République algérienne démocratique et populaire
Ministère de l'Enseignement supérieur et de la Recherche scientifique
UNIVERSITÉ IBN KHALDOUN DE TIARET

Thèse

Présenté à :

FACULTÉ DE MATHÉMATIQUES ET D'INFORMATIQUE
DÉPARTEMENT D'INFORMATIQUE

Afin d'obtenir le diplôme de : Master

Spécialité : génie logiciel

Présenté par : **MAZOUZ Fadila**

Sur le thème :

Génération Augmentée par la Recherche pour l'Analyse des Sentiments à Base d'Aspects

Soutenue publiquement le /06/2025 à Tiaret devant le jury composé de :

Mr KHARROUBI Sahraoui	M.C.A Université Ibn-Khaldoun Tiaret	Président
Mme BOUBEKEUR Aicha	M.A.A Université Ibn-Khaldoun Tiaret	Encadreur
Mr BENAOUDA Habib	M.A.A Université Ibn-Khaldoun Tiaret	Examinateur

2024-2025

Remerciements

En tout premier lieu,

Je remercie le bon dieu de m'avoir donné la force de mener

à bien ce travail. Au nom du Dieu,le Clément et le Miséricordieux, louange à ALLAH.

Je tiens à exprimer ma profonde gratitude à Mme. BOUBEKEUR Aicha, pour avoir encadré et dirigé mes recherches.

Je la remercie pour m'avoir soutenu et appuyé tout au long de ma thèse. Ses précieux conseils, son exigence et ses commentaires ont permis d'améliorer grandement la qualité de mes travaux.

Je présent également tous mes respects et mes sincères remerciements aux membres de jury Mr.KHAROUBI Sahraoui et Mr.BENAOUDA Habib qui ont accepté d'évaluer mon travail.

J'adresse mes remerciements à tous les professeurs, pour leurs conseils et leurs critiques qui ont guidé mes réflexions durant mes recherches.

Et je remerc également tous mes collègues et amis du département d'informatique de l'université Ibn Khaldoun Tiaret souhaite exprimer mes profondes gratitudes mes parents qui m'ont soutenus tout au long de mon projet.

Ainsi que toute la famille, les amis pour leur soutien indéfectible. Enfin, Je remercie tous ceux qui m'ont aidés de près ou de loin dans l'élaboration de ce travail.

Merci à tous !

DÉDICACE

Je dédie ce modeste travail

Je dédie ce mémoire a toutes les personnes qui ont été présentes à mes côtes et m'ont apporté leur soutien tout au long de ce parcours académique et professionnel, transformant chaque d'défi en une opportunité de grandir et d'apprendre.

-À ma mère, Tes encouragements constants, surtout lorsque mes pensées négatives étaient plus fortes que moi, m'ont donné la force de persévérer. Merci pour tout ce que tu m'as donné, même dans les moments les plus difficiles.

-À mon père, l'homme numéro un de ma vie, la personne qui a cru en moi plus que quiconque, l'homme de ma vie.

-À mes sœurs Salima, chaima , et Khadra mes frères Abd Elhadi et Khaier eddine ont toujours été si aimants et encourageants .

-À mes chères amies nassre eddine , Djamila ..qui m'a encouragé et soutenu de tout leurs cœur et tous ceux que je connais de près ou de loin.

-HADJLA-

Abstract

With advances in generative artificial intelligence, online reviews have become an essential source of knowledge for various scientific contributions and research. Sentiment analysis, and in particular aspect-based sentiment analysis (ABSA), enables detailed analysis of opinions by associating each sentiment with a specific aspect or characteristic of a product or service. This project fits into this context and aims to design and develop an ABSA tool applied to student reviews on the RateMyProfessors website. Unlike traditional analysis approaches, our solution relies on the use of large language models (LLMs), leveraging prompting, augmented generation and fine-tuning to adapt the analysis to the educational context. Experimental results show that well-oriented LLMs offer detailed and contextualised analysis while reducing the need for manual annotation. This work also opens up prospects for improving the robustness, adaptability and application of this approach to other fields.

Key words : Generative AI, LLM, ABSA, augmented/guided generation, prompting, fine- tuning.

Résumé

Avec les avancées de l'intelligence artificielle générative, les avis en ligne sont devenus une source de connaissances incontournable pour diverses contributions et recherche scientifiques. L'analyse des sentiments, et en particulier l'analyse des sentiments à base aspects (ABSA), permet l'analyse fine des opinions en associant chaque sentiment à un aspect ou caractéristique spécifique d'un produit ou service. Ce projet s'inscrit dans ce contexte et vise à concevoir et developper un outil d'ABSA appliquée aux avis d'étudiants du Site Web de RateMyProfessors. Contrairement aux approches d'analyses classiques, notre solution repose sur l'utilisation de larges modèles de langue (LLMs), exploitant des instructions d'incitation, de la génération augmentée et de fine- tuning, pour adapter l'analyse au contexte éducatif. Les résultats expérimentaux montrent que les LLM, bien orientés, offrent une analyse fine et contextualisée, tout en réduisant le besoin de l'annotation manuelle. Ce travail ouvre également des perspectives pour améliorer la robustesse, l'adaptabilité et l'application de cette approche à d'autres domaines.

Mots clés : IA générative, LLM, ABSA, génération augmentée/guidée, instruction d'incitation, finetuning.

ملخص

مع التقدم الذي أحرزته الذكاء الاصطناعي التوليدي، أصبحت الآراء عبر الإنترنت مصدراً لا غنى عنه للمعرفة في مختلف المساهمات والبحوث العلمية. تسمح تحليل المشاعر، ولا سيما تحليل المشاعر القائم على الجوانب ،(ABSA) بتحليل دقيق للآراء من خلال ربط كل مشاعر بجانب أو خاصية محددة لمنتج أو خدمة.

يندرج هذا المشروع في هذا السياق ويهدف إلى تصميم وتطوير أداة ABSA مطبقة على آراء الطلاب على موقع .RateMyProfessors على عكس مناهج التحليل التقليدية، تعتمد حلولنا على استخدام نماذج لغوية واسعة ،(LLMs) تستخدم تعليمات التحفيز والتوليد المعزز والضبط الدقيق، لتكييف التحليل مع السياق التعليمي. تظهر النتائج التجريبية أن نماذج اللغة الكبيرة ،(LLMs) إذا تم توجيهها بشكل جيد، توفر تحليلًا دقيقًا ومناسبًا للسياق، مع تقليل الحاجة إلى التعليق اليدوي. يفتح هذا العمل أيضًا آفاقًا لتحسين متانة هذا النهج وقابليته للتكيف وتطبيقه في مجالات أخرى.

الكلمات المفتاحية: الذكاء الاصطناعي التوليدي، ،LLM ،ABSA التوليد المعزز/الموجه، تعليمات التحفيز، الضبط الدقيق.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations et acronymes

SA : Sentiment Analysis

ABSA : AS basée sur les aspects

RAG : La génération augmentée par la recherche

LLMs : Large Language Models

TF-IDF : Term Frequency-Inverse Document Frequency

LSA : analyse sémantique latente

IA : Intelligence artificielle

ML : Machine Learning

KNN : Support Vector Machine

SVM : K-Nearest Neighbor

DT : Decision Tree

NB : Naive Bayes

LR : Logistic Regression

NLP : Natural Language Processing

TALN : Traitement Automatique De La Langue Naturelle

CNN : Réseau neuronal convolutif

DNN : Deep Neural Network

RNA : réseau neuronal artificiel

RNN : Récurrent Neural Networks

NLP : traitement du langage naturel

BERT : Bidirectional Encoder Representations from Transformers

TF : Term Frequency

TF-IDF : Term Frequency-Inverted Document Frequency

WE : Word Embedding

Introduction générale

A l'ère de l'Intelligence Artificielle avancée, le monde numérique virtuel est devenu un espace incontournable d'expression et de communication pour leurs utilisateurs. Chaque jour, des millions d'utilisateurs partagent spontanément leurs opinions, critiques et recommandations à propos de produits ou de services, constituant ainsi une source riche et précieuse d'informations.

Dans ce contexte, l'analyse des sentiments et plus spécifiquement l'analyse des sentiments à base d'aspects (Aspect-Based Sentiment Analysis, ou ABSA) s'impose comme une approche clé pour extraire, structurer et interpréter ce qui est partagé en ligne. Contrairement à l'analyse des sentiments traditionnelle, qui se limite à déterminer le sentiment global d'un texte (positif, négatif ou neutre), l'ABSA permet de capturer des opinions plus fines et nuancées en associant les sentiments exprimés à des aspects ou caractéristiques spécifiques d'un produit ou service (par exemple, la qualité de l'encadrement, l'engagement de l'apprenant ou l'ambiance dans un cours). Cette granularité fine offre une vision plus contextualisée des préférences et insatisfactions des étudiants, particulièrement utile pour des secteurs comme l'éducation, où l'expérience étudiante repose sur une multitude de critères. Les approches classiques en ABSA reposent souvent sur des modèles d'analyse de sentiments simples pour l'extraction des aspects, la détection de sentiments, ou encore la reconnaissance des relations entre aspects et opinions.

Ces approches exigent généralement un développement spécifique pour chaque composant, ainsi qu'un volume important de données annotées, ce qui limite leur portabilité et leur efficacité dans des domaines peu dotés en ressources. Dans ce mémoire, nous adoptons

une approche innovante en nous appuyant sur les larges modèles de langue (LLM), qui ont récemment démontré leurs performances remarquables dans diverses tâches de traitement du langage naturel (NLP).

En vue de cela, nous exploitons les capacités avancées du LM, en les guidant et adaptant au contexte ciblé via des techniques de prompting, de recherche et génération et de fine-tuning. Ces stratégies permettent d'exploiter les connaissances linguistiques préacquises pour effectuer une analyse fine, contextualisée et plus facilement généralisable, même dans des contextes spécialisés comme celui de l'université. L'objectif principal de ce présent travail est ainsi de concevoir et d'évaluer une méthode d'ABSA appliquée aux avis d'étudiants d'un milieu universitaire. En s'appuyant sur les LLM, nous cherchons à démontrer qu'il est possible d'améliorer la précision et la pertinence de la SA tout en réduisant les besoins en annotation manuelle et en développement de modèles spécifiques.

Ce mémoire est structuré comme suit : nous commencerons par volet théorique sur l'analyse des sentiments, les approches ABSA et les LLM. Nous décrirons par la suite, pour le volet pratique, notre méthodologie, les données utilisées et les choix techniques effectués. Les résultats expérimentaux seront ensuite présentés et discutés, avant de conclure par une synthèse des apports de cette ABSA et des perspectives futures.

Organisation du mémoire :

Chapitre 1 :"De l'analyse des sentiments à gros grains à une analyse des sentiments à granularité fine" : Ce chapitre introduit les concepts fondamentaux de l'analyse des sentiments et présente l'approche basée sur les aspects.

Chapitre 2 : "Large Language Models" : Il s'agit de la présentation détaillée de l'IA générative basée sur les LLMs, en particulier l'approche RAG (Retrieval-Augmented Generation), son architecture et ses apports en traitement du langage naturel.

Chapitre3 :"Mise en oeuvre de la solution ABSA" : Ce chapitre présente la démarche méthodologique adoptée pour développer notre système ABSA, en détaillant l'architecture, les modules fonctionnels et les choix techniques. Il inclut également l'évaluation expérimentale du système à travers des tests sur un corpus donné.

CHAPITRE I

De l'analyse des sentiments à gros grains à une analyse des sentiments à granularité fine

I.1 Introduction

L'analyse des sentiments (AS), également appelée exploration d'opinions, est le domaine d'étude qui analyse les opinions, les sentiments, les évaluations, les attitudes et les émotions des individus envers des entités et leurs attributs exprimés dans un texte écrit. Les entités peuvent être des produits, des services, des organisations, des individus, des événements, des problèmes ou des sujets. Ce domaine représente un vaste espace problématique.

De nombreux noms apparentés et des tâches légèrement différentes comme , AS, l'exploration d'opinions, l'analyse d'opinions, l'extraction d'opinions, l'exploration de sentiments, l'analyse de la subjectivité, l'analyse des affects, l'analyse des émotions et l'exploration d'avis » sont désormais regroupés sous le terme de AS.

I.2 Analyse des sentiments

La polarité des sentiments a une signification contextuelle dans l'AS. En se basant sur la somme des opinions positives et négatives exprimées à propos d'un événement, des méthodes automatiques calculent la polarité des sentiments. Généralement, les scores de sentiment quotidiens sont calculés en mesurant le nombre de mots positifs et négatifs dans une phrase. Une phrase comportant plus de mots négatifs (reflétant la violence, la colère, la tristesse) que de mots positifs (exprimant le bonheur, la célébration, la joie) est considérée comme négative.

Définition La polarité des sentiments des données textuelles est le résultat d'une analyse exprimée en termes de valeur numérique obtenue par la somme algébrique des opinions contenues dans chaque entité de la phrase, du document ou du message. La polarité des sentiments peut être définie comme positive, négative ou neutre[9].

Le score de polarité des sentiments (Sp) d'un message est la combinaison linéaire de toutes les polarités. Celle-ci est ensuite convertie en un rapport à la somme pour obtenir une valeur rationnelle comprise entre -1 et 1. Par conséquent, $Sp = (Wp - Wn) / (Wp + Wn)$

où Wp désigne le nombre de mots positifs et Wn représente un certain nombre de mots négatifs dans le message [1].

Plusieurs décideurs, notamment les entreprises et les organisations gouvernementales, peuvent en apprendre davantage sur l'opinion publique grâce à l'analyse des médias sociaux. Les utilisateurs utilisent des messages courts pour partager leurs opinions sur les plateformes de réseaux sociaux. L'AS sur les médias sociaux évalue les émotions et les opinions. Ces messages sur les médias sociaux, reflètent l'opinion, le sentiment et les émotions du public via une combinaison de texte, d'images et d'émoticônes [1].

L'analyse des médias sociaux a connu un essor rapide au cours de cette décennie pour répondre à la question « Que pensent les gens d'un événement ou d'un sujet ? ». L'AS, des opinions et des émotions est essentielle. Par exemple, évaluer le bien-être d'une communauté ; prévenir les suicides, etc. De plus, l'analyse des retours clients permet aux entreprises d'obtenir des informations précieuses sur leur niveau de satisfaction. Cependant, le format atypique de ces messages complique la compréhension du langage du bouche-à-oreille électronique.

Par exemple, le message « Profitez de mon dimanche de détente 😊 » est un mélange

de mots et d'émojis exprimant un sentiment positif pour dimanche. Un tel message est difficile à analyser, car il contient des symboles spéciaux et des émoticônes. Le contexte de la phrase ne peut être interprété que si les analyseurs sont familiarisés avec la signification de l'émoji. L'AS ne serait pas correcte tant que ces messages ne seraient pas traduits en texte clair tout en conservant leur contexte et leurs émotions.

Les utilisateurs des réseaux sociaux expriment toujours leurs sentiments à propos d'un produit ou d'un événement public. Un Contenu (message) Généré par un Utilisateur UGC et contenant un sentiment peut être défini comme un quadruple [1] :

$$\boldsymbol{UGC = (\ o\ ,\ f,\ s\ ,\ h\ )}$$

- $\boldsymbol{o}$ indique un objet cible.
- $\boldsymbol{f}$ représente une caractéristique de l'objet.
- $\boldsymbol{s}$ désigne la valeur sentimentale de l'opinion (+ve, −ve ou neutre).
- $\boldsymbol{h}$ signifie détenteur d'opinion.

Par exemple, un avis d'utilisateur sur les téléphones portables :

« Même si la durée de vie de la batterie de mon nouveau téléphone n'est pas longue, cela me convient. »

Lorsque nous analysons ce message, nous obtenons le quadruple comme suit :

- $\boldsymbol{o}$ nouveau téléphone.
- $\boldsymbol{f}$ durée de vie de la batterie.
- $\boldsymbol{s}$ n'est pas long (sentiment -ve).
- $\boldsymbol{h}$ pour moi.

AS implique des sous-tâches majeures : le prétraitement, la transformation et la classification. Bien que le prétraitement et la transformation puissent être réalisés par manipulation de texte, la classification est une tâche complexe. Elle implique la reconnaissance du sentiment et de son contexte. Le système doit interpréter le sens et analyser le sentiment de manière similaire à l'intelligence humaine. Une méthodologie systématique est nécessaire :

- Nettoyage et extraction du contenu textuel.
- Identification de la polarité du sentiment du texte extrait.

I.2.1 Architecture du système d'analyse des sentiments

Après avoir extrait le texte brut des messages des réseaux sociaux, des algorithmes informatiques peuvent classer automatiquement la polarité des sentiments. Les deux catégories suivantes peuvent être utilisées pour classer globalement les algorithmes d'AS [1] :

- Basé sur le lexique Cette méthode utilise un référentiel préconstruit de mots émotionnels pour correspondre au message. Une base de connaissances contenant des unités textuelles annotées avec des étiquettes de sentiments est appelée lexique des émotions. Elles s'appuient sur des ressources lexicales telles qu'un lexique, des banques de mots ou une ontologie.

- Basé sur l'IA Les approches basées sur l'IA utilisent des méthodes d'apprentissage automatique pour identifier les sentiments. L'apprentissage automatique utilise des algorithmes capables d'apprendre des données en exploitant la similarité des documents entre les messages texte.

La subjectivité, la polarité ou l'objet d'une opinion sont souvent déterminés par un système lexical utilisant un ensemble de critères créés par l'homme. Ces règles peuvent prendre en compte diverses méthodes de traitement du langage naturel (TALN) développées en linguistique computationnelle, comme l'étiquetage des parties du discours, la tokenisation, la recherche de radicaux et l'analyse syntaxique [1].

Contrairement aux systèmes basés sur un lexique, les méthodes basées sur l'IA s'appuient sur des algorithmes d'apprentissage automatique plutôt que sur des règles élaborées à la main. Généralement, le processus d'identification de la polarité d'un sentiment est défini comme un problème de classification, dans lequel un classificateur reçoit un texte et génère une étiquette de polarité, comme positive, négative ou neutre.

Les systèmes basés sur le lexique sont peu performants en raison du langage non standard des utilisateurs des réseaux sociaux. Les résultats sont plus précis, ce qui constitue l'un des principaux avantages des systèmes basés sur l'IA. Ils ressemblent à un système de notation humain, tout en classant la polarité des sentiments en tenant compte des informations contextuelles. Les algorithmes d'apprentissage automatique sont largement utilisés pour résoudre des problèmes complexes du monde réel. Il s'agit d'une stratégie prometteuse, largement utilisée dans les disciplines de l'IA, notamment le traitement du

langage naturel (TALN), l'analyse sémantique, l'apprentissage par transfert, la vision par ordinateur, et bien d'autres [1]. L'AS sur les réseaux sociaux est une tâche complexe en raison des difficultés suivantes :

- Les utilisateurs n'ont pas de style d'écriture formel.
- Les utilisateurs utilisent des éléments familiers et personnels dans leur langage.
- Les utilisateurs utilisent un mélange de texte, d'émojis, d'abréviations, d'images et de symboles.
- Les hashtags sont également souvent utilisés pour souligner leur importance.
- Des millions de messages sont générés chaque seconde.

