République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université A/Mira de Béjaia

Faculté des Sciences Exactes

Département d'Informatique

MÉMOIRE DE MASTER RECHERCHE

En Informatique

Option

Intelligence artificielle

Thème

Détection par approche d'analyse du langage naturel des contenus produits par intelligence artificielle générative

Présenté par : -Abid Lydia

-Bounif Yasmine

Soutenu le 01/07/2024 devant le jury composé de :

Présidente	**Dr D. BOULAHROUZ**	**Maître de conf. A**	**U. A/Mira Béjaia.**
Encadrants	**Pr K. AMROUN**	**Professeur**	**U. A/Mira Béjaia.**
	Dr N. ELSAKAAN	**Maître assistant B**	**U. A/Mira Béjaia.**
Examinateur	**Dr F. BOUCHEBBAH**	**Maître de conf. A**	**U. A/Mira Béjaia.**

Béjaia, Juillet 2024.

⁂ *Remerciements* ⁂

Avant tout, nous tenons à remercier Dieu, le tout-puissant, de nous avoir accordé la volonté, la patience et surtout la santé durant toutes nos années d'études. Que sa guidance nous accompagne tout au long de notre vie future.

Nous tenons à remercier nos encadrants, Dr. ELSAKAAN Nadim et Pr. AMROUN Kamal de nous avoir supervisées durant notre projet de fin d'étude, pour leur orientations, leur précieux conseils et encouragements qui nous ont permis de mener à bien ce travail.

Nous souhaitons également exprimer notre gratitude à nos parents et à nos familles, qui ont su nous supporter et encourager tout au long de notre vie, ainsi que pour leur patience, leur soutien et leur aide inestimable.

Nous tenons à exprimer également notre gratitude aux membres de jury pour avoir accepté d'examiner et de juger notre travail.

Enfin, nous remercions tous ceux et celles qui, de Près ou de loin, ont contribué à l'aboutissement de ce mémoire.

⁂ *Dédicaces* ⁂

Je dédie ce travail :

A mes chers parents, pour tous leurs sacrifices, leur amour et leur soutien

A mes frères, Yanis et Walid, pour leurs encouragements permanents

A toute ma famille pour leur aide et leur soutien

A mes amis et collègues, et tous qui m'ont aidé de près ou de loin

BOUNIF Yasmine

⁂ *Dédicaces* ⁂

Je dédie ce travail :

À mes chers parents, pour tous leurs sacrifices, leur amour et leur soutien moral

À mes sœurs et frère, pour leurs encouragements permanents

À mon petit neveu adoré et mon très cher beau-frère dont le soutien et l'encouragement constants ont été inestimables

À mes chers cousins et cousines

À mes amis et collègues, et à tous ceux qui m'ont aidé

ABID Lydia

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

ANN Artificial Neural Network

BART Bidirectional and Auto-Regressive Transformers

BERT Bidirectional Encoder Representations from Transformers

cGAN Conditional Generative Adversarial Network

CNN Convolutional Neural Network

DL Deep Learning

DCGAN Deep Convolutional Generative Adversarial Network

GAN Generative Adversarial Network

GPT Generative Pre-trained Transformer

GRU Gated Recurrent Unit

IA Intelligence Artificielle

IBM International Business Machines

LLM Large Language Model

LSTM Long Short Term Memory

ML Machine Learning

MLM Masked Language Model

NER Named Entity Recognition

NLG Natural Language Generation

NLP Natural Language Processing

NLU Natural Language Understanding

POS Parts Of Speech

PHD Artificial Neural Network

RNN Recurrent Neural Network

RVB Rouge Vert Bleu

T5 Text To Text Transfer Transformer

TF-IDF Term Frequency Inverse Document Frequency

ULMFiT Universal Language Model Fine-Tuning

WGAN Wasserstein Generative Adversarial Network

Introduction générale

L'arrivée rapide de l'intelligence artificielle (IA) générative a transformé la façon dont nous produisons et consommons du contenu en ligne. Les grands modèles de langage ont démontré une capacité impressionnante à générer des textes, des images et des vidéos de manière autonome, souvent indiscernable de ceux créés par des humains. Dans le domaine du traitement du langage naturel (NLP), ces avancées ont permis aux modèles de comprendre et de générer du langage de manière plus contextuelle et précise, ouvrant ainsi de nouvelles possibilités pour la traduction automatique, la rédaction de contenu personnalisé et d'autres applications où l'interprétation et la production de langage sont nécessaires.

La problématique principale de ce mémoire réside dans la nécessité urgente de développer des méthodes efficaces de détection et de classification des contenus produits par l'intelligence artificielle générative. Alors que les grands modèles de langage basés sur l'architecture transformer, démontrent une capacité impressionnante à produire des textes, des images et des vidéos indiscernables des créations humaines, cette avancée technologique pose des défis significatifs en termes de sécurité et d'intégrité des plateformes en ligne. Identifier avec précision ces contenus est donc essentiel pour contrer la propagation de désinformations, de contenus trompeurs et potentiellement nuisibles, assurant ainsi la fiabilité des informations diffusées et renforçant la confiance du public dans un environnement numérique complexe et en évolution constante.

La motivation derrière ce travail de recherche est de souligner les défis et les risques potentiels associés à l'utilisation croissante de l'intelligence artificielle générative dans la création de contenu, en mettant en évidence les menaces qui pourraient compromettre la sécurité et l'intégrité des informations publiées sur le web, y compris dans le domaine académique. En abordant ces préoccupations, il est essentiel de développer des stratégies de protection et de surveillance pour garantir un environnement en ligne sûr et fiable pour la diffusion de connaissances et d'informations.

Pour relever ces défis, nous proposons une solution basée sur une version simplifiée d'un grand modèle de traitement du langage naturel. Ce mémoire est structuré en quatre chapitres distincts :

- **Chapitre 1 : Généralités** : Ce chapitre établit les bases conceptuelles de l'intelligence artificielle générative et des modèles de langage pré-entraînés.
- **Chapitre 2 : État de l'Art** : Ce chapitre examine les méthodes existantes de détection des contenus générés par l'IA, en mettant en lumière les avancées récentes dans ce domaine.
- **Chapitre 3 : Approche Proposée** : Ce chapitre présente notre méthode de détection basée sur le modèle DistilBERT, en décrivant la méthodologie utilisée pour identifier les contenus générés par l'IA.
- **Chapitre 4 : Résultats et Évaluation** : Ce chapitre analyse les résultats de notre étude et évalue les performances de notre système de détection, en tirant des conclusions sur l'efficacité de l'approche proposée.

Chapitre

Chapitre 1 : Généralités

1 Introduction

Ces dernières années, l'intelligence artificielle a révolutionné le monde de la technologie, propulsant des objets intelligents vers une adoption généralisée, grâce aux capacités de l'apprentissage automatique et de l'apprentissage profond. Ces sous-domaines de l'IA, basés sur des réseaux de neurones inspirés du cerveau humain permettent aux machines de comprendre et de traiter des données complexes de manière autonome. L'IA générative permet même aux machines de créer du contenu original, tel que du texte, des images et de la musique, en utilisant des approches avancées tels que les grands modèles de langage et les transformateurs, marquant ainsi une nouvelle ère dans le développement de l'IA.

Dans ce chapitre, nous explorerons en profondeur les concepts clés intégrés dans le domaine génératif de l'IA. Nous commencerons par définir les principaux termes, notamment l'intelligence artificielle, l'apprentissage automatique, les réseaux de neurones et l'apprentissage profond. Ensuite, nous plongerons dans les architectures de réseaux de neurones et les techniques spécifiques utilisées dans l'IA générative, y compris les réseaux de neurones adversariaux (GANs) et les transformateurs. Nous examinerons également les grands modèles de langage tels que GPT, T5, BERT et d'autres, ainsi que les techniques d'ajustement fin utilisées pour adapter ces modèles à des tâches spécifiques.

2 Intelligence artificielle

L'intelligence artificielle est un domaine de l'informatique qui vise à développer des systèmes capables de simuler des comportements intelligents, tels que la compréhension du langage naturel, la résolution de problèmes complexes et la prise de décisions autonomes, en s'appuyant sur des algorithmes et des modèles mathématiques. Elle englobe divers sous-domaines, notamment

les réseaux de neurones, l'apprentissage automatique et profond qui ont conduit à des avancées significatives dans l'intelligence artificielle [3].

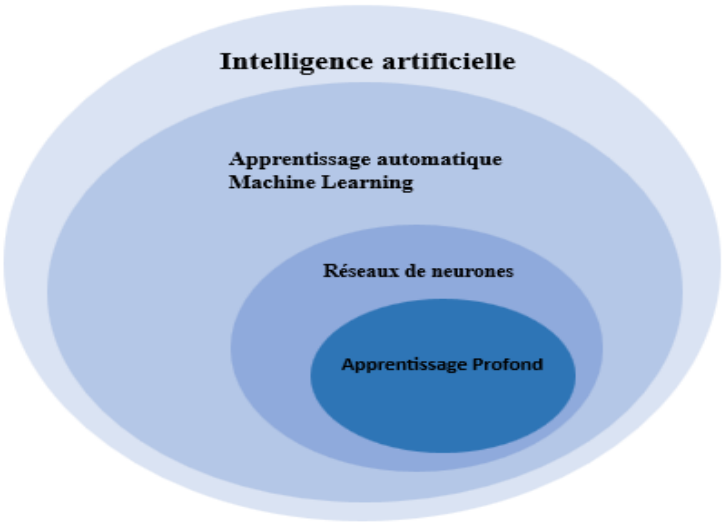


FIGURE .1 – Hiérarchie et composantes de l'intelligence artificielle

En représentant la composante dynamique de l'intelligence artificielle, l'apprentissage automatique se distingue comme le processus clé par lequel les systèmes informatiques acquièrent une capacité d'adaptation et d'amélioration autonomes, illustrant ainsi le potentiel évolutif de l'IA, comme le démontre la figure .1.

3 Apprentissage automatique

Également appelé Machine Learning (ML), ce terme a été introduit par Arthur Samuel en 1959, selon lui «L'apprentissage automatique est la discipline donnant aux ordinateurs la capacité d'apprendre sans qu'ils soient explicitement programmés». C'est une approche de l'intelligence artificielle ou les systèmes informatiques sont capables d'apprendre à partir de données sans être explicitement programmés. Elle permet aux machines de détecter des modèles complexes dans les données et de prendre des décisions basées sur ces derniers. C'est également utilisé pour comprendre et générer du texte, analyser le sentiment, traduire entre différentes langues et d'autres taches liées au langage. Que ce soit pour identifier la relation entre les entrées et les sorties ou pour découvrir des groupements et des associations inattendus dans les données, les algorithmes d'apprentissage offrent une flexibilité remarquable. En adaptant et en affinant leurs modèles à travers des itérations successives, les systèmes de ML peuvent traiter des informations de manière de plus en plus efficace, rendant possible l'automatisation de tâches qui étaient auparavant considérées comme

exclusivement du ressort de l'intelligence humaine. Cette capacité d'auto-amélioration et d'adaptation continue fait de l'apprentissage automatique un pilier crucial de l'avancée technologique [25] [6].

Modèles d'apprentissages L'apprentissage automatique s'appuie principalement sur des algorithmes qui peuvent être supervisés, non supervisés, ou semi-supervisés, offrant une flexibilité dans la manière dont les modèles apprennent à partir des données disponibles. Il existe plusieurs modèles d'apprentissage automatique qui sont classés selon la manière dont l'algorithme apprend, comme montré dans la figure .2. Le choix des approches dépend du type de données à traiter.

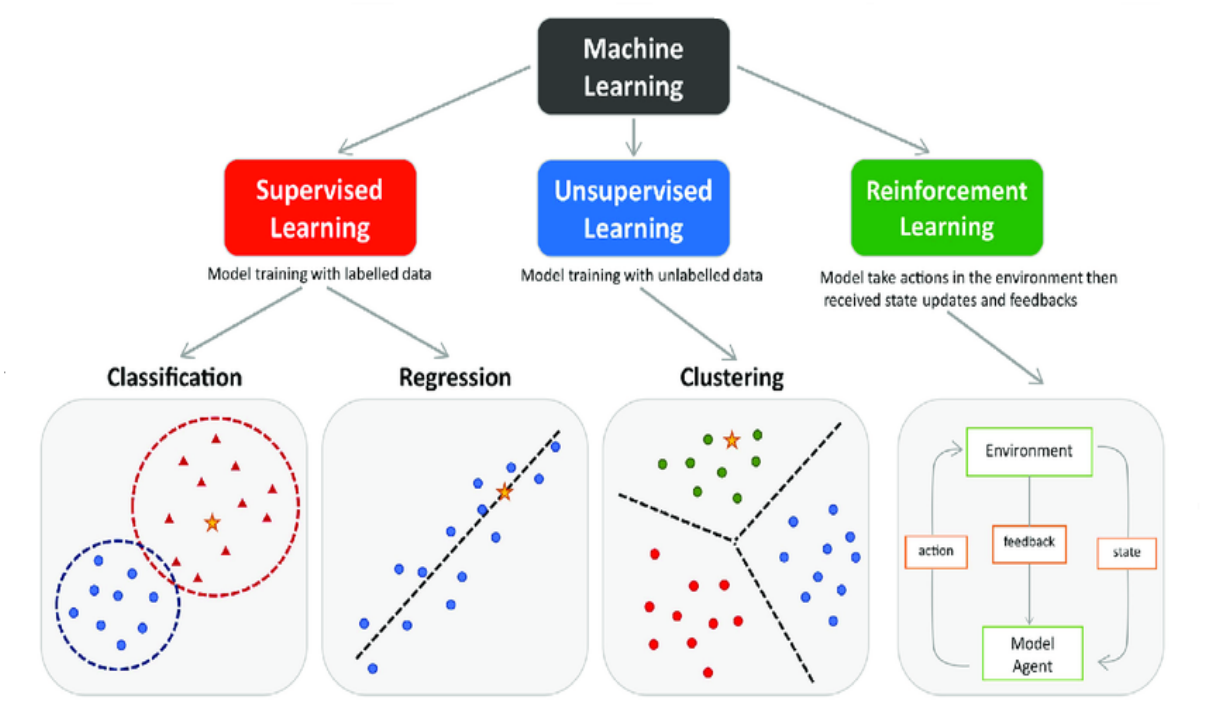


FIGURE .2 – Exemples de modèles d'apprentissages [24]

Apprentissage supervisé : Dans ce type d'apprentissage, on donne au système des exemples de données d'entraînement accompagnées de leurs étiquettes. Parmi ses tâches, on trouve : la classification et la régression [4].

Apprentissage non supervisé : Le principal objectif de l'apprentissage non supervisé est l'exploration de "patterns" ou motifs cachés dans les données, c'est pour cela qu'aucune étiquette n'est fournie à l'algorithme afin qu'il découvre seul une ou plusieurs structures dans les données. Il convient aux tâches telles que le clustering et la détection d'anomalies [25] [4].

Apprentissage semi supervisé : C'est une approche qui utilise à la fois des données étiquetées et non étiquetées pour entraîner un modèle. Employé dans les domaines comme la traduction, la détection de fraudes ou encore l'étiquetage de données [25].

Apprentissage par renforcement : Le système apprend par des récompenses ou des

punitions en réponse à ses actions. Utilisé dans les domaines tels que les jeux-vidéos et la robotique [4].

Apprentissage par transfert : C'est une technique en intelligence artificielle qui consiste à transférer des connaissances acquises dans un contexte donné vers un autre contexte similaire pour résoudre un problème spécifique. Cette technique permet d'adapter ou de transférer des connaissances acquises d'une tâche, d'un langage ou d'un domaine à un autre pour résoudre un problème spécifique. Cette approche est particulièrement utile pour les personnes qui n'ont pas accès aux ressources nécessaires pour entraîner des modèles à partir de zéro [23].

4 Apprentissage profond

L'apprentissage profond connu sous le nom de deep learning (DL) est une famille de méthodes de ML, permettant un apprentissage différent par niveau de détail, en utilisant des réseaux de neurones artificiels. Ces dernières années, l'apprentissage profond est devenu un outil important pour résoudre une grande variété de problèmes d'apprentissage automatique, en raison de sa capacité à apprendre des représentations de données complexes, l'apprentissage profond a révolutionné diverses disciplines, notamment la vision par ordinateur, le traitement du langage naturel, et la reconnaissance vocale ce qui a conduit à des avancées significatives dans l'intelligence artificielle.

Les modèles de DL sont généralement formés en utilisant d'énormes quantités de données permettant aux modèles de reconnaître des structures assez complexes. Cela permet de transformer des données brutes en formes abstraites de plus en plus raffinées à travers des couches successives, augmentant ainsi l'efficacité du modèle dans des tâches spécifiques.

Enfin, les algorithmes de DL reposent sur des réseaux de neurones, qui sont composés de couches de nœuds de traitement interconnectés [25].

5 Réseaux de neurones

Les réseaux de neurones artificiels sont des réseaux fortement connectés de processeurs élémentaires appelés neurones fonctionnant en parallèle. Toute structure hiérarchique de réseaux est évidemment un réseau. Chaque processeur élémentaire effectue des calculs sur les données qu'il reçoit, produit une sortie unique et la transmet aux neurones connectés, la connectivité élevée de ces réseaux permet la formation de hiérarchies complexes qui facilitent l'apprentissage à partir des données. En fait, les réseaux de neurones sont souvent utilisés pour des tâches telles que la classification, la prédiction et même la génération de texte, où ils sont capables d'apprendre des modèles à partir des données d'entrée et de générer des résultats précis ou de générer des résultats en réponse à de nouvelles informations. Par conséquent, les réseaux de neurones artificiels représentent des

outils puissants dans le domaine de l'intelligence artificielle, capables d'accomplir diverses tâches complexes avec d'excellentes performances [21].

