

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Master en Informatique

Option : Système d'Information et de Connaissances (S.I.C)

Thème

**Conception et développement d'une plateforme
éducative interactive basée sur l'optimisation des
apprentissages grâce aux résumés linguistiques.**

Réalisé par :

- Belhadj Mohammed Aymen

Présenté le 30 Juin 2025 devant le jury composé de :

- Mme Marouf radja (Présidente)
- Mme AMGHAR Djazia (Encadreur)
- Mr Sebbah Abderrezak (Examineur)

Année universitaire: 2024-2025

Remerciement

Tout d'abord, je remercie le dieu le tout puissant pour son aide et pour la volonté qui m'a donnée pour finir ce travail.

J'adresse mes remerciements à Madame la Professeure AMGHAR Djazia, qui était charge de fournir des orientations à chaque point de ce mémoire.

Mes remerciements vont également à Madame Marouf pour avoir acceptée de présider le jury de ma soutenance.

Je tiens aussi à remercier Monsieur Sebbah pour avoir accepté d'examiner mon travail

Dédicace

Je tiens à dédier cet humble travail

A mes parents pour leur amour, leur soutien et leurs encouragements. Leurs sacrifices ont été une source de motivation tout au long de mon parcours.

A mon frère et ma sœur pour leur soutien, leurs encouragements, leur présence et leur aide.

A mes amis, mes collègues pour leur amitié durant la période d'université.

Table des matières

<i>Liste des figures</i>	<i>VI</i>
<i>Liste des tableaux</i>	<i>VII</i>
<i>Liste des abréviations</i>	<i>VIII</i>
<i>Introduction Générale</i>	
<i>Contexte du projet</i>	<i>1</i>
<i>Problématique</i>	<i>1</i>
<i>Objectifs</i>	<i>1</i>
<i>Production Automatisée de Résumés Linguistiques</i>	<i>2</i>
<i>Intégration à la Plateforme Éducative</i>	<i>2</i>
<i>Organisation du mémoire</i>	<i>2</i>
<i>Chapitre 1 Apprentissage en ligne (E-learning) et plateformes éducatives</i>	<i>3</i>
<i>1. Introduction</i>	<i>4</i>
<i>2. Définition</i>	<i>4</i>
<i>3. Histoire d'apprentissage en ligne</i>	<i>4</i>
3.1 Origines : Cours par correspondance (1840-1873)	<i>4</i>
3.2 Introduction des machines d'enseignement (1920-1954).....	<i>5</i>
3.3 Utilisation des médias de diffusion (1925-1930s)	<i>5</i>
3.4 Ère des ordinateurs et premiers systèmes e-learning (1960s-1970s)	<i>5</i>
3.5 L'essor de l'e-learning : des ordinateurs personnels au web (1980-1990)	<i>5</i>
3.6 L'e-learning à l'ère des systèmes modernes et du mobile (2000-2020)	<i>6</i>
<i>4. Plateformes éducatives</i>	<i>6</i>
4.1 Types des plateformes éducatives.....	<i>7</i>
4.1.1 Learning Management System (LMS).....	<i>7</i>
4.1.2 Learning Content Management Systems (LCMS):.....	<i>7</i>
4.1.3 Massive open online courses (MOOC) :	<i>7</i>
<i>4. Tableau comparative entres les plateformes</i>	<i>7</i>
<i>5. Conclusion</i>	<i>8</i>
<i>Chapitre 2 Traitement Automatique du Langage Naturel (TALN)</i>	<i>9</i>
<i>1. Introduction</i>	<i>11</i>
<i>2. Différentes applications de TALN</i>	<i>11</i>
2.1 Traduction automatique	<i>12</i>