Les systèmes automatiques tentent d'extraire la polarité des sentiments à l'aide d'algorithmes et de techniques informatiques. Le langage courant rend difficile l'interprétation du contexte et des sentiments exprimés par les systèmes automatiques. La méthodologie d'AS présentée ici utilise des méthodes d'apprentissage automatique pour classer les sentiments au niveau de la phrase et agit directement à ce niveau.

I.2.2 Ingénierie des fonctionnalités des messages

Les messages générés sur les plateformes sociales sont mal formulés et peu structurés. Le système d'AS doit être adapté au style et aux spécificités de ce style d'écriture informel. Par exemple, le message « J'apprécie mon dimanche de détente 😊 » signifie un sentiment positif. Il comprend un mot et un émoji représentant le bonheur. L'ingénierie des caractéristiques est la méthode de sélection et de transformation des données brutes en caractéristiques exploitables en ML. L'inclusion de symboles et d'émojis joue un rôle important dans la détection des sentiments cachés. La première étape du traitement est l'ingénierie des caractéristiques. Il s'agit de l'extraction des phrases significatives afin que le message soit informatif, non redondant et soutienne les étapes suivantes d'apprentissage et de généralisation[1].

L'ingénierie des caractéristiques fonctionne comme suit :

- Les émojis ou émoticônes sont remplacés par la signification du référentiel de données locales communes' si un mappage est disponible. Dans le cas contraire, ils sont supprimés.

- La ponctuation telle que les parenthèses, les points et les virgules est supprimée.
- Les mots vides (un, une, et, etc.) sont supprimés.
- Les mots contenant des lettres et des chiffres spéciaux sont éliminés, tout comme les mots qui n'incluent pas uniquement des caractères alphabétiques.
- Les lettres minuscules sont utilisées pour tous les mots.

La traduction des émojis est un élément crucial de la sélection des fonctionnalités. Les émojis populaires utilisés dans l'AS peuvent être traduits en texte brut grâce à leur signification dans le référentiel Unicode Common Locale Data Repository (CLDR) [1].

Le texte brut converti peut ne pas être une phrase significative, mais il capture le sentiment et le contexte d'origine. Il pourrait servir de vecteur d'entrée aux algorithmes d'apprentissage automatique. Le texte brut extrait est transmis en entrée à un système automatique qui utilise un algorithme d'apprentissage automatique pour la classification. Ce système automatique de classification de la polarité des sentiments M est défini comme un quadruplet [1] :

$$\boldsymbol{M = \{\alpha, \lambda, \delta, \phi\}}$$

où

- $\alpha = \{et_1, et_2, \ldots, et_n\}$ est un ensemble de messages provenant des médias sociaux,
- $\lambda =\mid et_i - et_j \mid$ est une fonction qui calcule le score de similarité des messages,
- $\delta \rightarrow$ un algorithme pour classer la polarité des sentiments, et
- $\phi = \{SP, SN, SNe\}$ où est un ensemble d'étiquettes de sortie : positives, négatives et neutres.

  Pour une entrée donnée dans , l'algorithme produit une étiquette de sortie dans à l'aide de la fonction le diagramme suivant montre l'aperçu de la machine de classification des sentiments (Figure (I.1)).

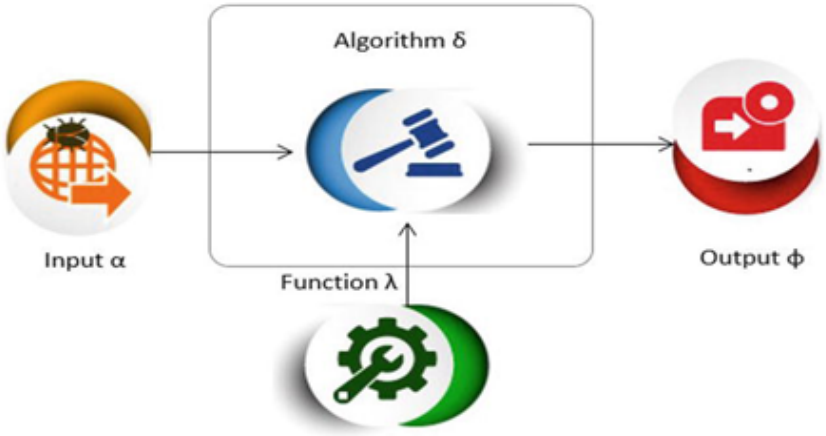

Figure I.1 : Système de classification de la polarité des sentiments.[1]

La polarité des sentiments des messages peut être efficacement prédite par un système d'intelligence artificielle en tenant compte du bon choix de :

1. La fonction de calcul du score de similarité de deux messages.
2. L'algorithme de classification adopté.

La sélection des algorithmes pour ces deux tâches est détaillée dans les sections suivantes :

I.2.2.1 Score de similarité des messages

Les algorithmes d'apprentissage automatique fonctionnent principalement sur des vecteurs d'entrée numériques. IL est essentiel de convertir le texte prétraité en vecteur numérique pour que les algorithmes prédisent correctement la polarité des sentiments. Le modèle d'apprentissage automatique est entraîné avec des ensembles de données étiquetés pour les messages positifs, négatifs et neutres. Un modèle bien entraîné prédit la polarité des sentiments des messages entrants en la comparant aux données d'entraînement. Comparer deux messages et attribuer un score numérique indiquant leur similarité est important pour une précision élevée. **TF-IDF** (« Term Frequency-Inverse Document Frequency ») est une approche de vectorisation largement utilisée en traitement de texte. Cette approche examine la fréquence relative des termes dans un texte via un pourcentage inverse de la phrase dans l'ensemble du corpus de documents. Elle est efficace pour la catégorisation de texte ou pour permettre aux machines d'interpréter les mots d'entrée représentés par des nombres. Dans TF-IDF, les mêmes textes doivent produire un vecteur plus proche. TF-IDF est le produit multiplicatif des scores de fréquence du terme et de

fréquence inverse du document du mot [1].

TF représente le « **nombre de fois où le mot apparaît dans le document / Nombre total de mots dans le document** ».

$$IDF = ln \frac{Nombre\ de\ documents}{Nombre\ de\ documents\ dans\ lesquels\ le\ mot\ apparat} \tag{1}$$

$$\boldsymbol{TFIDF(t,d,D) = TF(t,d) \times IDF(t,D)} \tag{2}$$

I.2.3 Classification des sentiments à l'aide de l'apprentissage automatique

Une tâche d'identification de la polarité des sentiments est généralement définie comme un problème de classification, où un classificateur reçoit un texte et génère une étiquette, telle que « positif, négatif ou neutre ». En général, des techniques d'apprentissage supervisé et non supervisé peuvent être utilisées pour classer du texte à l'aide d'une méthodologie d'apprentissage automatique. De nombreux documents étiquetés sont utilisés dans les approches supervisées. Les approches non supervisées sont employées lorsqu'il est difficile de trouver ces documents d'apprentissage étiquetés. Etant donné que des milliers de messages sont émis chaque minute sur les réseaux sociaux, nous pouvons en étiqueter manuellement quelques milliers et les utiliser pour l'apprentissage supervisé. Un ensemble de données d'apprentissage composé de trois étiquettes, à savoir positif, négatif et neutre, est préparé en sélectionnant des messages provenant de réseaux sociaux tels que Twitter, Facebook, etc. Seuls les documents d'apprentissage les plus similaires au document entrant sont utilisés par l'algorithme d'apprentissage automatique pour l'étiqueter [1][10],Parmi les algorithmes de classification de texte les plus courants [9][1][10] .

- Le K-Nearest Neighbor (KNN) : Grâce à l'apprentissage supervisé, cette méthode divise les objets en l'une des catégories prédéfinies d'un échantillon. Le KNN peut être utilisé pour classer la polarité des sentiments où les vecteurs numériques sont représentés sous forme de points dans un diagramme.

KNN est entraîné avec un ensemble de données étiquetées. L'étiquette de l'entrée entrante est prédite en fonction du vote majoritaire de ses voisins :

Algorithme : Classification KNN

1. Calculer la distance entre l'élément de données de requête et les éléments de données étiquetés.
2. Classer les échantillons étiquetés par distance croissante.
3. En utilisant la précision, déterminer les k voisins les plus proches qui constituent le nombre heuristique ment optimal. Pour ce faire, utilisez la validation croisée.
4. Attribuer une étiquette de classe à l'exemple de requête en fonction du vote majoritaire (voisins).
5. Une matrice de confusion peut être utilisée comme outil pour vérifier la précision de l'algorithme de classification KNN.

Lors de l'apprentissage, les éléments de données étiquetés forment les groupes en fonction de leur distance par rapport à leurs voisins. Un nouveau point de données se voit attribuer une étiquette en sélectionnant des voisins et leur vote majoritaire.

-Boost XGL :'Extreme Gradient Boosting est appelé XGBoost' - L'approche ML GBDT (« Gradient-Boosted Decision Tree ») est évolutive et distribuée. Elle offre une amélioration parallèle des arbres et constitue la technique ML la plus efficace pour les problèmes de classification et de régression.

-Machines à vecteurs de support (SVM) : La SVM est une méthode supervisée dans laquelle la méthode d'apprentissage examine les données et identifie des modèles. Les données sont affichées sous forme de points dans un espace à n dimensions. La valeur de chaque attribut est ensuite associée à une coordonnée spécifique, ce qui facilite la catégorisation.

-Bayes naïf (NB) : NB repose sur le théorème de Bayes, une approche permettant de déterminer la probabilité conditionnelle à partir d'informations antérieures et de la croyance naïve selon laquelle chaque attribut est indépendant des autres. Le principal avantage de Naive Bayes est qu'il fonctionne assez bien même avec de faibles quantités de données d'apprentissage, alors que la plupart des algorithmes de ML reposent sur d'énormes quantités de données d'apprentissage.

I.2.4 Classification des sentiments à l'aide de l'apprentissage profond

L'apprentissage profond, une sous-catégorie de l'apprentissage automatique, se concentre sur le développement de réseaux neuronaux capables d'apprendre à partir de données

Leurs algorithmes peuvent analyser de vastes ensembles de données et identifier des schémas complexes. Par conséquent, des modèles d'AS peuvent être développés pour prédire avec précision les sentiments exprimés dans des données textuelles. Ces algorithmes sont capables d'analyser les données textuelles et d'identifier les sentiments exprimés, ainsi que de percevoir le ton, l'émotion et l'intention sous-jacents. Ainsi, des informations précieuses peuvent être extraites de diverses sources de données textuelles, telles que les publications sur les réseaux sociaux et les avis clients.

Ces dernières années, les algorithmes d'apprentissage profond, notamment les réseaux neuronaux convolutifs **(CNN)** et les réseaux neuronaux récurrents **(RNN)**, se sont révélés très prometteurs pour améliorer la précision des modèles d'AS. Ces algorithmes excellent dans l'apprentissage de schémas de données complexes et peuvent être entraînés sur de vastes ensembles de données pour optimiser leurs performances. Les réseaux de neurones récurrents (RNN) et les réseaux de mémoire à long terme **(LSTM)** sont souvent utilisés pour capturer la nature séquentielle d'un texte, tandis que les réseaux de neurones convolutifs (CNN) extraient efficacement des caractéristiques cruciales par des opérations convolutives. De plus, des mécanismes d'attention sont utilisés pour se concentrer sur les sections pertinentes du texte. Cependant, il est essentiel de noter que les modèles d'apprentissage profond nécessitent des quantités importantes de données étiquetées et de ressources de calcul pour un apprentissage efficace[9].

L'apprentissage par transfert est devenu une piste d'exploration intéressante en AS. Il implique l'utilisation de modèles pré-entraînés pour améliorer les performances des modèles d'AS. Cette approche permet aux développeurs de capitaliser sur les connaissances acquises grâce à l'entraînement sur des ensembles de données conséquents, améliorant ainsi la précision des modèles d'AS face à des ensembles de données plus petits. Outre ces avancées notables, les chercheurs étudient également l'application de techniques d'apprentissage non supervisé, telles que le clustering et la modélisation thématique, pour renforcer la précision des modèles AS. Ces techniques s'avèrent essentielles pour identifier des tendances et des thèmes au sein des données textuelles, contribuant ainsi à l'affinement des modèles d'AS[1].

Il est important de noter que les modèles d'apprentissage profond nécessitent souvent d'importantes quantités de données d'apprentissage étiquetées, qui peuvent être difficiles d'accès dans divers domaines ou langages. Dans de tels scénarios, l'approche hybride pré-

sente des avantages notables. Plus précisément, elle implique l'utilisation d'un composant basé sur des règles pour la classification préliminaire des sentiments [1], utilisant des règles ou des lexiques prédéfinis, nécessitant ainsi un minimum de données étiquetées. Par la suite, un composant d'apprentissage automatique ou d'apprentissage profond est utilisé pour affiner les résultats à l'aide d'ensembles de données étiquetées plus petits, réduisant ainsi les besoins en données tout en obtenant des performances appréciables. Ainsi, l'approche hybride devrait jouer un rôle essentiel dans l'avenir d'AS. A mesure que ces techniques continuent d'évoluer, de nouvelles améliorations de la précision et de l'efficacité des modèles d'AS sont à anticiper, les rendant encore plus précieux pour les entreprises et les organisations cherchant à extraire des informations des commentaires clients et des données des réseaux sociaux.

I.2.5 Mesures de performance populaires pour la classification

Les performances d'un algorithme d'apprentissage doivent être évaluées avant de sélectionner le modèle pour des applications concrètes. L'AS est un problème de classification. Les résultats de tout algorithme de classification peuvent être évalués en créant une matrice de confusion et des indicateurs courants.

Sur la base des éléments de la matrice de confusion, un ensemble de mesures est généralement calculé pour évaluer les performances du modèle de classification[1].

I.2.5.1 Précision et taux d'erreur

La qualité d'un modèle de classification peut être évaluée à l'aide de ces indicateurs clés. Un « vrai positif », un « faux positif », un « vrai négatif » et un « faux négatif » sont respectivement désignés par les lettres TP, FP, TN et FN. Voici les définitions des termes « précision » et « taux d'erreur » en classification[1] :

$$\textbf{Précision} = (\textbf{TP} + \textbf{TN})/(\textbf{TN} + \textbf{FN} + \textbf{TP} + \textbf{FP}) \tag{3}$$

$$\textbf{Taux d' erreur} = \textbf{1}- \textbf{Précision} \tag{4}$$

I.2.5.2 Rappel, score F1 et précision

Ce sont également les principales mesures pour les ensembles de tests non équilibrés, et elles sont appliquées plus fréquemment que le taux d'erreur ou la précision. Pour la

classification binaire, la précision et le rappel sont spécifiés ci-dessous. La moyenne harmonique du rappel et de la précision est le score F1. Le score F1 est optimal lorsqu'il est égal à 1 (rappel et précision parfaits), et il est inférieur lorsqu'il est égal à 0 [1].

$$\textbf{Précision} = \textbf{TP}/(\ \textbf{TP} + \textbf{FP}\ ), \tag{5}$$

$$\textbf{Rappel} = \textbf{TP}/(\ \textbf{TP} + \textbf{FN}\ ), \tag{6}$$

$$\boldsymbol{F}\textbf{1- Score} = \textbf{2} \times \textbf{Prec} \times \textbf{Rec}/(\ \textbf{Préc} + \textbf{Rec}\ ) \tag{7}$$

Dans les problèmes de classification multi-classes, le calcul du rappel et de la précision peut se faire pour chaque étiquette de classe en évaluant les performances de chaque étiquette de classe individuellement ou simplement faire la moyenne des valeurs pour obtenir le rappel et la précision globaux.

I.2.6 Différents niveaux d'analyse des sentiments

La recherche sur L'AS a été principalement menée à trois niveaux de granularité : le niveau du document, le niveau de la phrase et le niveau de l'aspect [11] :

- Niveau du document : La tâche au niveau du document consiste à classer si un document d'opinion entier exprime un sentiment positif ou négatif. C'est ce qu'on appelle la classification des sentiments au niveau du document. Par exemple, dans le cas d'un avis sur un produit, le système détermine si l'avis exprime une opinion globale positive ou négative sur le produit. Ce niveau d'analyse suppose implicitement que chaque document exprime des opinions sur une seule entité (par exemple, un seul produit ou service). Par conséquent, il n'est pas applicable aux documents qui évaluent ou comparent plusieurs entités, pour lesquels une analyse plus fine est nécessaire.

- Niveau de la phrase : Le niveau suivant consiste à déterminer si une phrase exprime une opinion positive, négative ou neutre. Ce niveau d'analyse est étroitement lié à la classification de la subjectivité, qui distingue les phrases exprimant des informations factuelles (phrases objectives) des phrases exprimant des points de vue et des opinions subjectifs (phrases subjectives). Cependant, la subjectivité n'est pas équivalente au sentiment ou à l'opinion car, comme nous l'avons vu précédemment, de nombreuses phrases

objectives peuvent impliquer des sentiments ou des opinions - par exemple, "Nous avons acheté la voiture le mois dernier et l'essuie-glace est tombé". Inversement, de nombreuses phrases subjectives peuvent n'exprimer aucune opinion ou aucun sentiment - par exemple, "Je pense qu'il est rentré chez lui après le déjeuner".

- Niveau de l'aspect : Ni les analyses au niveau du document ni celles au niveau de la phrase ne permettent de savoir exactement ce que les gens aiment ou n'aiment pas. En d'autres termes, elles ne permettent pas de savoir sur quoi porte chaque opinion, c'est-à-dire la cible de l'opinion. Par exemple, si nous savons seulement que la phrase "J'aime l'iPhone 5" est positive, son utilité est limitée si nous ne savons pas que l'opinion positive porte sur l'iPhone 5. On pourrait dire que si nous pouvons classer une phrase comme positive, tout ce qui se trouve dans la phrase peut adopter l'opinion positive. Cependant, cela ne fonctionne pas non plus, car une phrase peut avoir plusieurs opinions - par exemple, "Apple s'en sort très bien dans cette économie médiocre". Cela n'a pas beaucoup de sens de classer cette phrase comme positive ou négative parce qu'elle est positive à propos d'Apple mais négative à propos de l'économie. Pour obtenir des résultats aussi fins, nous devons passer au niveau de l'aspect.