5.1 Topologies des réseaux de neurones

Les réseaux de neurones sont aussi définis par leurs topologies, c'est à dire la manière dont les neurones sont connectés et organisés entre eux.

5.1.1 Propagation Avant (Feed Forward) :

Il existe principalement deux types de cette topologie, les réseaux multicouches et les réseaux à connexions locales.

Réseau multicouches : Dans cette topologie illustrée dans la figure .3, un neurone de la couche (N-1) est connecté à tous les neurones de la couche N.

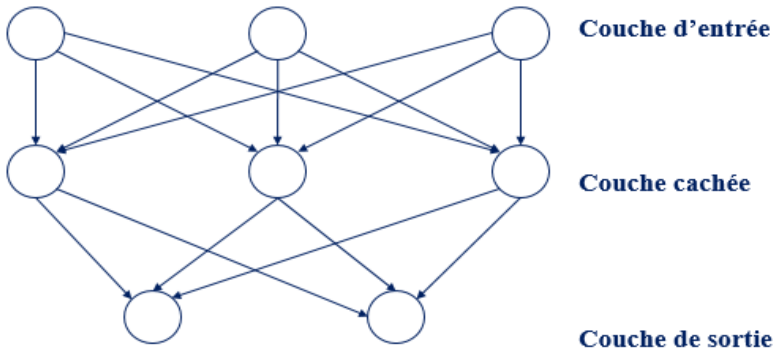


FIGURE .3 – Réseau de neurones multicouches

Réseau à connexions locales : Dans cette topologie présentée dans la figure .4, un neurone de la couche N n'est pas forcément connecté à tous les neurones de la couche N+1.

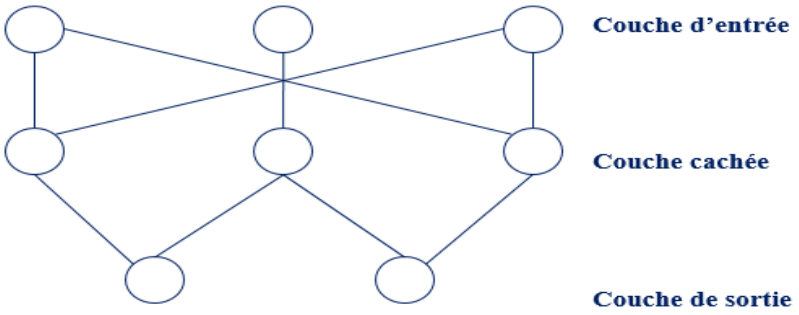


FIGURE .4 – Réseau à connexions locales

5.1.2 Réseaux Récurrents

Dans cette topologie montrée dans la figure .5, l'information d'activation est ramenée en arrière (couche N vers couche N-1, et les connexions sont le plus souvent locales [7].

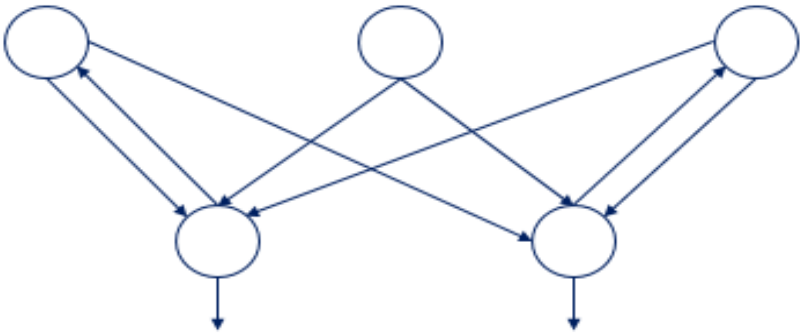


FIGURE .5 – Réseau récurrent

5.1.3 Réseaux à connexions complètes

Dans cette topologie représentée dans la figure .6, chaque neurone est connecté à tous les neurones du réseau (y compris lui-même), c'est un graphe complet [7].

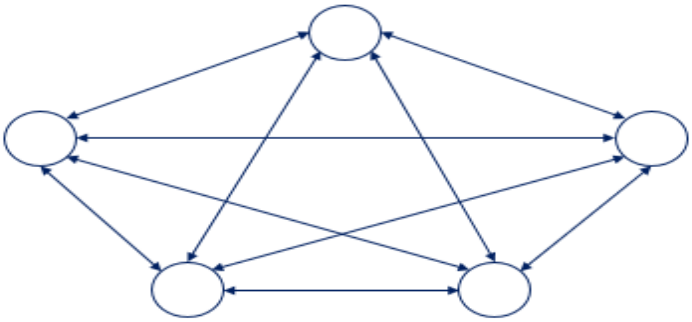


FIGURE .6 – Réseau à connexions complètes

5.2 Mécanisme d'attention

Le mécanisme d'attention est une composante clé des réseaux neuronaux, en particulier dans le contexte des modèles de traitement de séquences comme le transformateur. Il permet au modèle de se concentrer sur des parties spécifiques de l'entrée lors de la génération de la sortie, en attribuant des poids d'importance à différentes parties de l'entrée. Cette capacité fondamentale est amplifiée dans l'auto-attention, qui établit des relations entre différentes positions d'une même séquence pour calculer une représentation globale. Dans ce contexte, chaque élément de la séquence évalue son importance par rapport à tous les autres éléments de la même séquence, permettant ainsi aux modèles comme les transformateurs de capturer des dépendances complexes et à longue distance de manière efficace [33].

5.3 Architectures des réseaux de neurones

Les architectures de réseaux de neurones forment le cœur des systèmes d'apprentissage automatique modernes, exploitant des structures organisées en couches pour traiter et apprendre à partir des données. Parmi les architectures les plus courantes, on trouve les réseaux de neurones artificiels, les réseaux de neurones convolutifs et les réseaux de neurones récurrents. On trouve également les architectures des modèles génératifs qui sont une classe d'algorithmes de DL qui peuvent produire de nouvelles données similaires aux données d'entraînement, les types les plus populaires de cette catégorie sont les GANs, les transformateurs et les auto-encodeurs.

5.3.1 Réseaux de neurones artificiels

Les Réseaux de Neurones Artificiels (ANN) sont des modèles informatiques constitués de neurones interconnectés, inspirés par le fonctionnement du cerveau humain. Chaque neurone reçoit des signaux d'entrée, les traite en effectuant des opérations mathématiques sur ces signaux pondérés par des coefficients, puis produit une sortie en fonction du résultat obtenu. Les neurones sont organisés en couches, comprenant une couche d'entrée pour recevoir les données, une ou plusieurs couches cachées pour le traitement intermédiaire, et une couche de sortie pour produire le résultat final. Les connexions entre les neurones sont modélisées par des poids qui sont ajustés au fur et à mesure de l'apprentissage du réseau, permettant ainsi au réseau de s'adapter aux données et d'apprendre à effectuer des tâches spécifiques. Les ANNs sont largement utilisés dans de nombreux domaines tels que la robotique, la médecine, et la finance [7].

5.3.2 Réseaux de neurones convolutifs

Les réseaux de neurones convolutifs (CNNs) sont conçus spécifiquement pour la reconnaissance et la classification d'images. Ils comportent plusieurs couches de réseaux neuronaux qui extraient des informations des images et déterminent la classe à laquelle elles appartiennent.

Les réseaux neuronaux ordinaires sont des approximateurs mathématiques universels qui prennent une entrée, la transforment par une série de fonctions et en déduisent la sortie. Toutefois, ces réseaux neuronaux ordinaires ne sont pas adaptés à l'analyse d'une image. Pour un programme, toute image n'est qu'un ensemble de nombres du système de couleurs RVB dans un format vectoriel. Si un réseau neuronal peut comprendre le modèle, il peut former un CNN et détecter des images [7].

Les couches d'un réseau CNN comportent des neurones disposés en trois dimensions (hauteur, largeur et profondeur) et se composent de trois types de couches qui sont :

Couche de convolution : elle permet d'extraire les caractéristiques importantes d'une image en appliquant des filtres de convolution.

Couche de pooling : cette couche réduit la dimensionnalité de l'image après l'étape de convolution en sélectionnant les valeurs les plus pertinentes ce qui permet de réduire le surapprentissage.

Couche entièrement connectée : elle prend en entrée les résultats des couches de convolution et de pooling et effectue avec des prédictions ou classifications.

5.3.3 Réseaux de neurones récurrents

Les réseaux de neurones récurrents (RNNs) sont des réseaux où les connexions entre les neurones peuvent former un cycle. Les RNNs utilisent la mémoire interne pour leur traitement. Ce sont une classe d'ANN qui se caractérise par des connexions entre les couches cachées qui se propagent dans le temps afin d'apprendre des séquences. Leurs cas d'utilisation comprennent plusieurs domaines tels que les prévisions basées sur des séries temporelles, les prévisions météorologiques, la traduction linguistique, la reconnaissance de la parole, etc. Plusieurs variantes des RNNs existent, parmi lesquelles on trouve LSTM et GRU qui sont les plus connues.

Les réseaux LSTM (Long Short-Term Memory) sont conçus pour gérer les données séquentielles en préservant des informations à long terme. Un LSTM a pour rôle de reconnaître une entrée importante grâce à sa porte d'entrée, ensuite la stocker avec la porte d'oubli et d'extraire les informations pertinentes au besoin avec la porte de sortie.

les réseaux GRU (Gated Recurrent Units) sont également un type de RNNs tout comme LSTM, mais avec certaines différences, notamment, un réseau GRU ne possède que deux portes,

celle de mise à jour et celle de réinitialisation, les GRUs n'ont pas de mémoire différente de l'état caché exposé, ils ne possèdent également pas de porte de sortie [7].

5.3.4 Transformateurs

Les transformateurs sont une architecture neuronale capable de gérer des informations distantes, contrairement aux LSTM, les transformateurs ne sont pas basés sur des connexions récurrentes qui peuvent être difficiles à paralléliser, ce qui signifie qu'ils peuvent être plus efficaces au niveau de la mise en œuvre. Ils sont constitués d'empilements de blocs transformateurs, dont chacun est un réseau multicouches qui mappe des séquences de vecteurs d'entrée (x1,...,xn) à des séquences de vecteurs de sortie (z1,...,zn) de même longueur. Ces blocs sont créés en combinant des couches linéaires simples d'auto-attention et des réseaux à réaction qui représentent la clé d'innovation des transformateurs.

Cette architecture exploite le mécanisme d'auto-attention qui permet à un réseau d'extraire et utiliser des informations provenant de contextes arbitrairement grands. L'architecture du transformateur est à la base de la plupart des systèmes de traitement de langages naturels modernes, lorsqu'elle est utilisé pour la modélisation causale du langage, l'entrée d'un transformateur est une séquence de mots, et la sortie est une prédiction du mot suivant, ainsi qu'une séquence d'intégration qui représente la signification contextuelle de chacun des mots saisis. Un bloc transformateur comprend quatre types de couches : la couche d'auto-attention, une couche d'anticipation, des connexions résiduelles et des couches de normalisation [16].

5.3.5 Réseaux génératifs antagonistes

Les réseaux génératifs antagonistes (Generative Adversarial Networks) sont des techniques d'apprentissage profond qui utilisent une forme spécifique d'architecture de réseau neuronal développés par Goodfellow et al. [14] en 2014, et depuis lors, ils ont été l'un des domaines de recherche les plus actifs en DL. Les GANs ont émergé comme une technique utile pour produire des données réalistes dans diverses disciplines allant de la vision par ordinateur et des graphiques au traitement du langage naturel et la synthèse audio. Les GANs sont composés de deux réseaux neuronaux adversaires, un réseau générateur et un réseau discriminateur. Le premier apprend à produire des données indiscernables des données réelles, tandis que le second apprend à différencier entre les deux.

Plusieurs types d'architectures GAN ont été proposés, notamment les GANs convolutionnels profonds, les WGANs et les GANs conditionnels. Les DCGANs sont un type de GANs qui utilisent des CNNs dans les réseaux générateur et discriminateur pour produire des images de haute qualité. Les WGANs sont un type de GANs qui utilisent la distance de Wasserstein au lieu de la divergence Jensen-Shannon traditionnelle pour évaluer la distance entre les distributions produites et réelles.

Les cGANs sont un type de GANs qui conditionnent le réseau générateur et le réseau discriminateur sur des informations supplémentaires, telles que des étiquettes de classe ou des vecteurs d'attributs [19].

5.3.6 Auto-encodeurs

Les auto-encodeurs sont des architectures de réseaux neuronaux utilisées pour apprendre des représentations efficaces des données d'entrée sans supervision directe. Ils sont puissants pour détecter les caractéristiques et peuvent être utilisés pour pré-entraîner des réseaux de neurones profonds de manière non supervisée. Certains peuvent même générer de nouvelles données similaires à celles d'entraînement, en apprenant à reproduire leurs entrées en sortie. Cette tâche, en apparence simple, devient complexe lorsqu'on impose des contraintes au réseau, comme limiter la taille de la représentation interne ou ajouter du bruit aux entrées pour les rétablir ensuite. Ces contraintes incitent l'auto-encodeur à découvrir des méthodes efficaces de représentation des données.

Un auto-encodeur se compose d'un encodeur, qui convertit les entrées en une représentation interne, et d'un décodeur, qui convertit cette représentation interne en sorties. Il partage souvent la même architecture qu'un perceptron multicouches, mais le nombre de neurones dans la couche de sortie est identique au nombre d'entrées. Lorsque la dimension de la couche cachée est inférieure à celle de l'entrée, l'auto-encodeur est qualifié de sous-complet, et lorsqu'elle est supérieure, il est dit sur-complet. On distingue différents types d'auto-encodeurs, notamment les auto-encodeurs empilés, les auto-encodeurs contractants qui cherchent à retrouver des codages similaires pour des entrées similaires, les auto-encodeurs récurrents, les auto-encodeurs antagonistes génératifs, etc. [5][7].

6 Traitement automatique du langage naturel

Le traitement automatique du langage naturel ou Natural Language Processing en anglais (NLP) est un sous-domaine de l'intelligence artificielle qui concerne le traitement informatique et la compréhension des langues humaines. Le NLP a vu le jour dans les années 1950, à l'intersection de l'IA et de la linguistique, et est aujourd'hui une combinaison de divers domaines. De grandes quantités de texte sont générées quotidiennement par diverses plateformes de médias sociaux et applications web, ce qui rend difficile le traitement et la découverte des connaissances ou des informations qui y sont cachées, en particulier dans les délais impartis. Cela a ouvert la voie à l'automatisation en utilisant des techniques et des outils d'IA pour analyser et extraire des informations de documents, en essayant d'émuler ce que les êtres humains sont capables de faire avec un volume limité de données textuelles.

En outre, le NLP vise également à apprendre aux machines à interagir avec les êtres hu-

mains en utilisant le langage naturel, ce qui permet de créer des interfaces utilisateur avancées qui peuvent être basées sur du texte ou même de la parole. Les tâches de NLP peuvent être classées en deux catégories : l'analyse syntaxique et l'analyse sémantique. L'analyse syntaxique consiste à comprendre la structure des mots, des phrases et des documents. Parmi les tâches relevant de cette catégorie figurent la segmentation morphologique, la segmentation des mots, l'étiquetage de la partie du discours (POS) et l'analyse syntaxique . L'analyse sémantique, quant à elle, traite du sens des mots, des phrases et de leur combinaison et comprend la reconnaissance des entités nommées (NER), l'analyse des sentiments, la traduction automatique, etc [25].

Comme l'indique la figure .7, le NLP est classé en deux grandes parties à savoir la compréhension du langage naturel (NLU) ou la Linguistique et la génération du langage naturel (NLG).

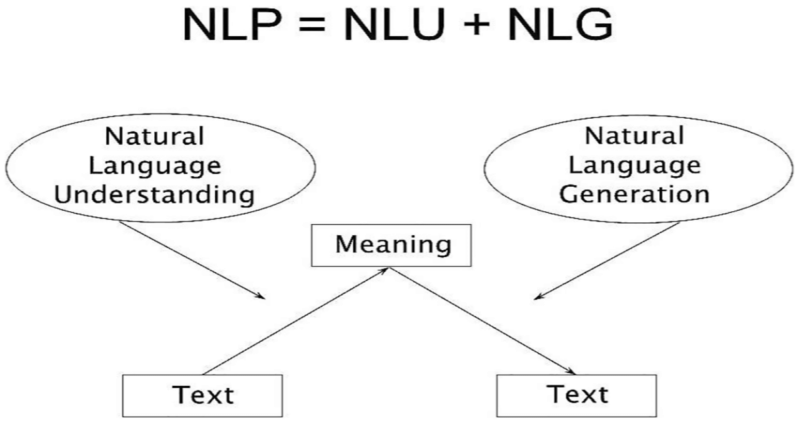


FIGURE .7 – Relation entre NLP, NLU et NLG [25]

6.1 Compréhension du langage naturel

La Compréhension du langage naturel (ou Natural Language Understanding en anglais) fait partie du NLP et permet aux machines de comprendre la communication humaine. La NLU aide les agents conversationnels à être précis avec les utilisateurs en analysant les conversations pour générer des résultats spécifiques. Elle permet de comprendre les émotions, les descriptions et le langage des utilisateurs, facilitant ainsi la collecte d'informations pertinentes comme l'organisation, le produit et le contexte de l'utilisateur. La NLU effectue diverses tâches telles que l'extraction de relations, la reformulation, l'analyse sémantique, l'analyse de sentiment et la gestion des dialogues pour améliorer la compréhension des agents conversationnels [25].

6.2 Génération du langage naturel

La Génération de Langage Naturel (ou Natural Language Generation en anglais) est une technique d'IA utilisée pour créer des récits écrits ou parlés à partir des données. Les algorithmes de

NLG analysent le contenu, le comprennent, le structurent et génèrent des phrases grammaticalement correctes pour livrer du contenu aux utilisateurs. La NLG est essentielle pour les agents conversationnels afin de créer des énoncés de communication, en combinant le NLP pour une interaction plus naturelle. Cette technique nécessite l'expertise linguistique pour générer efficacement du contenu adapté à la communication avec les utilisateurs, contribuant ainsi à humaniser l'expérience avec les agents conversationnels [25].

