'un modèle de classification des émotions, basé sur le fine-tuning de BERT et appliqué à la base dair-ai/emotion, reposent sur une combinaison équilibrée de technologies avancées, de choix méthodologiques éclairés, et de considérations éthiques. L'avenir de cette tâche dépendra de notre capacité à intégrer ces éléments dans des solutions robustes, explicables et adaptées aux besoins diversifiés des utilisateurs.

## **DEUXIEME PARTIE – DEVELOPPEMENT DU PROJET**

### **Introduction**

Pour donner suite à l'État de l'Art, rédigé et présenté en décembre 2024, le développement du Projet et de son Programme associé s'en est suivi jusqu'à sa présentation finale en fin mars.

Dans cette deuxième partie de rapport, nous allons dans un premier temps revenir sur quelques remarques effectuées par le jury de mi-parcours ainsi que les améliorations effectuées dans l'état de l'art ainsi que dans nos façons de voir et travailler les choses. Nous développerons les différents aspects de la conception du programme informatique, c'est à dire détailler les différentes étapes que l'on a voulu mettre en œuvre tout au long de la programmation. Nous expliciterons aussi les différents stades et évolutions que le code a subi pendant son développement, ainsi que certaines idées, ajoutées ou non au programme, via des algorithmes et des méthodes divers. Les résultats seront présentés sous forme de grille montrant différents essais et tâtonnements conduits par la recherche des meilleurs hyperparamètres (fine-tuning), résultats que l'on analysera. Pour refermer le rapport, une conclusion claire sur le travail réalisé sera effectuée, ainsi qu'une comparaison finale entre le projet terminé et l'état de l'art.

## **Remarques du Jury et Améliorations Effectuées**

Lors de la présentation de mi-parcours de l'état de l'art, présentation du rapport et présentation du support visuel, quelques remarques nous ont été faites.

Les plus importantes d'entre elles et celles qui ont fait l'objet des plus importantes modifications dans les rapports ont été celles-ci : trop d'énumérations, besoin de plus cibler l'état de l'art, besoin de sources précises pour les chiffres donnés.

L'énumération étant due à la génération de certaines parties du rapport grâce à l'intelligence artificielle, a pu très vite être corrigée.

L'état de l'art présenté en décembre a présenté trop en général les différentes technologies qui ont conduit à BERT (langage utilisé pour le projet). La remarque visait donc à faire comprendre que le ciblage sur uniquement les points intéressants concernant le projet était nécessaires. Concrètement, en plus de la présentation générale des anciennes technologies qui ont servies de tremplin à BERT, une plus grosse partie, plus détaillée sur BERT était nécessaire et a donc été traitée en améliorant le passage sur BERT.

Le thème principal du projet « analyse des émotions dans un texte donné » a lui aussi été sous-traité lors du développement de l'état de l'art. Une importante partie lui a donc été dédiée lors de la mise à jour du rapport, de même qu'une nouvelle partie entièrement dédiée à l'attention.

## **Aspects de la Conception**

Nous allons dans cette partie développer les choix méthodologiques et techniques que nous avons pris lors du développement du projet.

Dans un premier temps, beaucoup d'éléments nous ont été affecté d'office : le modèle BERT, la base de données sur laquelle entraîner le modèle dair-ai/emotion et le type de donnée à traiter : phrases de style « tweet ».

Les autres aspects ont donc été à notre convenance. Le choix d'utiliser le langage python pour développer le code a été fait pour la simplicité que cela apportait. Grace à l'école nous avons accès au logiciel Pycharm, spécialisé donc pour Python, et que nous aimions particulièrement utiliser. Aussi, le langage python étant un des, si ce n'est le plus documenté pour ce que nous faisons, ça été la majeure raison de pourquoi nous avons envie de s'orienter sur ce langage-là.

En termes de méthodologie, vu que nous étions 2 dans le groupe, le travail a été coupé en deux parties égales. Nous parlerons plus du développement du code dans la partie suivante, mais à savoir que : que ce soit lors de la rédaction des rapports écrits, de présentation ou bien du développement du code, chacun de nous avait un nombre égal de partie à développer.

Pour conclure cette partie sur les aspects de la conception, nous dirons que les choix techniques commis d'office ont permis une bonne mise à l'étrier pour le début du développement du code, mais dont nous avons donc vu très vite les limites. Le dataset dair-ai/emotion ne contenant "que" 16 000 exemples, nous avons pu constater très vite qu'il y avait des limitations du a ce nombre d'exemple. Certaines émotions étaient beaucoup plus présentes dans le dataset que d'autres, ce qui nous a force à utiliser d'autres méthodes pour palier à ce problème (data augmentation). Aussi, et en toute logique d'un projet utilisant un modèle qui s'entraîne sur une base de données, plus on aurait eu de données, meilleurs auraient été nos résultats. Sans parler du problème de trouver une base de données avec une structure parfaitement similaire à celle déjà utilisée, cette idée est une bonne piste d'amélioration facile de nos résultats.

## **Développement Réalisé**

Cette partie servira à détailler la partie "développement" du programme et du projet.

La première partie du projet a été de s'approprier le code venant du site kaggle, fourni par le tuteur, et qui comportait une base à améliorer pour le projet. Ce code faisait donc déjà tourner DistilBert (BERT) sur le dataset dair-ai/emotion, notre tâche a donc été dans un premier temps de s'approprier toutes les méthodes déjà contenues dans ce code. Le fine-tuning a donc été testé assez rapidement puisque nous avons vite joué avec les différents paramètres pour mieux les comprendre et voir les conséquences sur les résultats. Nous détaillerons un peu plus en détail le fine-tuning dans la prochaine partie qui présentera différents résultats que nous avons obtenus en manipulant justement les hyperparamètres.

Emotion	Nombre d'exemples
joy (1)	5362
sadness (0)	4666
anger (3)	2159
fear (4)	1937
love (2)	1304
surprise (5)	572

L'image ci-dessus nous montre la répartition des différentes émotions associées aux phrases contenues dans le dataset. Par exemple : 2159 phrases sont associées à l'emotion "colère". On peut très vite constater que le répartissement des classes est inégal : seulement 572 sont associées à la surprise contre 5362 pour la joie (soit + de 9 fois plus).