Ce niveau d'analyse était auparavant appelé niveau des caractéristiques, qui est maintenant appelé analyse du sentiment basée sur l'aspect. Au lieu d'examiner les unités linguistiques (documents, paragraphes, phrases, clauses ou expressions), l'analyse au niveau de l'aspect examine directement une opinion et sa cible (appelée cible de l'opinion). La prise de conscience de l'importance des cibles d'opinion nous permet de mieux comprendre le problème de l'analyse du sentiment.

L'objectif de ce niveau d'analyse est donc de découvrir des sentiments sur les entités et/ou leurs aspects. Sur la base de ce niveau d'analyse, un résumé des opinions sur les entités et leurs aspects peut être produit.

Outre les différents niveaux d'analyse, il existe deux types d'avis différents : les avis réguliers et les avis comparatifs [1] :

- Une opinion régulière exprime un sentiment à l'égard d'une entité particulière ou d'un aspect de l'entité. Par exemple, "Le Coca a très bon goût" exprime un sentiment ou une opinion positive sur l'aspect du goût du Coca. Il s'agit du type d'opinion le plus courant.

- Un avis comparatif compare plusieurs entités sur la base de certains de leurs aspects communs. Par exemple, "Le Coca a meilleur goût que le Pepsi" compare le Coca et le Pepsi sur la base de leurs goûts (un aspect) et exprime une préférence pour le Coca.

I.2.7 Analyse de sentiments en tant que mini-NLP

L'AS est généralement considérée comme un sous-domaine du NLP. Depuis sa création, l'AS a considérablement élargi la recherche en NLP en introduisant de nombreux problèmes de recherche difficiles qui n'avaient pas été étudiés auparavant. Cependant, les recherches menées au cours des vingt dernières années semblent indiquer que, plutôt que d'être un sous-problème de la PNL, l'AS est plutôt une mini-version de la PNL complète ou un cas spécial de la PNL complète. En d'autres termes, chaque sous-problème du NLP est également un sous-problème d'AS, et vice versa. La raison en est que l'AS touche à tous les domaines fondamentaux du NLP, tels que la sémantique lexicale, la résolution des coréférences, la désambiguïsation du sens des mots, l'analyse du discours, l'extraction d'informations et l'analyse sémantique [1].

En ce sens, l'AS offre une excellente plate-forme pour tous les chercheurs en TAL afin de réaliser des progrès tangibles et ciblés sur tous les fronts du TAL, avec le potentiel d'avoir un impact énorme sur la recherche et la pratique. Un chercheur en TAL de n'importe quel domaine peut commencer à résoudre un problème correspondant à l'AS sans changer de sujet ou de domaine de recherche. La seule chose qu'il doit changer est le corpus, qui doit être un corpus d'opinion.

En général, l'AS est un problème d'analyse sémantique, mais elle est très ciblée et limitée parce qu'un système d'AS n'a pas besoin de "comprendre" entièrement chaque phrase ou document ; il doit seulement en comprendre certains aspects - par exemple, les opinions positives et négatives et leurs cibles. Grâce à certaines de ses caractéristiques particulières, l'AS permet d'effectuer des analyses linguistiques beaucoup plus approfondies afin d'obtenir de meilleures informations sur le NLP que dans le cadre général, car la complexité du cadre général du NLP est tout simplement écrasante.

I.3 Analyse des sentiments à base d'aspects

Les premières approches de l'analyse du sentiment ont traité la tâche comme un problème de classification au niveau de la phrase ou du document, où le sentiment global du texte entier ou de la phrase a été déterminé. Toutefois, cette approche ne permettait pas de saisir les opinions nuancées sur des aspects ou des entités spécifiques mentionnées dans le texte. L'ABSA est apparue comme une réponse à cette limitation, visant à fournir une compréhension plus détaillée du sentiment en l'associant à des aspects ou des caractéristiques particulières. L'ABSA a gagné en importance grâce à ses applications permettant de comprendre les opinions des utilisateurs sur des produits, des services ou des sujets d'une manière plus granulaire Selon[12], le problème général de AS comprend deux composantes clés : la cible et le sentiment. Dans le cadre de l'ABSA, la cible peut être décrite soit par une catégorie d'aspect c, soit par un terme d'aspect a, tandis que le sentiment implique à la fois une expression d'opinion détaillée - le terme d'opinion o - et une orientation générale du sentiment - la polarité p.

Ces quatre éléments de sentiment constituent l'axe central des recherches en ABSA[13]

- **La catégorie d'aspect (c)** définit un aspect unique d'une entité et est censée appartenir à un ensemble de catégories C, prédéfini pour chaque domaine spécifique.
- **Le terme d'aspect (a)** est la cible de l'opinion qui apparaît explicitement dans le texte,
- **Le terme d'opinion (o)** est l'expression utilisée par l'émetteur de l'opinion pour exprimer son sentiment à l'égard de la cible.
- **La polarité du sentiment(p)** décrit l'orientation du sentiment à l'égard d'une catégorie ou d'un terme d'aspect. Elle est généralement positive, négative ou neutre.

Les terminologies utilisées dans la littérature ABSA sont souvent interchangeables, mais elles peuvent parfois avoir des significations différentes selon le contexte. Par exemple, cible de l'opinion, cible, aspect, entité sont souvent utilisés pour désigner la cible sur laquelle porte l'opinion. Cependant, selon le contexte, cela peut désigner une catégorie d'aspect ou un terme d'aspect, ce qui peut provoquer des confusions inutiles et rendre les revues de littérature incomplètes [11].

Les terminologies les plus communément acceptées tout en veillant à bien distinguer

les concepts similaires ont été adopté par divers système ABSA . Ainsi, comme défini ci-dessus, le terme d'aspect et catégorie d'aspectsont utilisée pour différencier les deux formes d'aspects, et cible ou aspect ne sont utilisée que comme expressions générales pour désigner la cible d'une opinion. Prenons l'exemple suivant :

"Le cours est obsolète mais le professeur est excellent".

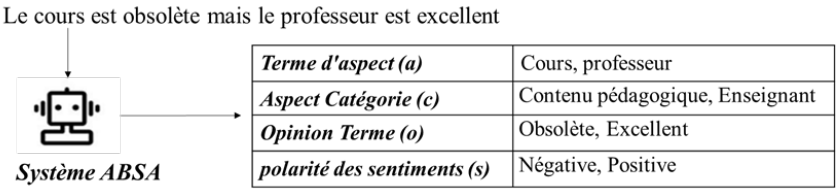

Figure I.2 : Quatre éléments de sentiment clés en l'ABSA

Dans cette phrase, la polarité du cours et du professeur est respectivement négative et positive. L'analyse pour comprendre la polarité d'une phrase, il est important de comprendre son contenu ainsi que son aspect. Il est donc essentiel de comprendre le contexte en reconnaissant l'aspect avant d'attribuer une polarité à une phrase donnée. Afin de déterminer la polarité d'une phrase, l'ABSA comprend deux étapes principales : l'extraction des aspects et la classification des sentiments. **Dans la phase d'extraction d'aspect**, le but est d'identifier les aspects ou caractéristiques mentionnés dans le texte qui sont pertinents pour l'AS. Cette étape repose souvent sur des techniques telles que la reconnaissance d'entités nommées ou l'analyse syntaxique. Une fois les aspects sont identifiés, **la phase de classification des sentiments** attribue des polarités de sentiment (positif, négatif ou neutre) à chaque aspect, indiquant le sentiment associé à cette caractéristique particulière. L'application de l'ABSA est particulièrement bénéfique dans les domaines où l'analyse de plusieurs aspects d'un produit ou d'un service est cruciale.

En combinant les éléments a (aspect), c (catégorie), o (opinion) et s (sentiment), l'ABSA (Aspect-Based Sentiment Analysis) se décline en plusieurs sous-tâches, chacune visant à capturer les différentes dimensions des opinions exprimées dans les textes. **Les tâches simples** incluent l'extraction des termes d'aspect (ATE), des termes d'opinion (OTE), ainsi que la détection des catégories d'aspect (ACD). Des variantes combinées, telles que l'extraction conjointe d'aspects et d'opinions (AOCE), ou l'extraction d'opinions orientées en fonction d'un aspect donné (AOOE), permettent d'approfondir l'ana-

lyse. D'autres approches visent à classifier directement le sentiment associé à un aspect (ABSC) ou à une catégorie (COSC). **Les tâches de type paire** consistent à extraire des relations binaires telles que les paires aspect-opinion (AOPE), aspect-sentiment (ASPE) ou catégorie-sentiment (CSPE). Viennent ensuite **les tâches triplet**, qui enrichissent l'analyse en associant simultanément un aspect, une opinion et un sentiment (comme dans ASTE ou AOSTE), ou en y intégrant également la catégorie (ACSTE). Enfin, **les tâches quadruples** (ACOSQE) représentent l'approche la plus complète, visant à extraire simultanément les quatre composantes : aspect, catégorie, opinion et polarité du sentiment. Ces sous-tâches peuvent être réalisées à l'aide de méthodes d'extraction, de classification, ou d'une combinaison des deux[11]tableau I.1.

I.3.1 Évolution de l'ABSA

Avec l'essor de l'apprentissage profond (DL) et ses performances remarquables dans divers traitements du langage naturel (NLP), les modèles basés sur CNN, les modèles RNN utilisant LSTM et GRU, les modèles LSTM basés sur l'attention, et les modèles à base de transformateurs sont largement utilisés pour l'AS basée sur les aspects dans des ensembles de données de pointe [14].

Les deux grandes catégories de modèles ABSA sont les modèles d'apprentissage profond (DL) et les modèles d'apprentissage automatique (ML)[11] [14]

Type	Abrév.	Nom de la tâche	Entrée	Sortie	Méthode
Simple	ATE	Extraction de termes d'aspect	P : Phrase	a	Extraction
	OTE	Extraction de termes d'opinion	P	o	Extraction
	ACD	Détection de catégories d'aspect	P	c	Classification
	AOCE	Co-extraction aspect-opinion	P	a, o	Extraction
	AOOE	Extraction d'opinion orientée aspect	P + a	o	Extraction
	ABSC	Classification de sentiment basée sur l'aspect	P + a	s	Classification
	COSC	Classification de sentiment orientée catégorie	P + c	s	Classification
Paire	AOPE	Extraction de paires opinion-aspect	P	(a, o)	Extraction & Classification
	ASPE	Extraction de paires aspect-sentiment	P	(a, s)	Extraction & Classification
	CSPE	Extraction de paires catégorie-sentiment	P	(c, s)	Extraction & Classification
Triplet	ACSTE (TASD)	Extraction de triplets catégorie-aspect-sentiment ou Détection de sentiment ciblé sur un aspect	P	(a, c, s)	Extraction & Classification
	AOSTE (ASTE)	Extraction de triplets opinion-aspect-sentiment ou aspect-sentiment-opinion	P	(a, o, s)	Extraction & Classification
Quad	ACOSQE	Extraction de quadruplets catégorie-opinion-aspect-sentiment	P	(a, c, o, s)	Extraction & Classification

Tableu I.1 : Classification des tâches d'analyse de sentiments

I.3.1.1 Techniques ML

L'apprentissage automatique relève du domaine de l'intelligence artificielle (IA) et se concentre sur le développement de modèles et d'algorithmes statistiques. L'apprentissage automatique permet aux ordinateurs d'apprendre et de faire des prédictions ou de prendre des décisions sans nécessiter de programmation explicite. L'application des techniques de ML a marqué une avancée significative dans l'ABSA. Des modèles d'apprentissage supervisé, tels que Naïve Bayesian, Support Vector Machine (SVM) et Artificial Neural Networks (ANN), ont été utilisés pour la classification des soustâches de l'ABSA. L'ingénierie des caractéristiques a joué un rôle crucial dans ces modèles, les chercheurs extrayant des caractéristiques pertinentes du texte, telles que les n-grammes, les sacs de mots, parties de discours, les structures syntaxiques et les lexiques de sentiments.

Les performances de ces méthodes dépendent fortement des caractéristiques élaborées manuellement, qui malheureusement demandent beaucoup de travail et sont comparativement moins efficaces. C'est pourquoi les chercheurs ont exploré des techniques révolutionnaires de DL pour les sous-tâches ABSA au cours des dernières années.

I.3.1.2 Techniques DL

Propulsés par les progrès rapides des techniques de réseaux neuronaux, les réseaux neuronaux profonds (RNP) ont atteint un succès notable dans diverses applications.

Ce succès a conduit la recherche ABSA à passer des techniques basées sur les caractéristiques aux méthodes DNN. L'essor des architectures DNN, telles que les réseaux neuronaux récurrents (RNN), les réseaux neuronaux convolutifs (CNN) et les transformateurs, a entraîné un changement de paradigme dans l'ABSA. Ces modèles se sont révélés plus performants pour capturer les informations contextuelles et pour apprendre des modèles complexes dans les données textuelles. Les modèles LSTM et CNN basés sur l'attention ont été largement proposés pour l'ABSA.

Ce passage des méthodes traditionnelles basées sur les caractéristiques à diverses architectures de réseaux neuronaux reflète les progrès substantiels de la recherche sur les réseaux neuronaux et leur applicabilité aux sous-tâches de l'ABSA. Chacune de ces approches basées sur les réseaux neuronaux apporte des atouts uniques, contribuant ainsi à l'évolution du paysage de l'ABSA. Toutefois, les modèles LSTM et CNN présentent deux limites principales. Tout d'abord, le modèle formé avec un ensemble de données n'était pas performant pour d'autres ensembles de données et d'autres domaines. Deuxièmement, ces modèles peinaient à atteindre des performances remarquables en termes de précision. C'est pourquoi les modèles à base de transformateurs sont apparus ces dernières années comme des solutions proposées pour les tâches ABSA.

I.3.1.3 Modeles basés sur des transformateurs

Le développement d'architectures de transformateurs, comme le démontrent des modèles tels que BERT (Bidirectional Encoder Representations from Transformers), a constitué une autre avancée significative dans le domaine de l'ABSA. Les modèles de transformateurs ont donné d'excellents résultats dans l'AS et d'autres tâches de traitement du langage

naturel, surpassant les méthodes antérieures dans la capture des dépendances à longue portée et des informations contextuelles. L'affinement des modèles de transformateurs pré-entraînés pour les tâches d'AS basées sur les aspects est devenu une pratique courante, permettant aux modèles de tirer parti d'un pré-entraînement à grande échelle sur diverses données textuelles. Les transformateurs possèdent un mécanisme d'auto-attention qui leur permet de reconnaître les dépendances entre les mots dans un texte. Contrairement aux réseaux neuronaux récurrents (RNN), qui traitent les séquences de manière séquentielle, les transformateurs peuvent traiter les mots en parallèle. Le mécanisme d'auto-attention peut être représenté par l'équation suivante[14] :

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}V\right)$$

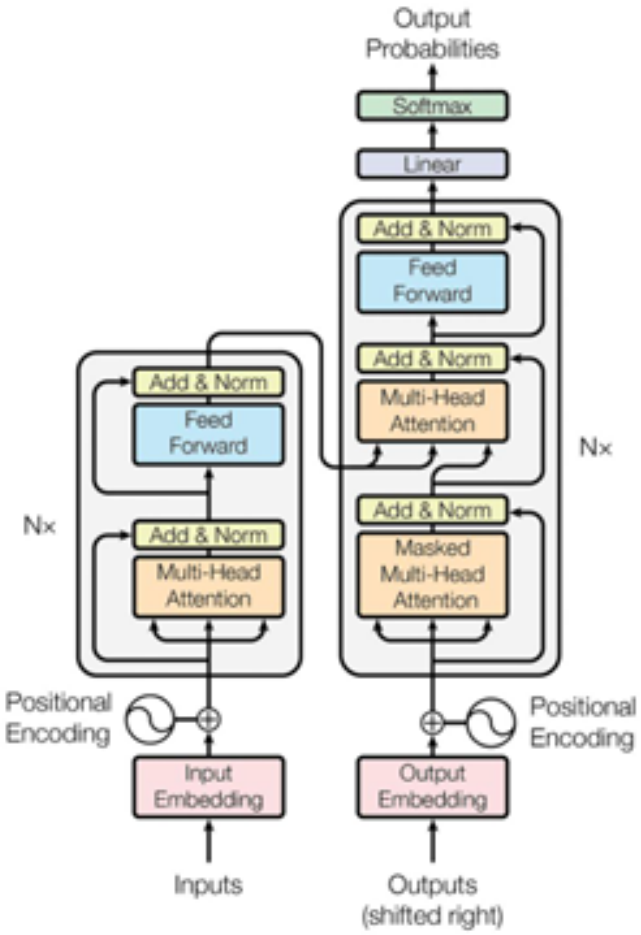

Figure I.3 : Architecture du transformateur [2]

La figure I.3 illustre comment un codeur et un décodeur constituent l'architecture d'un transformateur. Dans le contexte de l'ABSA, le codeur prend le texte(avis) et l'aspect en

entrée et les code dans une représentation à haute dimension. D'autre part, le décodeur produit la sortie, telle que la polarité du sentiment. Le mécanisme d'auto-attention des transformateurs permet au codeur et au décodeur d'enregistrer les dépendances globales de l'examen et de faciliter un flux d'informations efficace.

1. BERT : (Transformers Bidirectional Encoder Representations) est un modèle puissant basé sur les transformateurs qui a révolutionné les tâches de traitement du langage naturel. Il comprend la compréhension bidirectionnelle du contexte grâce à l'architecture des transformateurs. BERT a atteint des performances exceptionnelles dans divers domaines du NLP tels que l'AS, la classification des textes, la réponse aux questions et la traduction automatique. La compréhension contextuelle du langage par BERT a permis d'améliorer les moteurs de recherche, les chatbots et les assistants virtuels, permettant des réponses plus précises et de meilleures expériences pour l'utilisateur. Bien que les modèles basés sur les BERT aient connu un succès remarquable dans l'ABSA, des défis persistent, notamment la nécessité d'un pré-entraînement spécifique au domaine, le traitement des contextes ambigus et la prise en compte de la rareté des données étiquetées pour des domaines spécifiques. Par conséquent, pour surmonter ces limitations, les grands modèles de langage (LLM) tels que PaLM API, GPT-3 et flan-t5 peuvent être utilisés pour les sous-tâches ABSA [14][2].