2.2 Reconnaissance vocale	13
2.3 Classification de textes	14
2.4 Génération du résumés automatiques	15
3. Résumé Automatique.....	16
3.1 Types de résumé automatique	16
3.1.1 Résumé Extractif.....	16
3.1.2 Résumé abstraictive	17
3.2 Algorithmes de Résumé Automatique.....	18
3.2.1 TextRank.....	19
3.2.2 Transformers	20
3.2.2.1 Composants principaux des Transformers	20
3.2.2.1.1 Mécanisme d'attention auto-régressive (Self-Attention).....	21
3.2.2.1.2 Attention multi-têtes masquée (Masked Multi-Head Attention).....	21
3.2.2.1.3 Attention multi-têtes (Multi-Head Attention)	21
3.2.2.1.4 Encodage positionnel	21
3.2.2.1.5 Structure encodeur-décodeur.....	21
3.2.2.1.6 Réseaux Feed-Forward	21
3.2.2.1.7 Normalisation des couches et connexions résiduelles	21
1. Large Language Models (LLMs).....	23
4.1 Définition	23
4.2 Fonctionnement des LLMs.....	23
4.2.1 Pré-entraînement de LLMs	23
4.2.2 Fine-tuning de LLMs	23
4.3 Architecture des LLMs	23
2. Modèles les plus connus de LLM.....	24
5.1 Generative Pre-trained Transformer (GPT)	24
5.2 Bidirectional and Auto-Regressive Transformers (BART)	26
5.3 Bidirectional Encoder Representations from Transformers (BERT)	28
5.4 Large Language Model Meta AI (LLAMA).....	30
6. Conclusion.....	34
Chapitre 3 expérimentation et la conception	35
1. Introduction	38
2. Environnement et outils de développement.....	38
2.1 Environnement matériel.....	38
2.2 Environnement logiciel.....	38
2.3 bibliothèques utilisées	40
2.4 Langage de programmation.....	42

2.5 Ensemble de données	44
3. Flux du travail propose.....	48
3.1 Schéma de résumé un document s.....	48
3.2 Schéma de génération des questions	49
3.3 Schéma de chatbot.....	50
4. LLAMA.....	50
5. Évaluation	55
5.1 Modèle BART.....	55
5.2 Modèle basée sur TextRank.....	57
5.3 Evaluation final.....	58
5.3.1 BERTScore	58
5.3.2 FREScore.....	60
6. Conception par UML.....	61
6.1 Définition des acteurs.....	61
6.2 Besoins fonctionnels	61
6.3 Diagramme de cas d'utilisation	62
6.3.1 Acteur Administrateur	62
6.3.2 Acteur Enseignant	63
6.3.3 Acteur Etudiant	64
6.4 Diagramme de classe.....	65
3. Présentation des interfaces	66
4. Conclusion	70
Conclusion générale et perspectives	71
Bibliographie	73