On rappelle que pour un entraînement optimal d'un modèle (et ce quel que soit le modèle), il faut qu'il ait un nombre égal de données d'entraînement dans chacune des catégories dans lesquels il s'entraîne. Pour donner une image : une personne allant à la salle de sport va en théorie essayer de travailler toutes les parties du corps de manière relativement égale et ne va pas seulement travailler une seule partie du corps, ce qui pourrait occasionner un déséquilibre important à de nombreux niveaux.

La première solution a été de ne prendre que 572 exemples dans chacune des émotions, ce qui équilibrait parfaitement les données entre elles, et en prenant le maximum d'exemples de la classe la moins représentée. Le résultat était satisfaisant mais sans plus, car nous n'utilisions alors que 3432 exemples des 16 000 exemples que contient la base de données complète.

La deuxième solution, bien plus intéressante, et qui a nécessité de notre part des recherches supplémentaires pour la configurer au mieux : le processus de Data Augmentation.

La Data Augmentation est une technique utilisée surtout pour l'apprentissage automatique. Son but est globalement d'améliorer la robustesse et la généralisation du modèle, elle permet d'accroître la diversité des données d'entraînement sans en collecter de nouvelles. Nous l'avons ici utilisée pour palier au problème du nombre trop peu élevé de cas « surprise » dans le dataset (pour rappel 572 exemples de « surprise » seulement contre 1304 pour le deuxième moins représenté « love » et contre 5362 pour le plus représenté « Joy »).

Le processus du Data Augmentation est simple : prendre une phrase (réellement présente dans le dataset) et la dupliquer en y appliquant des transformations aléatoires mais cohérentes.

Exemple avec une phrase venant de « dair-ai/emotion » et venant de la classe « surprise » :

**Phrase originale :**

**"I can't believe he actually said that !" (*Je ne peux pas croire qu'il ait réellement dit ça !*)**

**Application de différentes transformations en Python :**

1. **Synonymisation (remplacement par des synonymes)**  
→ *"I can't believe he really stated that !"*
2. **Permutation des mots (changement d'ordre syntaxique)**  
→ *"He actually said that ? I can't believe it !"*
3. **Suppression et ajout de mots (bruit linguistique)**  
→ *"I seriously can't believe he just said that !"*
4. **Traduction automatique aller-retour (via Google Translate ou MarianMT en Hugging Face)**  
→ *"I can't believe he truly mentioned that !"*
5. **Paraphrase avec GPT ou T5 finetuné sur la paraphrase**  
→ *"I'm shocked that he really said that !"*



Grâce à ces manipulations, un même modèle peut apprendre à reconnaître des variations des données qu'il n'aurait peut-être pas rencontrées lors de son entraînement initial, ce qui améliore considérablement sa performance et sa capacité à généraliser sur des données inédites.

Nous avons donc réussi à générer de nouvelles données permettant ainsi de palier un peu au problème de la sous-représentation de certaines classes du dataset. Nous discuterons des chiffres un peu plus en détail dans la partie « résultats » du rapport, mais à savoir que nous avons donc utilisé la data augmentation sur la classe « surprise » et « love ».

Autre chose importante à noter concernant la Data Augmentation : Certaines phrases ne peuvent pas être dupliquées, alors nous ne pouvons pas vraiment généraliser les faits en disant « pour la classe surprise, nous avons généré 3 phrases pour chaque phrase de base comprise réellement dans le dataset », mais à comprendre cette phrase donc comme plutôt une indication, un objectif optimal quasi atteint.

Un exemple de phrase impossible à dupliquer serait :

"Je suis choqué que le président ait annoncé sa démission ce matin."

Les termes employés étant trop spécifiques et la phrase étant trop précise, il est impossible d'utiliser des synonymes ou bien de supprimer ou déplacer certains mots pour générer de nouvelles phrases.

La dernière partie du développement du programme a consisté à la programmation de l'affichage visuel. Notre Tuteur nous ayant conseillé d'utiliser Streamlit, c'est ce que nous avons fait et c'était des plus arrangeants car, associé au langage Python, Streamlit est un affichage sous forme de fenêtre web très bien documenté.

L'affichage final résulte donc en la comparaison du modèle initial sans notre fine-tuning à celui dont nous avons optimisé les hyperparamètres et fourni la Data Augmentation. Un espace de zone de texte a aussi été mise en place pour pouvoir analyser une phrase que l'on rédigerait nous-mêmes, pour pouvoir tester en temps réel le modèle, qui devra y repérer les émotions dominantes grâce à son précédent entraînement sur la base de données dair-ai/emotion.

## Résultats Obtenus

Nous présenterons dans cette section quelques résultats que nous avons obtenus lors de nos différents essais pour obtenir de meilleurs résultats.

Voici les hyperparamètres que l'on a utilisé pour tous les essais :

Le but ici est surtout de comparer les résultats obtenus au nombre de données utilisées, qu'elles soient issues uniquement du dataset ou bien du dataset + du procédé de Data Augmentation. A chaque tableau, une analyse sera attribuée.

<u>Essai</u>	<u>Taille du dataset</u>	<u>Batch size</u>	<u>Epochs</u>	<u>Accuracy</u>	<u>Temps d'apprentissage</u>
<u>Essai 3</u>	<u>3432 (572 exemples/classe)</u>	<u>16</u>	<u>3</u>	<u>0.75</u>	<u>~10 minutes</u>
<b>Essai 5</b>	<b>16000 (complet)</b>	16	3	<b>0.92</b>	~2 heures 30 minutes

Voici quelques données indicatives du « point de départ », c'est-à-dire que dans ce cas, on a pris uniquement 3432 exemples au total, soit 572 exemples de chaque classe, chiffre correspondant à la valeur maximale du nombre d'exemple de la classe la moins représentée. C'est le cas où on aura le moins de données pour l'entraînement et où donc, les résultats seront les moins bons.

On peut donc y voir une Accuracy de 0.75 qui est à peine acceptable avec un temps d'apprentissage de 10 minutes.

L'essai 5 a utilisé toutes les 16 000 données du dataset originel, causant ainsi une représentation très déséquilibrée des données pendant l'entraînement (cf. : le nombre de données pour chaque classe). Le temps d'apprentissage s'étant proportionnellement augmenté, l'Accuracy a aussi augmenté pour atteindre un chiffre excellent de 0.92.

Les prochains tableaux seront des essais sous Data Augmentation.