2. **GPT :** est un LLM de pointe et probablement un moment clé dans le domaine du traitement du langage naturel (NLP). Développé par l'OpenAI, le GPT utilise une architecture de transformation pour générer un texte cohérent et contextuellement précis. Le GPT est capable de produire des textes fluides tels que la traduction, la génération de texte et la réponse à des questions d'une manière semblable à celle des humains, car il a été affiné en utilisant un vaste corpus de données textuelles pour diverses tâches de TAL. GPT1, GPT-2 et GPT-3 ont été mis à disposition par OpenAI. En outre, la dernière innovation, ChatGPT, a stupéfié tout le monde par ses caractéristiques sophistiquées et s'est rapidement hissée au sommet des médias sociaux et des systèmes d'information. Le ChatGPT est capable d'accomplir diverses tâches à partir d'ensembles de données d'avis de clients, telles que l'AS, le résumé de texte, la classification de texte, la détection de mots offensants, etc. Bien que le ChatGPT ait démontré des capacités remarquables de génération de langage et qu'il ait été appliqué à diverses tâches de NLP, sa mise en œuvre pour ABSA reste limitée et nécessite une étude scientifique plus approfondie. Les derniers

modèles LLM n'ont pas encore été explorés en profondeur dans le domaine ABSA dans les études susmentionnées [14].

I.3.2 Datasets de Références pour ABSA

Les SemEval (Semantic Evaluation) sont des compétitions internationales organisées chaque année pour évaluer les progrès des systèmes d'analyse sémantique. Parmi ces compétitions, plusieurs tâches se concentrent sur AS, notamment les aspects, les opinions, et les relations associées. Les SemEval 2014, 2015 et 2016 ont particulièrement contribué à l'ABSA [15].

- SemEval 2014 Task 4 : Ce dataset couvre deux domaines principaux :
    - Restaurants : Avis client annotées pour les aspects (exemple : "service", "nourriture", "prix").
    - Laptops : avis produit annotées pour les aspects spécifiques à l'électronique (exemple : "batterie", "écran", "design").
- SemEval 2015 Task 12 : ABSA étendu
    - Ajout d'un nouveau domaine, comme les hôtels et les réseaux sociaux.
    - Incorporation de phrases plus complexes avec des opinions indirectes.
    - Annotation des opinions implicites.
- SemEval 2016 Task 5 : Domain Adaptation Challenge
    - Tâche centrée sur l'adaptation des modèles entre différents domaines.
    - Inclusion de quadruples sous forme implicite pour évaluer les limites des systèmes d'extraction.
    - Adaptation pour les quadruples et jetons implicites

Les datasets SemEval, bien qu'initialement centrés sur les aspects explicites, peuvent être enrichis pour supporter les quadruples et les jetons implicites via les applications des SemEval Datasets pour ABSA avancé :

- Benchmarking : Utilisés comme référence pour évaluer les modèles d'extraction de quadruples.
- Fine-tuning : Modèles pré-entraînés peuvent être ajustés sur ces datasets enrichis pour capturer des jetons implicites.

- Domain adaptation : Appropriés pour des scénarios cross-domain où l'implicite joue un rôle clé.

Les datasets SemEval restent incontournables pour toute recherche en ABSA. Avec des enrichissements ciblés pour les quadruples et les jetons implicites, ils permettent de répondre aux défis actuels de l'analyse sémantique complexe dans des textes réels

I.4 Conclusion

Ce chapitre a présenté d'une part les fondements de L'AS à gros grains, et d'autre part les principes de l'analyse à grains fins, ou ABSA. L'analyse classique permet une compréhension globale des opinions, tandis que l'ABSA offre une lecture plus détaillée en identifiant les sentiments associés à des aspects précis. Cette progression conceptuelle ouvre la voie à une exploration plus avancée avec l'IA, les LLMs et des méthodes associées à l'ABSA dans les chapitres suivants.

CHAPITRE II

Les Grands Modèles de Langage

II.1 Introduction

Les Grands Modèles de Langage (LLMs), basés sur l'architecture Transformer, ont révolutionné l'intelligence artificielle et le traitement automatique du langage naturel. Grâce à leur entraînement sur de vastes corpus textuels et à des technologies telles que l'attention et l'apprentissage par transfert, ils excellent dans de nombreuses tâches linguistiques. Ce chapitre présente leur architecture, leurs applications, les techniques d'optimisation et les méthodes d'évaluation, afin de mieux comprendre leur fonctionnement et leur potentiel.

II.2 Les Grands Modèles de Langage

LLM peut être défini comme une fonction qui recherche, en considérant une série de tokens (tels que des mots, des fragments de mots, des signes de ponctuation, des émojis, etc.), pour prédire quels tokens sont les plus suceptibles d'apparaitre par la suit.

Aujourd'hui, de nombreux LLM sont accessibles au public. Gemini [1] de Google, plus

1. https://gemini.google.com.

efficace et compact avec jusqu'a 540 milliards de paramètres, est accessible gratuitement via un navigateur Web et une API, avec un ensemble des données d'entraînement diversifié [16] . Le LLaMA [2] de Meta, destiné à l'avancement de la recherche en IA, propose divers modèles avec jusqu'à 70 milliards de paramètres accessibles aux chercheurs via l'application [17] . Les modèles GPT d'OpenAI utilisent l'architecture de transformateur pour la génération de sortie dynamique [2] , avec différentes versions accessibles différemment : GPT3.5 est gratuit via une interface Web [3], tandis que GPT-4 nécessite un abonnement, l'utilisation de l'API étant également payante à l'utilisation en fonction de la tokenisation pour le traitement du langage naturel.

Les API des LLM offrent aux développeurs un moyen d'intégrer les capacités avancées de ces modèles dans leurs propres applications. Ces interfaces peuvent être utilisées avec n'importe quel langage de programmation capable d'effectuer des requêtes HTTP. En pratique, une requête est envoyée à l'API avec une invite textuelle et divers paramètres permettant de configurer le comportement du modèle, comme le choix de la version du LLM ou le réglage de la température, qui influence le degré de créativité ou de variabilité dans la réponse générée.

II.2.1 Fonctionnement des grands modèles de langage

Les LLMs s'appuient sur des techniques d'apprentissage profond et exploitent d'importants volumes de données textuelles. Leur architecture repose principalement sur le modèle Transformer, et en particulier sur sa variante générative pré-entraînée (Generative Pretrained Transformer, GPT), reconnue pour son efficacité dans le traitement de données séquentielles telles que les textes.

Ces modèles sont constitués de nombreuses couches de réseaux neuronaux, dont les paramètres sont ajustés au cours de l'entraînement afin d'optimiser leur capacité de généralisation. Un composant fondamental de cette architecture est le mécanisme d'attention, qui permet au modèle de pondérer dynamiquement les différentes parties de l'entrée en fonction de leur pertinence contextuelle. Ce mécanisme, également organisé en couches, permet au modèle de se concentrer sélectivement sur les segments les plus informatifs des données, améliorant ainsi la compréhension du contexte et la cohérence des réponses

2. https://llama.meta.com.
3. https://chat.openai.com.

générées.

Les principaux usages des LLM en entreprise sont :

- **Production de texte :** Les LLMs sont capables de générer automatiquement des contenus textuels de longueur moyenne ou grande tels que des courriels, des articles de blog ou des rapports, en réponse à des prompts personnalisables. La génération augmentée par récupération (RAG) constitue une approche particulièrement efficace dans ce contexte, en enrichissant les réponses générées avec des connaissances extraites dynamiquement de sources externes.
- **Synthèse de documents :** Ces modèles peuvent résumer efficacement des textes longs, comme des articles scientifiques, des rapports d'entreprise ou des historiques clients, en produisant des synthèses adaptées à différents formats de restitution tout en conservant les informations clés.
- **Assistants conversationnels intelligents :** Les LLMs sont utilisés dans la conception de chatbots capables de répondre aux requêtes des utilisateurs en langage naturel, de réaliser des tâches en arrière-plan (back-end) et de fournir un accès automatisé à des services d'information client.
- **Génération et assistance au codage :** Les LLMs accompagnent les développeurs dans la rédaction de code, la détection d'erreurs et de vulnérabilités, ainsi que dans la traduction de code entre différents langages de programmation, facilitant ainsi le développement logiciel.
- **Analyse des sentiments :** Grâce à leur compréhension fine du langage, les LLMs peuvent analyser le ton et les émotions exprimées dans les textes clients, ce qui permet aux entreprises de mieux cerner la perception de leur marque et d'adapter leur stratégie de communication.
- **Traduction automatique multilingue :** Les LLMs proposent une couverture linguistique étendue, assurant des traductions fluides et contextuellement appropriées, ce qui renforce leur pertinence dans des contextes internationaux ou multilingues [18].

II.2.2 Terminologies sur LLM

La compréhension des LLMs repose sur la maîtrise de certains termes techniques essentiels, qui reflètent à la fois leur architecture, leur échelle et leur fonctionnement [19] :

-Paramètres : Les LLMs sont constitués de plusieurs milliards, voire de centaines de milliards de paramètres. Ces paramètres représentent les poids ajustés lors de l'entraînement du modèle, et déterminent sa capacité à capturer des relations complexes dans les données linguistiques. Leur nombre constitue un indicateur direct de la complexité et de la puissance du modèle.

- Tokens : Les données textuelles utilisées pour l'entraînement sont décomposées en unités appelées tokens, qui peuvent représenter des mots, des sous-mots ou des caractères. Les corpus d'entraînement des LLMs incluent généralement des billions de tokens, permettant au modèle d'acquérir une connaissance approfondie du langage.

- FLOP (Floating Point Operations) : Dans le contexte des architectures Transformer, l'entraînement d'un LLM nécessite environ six opérations en virgule flottante (FLOP) par paramètre pour chaque token traité. Cette métrique permet d'estimer la charge computationnelle globale requise pour entraîner un modèle.

- Calcul d'entraînement et émergence des capacités : Les recherches indiquent que certaines capacités complexes des LLMs, telles que le raisonnement logique ou la compréhension contextuelle avancée, ne se manifestent qu'au-delà d'un certain seuil de calcul. Cette observation suggère que l'émergence de comportements intelligents est liée à l'échelle du modèle et à la quantité de ressources mobilisées durant son entraînement.

II.2.3 Principaux défis posés par les LLM

LLMs, reposant sur l'architecture Transformer développée par Google, se distinguent par leur capacité à traiter et générer du langage naturel à partir de vastes corpus textuels, comptant des milliards de mots. Bien que cette approche leur confère une compréhension remarquable du langage humain, elle soulève plusieurs défis majeurs en matière de scalabilité et de coût.

Les modèles de grande envergure tels que Gemini et Gemma de Google, ou encore les modèles GPT d'OpenAI, exigent des ressources computationnelles considérables pour l'entraînement et l'inférence, ce qui complique leur déploiement à grande échelle. En outre,

certaines recherches suggèrent que les performances des LLMs tendent à stagner malgré l'augmentation continue de la puissance de calcul, remettant ainsi en question la corrélation directe entre taille du modèle, ressources mobilisées et qualité des résultats obtenus[19].

Au-delà des aspects techniques, les LLMs soulèvent d'importantes préoccupations d'ordre éthique. Ces modèles peuvent reproduire, voire amplifier, les biais présents dans leurs données d'entraînement, ce qui pose des questions cruciales quant à leur neutralité, leur équité et leur fiabilité.

Par ailleurs, les LLMs sont susceptibles de générer des informations inexactes, incohérentes, voire totalement erronées un phénomène connu sous le nom d'« hallucination ». Ces dérives rendent nécessaire une vérification humaine rigoureuse des contenus produits, en particulier dans des contextes sensibles où la précision de l'information est critique [19].

II.2.4 Architecture LLM

L'architecture encodeur-décodeur [20], [21] se compose de deux parties principales : l'encodeur et le décodeur (figure I.3) Chaque partie a un rôle spécifique dans le traitement de l'information.

L'encodeur [22] Prend en entrée une séquence de mots ou de phrases et les convertit en une série de représentations vectorielles. Ces représentations contiennent les informations contextuelles de l'entrée. Dans le modèle Transformer, l'encodeur est constitué de plusieurs couches identiques empilées les unes sur les autres. Chaque couche comprend principalement deux sous-couches : une sous-couche de self-attention multi-têtes et une sous-couche de réseau feed-forward. Les connexions résiduelles autour de chaque sous-couche, suivies d'une normalisation, permettent de préserver l'information et d'accélérer la convergence du modèle. Le décodeur [23] génère la sortie en se basant sur les représentations fournies par l'encodeur. Il est également composé de plusieurs couches, similaires à celles de l'encodeur, mais avec une couche supplémentaire d'attention qui aide à se concentrer sur les parties pertinentes de l'entrée de l'encodeur pour chaque étape de la génération de la sortie. Cette attention inter-couches permet au décodeur de prendre en compte l'ensemble de l'entrée tout en générant chaque mot de sortie, améliorant ainsi la cohérence et la pertinence du texte génère.

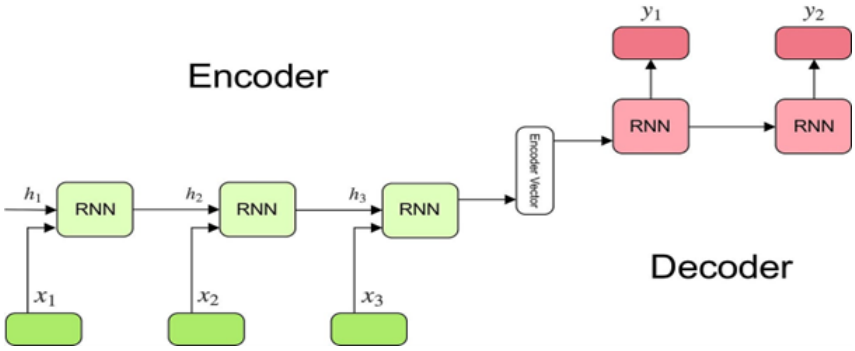

Figure II.1 : : Les modèles encodeur décodeur [3]

b. Composants de base des LLM

L'architecture transformer repose principalement sur les couches d'etaillées ci-dessous. La figure II.2 illustre l'interaction entre ces couches.

-Couche d'Embedding

Chaque mot ou token d'entrée est converti en un vecteur dense qui capture sa signification sémantique et syntaxique. Ces embeddings sont essentiels pour permettre au modèle de traiter le texte de manière significative.

-Mechanism d'Attention (self attention mechanism) Au cœur des LLMs se trouve le mécanisme de self-attention [24], qui permet au modèle de pondérer l'importance relative des différents mots dans une phrase. Il fonctionne en transformant la séquence d'entrée en trois vecteurs (query, key, value), et calcule une somme pondérée des valeurs, basée sur la similarité entre les vecteurs de query et de key. Ce processus aide le modèle à se concentrer sur les informations pertinentes et à capturer des dépendances à longue portée.

- Couches Cachees (Feed-Forward)

Après le traitement par self-attention, les sorties sont passées à travers des réseaux feed-forward qui modifient les représentations. Ces couches sont répétées plusieurs fois dans le modèle, permettant la conceptualisation d'abstractions de plus haut niveau.

-Attention Multi-Têtes (Multi-head Attention)

L'attention multi-têtes améliore le mécanisme de self-attention en divisant les embeddings en plusieurs sous-ensembles traites parallèlement. Cette technique permet au modèle de se concentrer simultanément sur différentes parties d'une séquence d'entrée, enrichissant ainsi la représentation contextuelle du texte.

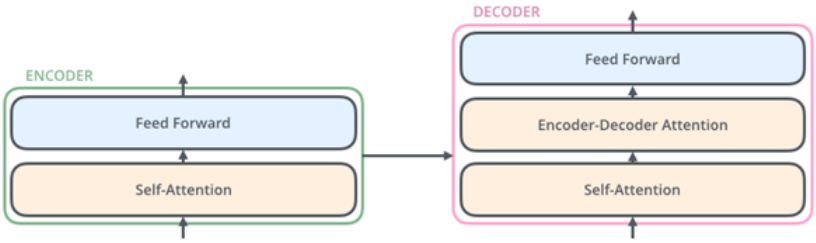

Figure II.2 : Les couches de l'architecture transformer [4]

II.2.5 Optimisation des performances des LLMs

Les LLMs, constituent une avancée majeure dans le domaine du TALN. Toutefois, pour en exploiter pleinement le potentiel, il est nécessaire de recourir à des techniques d'optimisation spécifiques visant à améliorer leurs performances et à en élargir les cas d'usage. Parmi ces techniques figurent le prompt engineering, le fine-tuning, l'apprentissage par renforcement (reinforcement learning, RL) et la RAG :

II.2.5.1 Ingénierie des prompts (Prompt Engineering)

Le prompt engineering est une discipline émergente qui vise à concevoir et à optimiser des prompts dans le but de maximiser l'efficacité des LLMs dans une variété d'applications et de contextes de recherche. Cette pratique s'avère cruciale pour tirer pleinement parti des capacités des LLMs, tout en mettant en évidence leurs limites potentielles. Chercheurs et développeurs y ont recours pour améliorer les performances des modèles sur des tâches diverses, allant de la simple réponse à des questions factuelles à des formes de raisonnement plus complexes, telles que le raisonnement arithmétique.

L'utilisation de prompts simples peut produire des résultats significatifs, mais la qualité de ces derniers dépend largement de la quantité et de la précision des informations fournies au modèle. Un prompt efficace peut inclure des éléments tels que des instructions spécifiques, des questions, du contexte pertinent, des entrées utilisateur, et des exemples illustratifs. Ces éléments sont cruciaux pour guider précisément le modèle, a obtenir des réponses plus précises et adaptées aux besoins spécifiques de l'utilisateur[24].

Le Prompt Engineering permet de tirer parti d'une diversité de techniques pour générer une infinité de contenus, allant d'articles de presse soigneusement rédigés à des poèmes adoptant un style et un ton spécifiques.

Chacune des techniques de prompts engineering, ci-dessous, possède des spécificités qui contribuent à diversifier et améliorer les réponses a générer par les LLM :

Few-Shot Learning Dans le domaine du TALN, le few-shot learning permet aux LLM envergure, qui ont été Pré-entraînes sur d'importants ensembles de données textuelles, de généraliser leur capacité à comprendre et à exécuter des taches connexes mais non vues auparavant avec seulement quelques exemples, sans mises à jour des poids. Cette méthode implique de donner au modèle K exemples de contexte et de complétion, suivis d'un exemple final de contexte pour lequel le modèle doit générer la complétion. K est généralement dans une fourchette de 10 à 100, selon le nombre d'exemples qui peuvent s'adapter dans la fenêtre de contexte du modèle[5].

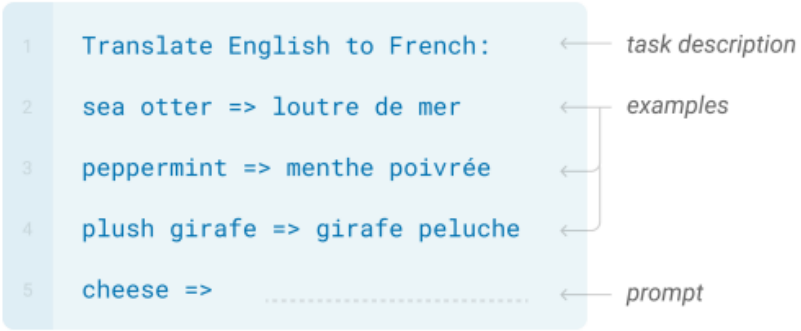

Figure II.3 : : Illustration d'un exemple en Few-shot [5].