7 Techniques utilisées en traitement du langage naturel

Pour analyser et interpréter efficacement le langage humain, le domaine du traitement automatique du langage naturel emploie une variété de techniques afin de transformer le texte humain en données structurées compréhensibles par les machines.

7.1 Fréquence du Terme - Fréquence Inverse du Document (TF-IDF)

TF-IDF désigne une méthode statistique déterministe utilisée pour évaluer l'importance d'un mot dans un document par rapport à une collection ou un corpus de documents, le rendant proportionnellement plus significatif en fonction de sa fréquence dans le document. Cette approche permet de filtrer les mots courants tout en mettant en avant les termes spécifiques et pertinents pour le document analysé. Ainsi, le poids TF-IDF est élevé pour un mot fréquent dans un document particulier mais rare dans le reste du corpus, ce qui permet d'identifier et de distinguer les mots-clés d'un document [10].

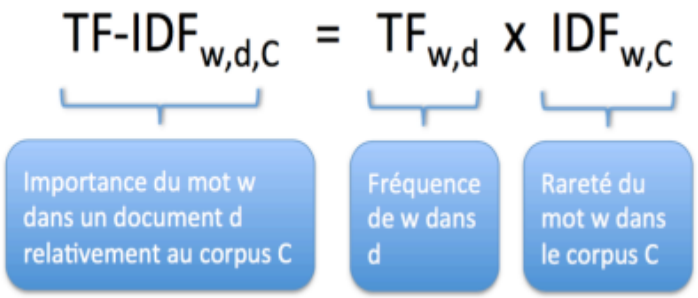


FIGURE .8 – Formule TF-IDF pour l'analyse textuelle [10]

7.2 N-Grammes

Les n-grammes servent à identifier les rapports entre les différentes séquences de texte. Ils s'avèrent très utiles dans un grand nombre de tâches de NLP, comme la modélisation linguistique, la détection de phrases clés ou encore la production de textes. Ils peuvent par exemple servir dans le cadre de la prédiction de mots suivants à partir de séquences antérieures, facilitant ainsi l'émission de suggestions lors de la saisie dans des moteurs de recherche [17].

7.3 Plongements lexicaux

Les plongements lexicaux ou Word Embeddings en anglais, représentent une classe de techniques où des mots ou des phrases de la langue sont mappés à des vecteurs de nombres réels. Conceptuellement, ces techniques cherchent à traduire la sémantique et les relations syntaxiques des mots en une forme géométrique. En d'autres termes, les mots sont représentés par des points dans un espace multidimensionnel de manière à ce que la proximité entre ces points reflète la proximité sémantique entre les mots. Cette approche permet de capturer des nuances complexes du langage humain, rendant possible le traitement de tâches telles que la détection de la similitude sémantique, la traduction automatique, et la reconnaissance des entités nommées avec une précision élevée [10].

8 Intelligence artificielle générative

L'IA générative est un type d'intelligence artificielle qui génère du nouveau contenu en modélisant les caractéristiques des données tirées des grands jeux de données qui alimentent le modèle, selon IBM : "L'IA générative est une catégorie de technologie d'apprentissage automatique qui apprend à générer de nouvelles données à partir d'un ensemble de données d'entraînement". Contrairement aux systèmes d'IA traditionnels qui peuvent reconnaître les modèles ou classifier le contenu existant ou même traduire un texte donné, ces modèles génératifs qui sont autant créatifs que innovant peuvent créer du nouveau contenu sous plusieurs formes, comme du texte, une image, un fichier audio ou du code logiciel [31] [21].

8.1 Principaux cas d'usages de l'IA générative

En théorie, l'IA générative peut être utilisée pour produire tout type de contenus. Dans la pratique, elle est principalement appliquée aujourd'hui pour :

- Implémenter des chatbots (réponses à des questions techniques, etc.)

- Écrire des ébauches de réponses (mails), faire des listes, résumer des textes, rédiger des notes de synthèse et des plans de documents.
- Écrire, auditer, expliquer du code.

8.2 Risques liés à l'IA générative

Alors que les possibilités technologiques de l'IA générative offrent de grandes opportunités, elles suscitent également des inquiétudes. L'intelligence artificielle générative peut assister les auteurs de menace dans la création d'exploits malveillants et éventuellement améliorer l'efficacité de leurs attaques informatiques. Il est très préoccupant qu'elle puisse permettre aux auteurs de menace d'exercer une influence importante. Nous serons donc confrontés à des opportunités révolutionnaires, mais aussi à des défis et à des risques que nous devons affronter et auxquels nous devons faire face. Ci-dessous, vous trouverez quelques-uns des dangers auxquels il est important de faire attention [31].

• La confidentialité des données : Il est possible que les utilisateurs fournissent par erreur des informations confidentielles concernant l'organisation ou des informations nominatives dans des demandes et des invites. Les auteurs de menaces pourraient recueillir ces informations confidentielles afin de dérober l'identité d'une personne ou de diffuser des informations erronées.

• Perte de droits d'auteur : Les outils d'intelligence artificielle générative peuvent donner aux auteurs de menace des moyens avancés de voler les données plus rapidement et en série. Cela représente donc un danger pour la propriété intellectuelle des universitaires et des chercheurs, ainsi que pour la crédibilité et la confiance dans le système éducatif et de recherche.

• La mésinformation et la désinformation : se réfèrent respectivement à la diffusion involontaire et intentionnelle de fausses informations. Le contenu produit par l'intelligence artificielle peut ne pas être clairement identifié comme tel, ce qui pourrait engendrer de la confusion (mésinformation) ou de la déception (désinformation). Les auteurs de menaces peuvent utiliser cette ressource pour commettre des fraudes et mener des campagnes frauduleuses contre des individus et des organisations.

• Ensembles de données empoisonnés : les acteurs malveillants peuvent injecter du code malveillant dans les ensembles de données utilisés pour entraîner les systèmes d'IA générative, ce qui peut avoir un impact négatif sur l'exactitude et la qualité des données générées. Cela pourrait également accroître le risque d'attaques à grande échelle sur la chaîne d'approvisionnement.

• Contenu biaisé : la plupart des ensembles de données de formation pour le LLM proviennent de l'Internet ouvert. Le contenu généré est donc soumis à des biais fondamentaux, puisque seule une fraction de toutes les données mondiales est accessible en ligne et utilisable à des fins d'IA. Le contenu généré peut également être nuisible si l'ensemble de données de formation ne fournit pas une représentation équitable des points de données.

• Code malveillant : les acteurs malveillants qualifiés peuvent contourner les limites des outils d'IA générative pour créer des logiciels malveillants et les utiliser dans des cyberattaques ciblées. Ceux qui ont peu ou pas d'expérience en codage peuvent utiliser l'intelligence artificielle pour écrire facilement des logiciels malveillants fonctionnels susceptibles de nuire à une entreprise ou une organisation.

9 Grands modèles de langage

Les grands modèles de langage, ou Large Language Models (LLM) en anglais, sont des modèles d'intelligence artificielle, généralement basés sur l'architecture Transformer, conçus pour comprendre et générer le langage humain, le code, et d'autres formes de texte. Entraînés sur de vastes ensembles de données textuelles, ils capturent les nuances et les complexités du langage humain, réalisant diverses tâches linguistiques avec précision, fluidité et style. Ces modèles reposent sur l'architecture Transformer, leur permettant d'exécuter des tâches linguistiques complexes sans nécessiter beaucoup d'ajustement fin. Ils peuvent être affinés pour des tâches spécifiques via l'apprentissage par transfert [22].

Les LLMs sont classés en modèles autorégressifs, autoencodeurs, ou une combinaison des deux.

Modèles autorégressifs : Comme GPT, ces modèles prédisent le mot suivant d'une phrase en se basant sur les précédents.

Modèles autoencodeurs : Ces modèles, tel que BERT, construisent une représentation bidirectionnelle d'une phrase.

Combinaison d'autorégressif et d'autoencodeur : Ces modèles comme T5 sont plus polyvalenst car ils utilisent les deux à la fois [22].

9.1 Modèle GPT

GPT (Generative Pre-trained Transformer) est un modèle autorégressif développé par OpenAI en 2018 qui utilise l'attention pour prédire le prochain mot d'une séquence en se basant sur les mots précédents. Les algorithmes GPT sont principalement utilisés pour la génération de texte et sont connus pour leur capacité à générer des textes à consonance naturelle et humaine. Le modèle GPT excelle dans la génération de texte libre aligné sur l'intention de l'utilisateur. GPT utilise le décodeur du transformateur et ignore l'encodeur pour devenir exceptionnellement performant dans la génération de texte, un mot à la fois. Les modèles basés sur GPT sont donc les meilleurs pour travailler et générer rapidement de grandes quantités de texte, par rapport à d'autres LLM qui se concentrent sur le traitement et la compréhension du texte. Les architectures dérivées du GPT sont idéales pour les applications qui requièrent la capacité d'écrire librement du texte [22].

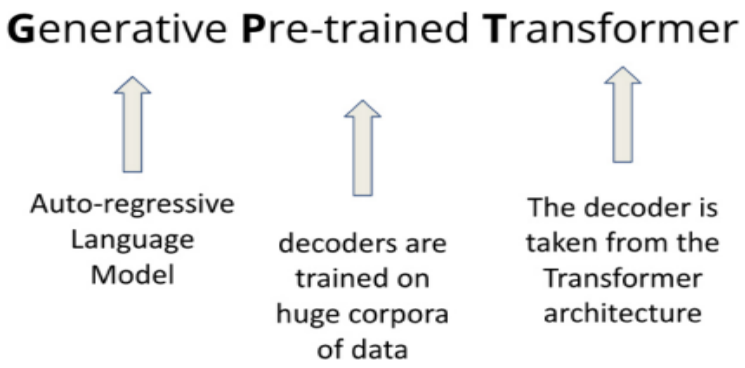


FIGURE .9 – Représentation du modèle GPT [22]

9.2 Modèle BERT

BERT, acronyme de Bidirectional Encoder Representations from Transformers, est un modèle de traitement du langage naturel développé par Google en 2018. Ce modèle repose sur l'architecture des transformateurs bidirectionnelle qui permet de capturer des informations contextuelles à la fois avant et après chaque mot dans une phrase afin d'identifier les relations entre les mots d'une phrase. BERT est pré-entraîné sur de vastes corpus de texte non étiqueté dans une tâche appelée "modèle de langage masqué" (MLM), où il apprend à prédire les mots manquants dans une phrase [22].

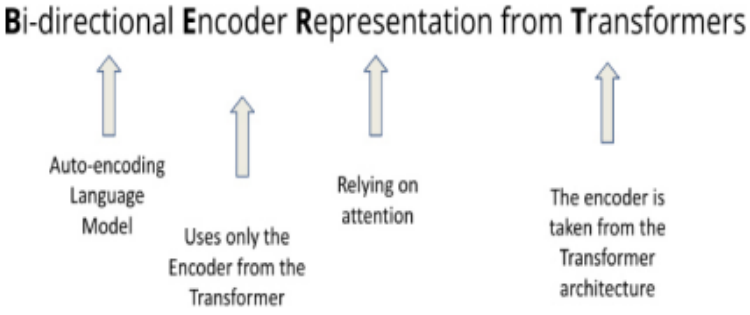


FIGURE .10 – Représentation du modèle BERT [22]

9.3 Modèle T5

Text-To-Text Transfer Transformer (T5) est un modèle de traitement du langage naturel développé par Google research en 2019 qui se distingue par son architecture purement encodeur/décodeur. Contrairement à BERT et GPT, T5 a été conçu pour exécuter une large gamme de tâches en traitement du langage naturel, allant de la classification de texte à la génération de texte. T5 utilise à la fois l'encodeur et le décodeur du transformateur, ce qui lui confère une grande polyvalence dans le traitement et la génération de texte. Les modèles basés sur T5 sont capables

d'effectuer diverses tâches en traitement du langage naturel en construisant des représentations du texte d'entrée à l'aide de l'encodeur et en générant du texte à l'aide du décodeur. Les architectures dérivées de T5 sont idéales pour les applications qui nécessitent à la fois la capacité de traiter et de comprendre du texte et de générer du texte librement. En résumé, T5 est un modèle LLM polyvalent qui excelle dans une variété de tâches en traitement du langage naturel grâce à son architecture encodeur/décodeur basée sur le transformateur [22].

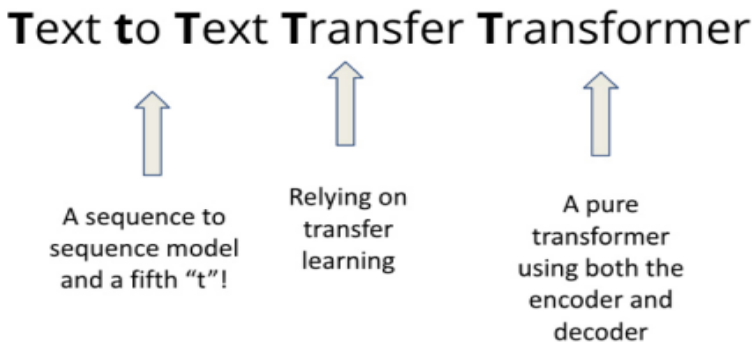


FIGURE .11 – Repésentation du modèle T5 [22]

9.4 Modèle BART

BART (Bidirectional and Auto-Regressive Transformers) est un modèle de traitement du langage naturel développé par Facebook AI et publié en 2019 qui combine des éléments bidirectionnels et auto-régressifs pour améliorer la qualité de la génération de texte. BART est capable de comprendre et générer du langage humain, du code et bien plus encore. Ces modèles sont entraînés sur de vastes quantités de données textuelles, ce qui leur permet de capturer les complexités et les nuances du langage humain. En termes de fonctionnement, BART est pré-entraîné sur un grand corpus de données textuelles et sur des tâches spécifiques de modélisation du langage. Pendant la phase de pré-entraînement, ce modèle cherche à apprendre et à comprendre le langage général ainsi que les relations entre les mots. Cette phase de pré-entraînement est cruciale pour permettre à BART d'acquérir une compréhension profonde du langage et des structures linguistiques, ce qui lui permet d'exceller dans des tâches telles que la génération de texte et la classification avec précision et style [22].

10 Ajustement fin des grands modèles de langage

L'ajustement fin (ou Fine-Tuning en anglais) est un processus clé de l'apprentissage par transfert qui optimise les performances des modèles de réseaux neuronaux pré-entraînés sur des tâches de classification spécifiques. La technique implique des modifications légères mais stratégiques des poids des couches réseau afin de les adapter précisément à la tâche cible. Le réglage fin est particulièrement apprécié dans le domaine du traitement du langage naturel (NLP), où les modèles sont initialement pré-entraînés sur de grandes bases de données à usage général pour acquérir une compréhension approfondie du langage. Le modèle est ensuite affiné sur un ensemble de données plus restreint, ciblant une tâche spécifique (par exemple la classification de textes). Cette approche permet d'exploiter les représentations linguistiques universelles apprises lors du pré-entraînement, favorisant ainsi une meilleure généralisation et optimisation sur de nouvelles tâches spécifiques sans nécessiter un entraînement exhaustif à partir de zéro. En résumé, le Fine-Tuning facilite une adaptation efficace des modèles pré-entraînés à des contextes particuliers, en tirant parti des connaissances générales préalablement acquises pour améliorer significativement la performance sur des tâches ciblées[21].

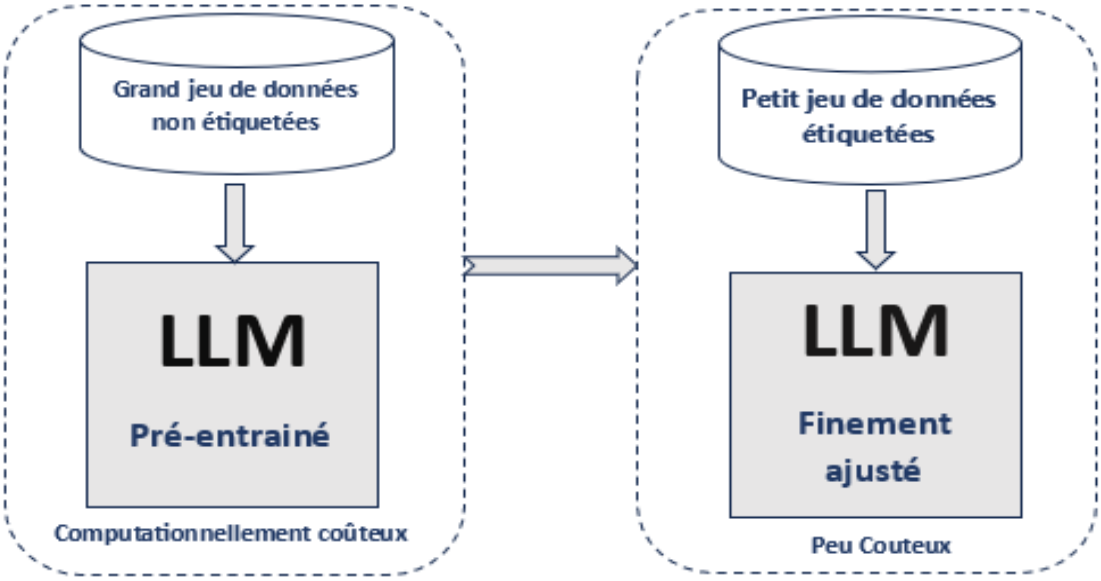


FIGURE .12 – Fine-tuning d'un LLM avec transfert learning

10.1 Types de Fine-Tuning

Il existe principalement trois types de fine-tuning visant à améliorer les performances du modèle en fonction de différents besoins et contextes spécifiques.