Essai	Taille du dataset	Batch size	Epochs	Accuracy	Temps d'apprentissage	Objectifs du Data Augmentation	Nombre d'échantillon pour chaque Label/Classe
<b>Essai 6</b>	<b>16000 (complet)</b>	16	3	<b>0.93</b>	2 heures 13 minutes	Classe « Love » : 1 phrase générée pour une vraie phrase  Classe « Surprise » : 3 phrases générées pour une vraie phrase.	Tristesse : 4666  Joie : 5362  Amour : 1916  Colère : 2159  Peur : 2113  Surprise : 1773

L'essai ci-dessus a tourné sur 3 Epochs et a duré 2h et 13 minutes, il introduit la Data Augmentation dans notre programme. Pour rappel les Objectifs de la Data Augmentation ne sont que des indications car certaines phrases peuvent ne pas être dupliquées comme expliqué précédemment. Les Objectifs lors de cet essai étaient : Générer une phrase pour chaque phrase de la catégorie « Love » et Générer 3 phrases pour chaque phrase de la catégorie « Surprise ». Autrement dit : doubler le nombre de données appartenant à la catégorie « Love » et Quadrupler les données de la catégorie « Surprise ». L'objectif étant pour rappel d'avoir un nombre de données le plus égal entre les différentes catégories et au minimum de minimiser les écarts entre ces nombres de données.

On passe donc de 572 à 1773 phrases pour la catégorie « surprise » et de 1304 à 2113 phrases pour la catégorie « Love ».

L'écart avec le nombre de données que comportent les autres catégories s'est donc considérablement réduit (dans l'ordre : 4666, 5362, 1916, 2159, pour les catégories tristesse, joie, colère, peur). Nous sommes aussi passé d'un échantillon original de 16 000 données à 17 989.

L'objectif de doubler les données appartenant à la catégorie « Love » est quasi atteint tout comme l'objectif de quadrupler les données de la catégorie « Surprise ».

Nous aurions pu travailler davantage sur la Data Augmentation pour avoir encore de meilleurs résultats et pousser la génération de nouvelles données, mais nous avons décidé

d'en rester là, notre sujet principal étant la maîtrise de BERT sur la classification de sentiments.

	Precision	recall	f1-score
Sadness	0.96	0.97	<b>0.96</b>
Joy	0.96	0.93	<b>0.95</b>
Love	0.77	0.89	0.82
Anger	0.94	0.91	<b>0.92</b>
Fear	0.93	0.85	<b>0.89</b>
Surprise	0.68	0.91	<b>0.78</b>
Accuracy			<b>0.93</b>
Macro avg	0.87	0.91	0.89
Weighted avg	0.93	0.93	0.93

Ce tableau représente plus en détail différentes autres métriques d'évaluation. Le score d'Accuracy et le score des autres métriques sont excellents, mais quelques détails quand même à noter.

Sadness et Joy présentent des scores particulièrement excellents, probablement dus à leur représentation très élevée dans les données d'entraînement. Anger et Fear ont de moins bons résultats, restant quand même excellents. Pour confirmer la théorie que le nombre de données impacte directement les résultats, les deux « moins bons » résultats appartiennent à Love et Surprise, qui bien qu'aidé par la Data Augmentation n'obtiennent pas d'aussi bons scores que les autres catégories.

Pour une analyse plus en profondeur :

**Love** a un F1-score de **0.82**, avec une précision plus faible (**0.77**) mais un rappel élevé (**0.89**). Cela suggère que le modèle détecte bien les vrais exemples de "love", mais fait plus d'erreurs en classant d'autres émotions comme "love".

**Surprise** est la classe **la plus difficile à classifier**, avec une précision relativement faible (**0.68**) et un rappel très élevé (**0.91**). Cela signifie que lorsqu'un texte est classé comme "surprise", il y a plus de chances d'erreur (beaucoup de faux positifs), mais le modèle capture bien les cas réels de surprise.

Ces deux remarques sur Love et Surprise font directement écho à l'utilisation de la Data Augmentation, qui, on le rappelle ne fait que générer de nouvelles données avec celles déjà existante, ce qui peut induire des phrases pas parfaites.

L'Accuracy de 0.93 de l'essai 6 fait donc à peine mieux que l'Accuracy de l'essai 5, qui avait utilisé l'ensemble déséquilibré du dataset original. La question de devoir utiliser ou non la Data Augmentation se pose donc, mais force est de constater que dans un souci

d'optimisation des ressources à disposition (« seulement » 16 000 exemples déséquilibrés dans le dataset), les résultats sont un peu meilleurs.

Point important à souligner : l'analyse des résultats jusqu'à présent peut faire penser que le sujet du projet s'est transformé en « Data Augmentation sur de L'Analyse de Sentiment avec BERT » au lieu de « Analyse de Sentiment avec BERT ».

L'accent est mis sur la Data Augmentation car c'est, au global, ce qui a le plus amélioré le résultat global du programme.

Les résultats permis par l'utilisation de BERT sur dair-ai/emotion sont déjà excellents et ne nécessitent pas énormément d'analyse complémentaire, d'où l'accent qui est mis sur la Data Augmentation, qui apporte un vrai plus aux résultats déjà bons.

L'Accuracy de + 0.90 des derniers essais indique que BERT fonctionne à merveille dans cette tâche et nous confirmerons ces chiffres avec des exemples de phrases où les émotions ont été correctement identifiées un peu plus tard dans cette même section d'analyse des résultats.

Essai	Taille du dataset	Batch size	Epochs	Accuracy	Temps d'apprentissage	Objectifs du Data Augmentation	Nombre d'échantillon pour chaque Label/Classe
Essai 12	16000 (complet)	16	10	0.92	5 heures 38 minutes	Classe « Love » : 1 phrase générée pour une vraie phrase Classe « Surprise » : 3 phrases générées pour une vraie phrase.	Label 0: 4666 Label 1: 5362 Label 2: 1916 Label 3: 2159 Label 4: 2113 Label 5: 1773

Lors de l'Essai 12, 10 Epochs ont été lancés et les autres paramètres sont restés similaires. L'Accuracy a diminué de 0.01 par rapport à l'exact même Essai mais avec 3 Epochs.

	precision	recall	f1-score
sadness	0.95	0.97	0.96
joy	0.98	0.90	0.94
love	0.74	0.96	0.84
anger	0.93	0.91	0.92
fear	0.92	0.84	0.88
surprise	0.67	0.88	0.76
accuracy			0.92
macro avg	0.86	0.91	0.88
weighted avg	0.93	0.92	0.92

Les deux essais classent de manière quasi similaire les différentes classes :

**Les classes sadness, anger et fear sont bien détectées dans les deux essais**, avec des scores F1-scores similaires et relativement élevés (0.92 à 0.96).