-One-Shot Learning (Apprentissage avec un seul exemple) : Le One-Shot Learning est similaire au few-shot, il implique de donner un seul exemple au modèle, accompagné d'une description de la tâche, utile pour des taches nécessitant un style ou un ton spécifique.

Composantes du Few-Shot Learning :

- Description de la Tâche : Une brève explication de ce que le modèle est censé accomplir, par exemple, "Traduire de l'anglais vers le français".
- Exemple : Un exemple est fourni pour montrer au modèle le type de prédiction attendu,tels que "sea otter => loutre de mer".
- Prompt : Le début d'un nouvel exemple que le modèle doit compléter en générant le texte manquant, par exemple, "cheese => ".

L'exemple présente dans la Figure II.4 met en évidence le concept.

Figure II.4 : Ilustration d'un exemple en One-Shot [5].

- Zero-Shot Learning (Apprentissage sans exemple) : Le Zero-Shot Learning ne fournit aucune démonstration. Le modèle reçoit uniquement une instruction décrivant la tâche ce qui le rend idéal pour des tâche générales De même, l'exemple fourni dans la Figure II.5 illustre le principe du zero-shot.

Figure II.5 : Ilustration d'un exemple en Zero-shot [5].

II.2.5.2 Génération Augmentée par la recherche

La technologie de RAG représente une avancée majeure dans le domaine de l'intelligence artificielle, développée initialement par des chercheurs de Meta IA, elle consiste en une fusion innovante entre les modèles de génération de texte et les techniques de récupération d'informations. Cette approche permet aux LM de tirer parti des vastes bases de données ou corpus pour enrichir leurs réponses, offrant ainsi des réponses plus précises et informées. RAG combine efficacement les capacités de réponse directe d'un LM avec la richesse des informations disponibles dans des documents externes, ce qui est particulièrement utile dans des applications telles que les chabots, ou la précision et la pertinence de l'information sont cruciales. Cette méthode est essentielle pour améliorer l'interaction des utilisateurs avec les systèmes bases sur l'IA, en fournissant des réponses

non seulement cohérentes mais aussi profondément ancrées dans des faits vérifiables et des d´emails spécifiques.RAG combine deux composants néccessaires II.5 :

Composant de recherche (Retrieval Component) : Le composant de recherche est responsable de la recherche et de la sélection des informations pertinentes à partir d'une base de données ou d'un corpus de documents.

Ce composant utilise des techniques telles que l'indexation de documents, l'extension de requêtes et le classement pour identifier les documents les plus appropriés en fonction de la requête de l'utilisateur.

- Composant de génération (Generation Component) : Une fois les documents pertinents récupérés, le composant de génération prend le relais. Il s'appuie sur de grands modèles de langage (LLM) tels que GPT-3 pour traiter les informations récupérées et générer des réponses cohérentes et contextuellement précises.

Ce composant est responsable de la conversion des faits récupérés en réponses lisibles par l'homme.

La RAG implique souvent une boucle d'interaction entre les composants de récupération et de génération. La récupération initiale ne renvoie pas toujours la réponse parfaite, de sorte que le composant de génération peut affiner et améliorer la réponse de manière itérative en se référant aux résultats de la récupération.

La mise en œuvre réussie de cette approche nécessite souvent un réglage fin des LLM sur des données spécifiques au domaine. Le réglage fin adapte le modèle pour comprendre et générer du contenu pertinent pour le domaine de connaissances spécifique, améliorant ainsi la qualité des réponses.

Les modèles de récupération convertissent souvent les documents et les requêtes en représentations de l'espace latent, ce qui facilite la comparaison et le classement des documents en fonction de leur pertinence par rapport à une requête. Ces représentations sont cruciales pour une récupération efficace.

Les composants de RAG utilisent généralement des mécanismes d'attention. Les mécanismes d'attention aident le modèle à se concentrer sur les parties les plus pertinentes des documents d'entrée et des requêtes, améliorant ainsi la précision des réponses.

a. Avantages de l'approche RAG : Le principal avantage de cette approche est sa capacité à combiner les forces de la recherche et de la génération afin de fournir des réponses de haute qualité. L'approche de RAG offre plusieurs avantages pour répondre

aux questions :

- Accès à une large base de connaissances (Access to à Wide Knowledge Base) : Cela permet au modèle de fournir des réponses qui peuvent ne pas être présentes dans ses données de préformation, ce qui le rend très informatif.

- Compréhension contextuelle (Contextual Understanding) : Le composant de génération utilise le contexte fourni par les résultats de récupération pour générer des réponses qui sont non seulement factuellement exactes mais également contextuellement pertinentes. Cette compréhension contextuelle conduit à des réponses plus cohérentes et plus précises.

- Raffinement itératif (Iterative Refinement) : La boucle d'interaction entre la récupération et la génération permet au système d'affiner de manière itérative ses réponses. Si la réponse initiale est incomplète ou incorrecte, le composant de génération peut effectuer des recherches ou des clarifications supplémentaires en fonction des résultats de la récupération, ce qui conduit à des réponses améliorées.

- Adaptabilité à diverses requêtes (Adaptability to Diverse Queries) : La RAG peut gérer une large gamme de requêtes utilisateur, y compris des questions complexes et à multiples facettes. Il excelle dans les scénarios où les moteurs de recherche simples basés sur des mots-clés peuvent ne pas être à la hauteur [7].

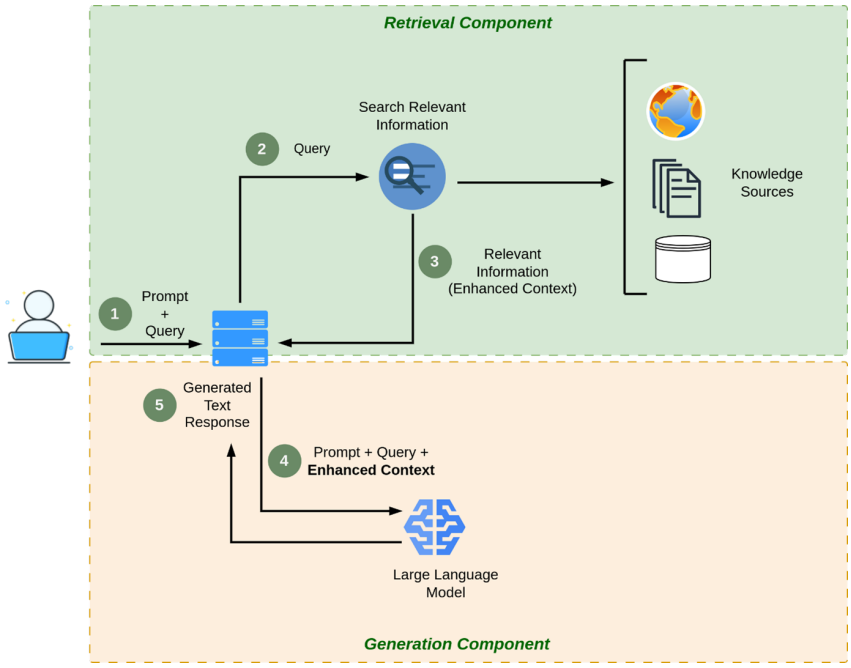

Figure II.6 : Architecture RAG [6].

b.Processus d'implémentation d'une architecture RAG

-Préparation des données :

Collecte des sources Cette étape consiste à rassembler l'ensemble des documents pertinents qui serviront de base de connaissances, tels que des manuels d'utilisation, fichiers PDF, bases de données produits, ou encore des FAQs.

-Indexation et stockage :

Segmentation et encodage (chunking et embeddings) : Les documents collectés sont divisés en unités textuelles plus petites appelées chunks. Chaque segment est ensuite converti en représentation vectorielle (embedding) à l'aide de modèles de langage pré-entraînés, permettant une compréhension sémantique fine.

Stockage dans une base vectorielle : Les embeddings ainsi générés sont stockés dans une base de données vectorielle, facilitant une recherche rapide et efficace des informations lors des requêtes ultérieures.

-Récupération d'information :

Traitement des requêtes utilisateur : Lorsqu'un utilisateur soumet une question, celle-ci est également transformée en embedding à l'aide du même modèle d'encodage utilisé pour les documents.

Recherche par similarité : Le système calcule les scores de similarité entre l'embedding de la requête et ceux des chunks disponibles, afin d'identifier les segments les plus pertinents à restituer.

-Génération de la réponse :

Synthèse par le modèle de langage : Les informations extraites sont combinées avec la requête initiale et transmises à un modèle de langage (LLM), qui génère une réponse cohérente, contextualisée, et adaptée au format souhaité.

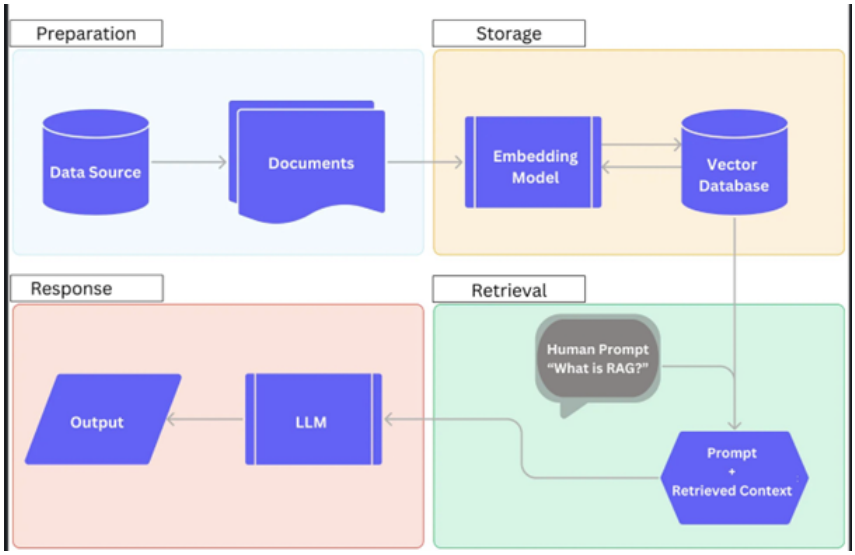

Figure II.7 : Processus d'implémentation des RAG [7].

Les étapes d'écrites ci-dessus (Figure II.7) correspondent à l'implémentation de la méthode RAG, qui est l'une des formes de base de la RAG. Il existe d'autres améliorations et variantes de RAG qui intègrent des techniques plus complexes et des optimisations supplémentaires pour répondre à des besoins spécifiques [25].

II.2.5.3 Défis de la génération augmentée par la récupération

: L'adoption de RAG représente une avancée significative dans le traitement du langage naturel et la recherche d'informations. Cependant, comme tout système d'IA complexe, RAG présente un ensemble de défis qui doivent être relevés pour exploiter pleinement son potentiel :

- Qualité et pertinence des données : RAG s'appuie fortement sur la disponibilité de données de haute qualité et pertinentes pour les tâches de récupération et de génération. Les défis dans ce domaine comprennent :

* **Données bruyantes :** des sources de données incomplètes, obsolètes ou inexactes peuvent conduire à la récupération d'informations non pertinentes, ce qui a un impact sur la qualité des réponses générées.
* **Biais et équité** : les biais présents dans les données de formation peuvent conduire à une récupération et une génération biaisées, perpétuant ainsi des stéréotypes ou des informations erronées.

- Complexité de l'intégration : L'intégration transparente des composants de récupération et de génération n'est pas triviale, car elle implique de relier différentes architectures et différents modèles. Les défis comprennent :

* **Compatibilité des modèles :** garantir que les modèles de récupération et de génération fonctionnent harmonieusement, en particulier lors de la combinaison de méthodes traditionnelles (par exemple, TF-IDF) avec des modèles neuronaux (par exemple, GPT-3).
* **Latence et efficacité :** équilibrer le besoin de réactivité en temps réel avec les ressources de calcul requises pour la récupération et la génération.

- Évolutivité

La mise à l'échelle des systèmes RAG pour gérer de gros volumes de données et de demandes d'utilisateurs peut s'avérer difficile :

* **Efficacité de l'indexation :** à mesure que le corpus de documents augmente, le maintien d'un index efficace et à jour devient crucial pour la vitesse de récupération.
* **Mise à l'échelle du modèle :** le déploiement de modèles neuronaux à grande échelle pour la récupération et la génération peut nécessiter des ressources de calcul substantielles.

- Paramètres d'évaluation

L'évaluation des performances des systèmes RAG présente des difficultés :

* **Absence de normes de référence :** Dans certains cas, il n'existe pas de norme de référence claire pour évaluer la pertinence et la qualité des documents récupérés.
* **Diversité des besoins des utilisateurs :** Les utilisateurs ont des besoins d'information variés, ce qui complique le développement d'indicateurs d'évaluation universels.

- Adaptation de domaine :

L'adaptation des systèmes RAG à des domaines ou secteurs spécifiques peut s'avérer complexe :

* **Connaissances spécifiques au domaine :** Intégration des connaissances et du jargon spécifique au domaine dans la récupération et la génération.
* **Disponibilité des données d'apprentissage :** Disponibilité des données d'apprentissage spécifiques au domaine pour affiner les modèles[26].

II.2.5.4 Ajustements fin (fine tuning)

Le fine-tuning est une technique de deep learning qui permet d'adapter un modèle pré- entraînée à des tâches spécifiques en ajustant ses paramètres. Cette méthode est utilisée pour affiner les capacités des modèles génératifs, afin qu'ils répondent mieux aux exigences spécifiques sans avoir recours à un entraînement complet. Elle est particulièrement utile pour les modèles ayant de nombreux paramètres, comme LLMs utilisés NLP. Enré-entraînant un modèle sur des données spécifiques, les entreprises peuvent améliorer l'efficacité du modèle, produisant des résultats plus rapides et mieux adaptés à des applications telles que le support client ou domaines bancaires.

Le fine-tuning est une des possibilités permettant de conserver un fort pouvoir de généralisation des LLM avec des objectifs métiers sur un secteur (banque, assurance, médical) précis. Il faut tout de même garder à l'esprit que le fine-tuning nécessite des compétences techniques en Data, ainsi que de la puissance de calcul et des GPU à disposition afin de pouvoir obtenir des résultats cohérents [7].

a. Processus et Méthodologies du Fine-tuning : L'ajustement fin, ou fine-tuning, commence par utiliser les poids d'un modèle pré-entraîné comme base pour un entraînement supplémentaire sur un ensemble de données spécifique. Ce processus peut impliquer diverses méthodes d'apprentissage, telles que supervisé, par renforcement, ou semi-supervisé, adaptées aux cas d'utilisation spécifiques du modèle.

Les données utilisées reflètent les tâches particulières et le domaine pour lequel le modèle est optimisé, permettant de transmettre des connaissances précises. Le fine-tuning peut mettre à jour tous les poids du réseau (Ajustement fin complet) ou se concentrer seulement sur certains (Paramètre Efficient Fine-Tuning), pour optimiser les performances tout en évitant que les connaissances antérieures seront perdues, ou bien mettre à jour les poids du modèle en se basant sur le score d'un modèle de récompense (Apprentissage par Renforcement à partir de Feedback Humain)

- Ajustement fin complet : Cette méthode actualise l'ensemble des poids du réseau neuronal, ce qui est conceptuellement similaire au pré -entraînement mais diffère principalement par l'état initial des paramètres et l'ensemble de données utilisé. Pour prévenir l'oubli catastrophique, on ajuste soigneusement les hyper-paramètres comme le taux d'apprentissage, assurant que le modèle conserve une bonne généralisation tout en s'adaptant

à de nouvelles tâches.

Le Supervised Fine-Tuning (SFT) est une forme spécifique d'ajustement fin complet qui utilise des ensembles de données annotées pour entraîner le modèle de manière plus ciblée.

- Parameter Efficient Fine-Tuning (PEFT) : Le fine-tuning complet est très gourmand en ressources de calcul, rendant cette approche souvent trop coûteuse et impraticable. Le PEFT consiste à actualiser uniquement un sous-ensemble de paramètres sélectionnes, réduisant ainsi les besoins en ressources computationnelles et en mémoire. Ces méthodes PEFT sont généralement plus stables que l'ajustement fin complet, en particulier pour les applications en traitement du langage naturel[27].

Cette approche inclut diverses techniques :

- Ajustement fin partiel : Mise à jour seulement de certains paramètres critiques, souvent les couches externes, tout en "gelant" les autres. Cela diminue la charge computationnelle tout en conservant la pertinence du modèle pour la tâche cible.
- Ajustement fin additif : Ajout de nouvelles couches ou paramètres au modèle existant, formant des modules adaptateurs qui sont ensuite entraînes indépendamment des poids pré-entraînes ,lesquels restent inchangés.
- Low Rank Adaptation (LoRA) : LoRA utilise la reparametrisation pour optimiser les poids,transformant les matrices de grande dimension en représentations de rang inférieur[28].

- Apprentissage par Renforcement à partir de Feedback Humain (RLHF) : Outre les méthodes traditionnelles de fine-tuning telles que l'ajustement fin complet et le Paramètre Efficient Fine-Tuning (PEFT), il est essentiel de reconnaître l'importance de l'Apprentissage par Renforcement à partir de Feedback Humain (RLHF). Le RLHF partage les principes de base du fine-tuning, en ce sens qu'il utilise également la mise à jour des poids d'un modèle pré-entraîne. Toutefois, il diffère dans sa capacité à ajuster dynamiquement ces poids en fonction des retours directs sur les réponses générées par le modèle. Cela permet une adaptation plus précise du modèle aux attentes et préférences des utilisateurs, en améliorant la qualité et la pertinence des interactions [29].

II.2.5.5 RAG vs. Fine-tuning

L'exploration des distinctions entre la RAG et le fine-tuning requiert une compréhension claire des contextes dans lesquels chaque approche se révèle la plus pertinente. Les études récentes soulignent que la méthode RAG s'avère particulièrement efficace pour l'intégration de connaissances actualisées dans les modèles, tandis que le fine-tuning permet d'optimiser les performances en affinant les connaissances internes du modèle, en adaptant le format des sorties, et en améliorant sa capacité à suivre des instructions complexes. Ces deux stratégies ne sont pas exclusives l'une de l'autre : elles peuvent être combinées dans un cadre itératif afin de renforcer l'utilisation des modèles, notamment pour des applications évolutives nécessitant un accès dynamique à des informations en constante évolution, ainsi qu'une génération de réponses personnalisées respectant des contraintes précises de format, de ton et de style. Par ailleurs, l'ingénierie des prompts constitue un levier complémentaire, permettant d'exploiter efficacement les capacités latentes du modèle. La Figure II.8 prsente les approches d'exploitation des LLM selon deux axes :l'axe des abscisses est le niveau des connaissances externes requis et l'axe des ordonnées concerne le degré d'adaptation du modèle. Les méthodes de Prompt Engineering (comme le few-shot) requièrent peu d'adaptation et peu de connaissances externes. Les approches RAG occupent une position intermédiaire, combinant récupération d'informations et génération sans entraînement lourd. Enfin, les techniques de Fine-tuning nécessitent à la fois une adaptation poussée du modèle et une forte intégration de connaissances externes, offrant des performances optimisées au prix d'un coût computationnel élevé.