10.1.1 Pré-entraînement général de modèle de langage

Le modèle est d'abord entraîné sur un vaste corpus de texte général (comme Wikipedia) pour apprendre une représentation riche et universelle du langage.

10.1.2 Fine-Tuning spécifique à la tâche

Le modèle pré-entraîné est ensuite ajusté finement sur un ensemble de données spécifique à la tâche, permettant au modèle d'apprendre les nuances et spécificités de cette tâche particulière.

10.1.3 Techniques clés pour le Fine-Tuning

- **Fine-Tuning discriminatif** : Différents taux d'apprentissage sont appliqués à différentes couches du modèle, permettant un ajustement plus précis et évitant la perte de connaissances générales apprises lors du pré-entraînement.
- **Taux d'apprentissage triangulaires inclinés** : Un programme de taux d'apprentissage qui augmente initialement puis diminue, favorisant une convergence rapide vers une bonne solution avant un affinage plus détaillé.
- **Décongélation graduelle** : Les couches du modèle sont décongelées et ajustées finement de manière séquentielle, commençant par les couches supérieures, pour prévenir l'oubli catastrophique.

10.2 Relation avec la classification de texte

Dans le contexte de la classification de texte, le Fine-Tuning est essentiel pour adapter les modèles de langage universels aux nuances et spécificités d'un corpus particulier. Cela permet d'obtenir des performances élevées même avec un nombre limité d'exemples d'entraînement. L'approche ULMFiT (Universal Language Model Fine-Tuning), par exemple, illustre l'efficacité du Fine-Tuning dans la classification de texte, en offrant une méthode robuste pour ajuster un modèle de langage pré-entraîné à différentes tâches de NLP[21].

Chapitre

Chapitre 2 : Etat de l'art

Dans ce chapitre, nous plongeons dans le domaine du traitement automatique du langage naturel pour ensuite explorer en détail les méthodes de détection de contenu texte généré par l'IA. Ces méthodes peuvent être classées en trois catégories distinctes en fonction de leurs caractéristiques particulières, chacune pouvant être examinée sous des perspectives que nous pouvons catégoriser en quatre scénarios distincts basés sur leur application. Ces catégorisations mettent en évidence les différents niveaux d'informations disponibles pour les détecteurs, allant d'une connaissance limitée à un accès complet, et démontrent les différents scénarios rencontrés dans la détection du contenu.

1 Évolution du traitement automatique du langage naturel

L'étude approfondie menée par U. Naseem et al. [20] offre un aperçu détaillé de l'évolution du traitement automatique du langage naturel à travers les différentes étapes de son développement. Cette analyse met en lumière les progrès significatifs réalisés dans le domaine, depuis les premières approches basées sur des règles linguistiques jusqu'aux modèles de représentation de mots de pointe. Les premières approches du NLP étaient principalement fondées sur des règles linguistiques et des modèles manuels pour tenter de décoder et interpréter le langage humain. Cependant, ces méthodes se sont rapidement heurtées à des limitations en raison de la complexité et de la variabilité du langage naturel. Avec l'avènement de l'informatique et de l'intelligence artificielle, le domaine du NLP a connu une transformation importante. L'introduction de méthodes statistiques et de modèles d'apprentissage automatique a permis d'améliorer considérablement la capacité des systèmes à traiter de grandes quantités de données textuelles et à extraire des informations pertinentes de manière plus efficace.

L'apparition de l'apprentissage profond a révolutionné le domaine du NLP en permettant la création de modèles neuronaux sophistiqués capables de saisir les subtilités du langage naturel. Des architectures telles que les réseaux de neurones récurrents et les transformateurs ont considérablement amélioré les performances des systèmes de NLP dans des tâches complexes telles que la traduction automatique et l'analyse de sentiment. Aujourd'hui, le NLP bénéficie de l'utilisation

de modèles de représentation de mots avancés tels que Word2Vec, GloVe et GPT. Ces modèles permettent de convertir le texte en vecteurs numériques tout en préservant sa sémantique, ouvrant ainsi de nouvelles perspectives en matière de traitement automatique du langage naturel. Les progrès réalisés dans le domaine du NLP ont ouvert la voie à une multitude d'applications innovantes dans divers domaines tels que la traduction automatique, l'analyse de sentiment et la génération de texte. Ces avancées positionnent le NLP comme un domaine de recherche et de développement essentiel dans le domaine de l'intelligence artificielle, avec des perspectives prometteuses pour l'avenir.

2 Scénarios de détection

Dans le domaine de la détection de contenu texte généré par l'IA, l'approche boite blanche implique soit un accès complet et transparent aux éléments internes du modèle de langage, tels que ses paramètres, ses couches cachées, ses mécanismes d'attention, etc, ce qui n'est pas évident lorsque les modèles sont inaccessibles, ou en ayant un accès partiel aux logits du modèle. Cela signifie que les détails internes du modèle sont connus et exploités pour effectuer la détection.

En revanche, une approche boite noire ne nécessite pas une compréhension détaillée du modèle de langage, il y a deux cas lorsque le modèle source est connu et lorsqu'il ne l'est pas. Au lieu de cela, elle se concentre sur l'utilisation des entrées et sorties du modèle pour la détection, sans nécessiter de connaissance interne du fonctionnement du modèle. Ces éléments ne sont pas directement accessibles ou visibles lorsqu'on utilise le modèle, mais ils sont responsables de son fonctionnement et de sa performance dans la génération de texte [37].

3 Méthodes de détection

Les méthodes de détection de contenu texte généré par des modèles de langage sont classifiées en fonction de la transparence du modèle (boîte noire ou blanche) et de la connaissance de la source (connue ou inconnue). Il distingue les approches suivantes :

- Basées sur l'apprentissage, qui nécessitent un entraînement avec des données étiquetées.
- Des méthodes zéro-shot, qui fonctionnent sans apprentissage spécifique sur des exemples cibles.
- Des techniques en filigrane qui intègrent des signaux distinctifs dans le contenu pour faciliter la détection.

Pour chaque catégorie, les méthodes sont divisées en fonction de la disponibilité d'informations sur le modèle : accès complet, partiel ou aucune information sur la structure interne du modèle, illustrant ainsi la diversité des stratégies utilisées pour identifier le contenu généré par l'IA.

Le tableau suivant schématise de manière détaillée ces différentes catégories et approches, offrant une vue complète sur les méthodes de détection de contenu généré par IA en fonction de leur transparence et de la connaissance de la source.

Type de boîtes	**Aspect**	**Basé sur l'apprentissage**	**Zero shot**	**En filigrane**
Boîte noire	Source connue	Le modèle GPT Paternity Test (GPT-Pat) [39]	Le modèle DetectGPT [18]	La méthode proposée par X. Yang et al. [38]
	Source inconnue	Le modèle Ghostbuster, Notre méthode basée sur DistilBERT [34]	Le modèle de E. Tulchinski et al. [32]s	-
Boîte blanche	Accès complet	AI-GENERATED TEXT CLASSIFIER [27]	Le modèle de Jinyan Su et al. [29]	Le modèle de Y. Fu et al. [12]
	Accès partiel	Le modèle de V. Vora et al.[35]	Divergent N-Gram Analysis (DNA-GPT) [36]	-

Tableau .1 – Tableau de classification des méthodes

Passons maintenant à l'analyse des trois catégories principales de méthodes de détection de contenu texte généré par l'IA, ainsi que quelques exemples spécifiques de méthodes dans chacune de ces catégories.

3.1 Classificateurs basés sur l'apprentissage

Les classificateurs basés sur l'apprentissage constituent une approche fondamentale dans la détection de contenu texte généré par l'IA. Ces méthodes utilisent des modèles pré-entraînés sur des données binaires comprenant des distributions de textes générés par l'humain ou par l'IA. Les classificateurs peuvent être différenciés en fonction de leur capacité à opérer en tant que boite blanche ou boite noire [37].

3.1.1 Méthodes en boite blanche avec tous les paramètres du modèle

Le modèle AI-GENERATED TEXT CLASSIFIER proposé par P.Sarzaeim et al. [27] détermine la probabilité qu'un texte soit généré par une intelligence artificielle. Il utilise un outil d'extraction de caractéristiques textuelles qui est le vecteur de comptage qui convertit le texte en vecteurs numériques de longueur fixe, il a été formé sur 50 vecteurs pour chaque instance de l'ensemble de données, et un réseau de neurones multi-couches (3 couches de 100 neurones chacune) qui a comme rôle d'émettre une classification binaire. L'application comprend une interface utilisateur, un outil de détection de texte IA, une enquête de feedback et une base de données qui stocke les textes saisis en entrée et les commentaires des utilisateurs concernant le résultat, comme illustré dans la figure .1 ci-dessous. Les données ont été collectées manuellement à partir d'articles scientifiques sur Google Scholar et il a été demandé à chatGPT de générer des textes similaires. Ils ont utilisé un apprentissage supervisé étiquetant les données comme texte humain (0) et généré par l'IA (1). Une des limites de cette méthode réside dans le fait que les utilisateurs peuvent ne pas donner leur avis de manière précise ou honnête, ceci peut potentiellement conduire à la transmission d'informations erronées, introduisant ainsi des données erronées ou trompeuses dans l'analyse. Il est donc crucial de prendre en compte ce facteur lors de l'interprétation des résultats obtenus, et d'explorer des approches complémentaires pour garantir une compréhension plus juste et équilibrée du sujet.

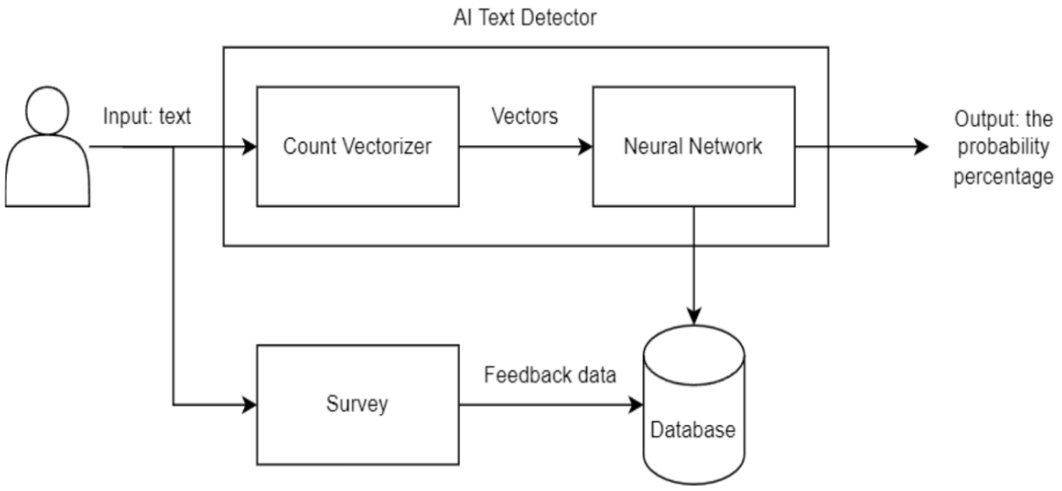


FIGURE .1 – Schéma de l'architecture du modèle AI-GENERATED TEXT CLASSIFIER [27]

3.1.2 Méthodes en boite blanche avec informations partielles du modèle

Le modèle présenté par V. Vora et al. [35] vise à identifier si un texte a été généré par une IA ou par un être humain, en suivant une méthode structurée en plusieurs étapes, la première étant le pré-traitement des données provenant de la compétition "Kaggle ML Olympiad Detect ChatGPT Answers", où ils ont procédé à un nettoyage des données notamment en supprimant les données non étiquetées et les lignes comprenant des valeurs manquantes. Les données sont divisées en ensemble d'entraînement 80% et de validation 20%. Ensuite vient l'étape d'extraction des caractéristiques qui comprend une analyse détaillée des aspects syntaxiques, sémantiques et stylistiques.

Les caractéristiques syntaxiques examinent la structure grammaticale, en se concentrant sur la distribution des parties du discours (POS) pour détecter les schémas distincts, tandis que les caractéristiques sémantiques évaluent la cohérence du sens, utilisant des techniques comme les plongements de mots pour comparer la similarité entre les phrases. L'objectif de l'extraction des caractéristiques est d'exploiter les graphes de connaissance existants pour dériver les caractéristiques informatives en alignant les entités, les concepts et les relations du graphe de connaissances avec le contenu textuel en sélectionnant des attributs pertinents basés sur le vocabulaire, la syntaxe, la sémantique et le style. Les caractéristiques stylistiques portent sur des attributs tels que la structure des phrases, le ton et la formalité, évaluant les différences de style à travers des paramètres comme la longueur des phrases.

Enfin, l'entraînement du modèle choisi qui est BERT, ce dernier repose sur plusieurs principes clés de l'apprentissage profond et du traitement du langage naturel. Il intègre un mécanisme d'attention pour comprendre le contexte des mots dans une phrase et des encodeurs bidirectionnels pour capturer la signification complète. BERT est pré-entraîné sur diverses tâches linguistiques, pour détecter les textes générés par l'IA, il est affiné sur des données spécifiques incluant à la fois

du texte humain et de l'IA. Ce processus lui permet d'apprendre à identifier les caractéristiques distinctives entre ces deux types de textes, fournissant ainsi une base solide pour la détection précise du texte généré par l'IA. Bien que l'article propose une méthode prometteuse pour cette tâche, il soulève également des questions sur la complexité de la mise en œuvre et la maintenance de graphes de connaissances qui peuvent nécessiter des ressources significatives.

3.1.3 Méthodes en boite noire avec source modèle connue

Le modèle GPT Paternity Test (GPT-Pat) [39] repose sur l'hypothèse que, étant donné un texte, ChatGPT peut générer une question correspondante et y répondre à nouveau. En comparant la similitude entre le texte original et le texte répondu, il est possible de déterminer si le texte a été généré par une machine. Pour réaliser cela, GPT-Pat utilise un réseau Siamois pour calculer la similitude entre le texte original et le texte généré en réponse avec un modèle de langage pré-entraîné (xlm-RoBERTa-base) pour convertir les textes en plongements sémantiques, et un classificateur composé de couches entièrement connectées pour la classification finale, le fonctionnement de cette méthode est schématisé dans la figure .2 . Le réseau est entraîné sur divers ensembles de données, notamment le HC3, qui comprend des questions et leurs réponses correspondantes fournies par des humains ou générées par ChatGPT. D'autres ensembles de données utilisés pour évaluer la généralisabilité de la méthode incluent Wiki, CCNews, CovidCM, et ACLAbs, chacun ayant été nettoyé pour éliminer les mots indiquant clairement s'ils sont écrits par des humains ou générés par une machine.

Cependant, cette approche dépend fortement de la capacité de ChatGPT à générer des réponses pertinentes et cohérentes, ce qui pourrait poser problème dans des cas où il produit des réponses inexactes ou hors sujet, de plus la nécessité d'une optimisation constante pour s'adapter aux évolutions des modèles de langue pourrait représenter un défi continu pour maintenir son efficacité.

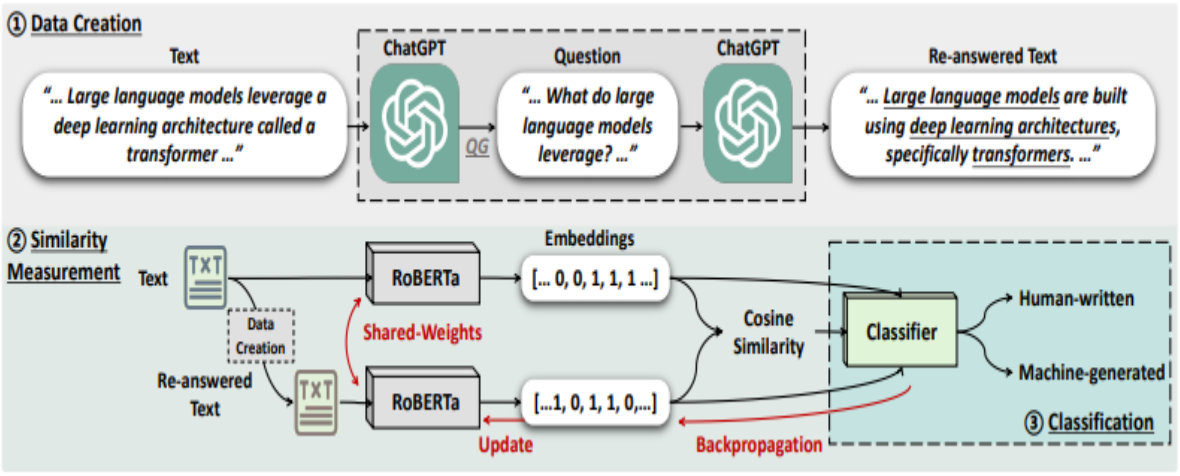


FIGURE .2 – Schéma opérationnel du GPT-Pat pour la détection de texte généré par IA [39]

3.1.4 Méthodes en boite noire avec source du modèle inconnue

Le modèle Ghostbuster, présenté dans l'article "Detecting Text Ghostwritten by Large Language Models" [34], est une avancée notable dans le domaine de la détection du texte généré par IA, offrant une méthode robuste et adaptable pour distinguer le contenu généré artificiellement dans une variété de contextes et de formats de texte. Pour y parvenir, Ghostbuster utilise une approche en plusieurs étapes, la première étant le passage à travers des modèles de langage plus faibles ce qui signifie que le système analyse les textes en utilisant des versions moins avancées ou moins puissantes de modèles de langage artificiels. Concrètement, cela implique l'utilisation d'un modèle unigramme qui regarde la probabilité d'apparition de chaque mot isolément, ou un modèle trigramme Kneser-Ney qui analyse la probabilité des mots en prenant en compte les deux mots précédents comme le montre la figure.3, et des versions plus simples de GPT-3, spécifiquement Ada et Davinci avant qu'ils soient ajustés pour des instructions spécifiques. Cette étape produit des vecteurs de probabilités pour chaque token (ou mot) du document analysé. Ensuite la sélection de caractéristiques à partir des vecteurs obtenus, Ghostbuster cherche à identifier les combinaisons de caractéristiques, des aspects statistiques ou probabilistes des mots dans le texte qui sont les plus susceptibles d'indiquer si un texte a été généré par un humain ou par une IA. Cela est réalisé grâce à une recherche structurée sur les combinaisons de ces caractéristiques et à l'utilisation d'opérations mathématiques simples pour résumer ces caractéristiques en un ensemble plus petit et gérable. Finalement, les caractéristiques sélectionnées sont utilisées pour entraîner un classificateur de régression logistique, un type d'algorithme d'apprentissage automatique, qui apprend à distinguer le texte généré par IA du texte écrit par des humains. Les ensembles de données utilisés pour évaluer Ghostbuster couvrent trois domaines principaux, les dissertations d'étudiants, l'écriture créative, et les articles de presse. Ces données comprennent à la fois du texte rédigé par des humains et du texte généré par des IA, fournissant ainsi une base solide pour tester sa performance [37].