**Le modèle a plus de mal avec "love" et "surprise" dans les deux essais**, ce qui indique que ces émotions sont plus difficiles à traiter.

Mais des différences minimales sont quand même à noter entre ces deux essais :

**Le 12eme essai a légèrement amélioré la classification de "love" (F1-score +0.02)**, grâce à une meilleure capacité à capturer les vrais positifs (rappel +0.07).

**L'essai 12 a une meilleure précision sur "Joy" (0.98 vs 0.96), mais un rappel plus faible (0.90 vs 0.93)**, ce qui signifie qu'il est plus conservateur dans ses prédictions mais fait plus d'erreurs en ne détectant pas tous les vrais cas.

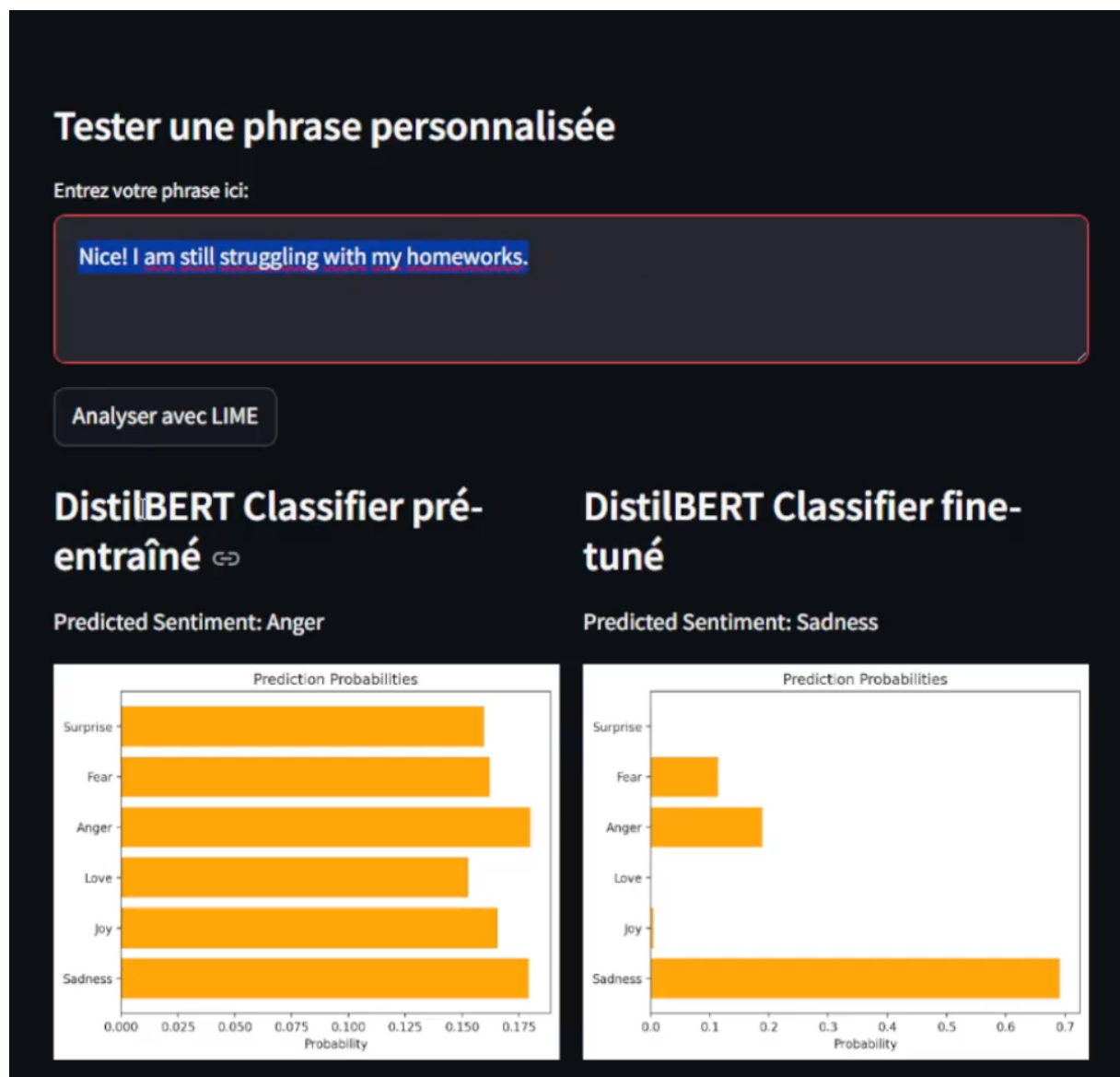
**La classe "surprise" est moins bien détectée dans le deuxième modèle (F1-score 0.76 vs 0.78), avec un rappel qui diminue légèrement (0.88 vs 0.91).**

Ce qu'on peut conclure sur la comparaison des deux derniers essais, qui ont pour seule différence le nombre d'époques et donc le temps d'apprentissage, c'est qu'il n'y a eu aucune vraie amélioration.

Cela peut indiquer plusieurs choses : le modèle a sûrement déjà convergé avec moins d'époques et certaines classes ayant d'infimes moins bons résultats pourraient faire penser au début d'un overfitting.

Un nombre réduit d'époque est donc favorisé.

## Analyse des résultats du Projet sur des phrases rédigées manuellement :



L'interface ci-dessus nous montre le résultat lorsque l'on rentre la phrase « Nice ! I am struggling with my homeworks » dans la zone de texte prévu pour une démonstration du programme en direct.

Les deux graphiques que nous comparons indiquent les probabilités que la phrase soit de d'une certaine classe (parmi Surprise, Fear, Anger, Love, Joy et Sadness, classes du dataset). La somme des probabilités indiquées est donc toujours égale à 1 et le sentiment indiqué pour la phrase est celui qui a eu la probabilité la plus élevée.

Le Classifieur utilisé pour générer le graphique de gauche utilise uniquement BERT dans sa version classifieur la plus générale.