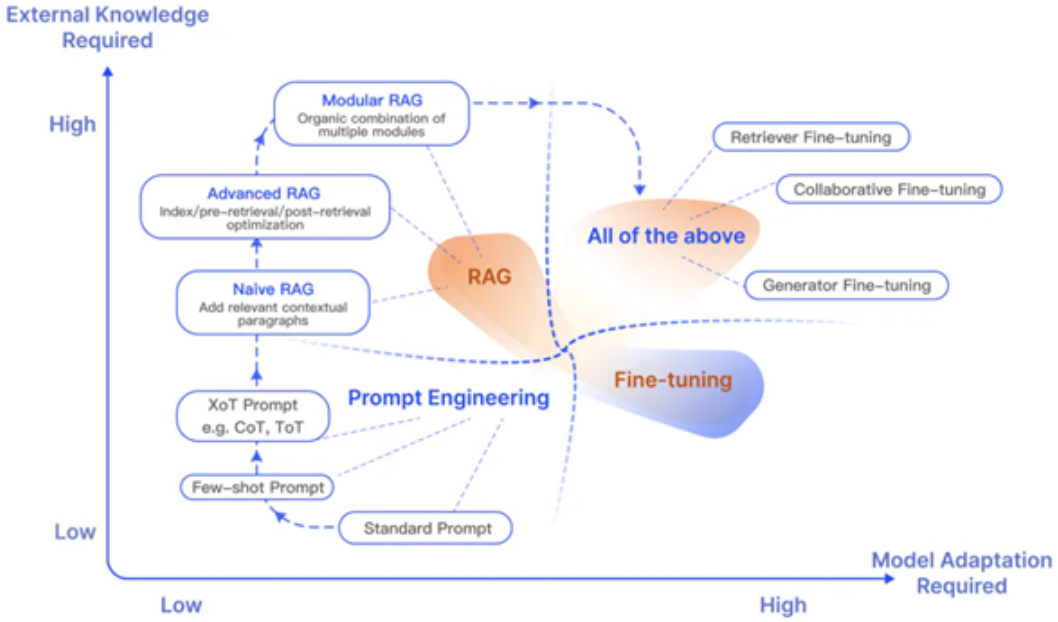

Figure II.8 : Comparaison des méthodes d'optimisation [8]

II.3 Evaluation des LLMs

L'évaluation des modèles de langage (LLMs) [30],[31] constitue une étape essentielle pour apprécier la qualité et la pertinence des réponses générées. Quelle que soit la méthode adoptée, cette évaluation peut être conduite selon deux approches : subjective, en sollicitant des experts pour juger de la qualité des réponses produites, ou objective, en recourant à des métriques quantitatives telles que la précision, la cohérence, la lisibilité, la pertinence, la perplexité, la robustesse ou encore la variabilité. Les résultats issus de ces évaluations permettent d'orienter les ajustements du modèle et de contribuer à l'amélioration continue de ses performances.

II.3.1 Types d' Évaluations

- **Évaluation Humaine** : Dans cette technique, un groupe de personnes évalue la qualité du texte généré par le modèle. Les évaluateurs humains peuvent être des experts du domaine ou des personnes ordinaires. Les évaluations sont généralement basées sur des critères tels que la clarté, la cohérence, la pertinence et la fluidité du texte.
- **LLM-comme-Juge :** Cette méthode utilise un autre LLM pour évaluer les sor-

ties du modèle testée[30]. Cette approche a été trouvée pour refléter largement les préférences humaines pour certains cas d'usage.

- **Évaluation Automatique :** Les métriques automatiques peuvent être utilisées pour évaluer la qualité du texte généré. Les métriques d'évaluation automatique les plus courantes incluent la précision, le rappel(**chapitre I** $\oint$ 3,6), et la perplexité. Ces métriques sont souvent utilisées pour évaluer des aspects spécifiques de la génération de textes, tels que la précision de la grammaire ou la cohérence thématique.

a. **Métriques d'évaluation automatique**

Les performances des LLMs parmi les mitriques qui évalue.

- ROUGE Score

Le ROUGE Score, ou Recall-Oriented Understudy for Gisting Evaluation, est un ensemble de métriques utilisées pour évaluer la qualité des résulttats. Il calcule le F1-score basé sur le nombre de mots consécutifs communs entre le texte de référence et le texte généré. Le score varie de 0 à 1, un score proche de zéro indiquant une faible similarité entre le candidat et les références, et un score proche de un indiquant une forte similarité. le ROUGE Score est basé sur le concept des n-grams, qui sont des séquences de n mots. Par exemple, un 1-gram est un seul mot, un 2-gram est une paire de mots, etc.

Les différents types de métriques ROUGE incluent :

- **ROUGE-N :** Mesure le chevauchement des n-grams entre les résultats système et de référence. Par exemple, ROUGE-1 se réfère au chevauchement des unigrammes (chaque mot), tandis que ROUGE-2 se réfère au chevauchement des bigrammes (deux mots consécutifs).
- **ROUGE-L :** Basé sur la longueur de la plus longue sous-séquence commune (LCS). Il calcule la moyenne harmonique pondérée combinant le score de précision et le score de rappel. Il ne nécessite pas de correspondances consécutives mais des correspondances en séquence.
- **ROUGE-W :** Une statistique basée sur LCS pondérée qui favorise les LCS consécutifs.
- **ROUGE-S :** Une statistique de co-occurrente basée sur les skip-bigrammes.

Un skip- bigramme est toute paire de mots dans l'ordre de la phrase.

- **ROUGE-SU :** Une statistique de co-occurrente basée sur les skip-bigrammes plus les unigrammes.

II.4 Conclusion

Ce chapitre s'est allé à exploré les fondements essentielles de l'IA avancée en abordant les stratégies de déploiement de RAG, de Prompt Engineering et de Fine-Tuning.

Dans la continuité des apports théoriques, le chapitre suivant met en lumière la façon dont la combinaison entre la récupération d'information et la génération de texte permet d'extraire et d'interpréter automatiquement les aspects, les opinions et les polarités exprimés dans les avis des utilisateurs.

CHAPITRE III

Mise en œuvre de la solution ABSA

III.1 Introduction

Ce chapitre décrit la mise en œuvre pratique de la solution ABSA en s'appuyant sur les capacités avancées du LLM exploité, en le guidant et l'adaptant au contexte ciblé via des techniques de prompting, de recherche et génération de sorties structurées et de fine-tuning.

Ce processus repose sur une conception minutieuse des invites (prompts), adaptée à chaque stratégie, afin d'assurer la cohérence, la précision sémantique et la pertinence des résultats générés.

Le tableau ci-dessous met en lumière notre choix de solution, en tenant compte des avantages et des limitations des trois principales approches d'exploitation des LLM présentées au chapitre 2. Le prompting est la méthode la plus simple, économique et rapide à mettre en œuvre, idéale pour des pratiques génériques. En revanche, le RAG offre un bon équilibre entre précision et flexibilité, notamment pour des cas où l'accès à des données à jour est important. Enfin, le Fine Tuning est la solution la plus puissante mais aussi la plus coûteuse et complexe, adaptée à des

besoins très spécialisés nécessitant un comportement ou une tonalité sur mesure du modèle. Le choix entre ces trois méthodes dépend donc étroitement des exigences du projet à réaliser, du budget disponible et du niveau de personnalisation souhaité :

Tableu III.1 : Critéres de choix entre les principales d'exploitation de LLM

Critères d'évaluation	Prompting	RAG	Fine Tuning
Exigence du projet	Connaissances générales basées sur le LLM	Nécessite des données externes à jour	Sujets hautement spécialisés. Informations avec un style et un ton bien définis
Budget	Moins coûteux en ressources de calcul	Peut-être coûteux car implique la mise en place et la maintenance d'une base de données	Coûteux en infrastructure et calcul. Nécessite des ressources supplémentaires pour l'entraînement
Délai de mise en œuvre	Déploiement rapide avec un minimum de configuration	L'intégration de sources de données externes prend du temps	Nécessite beaucoup plus de temps pour l'entraînement et l'optimisation
Complexité d'implémentation	Faible complexité	Complexité moyenne	Complexité élevée

Transparence et interprétabilité	Difficile d'identifier la source de l'information	Élevée. Facile d'identifier la source de l'information	Difficile d'identifier la source de l'information
Exemples d'usage	Création de contenu général, Q& A basique, développement de prototypes	Création de contenu dynamique, résolution de requêtes complexes, assistance à la recherche	Support client spécialisé, génération de contenu ciblé, analyse avancée de données

Pour l'approche adoptée, nous avons choisi d'explorer deux combinaisons complémentaires : d'une part, l'intégration de l'approche Prompting avec la méthode RAG, et d'autre part, son intégration avec la méthode Fine-Tuning. Ce double choix stratégique vise à tirer parti de la flexibilité du Prompting, tout en exploitant la puissance de recherche contextuelle de RAG et la capacité de spécialisation du Fine-Tuning. Les deux solutions seront mises en œuvre pour la génération automatique de tuples ABSA, permettant ainsi de comparer leur efficacité respective dans l'analyse fine d'opinions structurées. Les détails de ces intégrations seront décrits dans les sections suivantes :

III.2 Architecture globale de la solution ABSA

L'AS traditionnelle consiste à évaluer et à catégoriser un texte, tel qu'un avis étudiant , en termes positifs, négatifs ou neutres, en fonction du sentiment ou du point de vue global éxprimé. Une forme plus granulaire, d'ABSA, se concentre sur la recherche et l'analyse des sentiments ou opinions exprimés à l'égard des avis partagés. Pour cette thématique, nous avons utilisé les techniques de génération augmentée ou guidée de tuples de sentiments d'ABSA sans exemples et avec exemples et de fine tuning. L'étape finale du processus consiste à évaluer les performances du modèle à l'aide de certaines métriques, qui sont les Précision, Rappel et F1-Score :

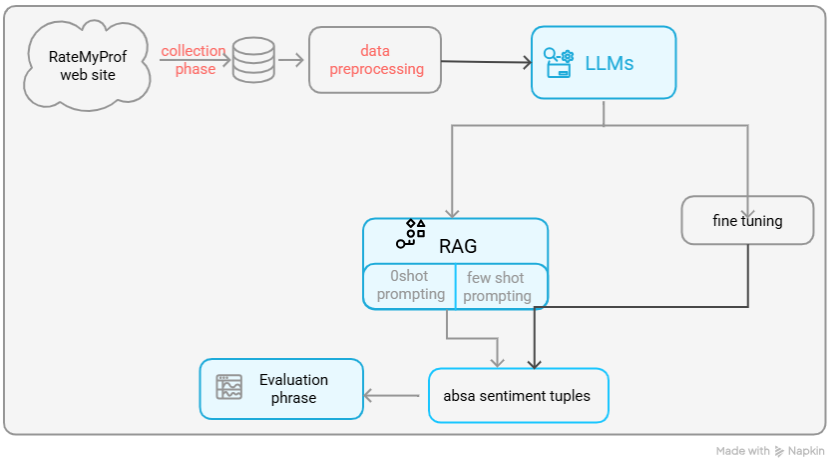

Figure III.1 : Architecture Globale du système ABSA

III.3 Pipeline de la solution ABSA

Le pipeline suivi pour la réalisation de la solution ABSA se compose des étapes clés suivantes :

1. **Collecte de données :** initialement, des données issues site web de RateMyProfessor sont collectées et prétraitées, incluant les commentaires et évaluations des étudiants sur leurs professeurs. Après nettoyage et structuration, ces données ont été exploitées dans les étapes suivantes.
2. **Choix du LLM :** Face à la variété des modèles disponibles, nous avons sélectionné des LLM open source, librement téléchargeable et utilisable localement. Ce choix s'est porté sur des critères d'accessibilité, de simplicité d'utilisation et de capacité à être déployé sans infrastructure lourde. Nous avons ensuite testé ses performances sur un échantillon de nos données afin d'évaluer sa capacité à répondre de manière exacte et cohérente à nos besoins.
3. **Choix de la méthode d'optimisation du LLM :** Choisir le bon LLM adapté à nos données et besoins spécifiques était complexe. Deux stratégies d'optimisation complémentaires sont adoptées et choisies pour leur efficacité et leur faisabilité avec nos contraintes techniques. La première méthode consistait à intégrer la génération augmentée ou guidée par récupération, en zéro shot ou few shot, permettant au LLM d'extraire et d'utiliser des données pertinentes

ou de contexte dans ses réponses. La seconde méthode exploitait la méthode affinée de fine tuning pour adapter plus finement le modèle aux spécificités de la tâche ABSA.

4. **Évaluation des résultats :** Pour mesurer la performance du modèle, une évaluation rigoureuse et une interprétation des résultats sont essentielles pour choisir l'approche la plus adaptée, en calculant les métriques classiques de Précision, Recall et F1-score.

III.4 Formulation du problème ABSA

Etant donné une collection de phrases ou avis textuels $D = \{S_1 \ , ..., S_N \}$. Chaque phrase S_i contient potentiellement plusieurs mots ou tokens $S_i = \{w_1, w_2, ..., w_n\}$, où w_i est le i-ème mot de la séquence textuelle. Le but est de construire une fonction linguistique augmentée et finetunée :

$$LLM_\theta \ : S_i \rightarrow Q_i = \{(a_j, c_j, s_j, o_j)\}(pour j = 1, k),$$

Où, l'entrée du modèle linguistique augmenté ou affiné LLM_θ est une phrase S_i souvent accompagnée d'un prompt sans example, avec exemple et de système pour guider le format de la réponse : $Input = (Prompt, UserQuery)$ et la sortie est une séquence de tuples de sentiment en quadruplets au format structuré
$Q_i = \{(a_j, c_j, s_j, o_j)\}(pour j = 1, k)$, séparés par « ; » :

- **Aspect** a_j : Entité ou caractéristique explicite mentionnée dans S.
- Catégorie c_j : Classe sémantique de A_j.
- Opinion o_j : Expression subjective associée à A_j.
- -Polarité s_j : Sentiment $\in \{positif, ngatif, neutre\}$.

III.5 Choix de la collection de données ou Dataset

Cet ensemble de données contient des avis et des métadonnées de professeurs provenant des 100 meilleures universités des États-Unis, collectés sur le site de RateMyProfessors.com [1].

1. Kaggle Dataset - RateMyProfessor T100 Universities Reviews

Les données comprennent des informations détaillées sur les professeurs, les avis des étudiants et les métadonnées universitaires couvrant la période de 1999 à 2025. Son exploitation va répondre aux objectifs suivants :

- Permettre une analyse avancée par apprentissage automatique des modèles de retours des étudiants.
- Suivre l'évolution temporelle de l'efficacité pédagogique et de la satisfaction des étudiants.
- Fournir des données structurées pour la recherche et l'analyse pédagogiques.
- Créer un historique des indicateurs de performance pédagogique.
- Fournir des informations basées sur les données sur la qualité de l'enseignement supérieur. Les caractéristiques importantes des données à explorer sont en deux catégories :
- Informations sur le professeur :
  - Nomet titre
  - Département et établissement
  - Note globale
  - Nombre total d'avis
  - Pourcentage de personnes prêtes à recommencer
  - Score de difficulté
- Détails de l'avis :
  - Notes individuelles des avis
  - Horodatage des avis
  - Informations sur le cours
  - Note reçue (si fournie)
  - Commentaires détaillés des étudiants
  - Votes utiles/inutiles

III.5.1 Processus de génération de tuples de sentiment ABSA

Prétraitement Le prétraitement des données commence par le chargement du fichier **des avis** , suivi de la suppression des lignes et colonnes vides afin de ga-

rantir l'intégrité du jeu de données. Ensuite, seuls les avis rédigés en anglais sont conservés grâce à une détection automatique de la langue. Une phase de nettoyage du texte est ensuite appliquée : elle inclut la suppression des balises HTML, des caractères spéciaux, des emojis et des espaces superflus, ainsi que la normalisation en minuscules. Ce prétraitement assure un texte propre et cohérent, prêt pour les étapes suivantes du pipeline ABSA.

-Recherche : La phase de recherche consiste à transformer tous les avis en vecteurs d'embeddings à l'aide d'un modèle comme SentenceTransformer. Un embedding est également généré pour un mot- clé ou un avis cible. Ensuite, les similarités cosinus sont calculées entre ce vecteur et tous les vecteurs des avis.
ici tous les avis similaires dépassant un certain seuil de similarité peuvent être retenus.
Les résultats retenus sont les avis similaires à la requête soumise, accompagnés de leur score de similarité. Ils sont enregistrés dans un fichier CSV en vue de leur utilisation lors de la phase de génération.

- Conception des Prompts :

*** Zero-Shot :** Aucune étiquette d'entraînement spécifique n'est nécessaire. le LM apprend à inférer le tuple aspect, polarité, opinion, catégorie directement.

*** Few-Shot :** l'approche few-shot fournit au modèle un petit nombre d'exemples annotés dans le prompt, ce qui lui permet de mieux comprendre la tâche attendue, le style de réponse et la structure de sortie. Ces exemples agissent comme des démonstrations guidées et permettent d'obtenir des résultats plus fiables et cohérents, surtout pour les textes ambigus ou riches en sous-entendus

- Génération : A cette phase de génération, le LLM qui reçoit en entrée un prompt structuré et retourne une sortie conforme au format ABSA attendu. L'objectif est de produire automatiquement, à partir d'un avis, : "aspect, opinion, polarité, catégorie".

III.5.2 Modèlés utilisés

Nous avons évalué notre approche en exploitant plusieurs LLM, chacun présentant des caractéristiques distinctes qui influencent leur performance dans des tâches spécifiques de compréhension et de génération de texte :

-Mistral : Mistral est une famille de ML open source développés par la société Mistral AI. Le modèle Mistral 7B, par exemple, est un transformer dense de 7 milliards de paramètres, entraîné pour exceller en génération de texte, compréhension et raisonnement.

- LLama2 :7b : LLaMA 2 (LLM Meta AI) est une série de modèles de langage développés par Meta (Facebook). Le modèle LLaMA 2 - 7B comporte 7 milliards de paramètres, conçu pour être plus performant et accessible que ses prédécesseurs.

- phi3 :3.8b-instruct : Phi-3 est une série de modèles compacts développés par Microsoft, dont Phi-3 Instruct 3.8B est une version spécifiquement entraînée pour suivre des instructions. Ces descriptions offrent un aperçu des capacités et des objectifs de conception de chaque modèle, fournissant un contexte pour comprendre leurs performances variées dans notre analyse comparative.