Liste des figures

<i>Figure 2.1 - Architecture et fonctionnement de google traduction</i>	<i>12</i>
<i>Figure 2.2: Architecture classique (et hybride) d'un Système de Reconnaissance Automatique de la Parole (SRAP).....</i>	<i>13</i>
<i>Figure 2.3: Architecture de classification de texte</i>	<i>14</i>
<i>Figure 2.4 - Modèle général pour l'approche linguistique de résumé automatique</i>	<i>15</i>
<i>Figure 2.5 - Architecture d'un système de résumé extractif</i>	<i>17</i>
<i>Figure 2.6 - Architecture d'un système de résumé abstrait</i>	<i>18</i>
<i>Figure 2.7 - Résumé de texte utilisant l'algorithme TextRank (Logique de base)</i>	<i>19</i>
<i>Figure 2.8 - Matrice de similarité</i>	<i>20</i>
<i>Figure 2.9 - Architecture du transformer.....</i>	<i>22</i>
<i>Figure 2.10 - Différence entre les approches extractives et abstractive</i>	<i>22</i>
<i>Figure 2.11 : Architecture de Encoder-Decoder, Decoder et Encoder.....</i>	<i>24</i>
<i>Figure 2.12 - Architecture du GPT</i>	<i>25</i>
<i>Figure 2.13 - Architecture du BART</i>	<i>26</i>
<i>Figure 2.14 -Schématique de BART</i>	<i>27</i>
<i>Figure 2.15 - Architecture de BERT</i>	<i>29</i>
<i>Figure 2.16 - Pré-entraînement et Ajustement fin de BERT</i>	<i>30</i>
<i>Figure 2.17 - Evolution de LLaMA.....</i>	<i>31</i>
<i>Figure 2.18 - Architecture du LLaMA.....</i>	<i>33</i>
<i>Figure 3.1 - Logo de vs code</i>	<i>39</i>
<i>Figure 3.2 - Logo de WAMP server</i>	<i>39</i>
<i>Figure 3.3 - Logo de MariaDB</i>	<i>39</i>
<i>Figure 3.4 - Logo de modelio.....</i>	<i>40</i>
<i>Figure 3.5 - Logo de python.....</i>	<i>43</i>
<i>Figure 3.6 - Logo de php.....</i>	<i>43</i>
<i>Figure 3.7 - Logo de HTML</i>	<i>43</i>
<i>Figure 3.8 - Logo de CSS</i>	<i>44</i>
<i>Figure 3.9 - Ensemble de donnes des texts.....</i>	<i>45</i>
<i>Figure 3.10 - Les 15 premières lignes de l'ensemble de donnes CNN/DailyMail</i>	<i>46</i>
<i>Figure 3.11 - Ensemble de PDF.....</i>	<i>46</i>
<i>Figure 3.12 - PDFs, AFC.pdf(a gauche) SI.pdf(a droite).....</i>	<i>47</i>
<i>Figure 3.13 - Schéma partie de resume un document pdf</i>	<i>48</i>
<i>Figure 3.14 - Schéma partie de génération des questions.....</i>	<i>49</i>
<i>Figure 3.15 - Schéma partie de chatbot</i>	<i>50</i>
<i>Figure 3.16 - Choix du modèle Llama via api groq</i>	<i>51</i>
<i>Figure 3.17 - Fonction d'extraction du texte du PDF.....</i>	<i>51</i>
<i>Figure 3.18- Fonction de resume du texte extrait.....</i>	<i>51</i>
<i>Figure 3.19 - Fonction de génération des questions et réponses.....</i>	<i>52</i>
<i>Figure 3.20 - Fonction de question/réponse.....</i>	<i>52</i>

<i>Figure 3.21 - Traitement document PDF.....</i>	<i>53</i>
<i>Figure 3.22 - Résumé du document AFC.pdf.....</i>	<i>53</i>
<i>Figure 3.23 - Génération des questions.....</i>	<i>54</i>
<i>Figure 3.24 - Partie des questions directes avec réponses et partie des QCM.....</i>	<i>54</i>
<i>Figure 3.25 - Interaction avec chatbot.....</i>	<i>55</i>
<i>Figure 3.26 - Bibliothèques utilisées.....</i>	<i>56</i>
<i>Figure 3.27 - : Résumé du document AFC.pdf.....</i>	<i>56</i>
<i>Figure 3.28 - Bibliothèque du TextRank.....</i>	<i>57</i>
<i>Figure 3.29 - Résumé de TextRank.....</i>	<i>58</i>
<i>Tableau 3.1 - Normes d'évaluation du BERTScore.....</i>	<i>58</i>
<i>Figure 3.30 - Graphe d'evaluation selon la mesure BERTScore.....</i>	<i>59</i>
<i>Figure 3.32 - Scores des modèles avec FREScore.....</i>	<i>60</i>
<i>Figure 3.33 - Diagramme de cas d'utilisation « Client ».....</i>	<i>62</i>
<i>Figure 3.34 – Diagramme de de cas d'utilisation « Enseignant ».....</i>	<i>63</i>
<i>Figure 3.35 - Diagramme de cas d'utilisation « Etudiant ».....</i>	<i>64</i>
<i>Figure 3.36 - Diagramme de classe.....</i>	<i>65</i>
<i>Figure 3.37 – Interface de connexion.....</i>	<i>66</i>
<i>Figure 3.38 – Interface d'accueil.....</i>	<i>66</i>
<i>Figure 3.39 – Interface d'ajouter un utilisateur.....</i>	<i>67</i>
<i>Figure 3.40 – Interface d'ajouter une catégorie.....</i>	<i>67</i>
<i>Figure 3.41 - Interface d'ajouter un cours.....</i>	<i>68</i>
<i>Figure 3.42 - Interface d'ajouter une ressource.....</i>	<i>68</i>
<i>Figure 3.43 – Interface du résumé un document PDF.....</i>	<i>69</i>
<i>Figure 3.44 – Interface de génération des questions.....</i>	<i>69</i>
<i>Figure 3.45 - Interface de chatbot.....</i>	<i>70</i>