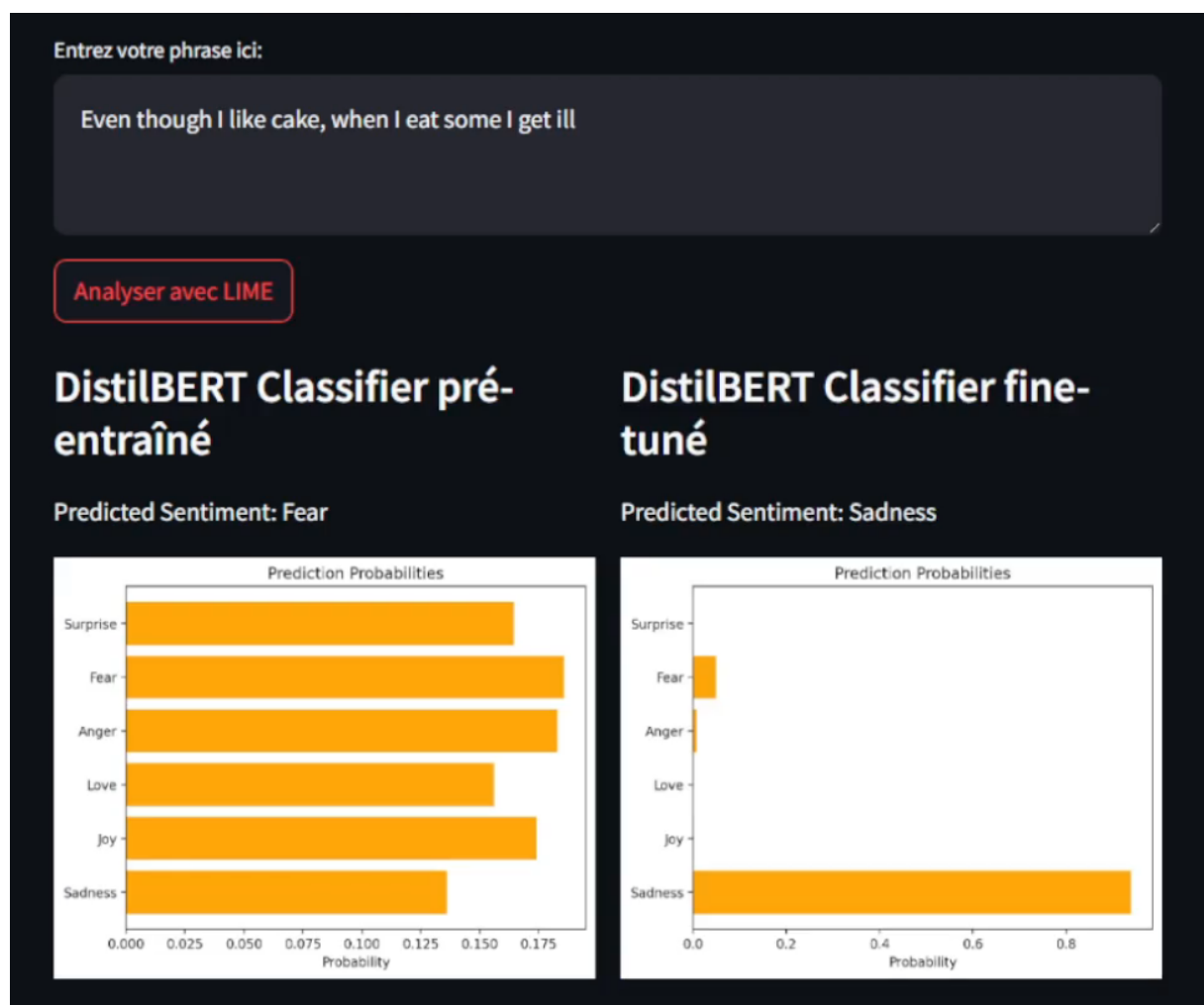
Le Classifieur utilisé pour générer le graphique de droite utilise notre propre version de BERT, spécialisé dans la détection de sentiment.

Ici la comparaison est flagrante, notre modèle fine-tuné indique le bon résultat pour la phrase et indique des valeurs cohérentes concernant les autres émotions.

Le sens de la phrase va évidemment vers la tristesse : la phrase traduite indique : « Super, je suis en train de galérer avec ces devoirs ! » et les sentiments autre que la tristesse qui se rapprochent le plus du sens de la phrase sont effectivement : la colère et la peur.

Une analyse mot par mot sera développée un peu plus tard dans cette même section, au moment de parler de l'explicabilité.

Le Classifieur de gauche n'a pas du tout réussi à trouver le sentiment dominant, donnant une probabilité quasi équivalente à toutes les émotions, ce qui ne peut en aucun cas marcher. Son sentiment annoncé final est quant à lui, aussi faux.



Le Deuxième exemple nous montre les résultats pour la phrase « Even though i like cake, when i eat some i get ill » à se traduire par : Même si j'aime bien les gâteaux, en manger me rend malade.



Les exactes mêmes remarques que l'exemple précédent peuvent être faites : Notre modèle fine-tuné est beaucoup plus performant que le modèle de base et prédit les bons sentiments. Une analyse plus détaillée de l'effet de chaque mot aura lieu dans la section sur l'explicabilité.

Pour conclure cette partie, nous sommes grandement satisfaits des résultats des analyses qui ont été faites sur les phrases que nous avons rédigées pour les exemples. Les sentiments détectés sont cohérents avec le sens de la phrase et la comparaison avec le modèle de base est sans équivoque et montre bien que les modifications que nous avons apportées ont eu un réel impact sur les résultats.

Sur toutes les phrases que nous avons testées, et phrases que nous vous encourageons à essayer, l'objectif principal du projet concernant la détermination du sentiment dominant dans la phrase, est un objectif accompli.

### Analyse des résultats sur l'explicabilité :



Voici les résultats développant l'explicabilité pour le premier exemple : « Nice ! I am struggling with my homeworks ».

L'explicabilité fonctionne ici grâce à LIME. LIME est une méthode qui explique les prédictions d'un modèle en créant des variations d'une instance et en observant quelles caractéristiques influencent le plus la décision. Il construit un modèle simple autour de cette instance pour rendre l'explication compréhensible. A noter aussi que LIME est indépendant des modèles et est une méthode qui fonctionne seule. Elle analyse donc arbitrairement chaque mot sans tenir compte de quoique ce soit d'autre. LIME est modèle-agnostique.

Les graphiques fonctionnent de cette façon : Pour chaque mot constituant la phrase, une valeur va lui être attribuée en lien avec l'émotion majoritaire repérée. Plus la valeur de celle-ci sera positive (plus la barre verte sera importante), plus le mot sera en rapport avec l'émotion attribuée.