III.6 Évaluation des résultats

a.Résultats obtenus pour Zéro Shot

Enter keyword : Book

Test de deux phrase :

```
# Evaluate with sample test cases
test_cases = [
("The campus is beautiful with historic buildings.",
[   {"aspect": "campus", "polarity": "positive", "category": "
   appearance", "opinion": "beautiful"},
{"aspect": "buildings", "polarity": "positive", "category": "
   architecture", "opinion": "historic"}
] ),
( "The library was huge but the staff were unfriendly.",
[  {"aspect": "library", "polarity": "positive", "category": "
   size", "opinion": "huge"},
{"aspect": "staff", "polarity": "negative", "category": "
   service", "opinion": "unfriendly"}
]    ) ]
Model Evaluation Metrics:
Accuracy: 0.75
Precision: 0.88
Recall: 0.75
F1 Score: 0.77
```

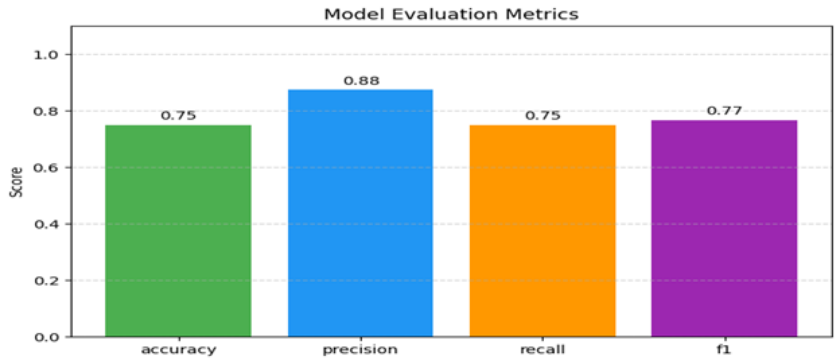

Figure III.2 : Métriques d'évaluation du modèle pour deux phrases en zero Shot

Top results :

```
result 1:
```

```
Review: 'Dont hesitate to follow one of the best student guide
    MIKE...he is so funny and do explain very well.....great
From Diane and Daniel from Montréal
Analysis: [
{"aspect": "student guide", "polarity": "positive", "category"
   : "recommendation", "opinion": "best"},
{"aspect": "personality of MIKE", "polarity": "positive", "
   category": "character trait", "opinion": "funny"},
{"aspect": "explaining skills of MIKE", "polarity": "positive"
   , "category": "knowledge sharing","opinion": "explain very
   well"}
]
Result 2:
Result 2:
Review: Of course unchanged since my last visit 40 years ago.
   Impressive buildings and atmosphere. Busy but a student
   buzz. Great book store.
Analysis: [
{"aspect": "consistency", "polarity": "positive", "category":
   "reliability", "opinion": "unchanged since last visit"},
{"aspect": "buildings and atmosphere", "polarity": "positive",
    "category": "ambiance", "opinion": "impressive"}],

[{"aspect": "crowdedness (student buzz)", "polarity": "neutral
    to positive",
  "category": "crowd level",
  "opinion": "busy but a student buzz"}]
]

Result 3:
Review: Many more talented have described it before me. Read
   them, The Coop library proposes many books in Harvard.

You can take a tour guided by students.
```

```
The best experience is not to visit it but to study there.
Analysis: [
{"aspect": "talented descriptions", "polarity": "positive", "
   category": "reputation/experience", "opinion": "many more
   talented have described"},
{"aspect": "book variety (Harvard)", "polarity": "neutral to
   positive", "category": "variety", "opinion": "Many books in
    Harvard" but lack of specific sentiment towards the book
   selection suggests neutral, though it may be viewed as a
   potential point of positivity depending on personal
   interest.
  ]
```

-Test de 07 phrases :

```
Model Evaluation Metrics:
Accuracy: 0.53
Precision: 0.88
Recall: 0.53
F1 Score: 0.62
```

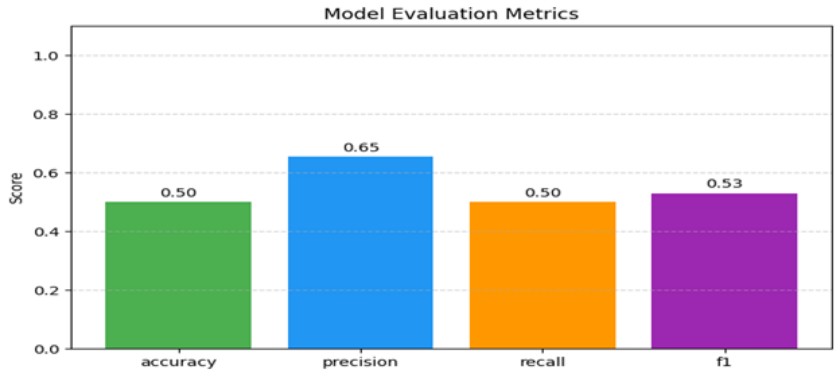

Figure III.3 : Métriques d'évaluation du modèle pour sept phrases en zero Shot

b.Résultats obtenus pour Few Shot :

Test de deux phrases.

```
  Model Evaluation Metrics:
Accuracy: 1.00
Precision: 1.00
Recall: 1.00
F1 Score: 1.00
```

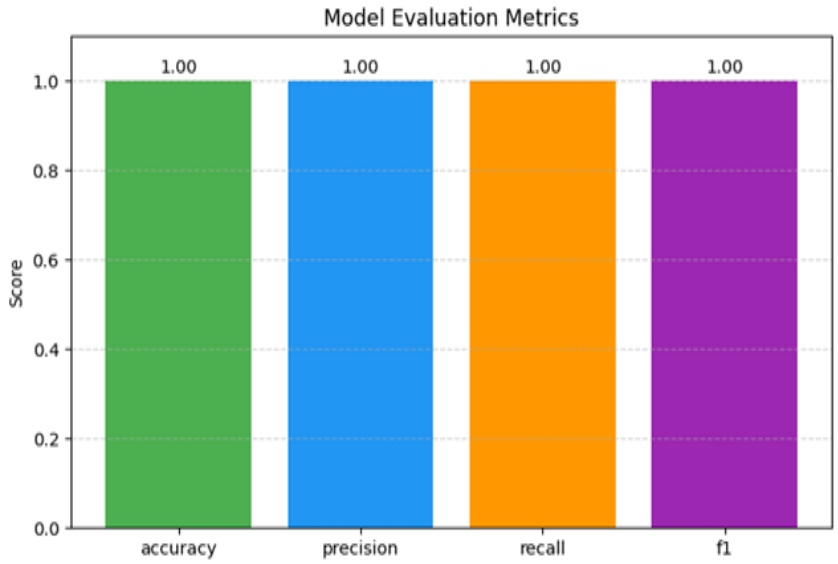

Figure III.4 : Métriques d'évaluation du modèle pour deux phrases en Few Shot

-Test de sept phrases.

```
Model Evaluation Metrics:
Accuracy: 0.89
Precision: 0.89
Recall: 0.89
F1 Score: 0.89
```

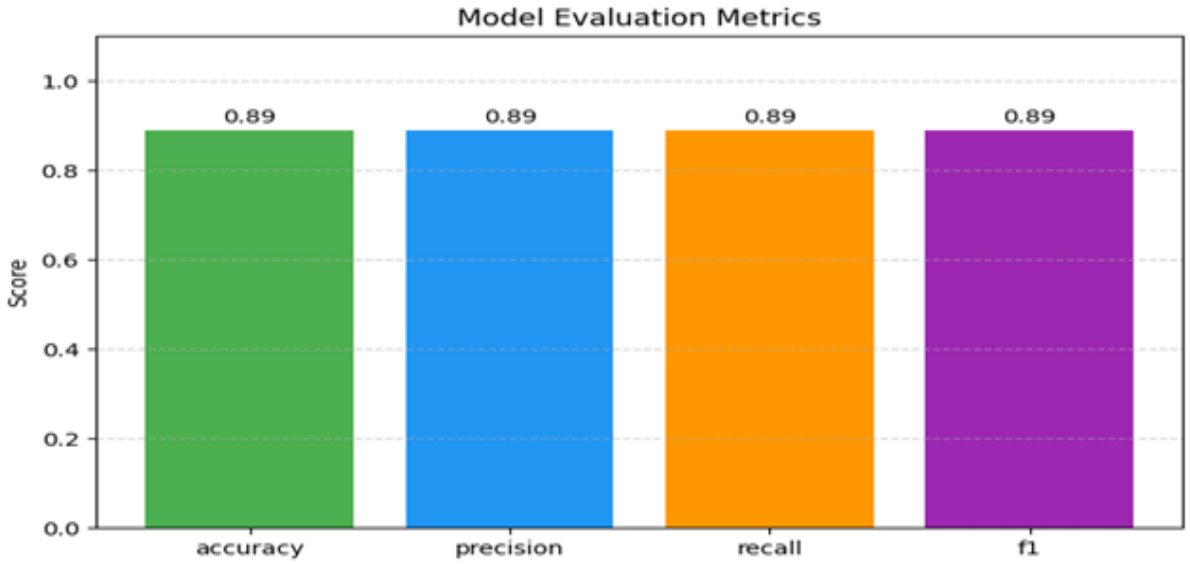

Figure III.5 : Métriques d'évaluation du modèle pour sept phrases en Few Shot

- c.Analyse des performances générales : Les résultats des trois modèles **Mistral, LLaMA2 et Phi3** ,sont strictement identiques, aussi bien au niveau des métriques (Accuracy, Precision, Recall, F1) que des sorties générées, et ce, pour toutes les tailles de test **(2, 5, 7, 11).**

Taille	Mistral-Accuracy	Mistral-Precision	Mistral-Recall	Mistral-F1	LLaMA2-Accuracy	LLaMA2-Precision	LLaMA2-Recall	LLaMA2-F1	Phi3-Accuracy	Phi3-Precision	Phi3-Recall	Phi3-F1
2	0,75	0,88	0,75	0,77	0,75	0,88	0,75	0,77	0,75	0,88	0,75	0,77
5	0,55	0,84	0,55	0,61	0,55	0,84	0,55	0,61	0,55	0,84	0,55	0,61
7	0,53	0,88	0,53	0,62	0,53	0,88	0,53	0,62	0,53	0,88	0,53	0,62
11	0,46	0,78	0,46	0,55	0,46	0,78	0,46	0,55	0,46	0,78	0,46	0,55

Tableu III.2 : Comparaison des performances des modèles ABSA

- d.Analyse comparative des résultats en : Zero-Shot vs Few-Shot :

Les résultats mettent en évidence une amélioration significative des performances du modèle Phi3 lorsqu'on passe du Zero-Shot (aucun exemple donné au modèle) au Few-Shot (quelques exemples fournis).

Observations clés

-Performance maximale en Few-Shot à 2 phrases : toutes les métriques atteignent

un score parfait (1.00), ce qui montre que le modèle est capable de généraliser parfaitement sur un échantillon minimal lorsque des exemples pertinents sont fournis.

-A mesure que la taille du texte augmente, les scores en mode Zero-Shot chutent rapidement, notamment en termes d'accuracy et de recall. En revanche, en Few-Shot, les performances restent élevées, même avec 11 phrases, ce qui prouve que l'exposition à quelques exemples améliore la robustesse et la cohérence du modèle face à des entrées plus longues.

- Le score F1 illustre parfaitement cet écart : il passe de 0.55 (ZS, 11 phrases) à 0.76 (FS, 11 phrases), soit une amélioration de +21 points.

Taille phrases	ZeroShot-Accuracy	ZeroShot-Precision	ZeroShot-Recall	ZeroShot	FewShot-Accuracy	FewShot-Precision	FewShot-Recall	FewShot-F1
2	0,75	0,88	0,75	0,77	1	1	1	1
5	0,55	0,84	0,55	0,61	0,75	0,88	0,75	0,81
7	0,53	0,88	0,53	0,62	0,89	0,89	0,89	0,89
11	0,46	0,78	0,46	0,55	0,76	0,77	0,76	0,76

Tableu III.3 : Comparaison des performances de Phi3 : Zero-Shot vs Few-Shot

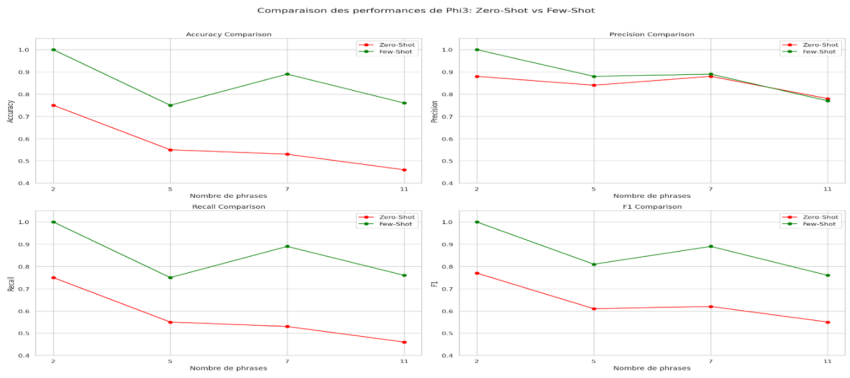

Figure III.6 : Comparaison des performances de Phi3 : Zero-Shot vs Few-Shot

e.Interface de l'assistant ABSA Cette application repose principalement sur composantes :

- L'interface de l'assistant : C'est le point de contact entre l'utilisateur et notre système.

- Le choix du fichier csv : Ensemble d'avis étudiants.

- Entrée du keyword : book

- Affichage des résultats. .

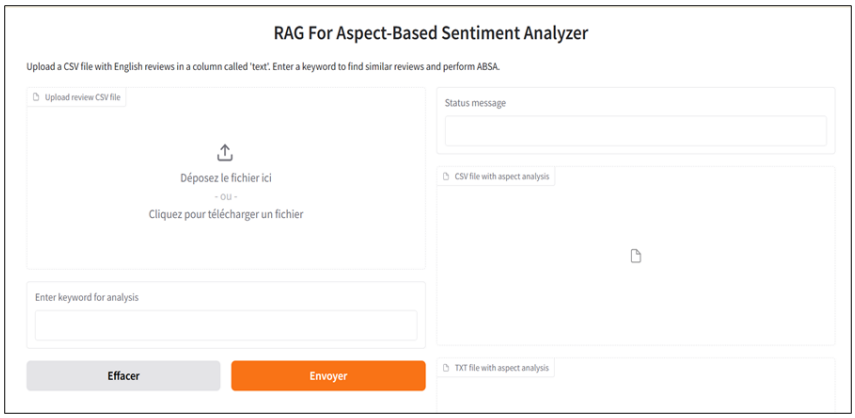

Figure III.7 : Aperçu d'ensemble sur l'interface utilisateur

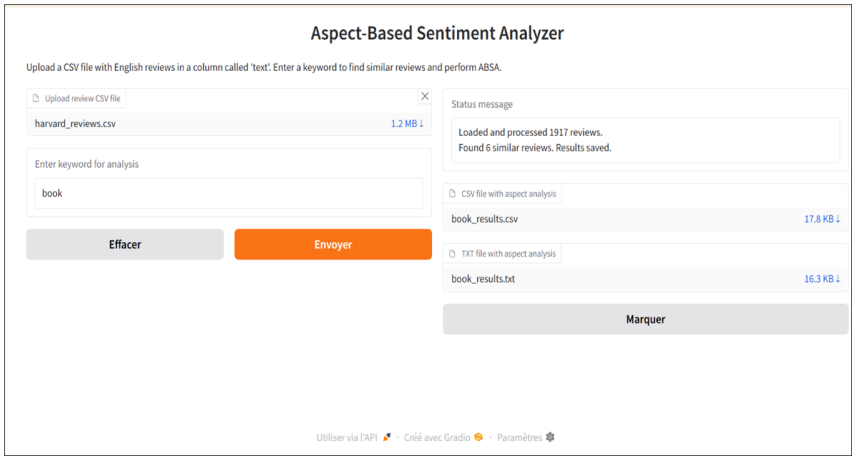

Figure III.8 : Aperçu d'ensemble sur l'interface d'ABSA

f. **Fine-tuning du modèle sur le jeu de données des avis d'étudiants** Dans cette expérimentation, le modèle de langue a été fine-tuné sur un corpus de commentaires d'étudiants à propos de leurs enseignants, extrait du fichier reviews.csv. Chaque entrée contient un commentaire libre annoté selon plusieurs aspects pertinents (par exemple : Clarity, Difficulty, Helpfulness), avec un sentiment associé (positif, négatif ou neutre).

L'objectif est de permettre une analyse de sentiment fine par aspect, afin de mieux comprendre les points forts et les axes d'amélioration dans les retours étudiants.

Les résultats montrent une nette amélioration des performances du modèle jusqu'à environ 4000 étapes, où l'on observe des valeurs maximales de précision ($86,07\%$) et de F1-score ($85,8\%$). La baisse de la perte d'entraînement ainsi que la hausse des métriques de performance suggèrent que le modèle a bien appris à distinguer les sentiments associés à différents aspects des commentaires.

Cependant, au-delà de cette étape, la perte de validation augmente légèrement, ce qui pourrait indiquer un début de surapprentissage. Les performances restent néanmoins relativement stables, ce qui confirme la robustesse LLM même après plusieurs milliers d'itérations.

Step	Training Loss	Validation Loss	Accuracy	Precision	Recall	F1
500	0.448800	0.509933	0.811333	0.772413	0.811333	0.772608
1000	0.470200	0.472133	0.819333	0.790169	0.819333	0.798852
1500	0.414000	0.436051	0.834000	0.823681	0.834000	0.827594
2000	0.279500	0.500166	0.843333	0.821689	0.843333	0.822488
2500	0.309200	0.431969	0.842667	0.828481	0.842667	0.833592
3000	0.235400	0.465099	0.858000	0.848980	0.858000	0.851246
3500	0.163900	0.476056	0.854000	0.847731	0.854000	0.850272
4000	0.305400	0.460894	0.860667	0.856114	0.860667	0.858000
4500	0.227600	0.576872	0.862667	0.852922	0.862667	0.855240
5000	0.138900	0.563358	0.853333	0.847315	0.853333	0.849852
5500	0.169000	0.631655	0.859333	0.852004	0.859333	0.854774

Tableu III.4 : Évalution des métriques (Accuracy, Precision, Recall, F1) en fonction des étapes

Les résultats montrent une nette amélioration des performances du modèle jusqu'à

environ 4000 étapes, où l'on observe des valeurs maximales de précision (86,07 %) et de F1-score (85,8 %). La baisse de la perte d'entraînement ainsi que la hausse des métriques de performance suggèrent que le modèle a bien appris à distinguer les sentiments associés à différents aspects des commentaires.