Bien que l'utilisation de modèles de langage plus faibles comme étape intermédiaire soit innovante, elle soulève des questions sur la capacité de la méthode à s'adapter aux progrès rapides dans le domaine des modèles de langage plus récents et plus sophistiqués qui peuvent produire du texte très semblable au style et nuances de l'écriture humaine, réduisant potentiellement l'efficacité des modèles plus faibles à détecter des anomalies ou des signes révélateurs de texte généré par IA.

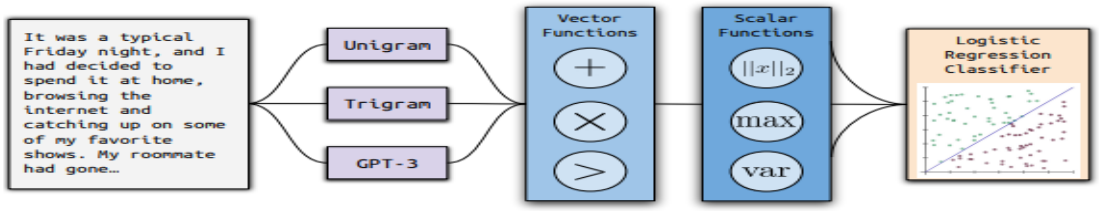


FIGURE .3 – Schéma de la procédure d'entraînement du modèle Ghostbuster [34]

3.2 Zéro-coup

Les détecteurs de zéro-coup exploitent les caractéristiques intégrées des modèles de génération pour se reconnaître automatiquement, sans nécessiter un ensemble de données spécifiquement étiquetées pour la tâche de détection. En utilisant des méthodes telles que l'analyse des poids du modèle et des activations neuronales, ces détecteurs peuvent identifier des signaux de génération automatique de texte. Cette approche offre l'avantage de ne pas nécessiter de données étiquetées pour la détection, mais peut présenter des défis en termes de précision et de généralisation dans des contextes variés [37].

3.2.1 Méthodes en boite blanche avec tous les paramètres du modèle

Jinyan Su et al. [29] ont proposé une nouvelle méthode qui s'intitule DetectLLM, comprenant deux variantes, DetectLLM-LRR qui calcule le ratio entre la vraisemblance logarithmique et une mesure de rang d'un texte, où un ratio élevé indique probablement un texte généré par machine. D'autre part, DetectLLM-NPR qui applique des perturbations mineures au texte et mesure la sensibilité de sa mesure de rang à ces changements, avec une sensibilité accrue suggérant également une origine artificielle. Ces méthodes ont été évaluées sur trois ensembles de données distincts : XSum pour les résumés d'articles, SQuAD pour les réponses à des questions basées sur des paragraphes de Wikipedia et WritingPrompts, pour des histoires créatives, utilisant des textes humains comme référence pour générer des paires de textes machine-humain. Cette approche permet d'exploiter les différences subtiles dans la manière dont les modèles de langage et les humains génèrent du texte, offrant un moyen efficace et précis de détecter les contenus générés automatiquement sans nécessiter de données d'entraînement spécifiques[37]. Toutefois, comme toute méthode, elle a

ses limites, notamment la dépendance aux perturbations, son fonctionnement repose encore sur la création de ces dernières. Cette dépendance peut restreindre son application dans des contextes où produire des perturbations s'avère peu pratique.

3.2.2 Méthodes en boite blanche avec informations partielles du modèle

La méthode présentée par X. Yang et al. [36] appelée Divergent N-Gram Analysis (DNA-GPT), permet de repérer les textes générés par les LLMs sans nécessiter de phase d'entraînement. L'étude évalue la méthode sur cinq ensembles de données, comprenant l'anglais et l'allemand, en utilisant des modèles tels que text-davinci-003, GPT-3.5-turbo, GPT-4, GPT-NeoX-20B et LLaMa-13B. Le processus commence par diviser le texte en deux parties, X et Y0, où X représente le début du texte et Y0 la suite à générer par les LLMs. Ces derniers sont ensuite utilisés pour produire différentes séquences basées uniquement sur X, résultant en un ensemble de textes générés. La méthode repose sur l'hypothèse que les LLMs maximisent la probabilité logarithmique tandis que les humains suivent un processus différent, ce qui crée un écart entre les distributions de probabilité des textes générés par les machines et ceux produits par les humains. En analysant cette différence, DNA-GPT peut efficacement distinguer les textes générés par les LLMs. Bien que cette méthode ne nécessite pas d'entraînement, elle peut être limitée dans les cas où les textes générés diffèrent considérablement des exemples utilisés pour l'évaluation, ce qui souligne l'importance d'avoir des données variées pour évaluer le modèle.

3.2.3 Méthodes en boite noire avec source modèle connue

La méthode DetectGPT proposée par E. Mitchell et al. [18] repose sur l'hypothèse initiale que les échantillons d'un modèle, tels qu'un générateur de texte, sont souvent associés à des scores de probabilité logarithmique plus bas, ce qui signifie qu'ils semblent moins naturels ou moins plausibles que les textes écrits par des humains. En utilisant une fonction de perturbation, DetectGPT génère des versions modifiées du texte tout en préservant le sens initial, puis calcule l'écart de perturbation entre le texte original et ses versions modifiées, tel que représenté dans la figurer .4. Cette méthode suppose que cet écart sera significativement plus grand pour les textes générés par un modèle que pour ceux écrits par des humains, hypothèse qui est ensuite testée empiriquement, en utilisant le dataset XSum. Elle examine la variation des probabilités associées aux textes afin de déterminer s'ils présentent des caractéristiques typiques des textes générés par des ordinateurs ou des humains, en exploitant un espace sémantique où les modifications subtiles préservent la signification. Son objectif principal est de discriminer les textes produits par des LLMs de ceux produits par des humains. Cependant, il est important de noter que cette méthode peut être limitée par l'exactitude de l'hypothèse initiale ainsi que le choix de la fonction de perturbation.

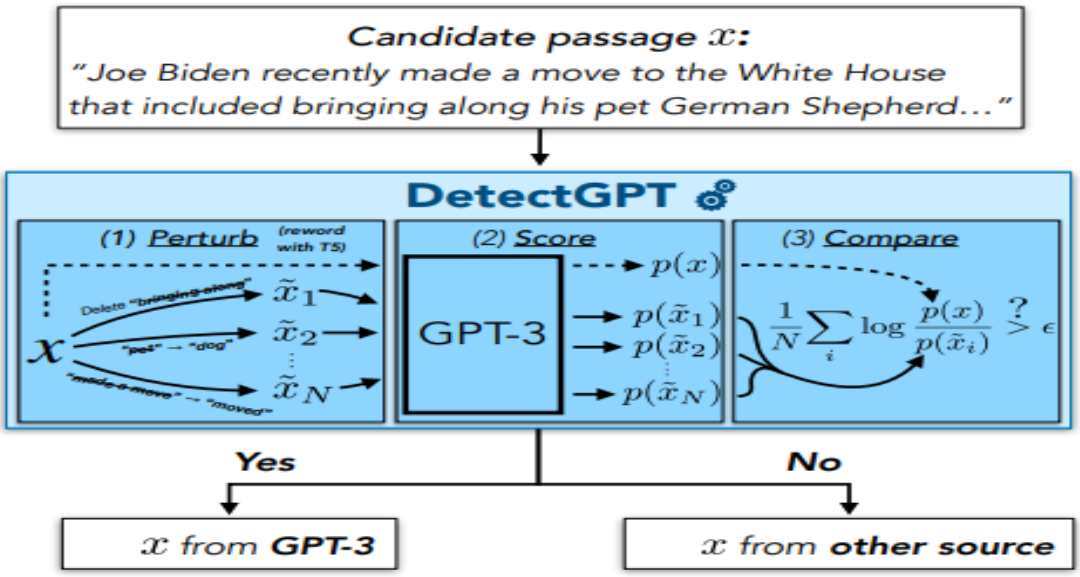


FIGURE .4 – Schéma de la méthode DetectGPT [18]

3.2.4 Méthodes en boite noire avec source du modèle inconnue

E. Tulchinski et al. [32] ont proposé une méthode de détection innovante en analysant de manière précise la dimension intrinsèque des échantillons de texte, en utilisant un ensemble de données multilingue comprenant des générations produites par GPT-3.5 ainsi que des textes naturels du même domaine, leur stratégie est fondée sur la théorie de la dimension de l'homologie persistante (PHD), un concept issu de l'analyse topologique des données. L'idée fondamentale est que les textes écrits par des humains présentent une certaine dimensionnalité intrinsèque qui diffère de celle des textes générés par l'IA. Pour cela, la méthode exploite la dimension intrinsèque de l'espace sous-jacent des ensembles d'embeddings pour un échantillon de texte donné. Il a été observé que la dimension intrinsèque moyenne des textes fluides dans une langue naturelle se situe autour de la valeur 9 pour plusieurs langues basées sur l'alphabet, et autour de 7 pour le chinois, tandis que la dimension intrinsèque moyenne des textes générés par l'IA pour chaque langue est d'environ 1,5 inférieure, avec une séparation statistique claire entre les distributions générées par les humains et celles générées par l'IA. Cette propriété permet de construire un détecteur de textes artificiels basé sur le score.

Le détecteur proposé s'appuie sur un estimateur de la dimension de l'homologie persistante (PHD) pour estimer avec précision la dimension des échantillons de texte. Cet estimateur a montré qu'il surpassait d'autres détecteurs de textes artificiels avec une marge importante dans le cadre d'utilisation général, pour une très large classe de générateurs ils ont également utiliser un algorithme efficace pour calculer la PHD, les défis liés à l'estimation de la dimension intrinsèque

en présence de bruit et d'irrégularités, et présente des expériences validant la méthode à travers différentes langues et domaines de texte.

3.3 En filigrane

Le filigrane de texte est une technique qui consiste à insérer des signes distinctifs ou des motifs cachés dans un texte. Ces signes sont conçus pour être détectables par des algorithmes spécifiques, permettant ainsi de vérifier l'origine ou l'authenticité du texte, sans altérer de manière significative sa lisibilité ou son contenu. Cette méthode trouve des applications variées, notamment pour protéger les droits d'auteur, et pour distinguer les textes générés par intelligence artificielle des textes écrits par des humains [37].

3.3.1 Méthodes en boite blanche avec tous les paramètres du modèle

Y. Fu et al. [12] présentent une méthode innovante axée sur l'intégration de signaux de détection robustes dans du texte généré par des modèles de langage. Les expériences ont été menées en utilisant différents modèles avec des tailles et des architectures différentes, notamment BART-base, BART-large, Flan-T5-small et Flan-T5-base. Ces modèles ont été évalués sur deux tâches principale, la summarization sur les jeux de données CNN/DailyMail et XSUM, ainsi que la génération de texte à partir de données sur les jeux de données DART et WebNLG. Cette approche comprend un algorithme de watermarking qui divise le vocabulaire en listes verte et rouge, permettant une sélection restreinte de mots pour la génération de texte. Il propose des variantes de hard et soft watermarks pour ajuster la distribution de probabilité, assurant une génération contrôlée et pertinente. En outre, le watermarking sémantique introduit dans cette méthode propose l'insertion de tokens sémantiquement liés dans une liste verte pour guider la génération de texte. Cette approche novatrice améliore la qualité et la pertinence du contenu généré en prenant en compte les relations sémantiques entre les tokens. De plus, elle ajuste les logits pour garantir la pertinence des mots choisis, renforçant ainsi la cohérence du texte produit. Cependant l'ajout de mots sémantiquement liés, bien qu'il puisse améliorer la qualité du contenu produit, il pourrait également compliquer la détection des watermarks. La présence de ces mots dans le texte généré rendrait plus difficile la distinction entre le texte original et généré.

3.3.2 Méthodes en boite noire avec source modèle connue

La méthode proposée par X. Yang et al. [38] est basée sur la substitution lexicale (LS) tenant compte du contexte. Elle utilise BERT pour générer des candidats LS pour un mot cible tout en considérant la similarité sémantique probabiliste avec le contexte, puis introduit un autre modèle BERT pour déduire la parenté sémantique entre les candidats et la phrase originale afin de filtrer

ceux qui ne conservent pas le sens d'origine. Des tests de "synchronicité" et "substituabilité" sont conçus pour localiser les mots aptes à porter un signal de filigrane tout en préservant le sens global. Un système de filigrane séquentiel progressif basé sur ces tests remplace progressivement les mots localisés par leurs candidats LS afin d'intégrer/extraire le message de filigrane. Cette approche vise à préserver l'intégrité sémantique sans dépendre de ressources lexicales statiques ou d'un apprentissage supervisé coûteux. Les expériences ont été menées sur des datasets de styles d'écriture variés comme des romans classiques (Les Hauts de Hurlevent, Dracula, Orgueil et Préjugés), WikiText-2, IMDB et AgNews, et les résultats surpassent les méthodes existantes en préservation sémantique et transférabilité entre styles de textes . Néanmoins, les modèles pré-entraînés comme BERT peuvent être biaisés par les données d'entraînement, ce qui peut mener à un choix de synonymes pas forcément correspondant au contexte. En outre, l'efficacité de cette méthode peut varier selon le type du texte et du domaine, ce qui soulève des questions sur sa généralisabilité.

4 Datasets utilisés

Dans cette section nous allons décrire les différents jeux de données utilisés dans les méthodes présentées dans la section précédentes.

Kaggle ML Olympiad Detect ChatGPT Answer : Les données proviennent d'un concours de KaggleML, nommé "Olympiad Detect ChatGPT Answers Kaggle", qui est une plateforme de concours de science des données, où des es ingénieurs en apprentissage automatique peuvent s'affronter pour créer les meilleurs modèles afin de résoudre des problèmes spécifiques ou d'analyser certains ensembles de données. Elles comprennent un large éventail de questions, allant des opinions aux requêtes de connaissances générales et aux faits scientifiques. Chaque question est associée à un ensemble de réponses générées à la fois par des humains et par ChatGPT, chacune étant étiquetée en conséquence.

HC3 : Le HC3 est un ensemble de données qui comprend des questions et leurs réponses associées, écrites soit par des humains, soit générées par ChatGPT. Les données humaines proviennent principalement de sets de données de questions-réponses (QA) publiques, tandis que d'autres sont extraites de Wikipedia. Les réponses générées par ChatGPT sont obtenues en saisissant les questions dans le modèle ChatGPT.

HC3 Plus : Est une extension de HC3 qui inclut non seulement des tâches de questions-réponses mais aussi des tâches de paraphrase, de résumé, et de traduction. Cela signifie que pour ces tâches, le texte généré doit rester fidèle au sens du texte source.

Wiki : Les datasets Wiki, tels que WikiText-2, WikiMatrix et Wiki40, contiennent une variété de données extraites de Wikipédia. Ils contiennent des articles, des sections, des paragraphes ou des

phrases provenant de différentes langues et sujets, comme pour le dataset Wiki40 qui est composé de plus de 40 langues.

CCNews : Le jeu de données CCNews est une collection de nouvelles provenant de diverses sources, comprenant probablement des articles de presse, des sites d'actualités en ligne.

CovidCM : Le jeu de données CovidQA regroupe des paires de questions-réponses relatives à la COVID-19, collectées à partir de 15 sites Web d'actualités en anglais sur 4 continents.

ACLabs : Le jeu de données ACLAbs comprend une sélection de données qui peuvent inclure des résumés, des titres, des auteurs, des mots-clés, des références bibliographiques etc.... issues d'articles scientifiques de l'Association for Computational Linguistics (ACL).

XSum : Le dataset XSum (extreme summarization est une collection d'articles de presse et de leurs résumés d'une phrase correspondants de la BBC. Il est utilisé pour entraîner et évaluer des modèles de résumé abstrait, qui consistent à résumer un texte donné, souvent avec un taux de compression élevé.

SQuaD : Le jeu de données SQuAD (Stanford Question Answering Dataset) est un ensemble de données de compréhension de texte à grande échelle qui comprend plus de 100 000 questions posées par des travailleurs en ligne sur un ensemble d'articles de Wikipédia. Les réponses à ces questions sont des segments de texte extraits du passage correspondant. Le dataset est conçu pour défier les machines à comprendre et répondre à des questions basées sur des passages de lecture.

Writing Prompts : Ce dataset est issu du forum WRITING-PROMPTS de Reddit, où les utilisateurs proposent des idées d'histoires et d'autres répondent avec des récits. Ce dataset offre ainsi une ressource riche pour l'entraînement de modèles de génération de récits.

PubMedQA : L'ensemble de données PubMedQA est un ensemble de données pour la recherche biomédicale qui comprend trois sous-ensembles.Le premier PQA-L qui contient des paires question-réponse annotées par deux annotateurs, le second PQA-U qui est construit à partir d'instances non annotées et le dernier PQA-A qui est utilisé pour la pré-formation en collectant des instances étiquetées de manière bruyante.