Exemple : « Struggling » qui se traduit par « (vraiment) galérer », se rapproche énormément des sentiments tristesse/Sadness et Anger/Colère, plutôt que par exemple la joie/Joy). Plus un mot sera associé négativement à une émotion (c'est-à-dire plus la barre rouge est grande), plus le mot sera éloigné de cette émotion. Exemple : « Nice » qui se traduit habituellement par « Gentil » (dans le contexte de la phrase comme « super ! »), ne s'accorde pas du tout avec l'émotion Colère/Anger, mais se rapprocherait plutôt de l'émotion Joy/Joie.

A gauche, les résultats pour le modèle classifieur DistilBERT de base sans nos modifications. A droite, ceux de notre propre modèle fine-tuné.

Pour le modèle de gauche, l'explication est faussée dès le départ car le sentiment majoritaire identifié et pris pour l'explication n'est pas le bon (Anger au lieu de Sadness). Si on enlève ce problème, on peut tout de même constater que LIME attribue des valeurs cohérentes à chaque mot en fonction de l'émotion Anger/Colère identifiée.

A droite, les résultats de LIME sont aussi cohérents : « Struggling » est grandement corrélé à la tristesse et « Nice » ne l'est pas.

L'utilité de LIME est de nous développer les raisons du « pourquoi cette phrase a été associée à cette émotion ? ». Les graphiques générés par LIME nous montrent des indications pour chaque mot et pour une certaine émotion (dans nos graphiques : uniquement pour l'émotion qui est identifiée comme la plus correcte pour la phrase). A savoir que le programme génère des indications pour chaque mot, chaque émotion et pour chaque phrase à chaque fois et que ce que l'on voit grâce à LIME est donc ce qu'on peut considérer comme une « petite partie de l'envers du décor », une petite partie de comment marche réellement le code en interne. Un bref aperçu de comment fonctionne le cœur du programme de détection des sentiments. C'est le principe de l'explicabilité.



Voici les graphiques correspondant à l'explicabilité pour la phrase « Even though i like cake, when i eat some i get ill »

LIME définit bien le mot le plus important de la phrase « ill » /malade comme appartenant à la classe Sadness/Tristesse.

Les résultats pour les autres mots sont eux aussi cohérents avec la classe dominante repérée par les deux modèles.

A noter que l'on n'a pas particulièrement parlé des mots type : « I » / Je, « Eat » /Manger, « am » /Être, du fait de leur trop grande généralité. Ces mots ne sont pas les plus déterminants pour identifier l'émotion dominante de la phrase et ont donc été laissé de côté pour l'analyse des résultats. Ces mots peuvent être associés aux « stopword » qui, lors d'analyse de texte peuvent être ignoré car ils ne comportent pas de valeurs sémantiques importantes, c'est-à-dire qu'ils ne sont pas particulièrement pertinents pour une analyse du texte.

En conclusion de la partie de l'analyse des résultats de LIME pour l'explicabilité, on peut dire que les résultats sont excellents et parfaitement cohérents. Les mots sont associés de manière cohérente aux émotions repérées par les modèles, émotion qui soit correctement ou non associée à la phrase. Les valeurs attribuées à chaque mot par rapport l'émotion sont elles aussi cohérente.

En définitive, l'explicabilité fournie par LIME nous donne une réelle bonne vision de la manière de pensée de l'algorithme. L'objectif principal concernant l'explicabilité pour notre projet étant la meilleure compréhension de l'effet de chaque mot pour la détermination de l'émotion dominante, est un objectif ici réussi.

## Conclusion

Pour rappel, la problématique de notre projet est la suivante :

*Comment développer et optimiser un modèle de classification de texte, en particulier en utilisant le finetuning de BERT, pour la classification des émotions sur la base de données dair-ai/emotion, tout en assurant l'explicabilité des décisions du modèle ?*

Dans cette partie conclusion, nous allons revenir sur chacun des points la concernant et vérifier que le projet a en effet réussi sur chacun de ses points.

Concernant le développement d'un modèle de classification de texte utilisant BERT : Bien qu'ayant utilisé une version « miniature » de BERT : DistilBERT, nous avons effectivement réussi à implémenter un modèle de classification de texte utilisant cette technologie.

Concernant l'optimisation (du modèle de classification de texte et de la base de données) :

En utilisant notamment la Data Augmentation, nous avons réussi à optimiser les ressources que nous avions. Pour rappel, la base de données dair-ai/emotion fournissant 16 000 exemples, réparties dans les différentes classes de manière assez déséquilibrée, nous avons réussi avec la Data Augmentation à réduire le déséquilibre entre les classes.

Concernant le finetuning du modèle : la recherche des hyperparamètres optimaux pour notre modèle sur cette base de données a été un des gros points lors du développement.

Nous n'avons réalisé que trop tard que nous n'avions pas pris assez de trace écrite pour en développer une analyse dans le rapport, mais à savoir que nous l'avons beaucoup travaillé et en discuterons lors de la soutenance finale.

Concernant la classification d'émotions sur la base de données dair-ai/emotion : En prenant pour preuve les résultats des phrases tests, nous avons réussi à attribuer correctement l'émotion dominante correspondante à chaque phrase.

Concernant l'explicabilité des décisions du modèle : l'utilisation de LIME nous a permis de comprendre l'importance de chaque mot dans la prise de décision concernant le choix de l'émotion dominante pour chaque phrase. Nous avons donc bel et bien fourni un module pertinent d'explicabilité.

Bien que nous ayons pu, comme pour chaque travaux ou projets, approfondir les recherches (comme par exemple : perfectionner encore le finetuning et la recherche d'hyperparamètres, pousser encore la data augmentation pour avoir un parfait équilibre entre les classes, chercher de nouvelles données en utilisant d'autres database que dair-ai/emotion pour approfondir encore l'entraînement et améliorer les résultats des métriques d'évaluation, pousser plus loin l'explicabilité en utilisant d'autres modules complémentaires à LIME), nous avons globalement répondu point par point à la problématique posée.

Le projet est un programme fonctionnel utilisant une version finetunée de BERT, réussissant à classer les émotions du dataset dair-ai/emotion, en y incorporant un module d'explicabilité

## **Présentation des Outils, Partie Technique de L'Etat de l'Art**

### **Langages et Environnements**

- **Python** : Langage de programmation polyvalent, populaire pour le développement, l'analyse de données, et l'intelligence artificielle.
- **Jupyter** : Environnement interactif pour écrire et exécuter du code (notamment Python), idéal pour l'analyse de données et les expériences.