Cependant, au-delà de cette étape, la perte de validation augmente légèrement, ce qui pourrait indiquer un début de surapprentissage. Les performances restent néanmoins relativement stables, ce qui confirme la robustesse de DistilBERT même après plusieurs milliers d'itérations.

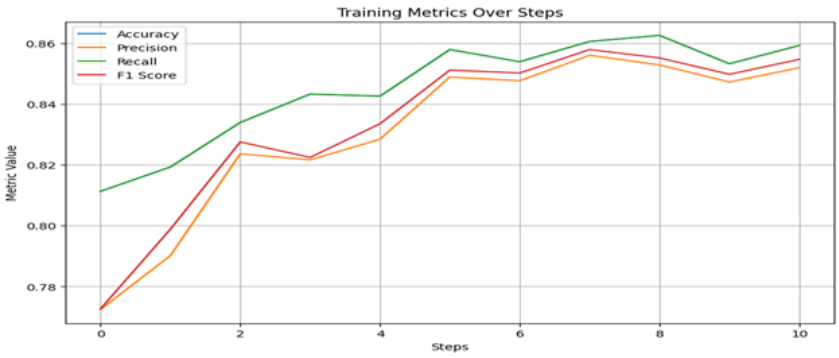

Figure III.9 : Évolution des métriques (Accuracy, Precision, Recall, F1) en fonction des étapes

Résultats

```
Give me a sentence: Master 2 for 2025 successfully completed."

Sentence: "Master 2 for 2025 successfully completed.""
Aspect-Based Sentiment Predictions:
- Difficulty: negative
- Clarity: positive
- Helpfulness: positive
Give me a sentence: : A rewarding conclusion after all the
   work "

Sentence: ": A rewarding conclusion after all the work ""
```

```
Aspect-Based Sentiment Predictions:
- Difficulty: negative
- Clarity: positive
- Helpfulness: positive
```

III.7 Conclusion

Ce chapitre a permis de concrétiser la solution proposée en détaillant les différentes étapes nécessaires à sa mise en œuvre. Pour cela, des données ont été exploitées, structurées sous forme d'avis (reviews). Dans le cadre de la modélisation, une sélection rigoureuse des modèles a été effectuée en s'appuyant sur la méthode du prompt engineering.

L'implémentation s'est déroulée selon une approche progressive. Le système ABSA, en tant que solution à granularité fine, s'est révélé particulièrement pertinent dans le cadre de l'AS des aspects, notamment lorsqu'une base documentaire riche et actualisée est disponible, comme des avis étudiants .

Par ailleurs, les approches zero-shot et few-shot ont été utilisées pour des tests exploratoires et la création de prototypes, en raison de leur rapidité de mise en œuvre, bien qu'elles soient généralement moins précises. Enfin, le fine-tuning a été identifié comme la méthode la plus fiable, à condition de disposer de données annotées, car elle permet d'atteindre une précision optimale sur des aspects spécifiques.

Conclusion générale

L'essor des plateformes éducatifs interactifs a profondément transformé la manière dont les institutions académiques perçoivent et interagissent avec leurs personnels et apprenants.
Les avis en ligne, notamment des étudiants universitaires, représentent une source de connaissances précieuses permettant de mieux comprendre les attentes, les préférences et les insatisfactions des consommateurs. Dans ce contexte, la tâche ABSA s'est imposée comme un solution importante pour capter des opinions fines et contextualisées, en associant chaque sentiment exprimé à un aspect spécifique du service ou du produit.

Ce projet a abordé une approche moderne et efficace pour régler le problème de l'ABSA en exploitant les capacités avancées des LLM. Plutôt que de recourir à des architectures complexes, notre solution repose sur l'orientation du modèle à l'aide d'instructions explicites structurés et sur sa spécialisation affiné pour le domaine ciblé. Ces stratégies permettent de profiter des connaissances linguistiques pré-acquises des LLM tout en les guidant et adaptant aux spécificités du secteur de l'université.

Les expérimentations menées sur des avis d'étudiants ont mis en évidence la pertinence de cette approche. Les résultats obtenus démontrent que les LLM, une fois guidés et adaptés, sont capables d'effectuer une analyse fine des sentiments avec un bon niveau de précision.

Cependant, ce travail ouvre également la voie à plusieurs perspectives d'amélioration. Il serait pertinent, par exemple, d'explorer l'intégration de techniques de détection auto-

matique d'aspects dans des contextes non annotés, de renforcer la robustesse des modèles face au langage informel typique des réseaux sociaux, ou encore d'évaluer l'approche sur d'autres domaines et d'autres langues, pour tester son pouvoir de généralisation. Par ailleurs, l'usage de modèles multilingues ou la combinaison avec des signaux multimodaux (images, vidéos) pourrait enrichir davantage l'analyse.

En conclusion, cette étude montre que les LLM représentent une avancée significative de l'ABSA appliquée aux données issues de l'internet. Ils offrent une nouvelle voie vers des analyses plus intelligentes, flexibles et contextuellement riches, au service d'une meilleure compréhension de la voix de l'utilisateur.

Bibliographie

[1] K. V. Rajan, "Sentiment analysis of social media using artificial intelligence," in *Advances in Sentiment Analysis-Techniques, Applications, and Challenges*. IntechOpen, 2024.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[3] "Understanding Encoder-Decoder Sequence to Sequence Model | by Simeon Kostadinov | TDS Archive | Medium." [Online]. Available : https://medium.com/data-science/understanding-encoder-decoder-sequence-to-sequence-model-679e04af4346

[4] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2 : Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742.

[5] "Prompt Engineering Guide | Prompt Engineering Guide." [Online]. Available : https://www.promptingguide.ai/fr

[6] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[7] M. A. Yang, Angelina, "A Practical Approach to Retrieval Augmented Generation Systems," Nov. 2023. [Online]. Available : https://mallahyari.github.io/rag-ebook/

[8] "What is Fine-Tuning ? | IBM." [Online]. Available : https://www.ibm.com/think/topics/fine-tuning

[9] A. Nazir, Y. Rao, L. Wu, and L. Sun, "Issues and challenges of aspect-based sentiment analysis : A comprehensive survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 2, pp. 845–863, 2020.

[10] D. Kumar, A. Gupta, V. K. Gupta, and A. Gupta, "Aspect-based sentiment analysis using machine learning and deep learning approaches," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 11, no. 5s, pp. 118–138, 2023.

[11] W. Zhang, X. Li, Y. Deng, L. Bing, and W. Lam, "A survey on aspect-based sentiment analysis : Tasks, methods, and challenges," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 11, pp. 11 019–11 038, 2022.

[12] B. Liu, *Sentiment analysis and opinion mining*. Springer Nature, 2022.

[13] H. Zhang, Y.-N. Cheah, O. M. Alyasiri, and J. An, "A survey on aspect-based sentiment quadruple extraction with implicit aspects and opinions," 2023.

[14] N. Mughal, G. Mujtaba, S. Shaikh, A. Kumar, and S. M. Daudpota, "Comparative analysis of deep natural networks and large language models for aspect-based sentiment analysis," *IEEE Access*, vol. 12, pp. 60 943–60 959, 2024.

[15] S. U. S. Chebolu, F. Dernoncourt, N. Lipka, and T. Solorio, "Survey of aspect-based sentiment analysis datasets," *arXiv preprint arXiv :2204.05232*, 2022.

[16] G. Team, R. Anil, S. Borgeaud, Y. Wu, J. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. Dai, A. Hauth *et al.*, "Gemini : A family of highly capable multimodal models. arxiv 2023," *arXiv preprint arXiv :2312.11805*, 2024.

[17] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama : Open and efficient foundation language models," *arXiv preprint arXiv :2302.13971*, 2023.

[18] "Qu'est-ce que le NLP (traitement automatique du langage naturel) ? | IBM." [Online]. Available : https://www.ibm.com/fr-fr/think/topics/natural-language-processing

[19] "Les grands modèles de langage (LLM) : définition, fonctionnement & application," Jan. 2025. [Online]. Available : https://www.callmenewton.fr/guide-ia/grands-modeles-de-langage/

[20] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8528–8535.

[21] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv :1406.1078*, 2014.

[22] S. Shang, J. Liu, and Y. Yang, "Multi-layer transformer aggregation encoder for answer generation," *IEEE Access*, vol. 8, pp. 90 410–90 419, 2020.

[23] M. Fujitake, "Dtrocr : Decoder-only transformer for optical character recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2024, pp. 8025–8035.

[24] "The Illustrated Transformer – Jay Alammar – Visualizing machine learning one concept at a time." [Online]. Available : https://jalammar.github.io/illustrated-transformer/

[25] "Creare un Chatbot in Python con LangChain e RAG," Nov. 2023. [Online]. Available : https://www.diariodiunanalista.it/posts/chatbot-python-langchain-rag/

[26] P. Zhao, H. Zhang, Q. Yu, Z. Wang, Y. Geng, F. Fu, L. Yang, W. Zhang, J. Jiang, and B. Cui, "Retrieval-augmented generation for ai-generated content : A survey," *arXiv preprint arXiv :2402.19473*, 2024.

[27] "L'intelligence artificielle générative," 2023.

[28] Z. Fu, H. Yang, A. M.-C. So, W. Lam, L. Bing, and N. Collier, "On the effectiveness of parameter-efficient fine-tuning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 37, no. 11, 2023, pp. 12 799–12 807.

[29] V. Lialin, V. Deshpande, and A. Rumshisky, "Scaling down to scale up : A guide to parameter-efficient fine-tuning," *arXiv preprint arXiv :2303.15647*, 2023.

[30] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, H. Wang, and H. Wang, "Retrieval-augmented generation for large language models : A survey," *arXiv preprint arXiv :2312.10997*, vol. 2, no. 1, 2023.

[31] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-eval : Nlg evaluation using gpt-4 with better human alignment," *arXiv preprint arXiv :2303.16634*, 2023.

Annexe

Environnement et outillés de développement

Le projet a été mis en œuvre en utilisant une combinaison de cloud et sur site.

A.1 Présentation de Python

Python est un langage de programmation à haut niveau, interprète et oriente objet, crée par Guido van possum et publie pour la première fois en 1991. Il est devenu l'un des langages de programmation les plus Populaires et les plus utilisés, apprécie pour sa lisibilité et son code clair. Python est souvent d´écrit comme Un langage proche de l'anglais simple, ce qui rend son code particulièrement facile à lire et à comprendre. un autre avantage majeur de Python est sa polyvalence. Il est utilisé dans divers domaines, y compris Le d´enveloppement web, le d´enveloppement de logiciels, l'analyse de données, l'apprentissage automatique, L'intelligence artificielle, la science des données, la bio-informatique, le calcul scientifique, et bien d'autres. Son usage est également t'es répandu dans le domaine du Traitement Automatique du Langage Naturel (NLP), grâce à sa lisibilité, sa simplicité d'utilisation et la variété de ses bibliothèques d´éditées. Les Applications de Python en NLP vont de la génération de texte à la traduction automatique, en passant par L'analyse de sentiments et le résume automatique de textes.

A.2 Bibliothèques Utilisées

Les bibliothèques en programmation sont des ensembles d'outils, de modules, de méthodes et de classes Qui étendent les fonctionnalités du langage utilise. La plupart des bibliothèques disponibles en Python sont Gratuites et open source. Voici les bibliothèques utilisées dans ce projet :

❖ **Transformers**

Transformers est une bibliothèque qui fournit des API et des outils pour télécharger et entrainer facilement des modèles prêt-entraines de pointe. Ces modèles supportent des taches courantes dans différentes modalités, telles que :

- **Traitement du Langage Naturel (NLP)** : classification de texte, reconnaissance d'entités nommées, Questions-réponses, modélisation de langage, résume, traduction, choix multiples, et génération de Texte.
- **Vision par Ordinateur :** classification d'images, d´détection d'objets, et segmentation.
- **Audio :** reconnaissance automatique de la parole et classification audio.
- **Multimodal :** questions-réponses sur table, reconnaissance optique de caractères, extraction d'informations à partir de documents scannes, classification vidéo, et questions-réponses visuelles. Transformers supporte l'interopérabilité des Framework entre Bytoch, TensorFlow et JAX, offrant la flexibilité d'utiliser différents Framework à chaque ´étape de la vie d'un modèle.

❖ **LangChain**

LangChain est un cadre pour d´envelopper des applications alimentées par de grands modèles de langage(LLMs). LangChain simplifie chaque étape du cycle de vie de l'application LLM, que ce soit pour :

- **Développement :** construire des applications en utilisant les blocs de construction et composants Open-source de Lang Chain, intégrations tierces et modèles.
- **Mise en production :** utiliser LangSmith pour inspecter, surveiller et évaluer vos chaines, optimisant Et d´éployant en toute confiance.
- **Déploiement :** transformer toute chaine en une API avec LangServe.

  LangChain comporte plusieurs modules pour assurer le bon fonctionnement des

composants nécessaires La création d'applications NLP, tels que l'interaction avec les modèles, la connexion et récupération des Données, les chaines, les agents, et la mémoire.

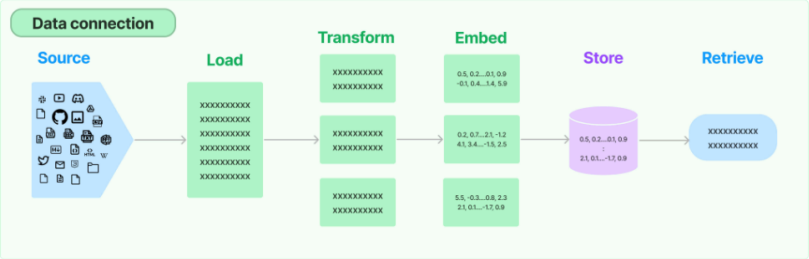

Figure III.10 : A.1 : Récupération d'informations avec la bibliothèque Langchain

❖ **Hugging Face Hub**

La bibliothèques huggingface- hub permet d'interagir avec le Hugging Face Hub, une plateforme de machine learning pour les créateurs et les collaborateurs. Elle permet de d´couvrir des modèles pré-entraines et des jeux de données pour vos projets, de jouer avec des applications de machine learning hébergées sur le Hub, et de créer et partager vos propres modèles et jeux de données avec la communauté.

❖ **Groq**

Groq est une bibliothèque permettant l'utilisation de Groq Cloud avec LangChain, facilitant l'utilisation des différentes fonctionnalités de LangChain avec les modèles de Groq Cloud.

❖ **Sentence Transformers :**

Embeddings, Retrieval, and Reranking Ce cadre fournit une méthode simple pour calculer les intégrations afin d'accéder, d'utiliser et de former des modèles d'intégration et de reclassement de pointe. Il calcule les intégrations à l'aide de modèles Sentence Transformer (démarrage rapide) ou pour calculer les scores de similarité à l'aide de modèles Cross-Encoder (également appelés reranker) (démarrage rapide). Cela ouvre la voie à un large éventail d'applications, notamment la recherche sémantique, la similarité textuelle sémantique et l'exploration de paraphrases.

❖**Re** : Ce module fournit des opérations de correspondance d'expressions régulières similaires à celles trouvées dans Perl.

❖ **Os** : L'instruction import os en Python importe le module os, qui vous permet d'inter-

agir avec le système d'exploitation. Ce module offre des fonctions permettant de manipuler des fichiers et des dossiers, de gérer des processus, d'interroger le système et de nombreuses autres opérations liées au système d'exploitation.

❖ **NumPy :** Ajoutant la prise en charge de grands tableaux et matrices multidimensionnels, ainsi qu'une grande collection de fonctions mathématiques de haut niveau pour fonctionner sur ces tableaux

❖ **Pandas :** Pour la manipulation et l'analyse des don- nées. En particulier, il propose des structures de données et des opérations pour manipuler des tableaux numériques et des séries chronologiques.

❖ **Scikit-learn :** Destinée a l'apprentissage automatique, elle comprend notamment des fonctions pour l'apprentissage supervisé. Elle est conçue pour s'harmoniser avec d'autres bibliothèques libres Python.

❖ **Matplotlib :** Pour créer des visualisations statiques, animées et interactives en Python.

A.3 Outils Utilisés

Google Colab, ou Google Colaboratory, est une plateforme gratuite offerte par Google permettant d'´écrire et exécuter du code Python directement dans le navigateur. Elle facilite l'accès aux ressources de calcul et aux bibliothèques d'apprentissage automatique usuelles sans nécessiter d'installation locale.

Voici quelques caractéristiques l'es de Google Colab :

- Accessibilité Google Colab est héberge sur le cloud, permettant de l'utiliser sans installer Python ou d'autres bibliothèques sur son ordinateur. Il suffit de se connecter 'a un compte Google pour y accéder. De plus, Google Colab est livre avec de nombreuses bibliothèques Python pré-installées, incluant des bibliothèques de data science et de visualisation telles que NumPy, Pandas, Scikit-learn, TensorFlow, PyTorch, Matplotlib, Seaborn, et Plotly.
- Accès aux ressources de calcul Google Colab offre un accès gratuit à des Processeurs graphiques (GPU) et a des unités de traitement de tenseur (TPU), ce qui est extrêmement utile pour les taches gourmandes en calcul.
- Facilite de collaboration Les notebooks crées dans Google Cola sont sauvegardes au-

tomatiquement sur Google Drive, permettant un stockage et un partage faciles avec d'autres d´enveloppeurs. Google Colab prend en charge la collaboration en temps réel, permettant à plusieurs utilisateurs de travailler simultanément sur le même notebook. Il permet également l'utilisation de GitHub pour le suivi des Versions de code et la collaboration.

- Flexibilité dans la programmation Google Colab permet d'accéder facilement à des données externes et d'inclure des cellules de texte explicatif pour documenter le travail. Les notebooks offrent plusieurs modes d'exécution, incluant l'exécution d'une seule cellule, d'une s´élection de cellules, ou de tout le notebook en une seule fois.
- En résume, Google Colab est un outil puissant pour le d´enveloppement en Python, particulièrement dans le domaine de l'apprentissage automatique. Il offre un environnement de d´enveloppement interactif, un accès aux GPU et TPU, un stockage sur Google Drive, une collaboration en temps réel, et de nombreuses autres fonctionnalités avancées.

  colab-xterm

  colab-xterm est un outil qui vous permet d'ouvrir une fenêtre de terminal à l'intérieur d'une cellule Google Colab, vous donnant un contrôle direct du système comme si vous travailliez sur une machine locale.

  - **Installer paquet :**

    !pip install colab-xterm https://pypi.org/project/colab-xterm/
  - **Charger extension** : % load-ext colabxterm
  - **Lancer Terminal :** % xterm
  - **Installer ollama depuis son site officiale :** curlhttps://ollama.ai/install.sh|sh