Reddit-ELI5 : Ce jeu de données est une compilation de questions et de réponses provenant du sous-forum de reddit Explain Like I'm Five (ELI5). Il met l'accent sur la clarté et la simplicité des réponses, encourageant les utilisateurs à les formuler de manière compréhensible pour un enfant de cinq ans. Cette approche vise à rendre les informations accessibles à un large public, en minimisant la dépendance à des connaissances préalables et en utilisant un langage simple et facile à comprendre.

CNN/daily : Ce dataset est utilisé pour le résumé automatique. Il contient des articles de presse et leurs résumés.

DART : Le dataset DART (Data-to-Text Generation) contient des données structurées sous forme de triplets RDF (Resource Description Framework), ainsi que leurs équivalents en texte

naturel. Ces données peuvent inclure des informations provenant de divers domaines. L'objectif principal de DART est de fournir un ensemble de données pour évaluer la capacité des modèles de génération de texte à transformer des informations structurées en texte naturel cohérent.

WebNLG : Le dataset WebNLG, quant à lui, se concentre également sur la génération de texte à partir de données structurées, mais il est spécifiquement créé pour le défi WebNLG qui consiste à convertir des triplets RDF en textes descriptifs naturels.

IMBD : IMDB est une base de données en ligne qui recueille des critiques de films, des émissions de télévision etc.. . Chaque critique est associée à une note, indiquant si elle est positive ou négative, en faisant un ensemble de données étiquetées.

AgNews : Le jeu de données AG News est un ensemble de données qui contient des extraits d'articles de presse provenant de l'Agence France-Presse (AFP) dans quatre catégories principales : "Monde", "Sport", "Affaires" et "Science/Technologie".

5 Analyse comparative

Dans cette section nous avons mené une analyse comparative des approches actuelles de détection des contenus générés par l'intelligence artificielle à travers une étude détaillée des travaux présentés dans la section précédente. Cette analyse est structurée en deux parties : d'une part, nous examinons les avantages et les inconvénients de chaque méthode pour évaluer leur pertinence dans divers contextes d'application. D'autre part, nous proposons une comparaison des caractéristiques et des performances des méthodes étudiées, mettant en évidence leurs spécificités et leurs domaines d'efficacité.

5.1 Avantages et inconvénients

Nous avons élaboré un tableau qui présente quelques avantages et inconvénients de chaque méthode analysée.

Méthode	**Avantages**	**Inconvénients**
AI-GENERATED TEXT CLASSIFIER [27] (2023)	— Ne nécessite pas d'entraînement. — Interprète les résultats.	— Complexité du déploiement. — Faux positifs et négatifs.
Le modèle de V. Vora et al.[35] (2023)	— Haute flexibilité. — Utilisation des graphes de connaissances.	— Dépendance aux Données de qualité. — Risque de surajustement.
GPT Paternity Test (GPT-Pat)[39] (2024)	— Robustesse aux attaques adaptatives. — Généralisabilité.	— Dépendance aux services de ChatGPT. — Complexité de mise en œuvre.
Ghobuster[34] (2023)	— Indépendance par rapport au modèle cible. — Généralisation.	— Dépendance aux données d'entraînement. — Détection de texte court

TABLEAU .2 – Avantages et inconvénients des différentes méthodes (Partie 1)

Méthode	**Avantages**	**Inconvénients**
La méthode de J. Su et al.[29] (2023)	— intuition pratique.	— Complexité dans l'implémentation. — Dépendance à l'accès complet au modèle LLM.
DNA-GPT [36] (2023)	— Ne nécessite pas d'entraînement. — Interprète les résultats.	— Dépendance à l'hypothèse de l'écart de vraisemblance. — Possibilité de faux positifs.
DetectGPT [18] (2023)	— Ne nécessite pas d'accès à des échantillons écrits par des humains. — Généralisation à de nouveaux domaines.	— Besoin de données complètes. — Complexité.
La méthode de E. Tulchinski et al. [32] (2023)	— Stabilité face au bruit. — Efficacité en termes d'échantillonnage.	— Dépendances à des paramètres spécifiques. — Complexité de calcul.
La méthode de Y. Fu et al. [12](2024)	— Amélioration du contenu généré. — Adaptabilité aux tâches de génération de texte conditionnel.	— Complexité accrue. — Défi de détection.
La méthode de X. Yang et al. [38] (2023)	— Robustesse et traçabilité. — Facilité d'adaptation.	— Complexité de mise en œuvre. — Limitation de généralisation.

Tableau .3 – Avantages et inconvénients des différentes méthodes

5.2 Comparaison des approches de détection étudiées

Nous avons réalisé une comparaison des différentes approches de détection par NLP des contenus produits par l'intelligence artificielle générative, en nous basant sur les travaux exposés dans la section précédente. Le tableau .3 résume les résultats de cette étude. Il est important de comprendre comment chaque modèle performe à travers plusieurs métriques telles que l'exactitude, la précision, la robustesse, et la diversité linguistique.

L'exactitude et la précision sont des indicateurs clés de la capacité d'un modèle à identifier correctement le contenu généré par IA. Par exemple, le modèle GPT-Pat [39] se distingue par une excellente exactitude, ce qui témoigne de sa haute précision et de sa capacité à détecter correctement les textes générés par IA.

La robustesse, qui évalue la résistance d'un modèle face aux tentatives de contournement comme la paraphrase, est une autre métrique importante. GPT-Pat [39] a démontré une robustesse significative. De même la méthode de E. Tulchinski [32] se révèle également très robuste, en plus de sa capacité à traiter jusqu'à dix langues.

La sélection d'une méthode de détection dépend fortement du contexte spécifique d'application et des exigences en matière de performance. Certains modèles tels que Gpt-Pat [39] et DNA-GPT [36] sont préférables dans des environnements où la précision et la robustesse sont prioritaires, tandis que la méthode de E. Tulchinski et al. [32] est plus adaptés pour des applications nécessitant une grande diversité linguistique et DetectLLM [29] pour une flexibilité dans le traitement de divers types de textes. Ces distinctions soulignent l'importance de bien choisir la méthode en fonction de l'environnement et des objectifs spécifiques, tout en tenant compte des forces et des limites de chaque approche.

Approche	Scénario	Dataset	Accuracy	Précision	Multi-lingue	Robustesse
AI-GENERATED TEXT CLASSIFIER [27] (2023)	Basé apprentissage Boite blanche-Tous les paramètres	Collecte manuelle de données	89.95%	-	Non	Peu robuste
Le modèle de V. Vora et al.[35](2023)	Basé apprentissage Boite blanche-informations partielles	Kaggle ML Olympiad Detect ChatGPT Answer	90.10%	-	Non	Robuste
GPT Paternity Test (GPT-Pat)[39] (2023)	Basé apprentissage Boite noire-Source connue	HC3 - Wiki CCNews CovidCM ACLabs	99.89% 95.32% 93.37% 96.76% 89.83%	99.84% 93.48% 96.70% 99.03% 100%	Non	Très robuste
Ghostbuster [34] (2023)	Basé apprentissage Boite noire-Source inconnue	Collecte manuelle	90.03%	90.03%	Non	Très robuste
DetectLLM de J. Su et al. [29](2023)	Zero shot Boite blanche avec tous les paramètres	XSum SQuaD Writing-Prompts	-	-	Non	Très robuste
DNA-GPT [36](2023)	Zero shot Boite blanche-informations partielles	PubMedQA Reddit-ELI5	90.67% 99.50%	-	Oui (2)	Modérément robuste
DetectGPT [18] (2023)	Zero shot Boite noire source connue	XSum SQuAD Writing-Prompts	97% 98% 98%	-	Oui (2)	Modérément robuste
La méthode de E. Tulchinski et al. [32] (2023)	Zero shot Boite noire Source inconnue	Wiki40b WikiM	73.1%	-	Oui (10)	Très robuste
La méthode de Y. Fu et al. [12](2024)	Filigrane Boite Blanche avec tous les paramètres	CNN/daily XSum DART WebNLG	-	-	Non	Peu robuste
La méthode de X. Yang et al. [38] (2023)	Filigrane Boite noire avec source connue	WikiText IMBD AgNews	-	-	Non	Peu robuste

Tableau .4 – Tableau comparatif des méthodes étudiées.

Chapitre

Chapitre 3 : Méthode de Détection Proposée

1 Introduction

Dans le domaine en rapide évolution du traitement du langage naturel, la distinction entre les textes générés par les humains et ceux produits par l'intelligence artificielle devient un enjeu crucial. Cette évolution pose des défis importants en matière d'authenticité de l'information, de sécurité et de légitimité des contenus, alimentant ainsi la nécessité de développer des systèmes de détection efficaces et fiables.

Dans ce chapitre, nous aborderons notre méthode pour détecter le contenu généré par l'IA en utilisant le traitement automatique du langage naturel, nous présenterons les différentes étapes de notre méthodologie, depuis l'acquisition des données jusqu'à l'évaluation des performances du modèle.

2 Modèle utilisé

Notre choix s'est porté sur le modèle BERT qui représente une évolution importante dans le domaine du traitement du langage naturel, offrant une approche bidirectionnelle pour la compréhension du langage qui capture les relations contextuelles entre les mots dans les deux sens. Cette caractéristique fondamentale permet à BERT de saisir les nuances du langage, fournissant ainsi une représentation plus riche et précise. Nous avons donc opté pour le modèle DistilBERT pour notre projet de détection de contenus textes générés par l'IA, qui représente une version allégée et plus rapide de BERT. Ce choix est justifié par plusieurs raisons. DistilBERT conserve les caractéristiques essentielles de BERT tout en répondant mieux aux contraintes de ressources computationnelles et de temps d'entraînement de notre projet. Offrant ainsi une solution efficace pour notre tâche de détection.

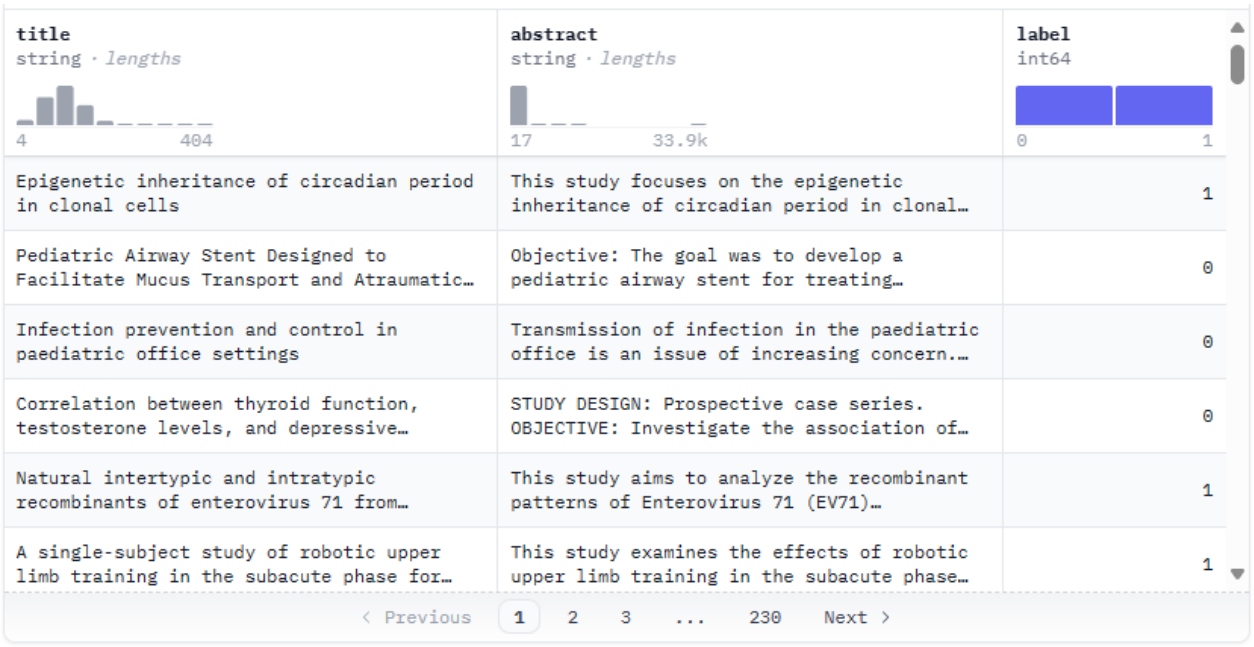
Une des raisons principales pour lesquelles notre choix s'est porté sur BERT est son efficacité éprouvée dans la compréhension et la génération de langage naturel. Ce modèle utilise le principe de l'attention masquée (masked attention) lors de son entraînement, ce qui lui permet de comprendre le contexte global et d'apprendre des représentations riches et précises pour chaque mot. Sa capacité à capturer les relations contextuelles complexes entre les mots en fait un outil puissant pour détecter et comprendre le langage généré par des systèmes d'IA. En outre, la disponibilité de modèles pré-entraînés et la facilité de fine-tuning pour des tâches spécifiques permettent une mise en œuvre efficace et rapide de solutions de NLP. En intégrant BERT dans notre processus de traitement, nous visons à améliorer la précision et la robustesse de notre système de détection de contenu généré par l'IA, offrant ainsi une protection contre les manipulations et les abus de langage automatisés.

Ces caractéristiques font de BERT un choix stratégique pour notre projet, où la compréhension fine et nuancée du langage est essentielle pour identifier et contrer les contenus générés de manière artificielle, contribuant ainsi à maintenir l'intégrité et la fiabilité des informations en ligne. [22].

3 Dataset employé

Nous avons utilisé le dataset AI-GA [26] pour entraîner notre modèle de détection de contenus générés par l'IA. Ce jeu de données, également connu sous le nom de "Artificial Intelligence Generated Abstracts" (Résumés Générés par Intelligence Artificielle), se compose de résumés et de titres. La moitié de ces résumés sont générés par une IA, tandis que l'autre moitié est originale. Il est largement utilisé dans la recherche en traitement automatique du langage naturel, offrant ainsi de nombreuses possibilités pour l'analyse et l'exploration.

Ce dataset comprend 28 662 échantillons, chacun contenant un résumé, un titre et une étiquette qui permet de distinguer un résumé original (étiqueté 0) d'un résumé généré par IA (étiqueté 1). Ce regroupement englobe à la fois des textes rédigés par des humains et des textes générés par un modèle de langage (LLM) utilisant GPT-4 et BARD couvrant divers genres tels que des essais, des histoires, de la poésie et du code Python. Il constitue une ressource précieuse pour l'investigation des méthodologies de détection de texte LLM, la figure.1 ci-dessous présente une visualisation du jeu de données AI-GA :



title string · lengths 4 404	abstract string · lengths 17 33.9k	label int64 0 1
Epigenetic inheritance of circadian period in clonal cells	This study focuses on the epigenetic inheritance of circadian period in clonal…	1
Pediatric Airway Stent Designed to Facilitate Mucus Transport and Atraumatic…	Objective: The goal was to develop a pediatric airway stent for treating…	0
Infection prevention and control in paediatric office settings	Transmission of infection in the paediatric office is an issue of increasing concern.…	0
Correlation between thyroid function, testosterone levels, and depressive…	STUDY DESIGN: Prospective case series. OBJECTIVE: Investigate the association of…	0
Natural intertypic and intratypic recombinants of enterovirus 71 from…	This study aims to analyze the recombinant patterns of Enterovirus 71 (EV71)…	1
A single-subject study of robotic upper limb training in the subacute phase for…	This study examines the effects of robotic upper limb training in the subacute phase…	1

< Previous 1 2 3 ... 230 Next >

FIGURE .1 – Visualisation du jeu de données AI-GA après encodage.

4 Approche proposée

Dans cette section, nous présentons notre approche basée sur le traitement du langage naturel utilisant le modèle DistilBERT, visant à identifier les textes générés par l'IA. Pour expliquer comment ce modèle peut être utilisé pour cette tache de classification, il est important de comprendre comment le modèle BERT comprend le langage naturel et quelles caractéristiques il prend en compte.

Pour détecter les contenus générés par l'IA, le modèle DistilBERT tire parti de diverses caractéristiques linguistiques et stylistiques. Il identifie des modèles stylométriques et des structures linguistiques atypiques souvent présentes dans les textes issus de générateurs de langue. Ces textes peuvent manifester des anomalies stylistiques, comme des répétitions inhabituelles ou des constructions syntaxiques non naturelles, que le modèle apprend à reconnaître. En plus, il analyse la fréquence des tokens, repérant des mots ou des phrases qui apparaissent de façon anormalement fréquente ou rare, un indicateur potentiel de contenu généré par machine. L'organigramme.2 ci-dessous illustre toutes les étapes de notre approche.

Le modèle examine également les séquences de tokens pour détecter des schémas répétitifs ou des transitions brusques, des traits courants dans les écrits produits par IA, mais rares dans les textes humains. Grâce à son architecture de transformateur et ses mécanismes d'attention, DistilBERT capture efficacement ces modèles complexes en tenant compte du contexte global de chaque mot.