---

### **Bibliothèques et Frameworks**

- **PyTorch** : Bibliothèque pour le Deep Learning axée sur la flexibilité et la recherche.
- **Streamlit** : Framework pour créer rapidement des applications web interactives pour visualiser des modèles ou des données.
- **TensorFlow** : Plateforme open source de bout en bout pour le machine learning et le Deep Learning, utilisée pour construire, entraîner, et déployer des modèles.
- **Keras** : API haut-niveau intégrée à TensorFlow, simplifiant la création et l'entraînement de modèles de Deep Learning.

---

### **Modèles et Architectures**

- **RNN (Recurrent Neural Network)** : Modèle conçu pour traiter des séquences comme du texte ou du son en tenant compte du contexte temporel.
- **BERT (Bidirectional Encoder Representations from Transformers)** : Modèle avancé pour le traitement du langage naturel, basé sur l'architecture Transformer.
- **DistilBERT** : Version allégée de BERT, plus rapide et efficace, idéale pour des applications en temps réel.
- **CNN (Convolutional Neural Network)** : Modèle conçu pour traiter les données visuelles comme les images.
- **Word2Vec** : Technique pour représenter les mots sous forme de vecteurs numériques, capturant leurs relations sémantiques.
- **GPT (Generative Pre-trained Transformer)** : Modèle génératif basé sur les Transformers, utilisé pour générer du texte cohérent et réaliste.

---

### **Concepts et Techniques**

- **TF-IDF** : Méthode pour mesurer l'importance d'un mot dans un document en fonction de sa fréquence dans le corpus.
- **Fine-tuning** : Processus d'ajustement des modèles pré-entraînés comme BERT ou DistilBERT pour des tâches spécifiques.
- **Self-Attention** : Mécanisme des Transformers permettant au modèle de se concentrer sur des parties pertinentes d'une séquence.

- **XAI (Explainable AI)** : Ensemble de techniques pour rendre les décisions des modèles d'intelligence artificielle compréhensibles pour les humains.
- **SHAP (SHapley Additive exPlanations)** et **LIME (Local Interpretable Model-agnostic Explanations)** : Outils pour expliquer les décisions prises par des modèles complexes.

---

### Applications

- **Analyse de sentiments** : Comprendre si un texte exprime une émotion positive, négative, ou neutre.
- **Traduction automatique** : Traduire des textes entre différentes langues.
- **Résumé automatique** : Condenser des textes longs en un résumé court et pertinent.
- **Classification de texte** : Classer automatiquement des textes dans des catégories spécifiques (ex. : spam vs non-spam, avis positif vs négatif).
- **Détection de spam** : Identifier les emails ou messages non sollicités.
- **Analyse multimodale** : Combiner texte, image, et audio pour des analyses plus riches.

---

### Données et Datasets





- **IMDB Reviews** : Dataset d'avis de films pour l'analyse de sentiments.
- **dair-ai/emotion** : Dataset de tweets classifiés par émotions.
- **MNIST** : Dataset d'images de chiffres manuscrits.
- **CIFAR-10** : Dataset d'images classées en 10 catégories (chats, avions, voitures, etc.).

---

### Perspectives futures

- **Modèles plus efficaces** : Développement de modèles plus légers et rapides (ex. : DistilBERT).
- **Analyse émotionnelle avancée** : Détection d'émotions complexes comme l'ironie ou le sarcasme.
- **Multimodalité** : Intégration de données texte, audio, et image pour une meilleure compréhension contextuelle.
- **Adaptabilité linguistique** : Meilleure prise en charge des langues et dialectes moins représentés.

- 
- <sup>i</sup> Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
- <sup>ii</sup> Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2017). *New avenues in opinion mining and sentiment analysis*. IEEE Intelligent Systems.
- <sup>iii</sup> Liu, B. (2012). *Sentiment analysis and opinion mining*. Synthesis Lectures on Human Language Technologies.
- <sup>iv</sup>  Calvo, R. A., & D'Mello, S. (2010). *Affect detection: An interdisciplinary review of models, methods, and their applications*. IEEE Transactions on Affective Computing.
- <sup>v</sup>  Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. NAACL.
- <sup>vi</sup>  Statista. (2023). Le Big Bang du Big Data : Evolution du volume de données numériques générées dans le monde. Statista. Disponible sur <https://fr.statista.com/infographie/17800/big-data-evolution-volume-donnees-numeriques-genere-dans-le-monde>.
- <sup>vii</sup> Schuller, B., Rigoll, G., & Lang, M. (2011). *Hidden Markov model-based speech emotion recognition*. IEEE Transactions on Affective Computing.
- <sup>viii</sup> Pang, B., & Lee, L. (2008). *Opinion mining and sentiment analysis*. Foundations and Trends in Information Retrieval.
- <sup>ix</sup>  Doshi-Velez, F., & Kim, B. (2017). *Towards a rigorous science of interpretable machine learning*. arXiv preprint.
- <sup>x</sup>  1. <sup>x</sup> Ruder, S., Peters, M., Swayamdipta, S., & Wolf, T. (2019). *Transfer learning in NLP*. ACL Tutorial.
- <sup>xi</sup> Manral, P. (2023). Navigating the evolution of NLP: A comprehensive deep dive. *LinkedIn Pulse*. Disponible sur <https://www.linkedin.com/pulse/navigating-evolution-nlp-comprehensive-deep-dive-2013-manral-6f7jf/>.
- <sup>xii</sup> Code & Cortex. (n.d.). TF-IDF avec la librairie scikit-learn : Comprendre et appliquer. Code & Cortex. Disponible sur <https://www.codeandcortex.fr/tfidf-librairie-scikit-learn/>.
- <sup>xiii</sup> DeepAI. (n.d.). *N-Gram*. DeepAI Machine Learning Glossary. Disponible sur <https://deepai.org/machine-learning-glossary-and-terms/n-gram>.
- <sup>xiv</sup> GeeksforGeeks. (n.d.). *Random Forest Algorithm in Machine Learning* [Illustration]. Disponible sur <https://media.geeksforgeeks.org/wp-content/uploads/20240701170624/Random-Forest-Algorithm.webp>.
- <sup>xv</sup> AWS. (2017). *Introduction to Gluon* [GIF]. Disponible sur <https://d2908q01vomqb2.cloudfront.net/f1f836cb4ea6efb2a0b1b99f41ad8b103eff4b59/2017/10/06/intro-gluon-1.gif>.
- <sup>xvi</sup> Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179-211.
- <sup>xvii</sup> Graves, A., Mohamed, A.-R., & Hinton, G. (2013). Speech recognition with deep recurrent neural networks. *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 6645-6649. IEEE.
- <sup>xviii</sup> Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in Neural Information Processing Systems*, 3104-3112.
- <sup>xix</sup> Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735-1780.
- <sup>xx</sup> Tang, D., Qin, B., & Liu, T. (2015). Document modeling with gated recurrent neural network for sentiment classification. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1422-1432.
- <sup>xxi</sup> ResearchGate. (2020). *RNN vs. LSTM: RNNs use their internal state (memory) to process sequences of inputs* [Illustration]. Disponible sur <https://www.researchgate.net/publication/341131167/figure/fig1/AS:887489082445828@1588605294853/RNN-v-s-LSTM-a-RNNs-use-their-internal-state-memory-to-process-sequences-of-inputs.jpg>.
- <sup>xxii</sup> Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