Liste des tableaux

<i>Tableau 1.1 - Tableau comparative entres les plateformes.....</i>	<i>8</i>
<i>Tableau 3.1 - Normes d'évaluation du BERTScore.....</i>	<i>58</i>

Liste des abréviations

AI	Artificial intelligence
API	Application Programming Interface
BART	Bidirectional Auto-Regressive Transformers
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pre-trained Transformer
LCMS	Learning Content Management Systems
LLaMA	Large Language Model Meta AI
LLM	Large Language Model
LMS	Learning Management System
MOOC	Massive open online courses
NLP	Natural Language Process
OCR	Optical Character Recognition
PLATO	Programmed Logic for Automatic Teaching Operations
RNN	Recurrent Neural Network
TALN	Traitement Automatique du Langage Naturel
TF-IDF	Term Frequency-Inverse Document Frequency
UML	Unified Modeling Language

Introduction Générale

Contexte du projet

Depuis plusieurs années, les technologies éducatives transforment les méthodes d'apprentissage en rendant l'éducation plus accessible et interactive. L'essor des plateformes d'apprentissage en ligne (**LMS**) a permis de surmonter les barrières géographiques et temporelles, offrant ainsi une flexibilité d'apprentissage adaptée aux besoins des apprenants. Cependant, face à la multiplication des contenus numériques, les étudiants et les professionnels sont confrontés à une surcharge informationnelle qui peut nuire à l'efficacité de l'apprentissage.

Dans ce contexte, l'intégration des résumés linguistiques dans les plateformes éducatives représente une solution pertinente pour améliorer la compréhension et l'assimilation des connaissances. En exploitant les avancées en traitement automatique du langage naturel (**NLP**) et en intelligence artificielle, il devient possible de générer automatiquement des résumés adaptés aux différents profils d'apprenants, facilitant ainsi la mémorisation et l'apprentissage personnalisé.

L'objectif de ce projet est de concevoir une plateforme éducative interactive capable de produire des résumés linguistiques automatiques pour divers types de contenus pédagogiques (documents PDF). Cette plateforme utilisera des modèles avancés de résumé automatique, permettant d'optimiser l'apprentissage en offrant des contenus synthétisés et adaptés au niveau de chaque utilisateur.

Problématique

Actuellement, face à la vaste quantité de ressources pédagogiques mises à disposition sur internet, les apprenants se trouvent fréquemment débordés par l'excès d'informations, ce qui risque de rendre leur apprentissage plus complexe. Des plateformes modernes telles que Moodle offrent un accès à une multitude de ressources, cependant, elles ne fournissent pas systématiquement des outils pour adapter ou personnaliser ces contenus en fonction des exigences des utilisateurs. Ce projet vise à résoudre ce problème en employant des synthèses automatiques pour faciliter la compréhension et la mémorisation des étudiants.

Objectifs

L'idée principale est de mettre en place une plateforme qui peut générer automatiquement des résumés clairs et personnalisés à partir de différents contenus éducatifs, comme des PDF. Pour cela, on utilisera des technologies avancées de traitement du langage (**NLP**) et des grands modèles de langage (**LLM**). L'objectif est de rendre l'apprentissage plus simple et accessible à tous, tout en gagnant du temps.

Production Automatisée de Résumés Linguistiques

Utilisation de Modèles NLP Sophistiqués : L'emploi de modèles avancés de traitement automatique du langage naturel, comme GPT, DeepSeek et les Transformers, permet de produire des résumés cohérents et appropriés pour le contenu éducatif.

Intégration à la Plateforme Éducative

La plateforme fusionnera ces caractéristiques pour fournir une expérience utilisateur harmonieuse, intégrant le système de synthèse dans un cadre éducatif interactif et facile d'utilisation.