Pour spécialiser DistilBERT dans la détection de contenu IA, nous avons effectué un fine-

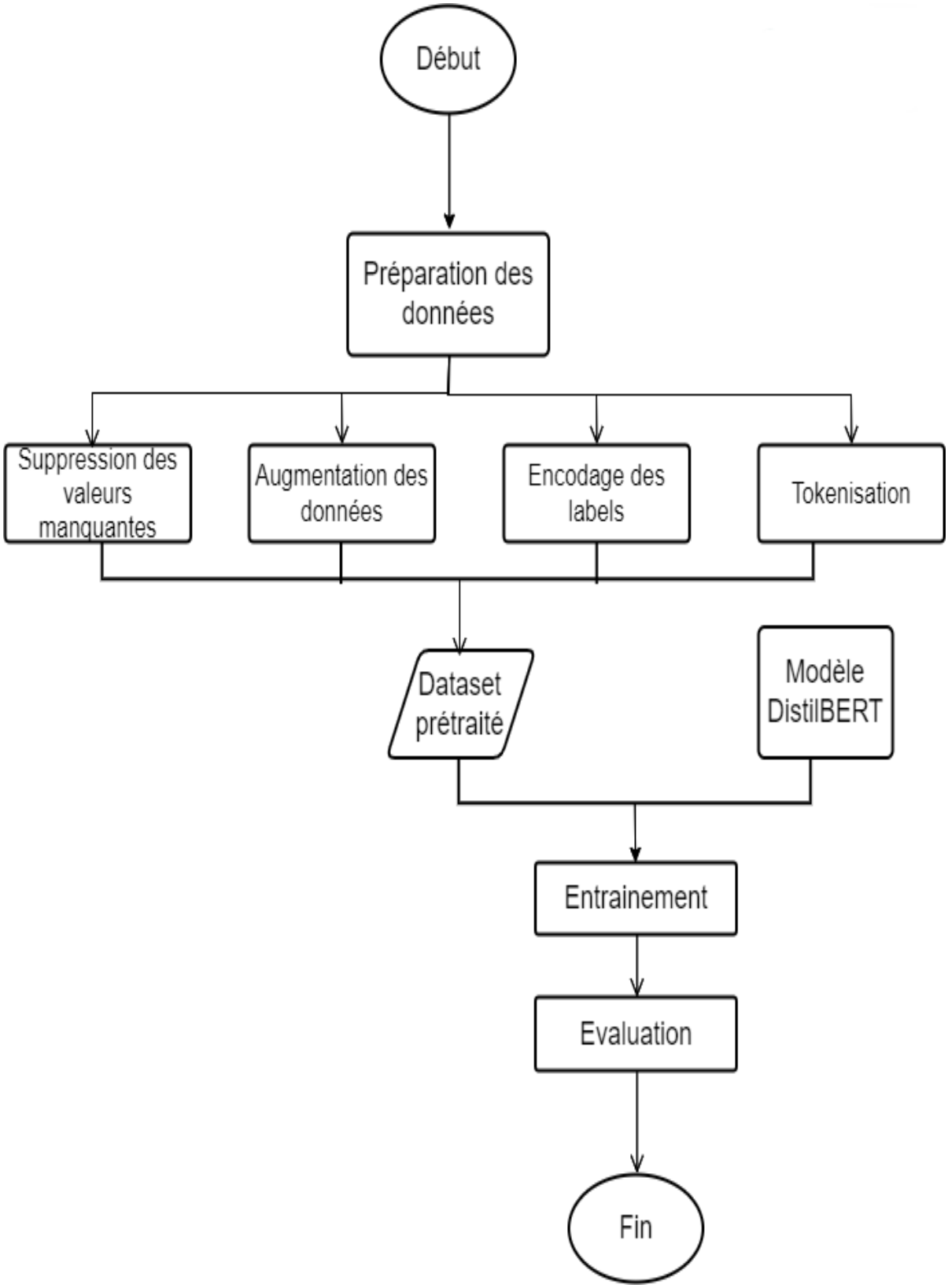


FIGURE .2 – Diagramme de l'approche proposée.

tuning des poids du modèle. Cette étape d'ajustement utilise des données d'entraînement étiquetées, distinguant les textes générés par IA de ceux rédigés par des humains, ce qui permet d'affiner la capacité du modèle à discerner les subtilités entre ces deux types de textes. Cette approche renforce la compréhension du langage par le modèle et améliore sa capacité à discriminer les textes, augmentant ainsi significativement sa performance de classification.

4.1 Préparation des données

C'est une étape essentielle qui prépare les données de manière à ce qu'elles soient adaptées à l'entraînement du modèle.

a- Suppression des valeurs manquantes : Nous avons procédé à l'élimination de toutes les observations de l'ensemble de données qui contiennent au moins une valeur manquante, afin d'assurer que seules les données complètes et utilisables sont utilisées pour le traitement.

b- Augmentation des données : utilisée pour enrichir artificiellement un jeu de données d'entraînement en introduisant des variations et des exemples supplémentaires. Cela inclut diverses techniques, pour notre part, nous avons effectué une synonymisation afin de créer des variantes de phrases avec des mots similaires, ce qui permet d'améliorer la performance et la généralisation des modèles.

c- Tokénisation : Nous avons séparé un texte en segments plus petits et significatifs. La division se fait en utilisant des marqueurs comme les espaces, la ponctuation et d'autres caractères délimitants pour découper le texte en tokens distincts. Chaque segment est généralement un mot. Cette étape a pour rôle de simplifier l'analyse et le traitement des textes par les algorithmes de NLP.

d- Encodage : Ce processus consiste à attribuer une valeur numérique spécifique à chaque catégorie ou label afin de permettre aux algorithmes de NLP de traiter les données textuelles de manière effective. Dans notre cas, le label "Human" est remplacé par "0" et le label "AI" par "1".

4.2 Entraînement du modèle

L'entraînement du modèle commence par l'initialisation de l'optimiseur AdamW, chargé d'ajuster les poids du modèle pendant l'entraînement. Il utilise des gradients pour mettre à jour les poids de manière itérative. Un gradient est un vecteur qui indique la direction et l'intensité du changement nécessaire pour minimiser la fonction de perte du modèle. Le taux d'apprentissage contrôle l'ampleur de ces ajustements à chaque étape, déterminant ainsi la vitesse à laquelle le modèle apprend.

AdamW accumule les informations des gradients passés, ce qui permet de stabiliser et d'accélérer l'apprentissage en utilisant une moyenne des gradients plutôt que de se baser uniquement sur les gradients actuels. Il réduit également l'impact des poids du modèle progressivement au cours de l'entraînement, ce qui aide à régulariser le modèle et à prévenir le surajustement.

Nous avons utilisé GridSearch pour déterminer les valeurs idéales des hyperparamètres avant d'entraîner le modèle. Une boucle d'entraînement est ensuite exécutée sur 2 époques, chaque époque consistant en un passage complet à travers l'ensemble d'entraînement, divisé en lots de taille 16. Durant chaque époque, le modèle est entraîné sur les données d'entraînement, puis évalué

sur l'ensemble de validation pour évaluer ses performances. GridSearch a permis d'optimiser les hyperparamètres en testant systématiquement différentes combinaisons et en sélectionnant celles qui donnent les meilleurs résultats de validation.

Les résultats de perte et d'exactitude sont enregistrés à chaque époque pour surveiller la performance du modèle au fil du temps. Ce processus d'entraînement, incluant le fine-tuning des poids du modèle, est essentiel pour améliorer sa capacité à différencier les textes rédigés par des humains de ceux générés par une IA.

4.3 Outils de développement

Pour mettre en place notre système, nous avons utilisé plusieurs outils, dont le langage Python sur Google Colab et la plateforme Hugging Face. Cette sous-section explique les caractéristiques de ces plateformes et donne une définition du langage Python.

Hugging Face : Hugging Face est une entreprise et plateforme en ligne spécialisée dans le traitement du langage naturel. Connue pour son modèle de langage Transformers, elle propose une bibliothèque open-source permettant d'intégrer facilement des modèles dans les applications. La plateforme offre des ressources variées, dont des modèles pré-entraînés, des tutoriels, des datasets, et un espace de collaboration pour des projets de ML [2].

Google Colaboratory : Google Colaboratory est un framework en ligne développé par Google qui permet d'écrire et d'exécuter des codes d'apprentissage automatique et profond. Il offre différentes versions de Python ainsi que différents environnements d'exécution. De plus, il permet de télécharger des ensembles de données volumineux directement depuis les serveurs vers Google Drive à très grande vitesse [1].

Langage Python : Python est un langage de programmation polyvalent souvent utilisé pour créer des modèles d'intelligence artificielle en raison de sa syntaxe claire et de sa vaste gamme de packages. Pour le développement de notre modèle , nous avons utilisé plusieurs packages et bibliothèques [9], notamment :

- **PyTorch :** Une bibliothèque open-source de machine learning largement utilisée pour la création et le déploiement de modèles neuronaux [8].
- **Scikit-learn (sklearn) :** Une bibliothèque d'apprentissage automatique simple et efficace pour la classification, la régression et le clustering [8].
- **Transformers :** Une bibliothèque pour la création et l'utilisation de modèles de transformation basés sur des réseaux de neurones [15].
- **ML_Things :** Une bibliothèque qui fournit des outils et des utilitaires pour le développement de modèles de machine learning[13].
- **NumPy :** Une bibliothèque fondamentale pour le calcul numérique en Python, offrant un support pour les tableaux multidimensionnels et les opérations mathématiques [8].

5 Conclusion

En conclusion, ce chapitre a détaillé le processus complet de développement d'un modèle de classification binaire basé sur l'architecture DistilBERT pour détecter les textes générés par l'IA. Nous avons commencé par décrire le modèle utilisé et le dataset employé, suivi par une explication de notre approche méthodologique incluant le choix et le pré-traitement des données. Ensuite, nous avons expliqué l'entraînement du modèle en utilisant des techniques avancées et des outils de développement spécifiques.

Chapitre

Chapitre 4 : Résultats et Evaluation

1 Introduction

Ce chapitre présente une analyse détaillée des résultats obtenus par notre modèle de détection de contenu généré par l'IA. Nous explorons diverses métriques d'évaluation qui reflètent l'efficacité et la précision de notre approche, fournissant ainsi une évaluation complète de la performance du modèle.

2 Défis rencontrés

Le développement de notre méthode de détection de contenu généré par l'IA n'a pas été sans obstacles. Voici quelques-uns des principaux défis auxquels nous avons été confrontés :

- Trouver un dataset approprié s'est avéré être l'un des défis majeurs. Étant donné la nouveauté de cette problématique, il existe peu de jeux de données étiquetés disponibles publiquement pour distinguer clairement entre les textes générés par des humains et ceux générés par des IA.
- Les ressources requises pour entraîner des modèles de traitement du langage naturel, en particulier ceux basés sur des architectures complexes comme BERT, sont considérables. Nous avons rencontré des limitations en termes de GPU et de RAM, ce qui a ralenti notre processus d'entraînement et a nécessité l'optimisation des ressources disponibles.
- Un autre défi significatif a été le déséquilibre des données sur un dataset testé précédemment, qui n'était pas utilisé dans notre approche finale. Ce dataset initial n'était pas équilibré, avec une surreprésentation des textes générés par l'IA par rapport aux textes écrits par des humains, cela a posé des problèmes de sur-apprentissage.
- L'ajustement des hyperparamètres du modèle DistilBERT a été une étape critique mais délicate. Trouver les valeurs optimales pour le taux d'apprentissage, la taille des lots et le

nombre d'époques a demandé plusieurs itérations et une évaluation rigoureuse pour éviter le sur-apprentissage tout en maximisant la précision.

3 Métriques d'évaluation

Cette section détaille les différentes métriques utilisées pour évaluer la performance de notre modèle. Ces indicateurs sont essentiels pour comprendre comment le modèle performe en termes d'exactitude, de précision, de rappel, et de F1-Score [11].

3.1 Exactitude (Accuracy)

Elle indique la proportion totale de prédictions correctes par rapport au total des cas testés. La formule pour calculer l'exactitude est :

$$\text{Exactitude} = \frac{\text{Nombre de prédictions correctes}}{\text{Nombre total de prédictions}}$$

Cette métrique donne une vue d'ensemble de l'efficacité du modèle.

3.2 Précision

Elle représente la proportion de prédictions positives qui sont correctement identifiées. C'est une mesure essentielle pour évaluer la qualité des prédictions positives du modèle, exprimée par :

$$\text{Precision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}}$$

3.3 Rappel

Il mesure la capacité du modèle à identifier tous les cas pertinents dans le dataset. Ce critère est crucial pour s'assurer que le modèle capture autant de cas positifs que possible, et est défini par :

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}}$$

3.4 F1-Score

Cette métrique combine la précision et le rappel en une seule mesure harmonique, très utile pour équilibrer ces deux aspects. Elle est calculée comme suit :

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

3.5 Matrice de confusion

La matrice de confusion est une table qui permet de visualiser les performances d'un modèle de classification. Elle présente un résumé des prédictions du modèle par rapport aux valeurs réelles dans un format tabulaire, comme suit :

		Classe Prédite	
		Positif (+)	Négatif (-)
Classe Réelle	Positif (+)	Vrais Positifs (VP)	Faux Positifs (FP)
	Négatif (-)	Faux Négatifs (FN)	Vrais Négatifs (VN)

- **Vrais Positifs (VP)** : Nombre d'échantillons de la classe positive correctement prédits comme positifs par le modèle.
- **Faux Négatifs (FN)** : Nombre d'échantillons de la classe positive incorrectement prédits comme négatifs par le modèle.
- **Faux Positifs (FP)** : Nombre d'échantillons de la classe négative (ou d'une autre classe) incorrectement prédits comme positifs par le modèle.
- **Vrais Négatifs (VN)** : Nombre d'échantillons de la classe négative correctement prédits comme négatifs par le modèle.

3.6 Moyenne arithmétique (Macro Average)

Est une méthode qui calcule d'abord la métrique pour chaque classe indépendamment, puis prend la moyenne arithmétique de ces résultats. Elle est particulièrement utile pour s'assurer que les classes moins fréquentes sont traitées équitablement. La formule générale pour le macro average est donnée par :

$$\text{Macro Average} = \frac{\sum_{i=1}^{N} \text{Metric}_i}{N}$$

où Metric_i est la métrique calculée pour la classe i et N est le nombre total de classes.

3.7 Moyenne pondérée (Weighted Average)

Cette méthode multiplie la métrique de chaque classe par le nombre d'échantillons dans cette classe avant de sommer les résultats et de diviser par le total des échantillons. Elle est utile lorsque certaines classes sont plus significatives du fait de leur prévalence. La formule pour le weighted average est :

$$\text{Weighted Average} = \frac{\sum_{i=1}^{N} n_i \times \text{Metric}_i}{\sum_{i=1}^{N} n_i}$$

où n_i est le nombre d'échantillons dans la classe i, et Metric_i est la métrique calculée pour cette même classe i.

4 Résultats et discussion

Cette section présente les résultats obtenus à travers les métriques définies ci-dessus, analysant la manière dont notre modèle se comporte face à divers scénarios et configurations de test.

4.1 Détails de la perte

Ce graphique .2 montre la courbe de perte pour les phases d'entraînement et de validation. La perte décroît rapidement dès les premiers epochs, ce qui est un indicateur positif de l'efficacité de l'apprentissage. La courbe de perte de validation diminue également puis remonte au bout du deuxième epoch, ce qui indique probablement un surapprentissage.

–

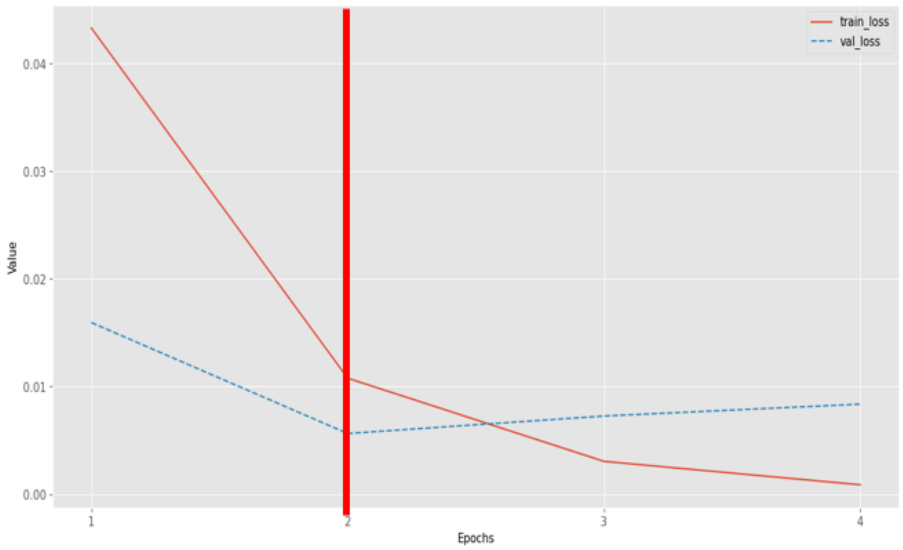


FIGURE .1 – Réduction de la perte pendant les phases d'entraînement et de validation avant l'arrêt anticipé.

Pour y remédier à cela, nous avons effectué un arrêt anticipé de l'entraînement. La courbe suivante représente les résultats obtenus.

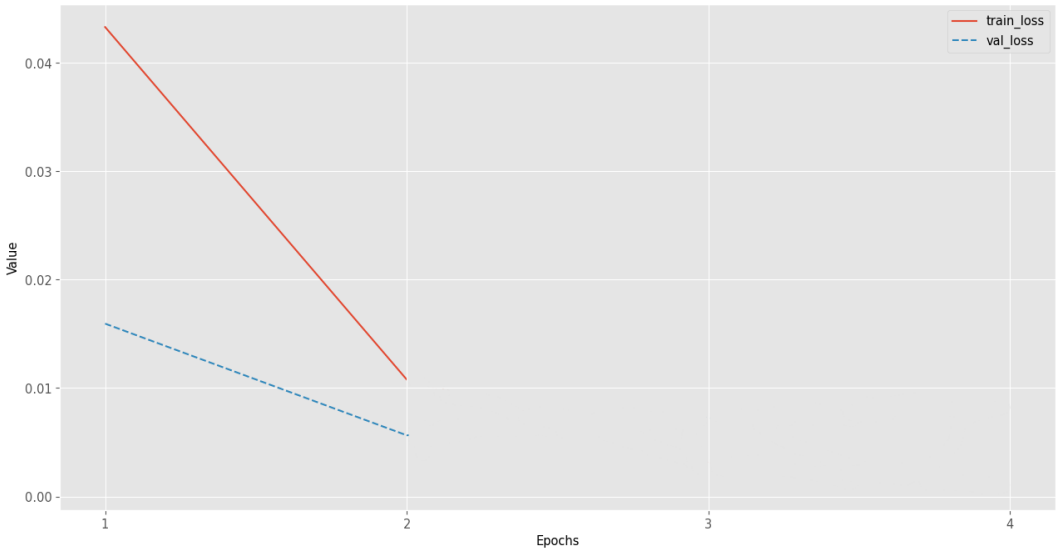


FIGURE .2 – Réduction de la perte pendant les phases d'entraînement et de validation après l'arrêt anticipé.