- 
- <sup>xxiii</sup> Jalammar, A. (n.d.). *Word2Vec Visualization* [Illustration]. Disponible sur <https://jalammar.github.io/images/word2vec/word2vec.png>.
- <sup>xxiv</sup> Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- <sup>xxv</sup> Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- <sup>xxvi</sup> Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter. *arXiv preprint arXiv:1910.01108*.
- <sup>xxvii</sup> Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8), 9.
- <sup>xxviii</sup>  Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21, 1-67.
- <sup>xxix</sup>  Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., ... & Le, Q. V. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.
- <sup>xxx</sup> Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.
- <sup>xxxi</sup> Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 328-339.
- <sup>xxxii</sup> Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1135-1144.
- <sup>xxxiii</sup> Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, 30.
- <sup>xxxiiii</sup> Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., ... & Rush, A. M. (2020). Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38-45.
- <sup>xxxv</sup>  Powers, D. M. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies*, 2(1), 37-63.
- <sup>xxxvi</sup>  Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4), 427-437.
- <sup>xxxvii</sup> Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, 21(1), 6.
- <sup>xxxviii</sup> Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, 62(1), 77-89.
- <sup>xxxix</sup> Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30(7), 1145-1159.
- <sup>xl</sup> Cambria, E., Schuller, B., Xia, Y., & Havasi, C. (2017). New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 32(5), 15-21.
- <sup>xli</sup> Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5), 1-22.
- <sup>xlii</sup> GeeksforGeeks. (n.d.). *Explainable AI Concept* [Illustration]. Disponible sur <https://media.geeksforgeeks.org/wp-content/uploads/20231201153509/Explainable-AI-Concept-1-660.png>.
- <sup>xliii</sup> Hugging Face Datasets. dair-ai/emotion. Disponible sur : <https://huggingface.co/datasets/dair-ai/emotion>.
- <sup>xliiii</sup> Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., & Potts, C. (2011). Learning word vectors for sentiment analysis. *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 142-150.



- 
- <sup>xliv</sup> Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. *CS224N Project Report*, Stanford.
- <sup>xlvi</sup> Yelp Dataset. Disponible sur : <https://www.yelp.com/dataset>.
- <sup>xlvi</sup> Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., ... & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:1911.02116*.
- <sup>xlvi</sup> Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2020). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4), 1234-1240.
- <sup>xlvi</sup> Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.
- <sup>xlvi</sup> Bolukbasi, T., Chang, K. W., Zou, J. Y., Saligrama, V., & Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. *Advances in Neural Information Processing Systems*, 29.
- <sup>li</sup> Lipton, Z. C. (2016). The mythos of model interpretability. *arXiv preprint arXiv:1606.03490*.
- <sup>li</sup> Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... & Song, D. (2020). Extracting training data from large language models. *USENIX Security Symposium*.
- <sup>lii</sup> Joshi, A., Bhattacharyya, P., & Carman, M. J. (2017). Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)*, 50(5), 1-22.
- <sup>lii</sup> Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper, and lighter. *arXiv preprint arXiv:1910.01108*.
- Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2020). TinyBERT: Distilling BERT for natural language understanding. *Findings of EMNLP 2020*, 4163-4174.
- <sup>liii</sup> Panch, T., Mattie, H., & Celi, L. A. (2019). The “inconvenient truth” about AI in healthcare. *NPJ Digital Medicine*, 2(1), 1-3.
- <sup>liii</sup> Pak, A., & Paroubek, P. (2010). Twitter as a corpus for sentiment analysis and opinion mining. *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC'10)*.
- <sup>liii</sup> Medhat, W., Hassan, A., & Korashy, H. (2014). Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4), 1093-1113.
- <sup>liii</sup> Zhou, X., & Zafarani, R. (2020). A survey of fake news: Fundamental theories, detection methods, and challenges. *ACM Computing Surveys (CSUR)*, 53(5), 1-40.
- <sup>liii</sup> Schmidt, A., & Wiegand, M. (2017). A survey on hate speech detection using natural language processing. *Proceedings of the Fifth International Workshop on Natural Language Processing for Social Media*, 1-10.
- <sup>liii</sup> Benton, A., Mitchell, M., & Hovy, D. (2017). Multi-task learning for mental health using social media text. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, 152-162.
- <sup>liii</sup> Tetlock, P. C. (2007). Giving content to investor sentiment: The role of media in the stock market. *The Journal of Finance*, 62(3), 1139-1168.
- <sup>liii</sup> Kaur, S., Kaul, A., & Sikka, G. (2020). Detecting sarcasm in multimodal sentiment analysis. *Neural Computing and Applications*, 32(3), 6503-6514.
- <sup>liii</sup> Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *Proceedings of the International Conference on Machine Learning (ICML)*.
- <sup>liii</sup> Alayrac, J.-B., Recasens, A., Schneider, R., Arandjelovic, R., Ramapuram, J., & Botvinick, M. (2022). Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*.
- <sup>liii</sup> Scao, T. L., Fan, A., Akiki, C., Pavlick, E., Chaudhary, V., & BigScience Workshop (2022). BLOOM: A 176B-parameter open-access multilingual language model. *Proceedings of NeurIPS 2022*.
- <sup>liii</sup> Barocas, S., Hardt, M., & Narayanan, A. (2017). Fairness and machine learning. *Fairness in Machine Learning Course Lecture Notes*.