4.2 Évolution de l'accuracy

Ce graphique .3 illustre l'évolution de l'accuracy pendant les deux epochs d'entraînement et de validation. On observe une progression constante de l'accuracy pour les données d'entraînement, indiquant une bonne capacité d'apprentissage du modèle. L'accuracy de validation, bien qu'elle commence plus bas, rejoint presque celle de l'entraînement.

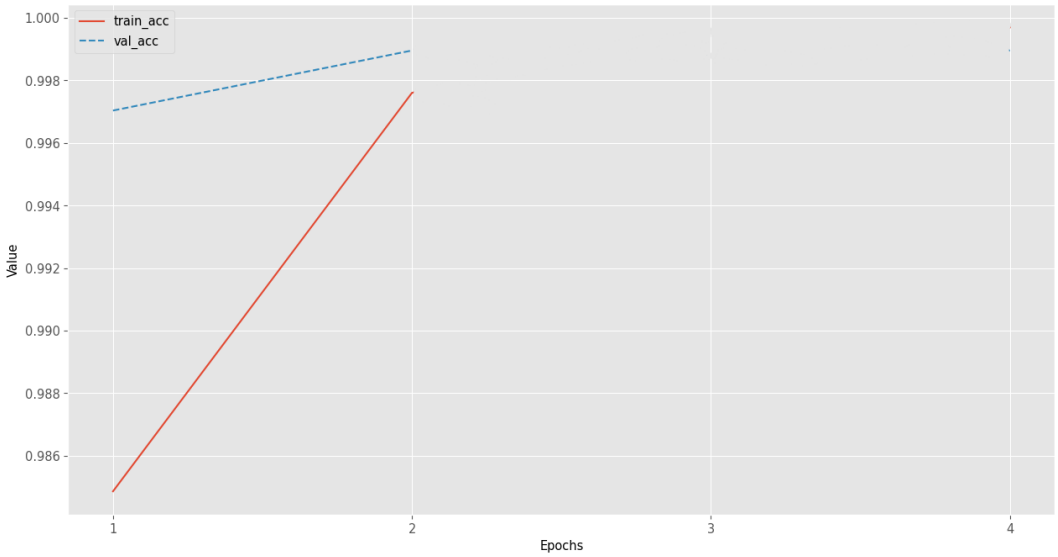


FIGURE .3 – Évolution de l'accuracy pendant l'entraînement et la validation.

4.3 Tableau de résultats globaux

Ce tableau.4 démontre les métriques de précision, rappel et F1-score pour les classes 'Humain' et 'IA' et les scores de Macro Average et Weighted Average, offrant une vue d'ensemble de la performance globale du modèle. Le Macro Average reflète l'équité du modèle dans le traitement de chaque classe, tandis que le Weighted Average illustre comment la prévalence de chaque classe influence la performance globale du modèle.

	precision	recall	f1-score	support
Human	1.00	0.93	0.96	42
AI	0.95	1.00	0.97	58
accuracy			0.97	100
macro avg	0.98	0.96	0.97	100
weighted avg	0.97	0.97	0.97	100

FIGURE .4 – Précision, rappel, et F1-score pour les catégories 'Humain' et 'IA'.

4.4 Matrice de confusion

La figure.5 ci-dessous présente une matrice de confusion normalisée illustrant les performances de notre modèle de classification entre les échantillons générés par des humains et ceux générés par une intelligence artificielle. Les valeurs dans la matrice sont des proportions, ce qui permet une évaluation plus intuitive des performances.

Le modèle affiche une précision remarquable avec 93% des échantillons humains correctement identifiés, tandis que 7% n'ont pas été classés comme générés par une IA. De plus, le modèle montre une excellence dans la classification des échantillons IA, avec un taux parfait de bonnes prédictions et aucune confusion avec les échantillons humains.

Globalement, ces résultats démontrent l'efficacité du modèle à distinguer entre les textes humains et ceux générés par une intelligence artificielle, avec une marge d'erreur minimale dans la classification des échantillons humains.

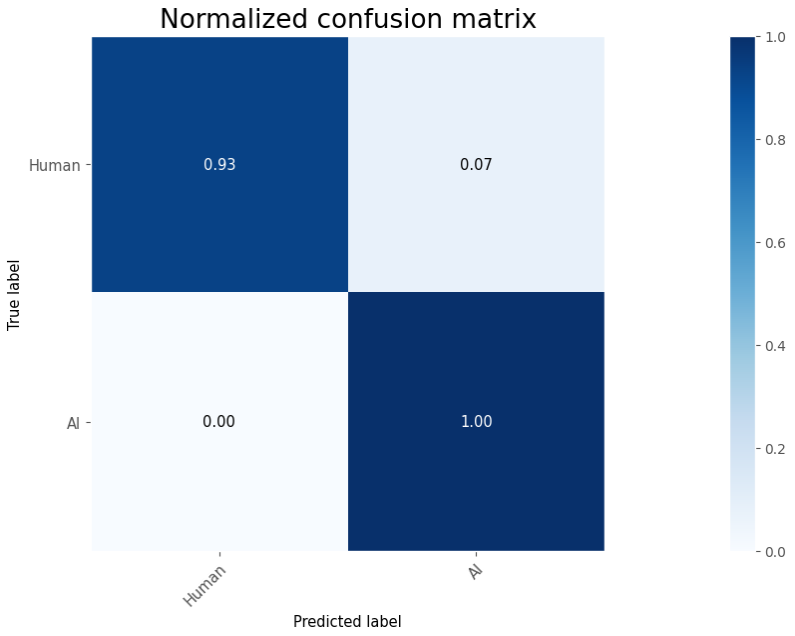


FIGURE .5 – Matrice de confusion.

5 Discussion des résultats

Dans ce projet, l'un des défis les plus importants était d'évaluer l'exactitude de notre modèle par rapport aux standards du marché existants, en le comparant avec certains classificateurs. Notre objectif était d'atteindre des niveaux de précision supérieurs à ceux de ces outils de référence. Malgré la limitation d'un ensemble de données plus petit par rapport à l'entraînement à l'échelle

industrielle, nous croyons fermement que notre prototype a le potentiel d'être développé et déployé à plus grande échelle pour une utilisation industrielle.

Notre méthode affiche des résultats remarquablement élevés avec une précision globale de 97%, surpassant plusieurs méthodes vues dans l'état de l'art. La précision est très élevée pour les textes IA (0.95) montrant qu'il y a très peu de faux positifs. Le rappel pour les textes humains est également élevé (0.93), ce qui signifie que la majorité des textes humains sont correctement identifiés, bien qu'il y ait quelques faux négatifs. Les scores F1 de 0.96 pour les textes humains et de 0.97 pour les textes IA indiquent un excellent équilibre entre précision et rappel.

6 Conclusion

Ces résultats sont significatifs pour le développement de modèles de classification de texte plus fiables et précis, pouvant être utilisés dans diverses applications telles que la détection de fausses informations, l'identification de messages indésirables, l'évaluation de la fiabilité de l'écriture scientifique et l'amélioration globale du traitement du langage naturel. Pour conclure, ce chapitre a évalué notre modèle de détection de contenu généré par l'IA à l'aide de diverses métriques importantes. Les résultats montrent globalement une bonne performance du modèle. Toutefois, des améliorations sont encore nécessaires pour optimiser sa capacité de généralisation et son efficacité.

Conclusion et perspectives

Ce mémoire a traité la problématique de la détection des contenus générés par l'intelligence artificielle, une question importante dans notre ère d'internet où la diffusion de fausses informations est fréquente.

Les méthodes décrites dans l'état de l'art ont souvent utilisé des grands modèles basés sur de différentes architectures pour la détection de contenus générés par l'IA. Ces approches ont combiné des techniques telles que l'apprentissage par transfert, l'augmentation des données et l'assemblage de modèles.

Dans ce contexte, notre contribution s'est basée sur l'utilisation d'un modèle léger, DistilBERT, pour le traitement du langage naturel. Nous avons développé une méthode utilisant ce modèle optimisé qui a montré de bons résultats dans la détection de textes générés par des machines. Contrairement aux grands modèles souvent utilisés dans les travaux précédents, notre approche a visé à offrir une solution plus accessible et moins gourmande en ressources, tout en maintenant des performances compétitives. Cependant, il est important de souligner que nos résultats proviennent d'un ensemble de données relativement restreint, principalement constitué de résumés d'articles de différents domaines.

En revanche, des travaux futurs sont nécessaires pour explorer davantage les possibilités d'amélioration. Il serait notamment bénéfique d'augmenter la taille du jeu de données pour inclure une variété plus large de textes et de générer des contenus à partir de différents LLM, afin d'assurer que notre modèle puisse détecter les contenus générés par une gamme plus large de grands modèles de langage et non seulement ceux produits par un seul type de LLMs. Il serait également très intéressant de se concentrer sur les contenus académiques tels que les mémoires, les articles et les thèses de doctorat, afin d'améliorer notre système pour garantir l'authenticité des travaux académiques soumis.

Bibliographie

[1] Google colab. `https://colab.research.google.com/`. [En ligne ; consulté le 26-juin-2024].

[2] Hugging face. `https://huggingface.co/`. [En ligne ; consulté le 26-juin-2024].

[3] *Artificial Intelligence*. CRC Press Taylor and Francis Group, Boca Raton, FL, version date : 20180209 edition, 2018.

[4] F. Bouchebbah. Chapitre 1 : Rappels sur l'analyse de données et le machine learning. Cours destiné aux étudiants de Master 2 en Intelligence Artificielle, Faculté de Sciences Exactes, Département d'Informatique, Université A. Mira de Béjaia, 2023. Version 1.

[5] F. Bouchebbah. Chapitre 4 : Architectures de deep learning. Cours destiné aux étudiants de Master 2 en Intelligence Artificielle, Faculté de Sciences Exactes, Département d'Informatique, Université A. Mira de Béjaia, 2023. Version 1.

[6] Azencott Chloé-Agathe. *Introduction au Machine Learning*. InfoSup. Dunod, 2018.

[7] Giuseppe Ciaburro and Balaji Venkateswaran. *Neural Networks with R : Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. Packt Publishing Ltd, 2017.

[8] Databird. Bibliothèque python. `https://www.data-bird.co/blog/bibliotheque-python`, 2024. [En ligne ; consulté le 26-juin-2024].

[9] Databird. Python : Un langage polyvalent. `https://www.data-bird.co/blog/langage-python#blog-body`, 2024. [En ligne ; consulté le 26-juin-2024].

[10] Axel de Goursac. Le natural langage processing, 2017. Copyright Myriad Data 2017.

[11] Nadim Elsakaan and Amroun Kamal. A comparative study of machine learning binary classification methods for botnet detection. In *Proceedings of the 2021 ACS International Conference on Computer Systems and Applications (AICCSA)*, chapter 3. Springer, 2022.

[12] Y. Fu, D. Xiong, and Y. Dong. Watermarking conditional text generation for ai detection : Unveiling challenges and a semantic-aware watermark remedy. 2024.

[13] Gmihaila. Ml things repository. `https://github.com/gmihaila/ml_things`, 2024. [En ligne ; consulté le 26-juin-2024].

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11) :139–144, 2020.

[15] Hugging Face. Transformers. `https://github.com/huggingface/transformers`, 2024. [En ligne ; consulté le 26-juin-2024].

[16] Daniel Jurafsky and James H. Martin. Transformers and large language models. *Speech and Language Processing*, 2023. Draft of February 3, 2024.

[17] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan : Training gans with vision transformers. 2021.

[18] E. Mitchell, Y. Lee, A. Khazatsky, C. D. Manning, and C. Finn. Detectgpt : Zero-shot machine-generated text detection using probability curvature. 2023.

[19] Krichen Moez. Les generative adversarial networks : Architecture, entraînement, et applications. *Laboratoire ReDCAD, Université de Sfax, Tunisie*, 2023. Disponible en ligne : URL de l'article.

[20] Usman Naseem, Imran Razzak, Shah Khalid Khan, and Mukesh Prasad. A comprehensive survey on word representation models : From classical to state-of-the-art word representation language models. *School of Computer Science, The University of Sydney, Australia. School of Information Technology, Deakin University, Australia. School of Engineering, RMIT University, Australia. School of Computer Science, University of Technology Sydney, Australia*, 2020.

[21] O.Campesato. *Transformer, BERT, and GPT : Including ChatGPT and Prompt Engineering.* Mercury Learning and Information, Boston, MA, 2024. Printed on acid-free paper in the United States of America.

[22] Sinan Ozdemir. *Quick Start Guide to Large Language Models : Strategies and Best Practices for Using ChatGPT and Other LLMs.* Addison-Wesley, 2023.

[23] Azunre Paul. *Transfer Learning for Natural Language Processing.* Manning Publications Co., Shelter Island, NY, 2021.

[24] Junjie Peng, Elizabeth C. Jury, Pierre Dönnes, and Coziana Ciurtin. Machine learning techniques for personalised medicine approaches in immune-mediated chronic inflammatory diseases : Applications and challenges. 2021.

[25] Pillai, Anitha S., and Roberto Tedesco. *Machine Learning and Deep Learning in Natural Language Processing.* CRC Press, Boca Raton, FL ; Milton Park, Abingdon, Oxon, 1 edition, 2024.

[26] Ateeq Qureshi. Ai and human generated text dataset. `https://huggingface.co/datasets/Ateeqq/AI-and-Human-Generated-Text`, 2023. [En ligne ; consulté le 23-mai-2024].

[27] P. Sarzaeim, A. M. Doshi, and Q. H. Mahmoud. A framework for detecting ai-generated text in research publications. 2023.

[28] Quinn Spencer. *Neural Networks Deep Learning and Machine Learning Outlined.* Quinn Spencer, 2018.

[29] J. Su, T. Yue Zhuo, D. Wang, and P. Nakov. Detectllm : Leveraging log rank information for zero-shot detection of machine-generated text. 2023.

[30] Z. Su, X. Wu, W. Zhou, G. Ma, and S. Hu. Hc3 plus : A semantic-invariant human chatgpt comparison corpus. 2024.

[31] Taulli Tom. *Generative AI : How ChatGPT and Other AI Tools Will Revolutionize Business*. Apress, New York, NY, 1 edition, 2023. Printed on acid-free paper.

[32] E. Tulchinskii, K. Kuznetsov, L. Kushnareva, D. Cherniavskii, S. Nikolenko, E. Burnaev, S. Barannikov, and I. Piontkovskaya. Intrinsic dimension estimation for robust detection of ai-generated texts. 2023.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017.

[34] V. Verma, E. Fleisig, N. Tomlin, and D. Klein. Ghostbuster : Detecting text ghostwritten by large language models. 2023.

[35] V. Vora, J. Savla, D. Mehta, A. Gawade, and R. Mangrulkar1. Classification of diverse ai generated content : An in exploration using machine learning and knowledge graphs. 2023.

[36] X. Yang, W. Cheng, Y. Wu, L. Petzold, W. Yang Wang, and H. Chen2. Dna-gpt : Divergent n-gram analysis for training-free detection of gpt-generated text. 2023.

[37] X. Yang, L. Pan, X.Zhao, H. Chen, L. Petzold, W. Yang Wang, and W. Cheng. A survey on detection of llms-generated contents. 2023.

[38] X. Yang, J. Zhang, K. Chen, W. Zhang, Z. Ma, F. Wang, and N. Yu. Tracing text provenance via context-aware lexical substitution. 2023.

[39] X. Yu, Y. Qi, K. Chen, G. Chen, Xi Yang, P. Zhu, W. Zhang, and N. Yu. Gpt paternity test : Gpt generated text detection with gpt genetic inheritance. 2023.

RÉSUMÉ

Ce travail se penche sur la détection des contenus textuels produits par l'intelligence artificielle générative à l'aide des techniques de traitement automatique du langage naturel. Nous avons réalisé un état de l'art des approches récentes pour distinguer les textes écrits par des humains de ceux produits par des modèles génératifs avancés. Suite à cette analyse, nous avons développé une approche en utilisant un grand modèle de langage pour améliorer la précision de la détection. Les résultats obtenus démontrent l'efficacité de certaines techniques tout en mettant en évidence les défis rencontrés, notamment la complexité croissante des modèles génératifs. Notre étude offre une vue d'ensemble des solutions pertinentes disponibles et propose des axes d'amélioration pour une détection plus fiable des contenus générés par IA.

Mots clés : NLP ; LLM ; BERT ; IA ; Détection d'IA ; apprentissage automatique ; IA générative ; apprentissage profond.

ABSTRACT

This work focuses on the detection of text content produced by generative artificial intelligence using natural language processing techniques. We have carried out a state-of-the-art review of recent approaches to distinguishing texts written by humans from those produced by advanced generative models. Following this analysis, we developed an approach using a large language model to improve detection accuracy. The results obtained demonstrate the effectiveness of some techniques while highlighting the challenges, notably the increasing complexity of generative models. Our study provides an overview of the relevant solutions available, and suggests areas of improvement for more reliable detection of AI-generated content.

Key words : NLP ; LLM ; BERT ; Generative AI ; machine learning ; Deep learning ; AI detction.