Organisation du mémoire

Mis à part cette introduction, le mémoire est organisé en trois chapitres :

Chapitre 1 : E-learning et plateformes éducatives

Ce chapitre présente l'introduction au e-learning et leur histoire, ainsi que l'histoire de E-learning et les types de plateformes éducatives. Il inclut une comparaison entre des plateformes éducatives.

Chapitre 2 : Traitement automatique du langage naturel (TALN)

Ce chapitre présente l'introduction de TALN, quelques différentes méthodes du TALN, ainsi qu'une définition du résumé automatique, les types et les méthodes du résumé automatique. Il inclut une définition de LLM, ainsi son fonctionnement, son architecture, les modèles les plus connus de LLM et une conclusion.

Chapitre 3 : Expérimentation et la conception

Ce chapitre présente une introduction de ce qu'on fait dans ce projet, l'environnement utilisé dans ce projet, ainsi le flux du travail pour le système, passant par une évaluation entre LLaMA, BART et TextRank. La deuxième partie c'était sur la conception, présentation des acteurs, des besoins et les diagrammes UML (cas d'utilisation et classe), et on termine par la présentation des interfaces et une conclusion de ce chapitre.

Chapitre 1 Apprentissage en ligne (E-learning) et plateformes éducatives

1. Introduction

Ce chapitre présente l'apprentissage en ligne, son histoire, ainsi que les différentes plateformes éducatives et leurs types. Ces éléments sont indispensables pour justifier le choix du type de plateforme effectués dans la suite du projet.

Nous avons commencé par définir l'apprentissage en ligne (E-learning), Décrire son parcours historique. Ensuite nous avons présenté les plateformes éducatives, leurs différents types et un tableau comparatif. Grâce à cette étude, nous avons pu identifier la plateforme open source la plus pertinente pour notre projet.

2. Définition

L'e-learning, représente une révolution dans l'accès à la connaissance, en exploitant les technologies numériques pour offrir un apprentissage flexible, interactif et accessible à un large public.

La définition de l'Union européenne est : « l'e-learning est l'utilisation des nouvelles technologies multimédia de l'Internet pour améliorer la qualité de l'apprentissage en facilitant d'une part l'accès à des ressources et à des services, d'autre part les échanges et la collaboration à distance ». [1]

L'e-learning repose sur plusieurs formats, tels que les cours en ligne ouverts et massifs (**MOOC**), les Learning Management System (**LMS**) comme Moodle, Open Edx et les plateformes interactives intégrant des outils multimédias (vidéos, quiz, forums, réalité virtuelle, etc.). Ces solutions permettent de diversifier les méthodes d'enseignement et d'adapter l'apprentissage aux besoins spécifiques des utilisateurs

3. Histoire d'apprentissage en ligne

L'évolution de l'apprentissage en ligne (e-learning) est un récit passionnant qui couvre plusieurs siècles, passant de techniques d'enseignement à distance basiques à des plateformes numériques avancées telles que Moodle.

3.1 Origines : Cours par correspondance (1840-1873)

L'e-learning trouve ses racines au 19e siècle avec les premiers cours par correspondance. En 1840, Sir Isaac Pitman révolutionne l'éducation à distance en enseignant la sténographie par courrier au Royaume-Uni, une initiative pionnière. En 1873, Anna Eliot Ticknor fonde une société à Boston pour encourager l'apprentissage à domicile par correspondance, malgré des critiques sur le manque d'interaction. Un an plus tard, en 1874, l'Illinois Wesleyan College fait

figure d'avant-gardiste en proposant des diplômes « in absentia », jetant ainsi les bases des programmes universitaires à distance. [2]

3.2 Introduction des machines d'enseignement (1920-1954)

Au début du 20e siècle, l'e-learning fait un bond en avant avec l'apparition des machines d'enseignement. En 1920, Sidney Pressey invente une « machine d'enseignement » pensée pour tester les élèves, jetant les bases de ce qu'on appelle l'instruction programmée. En 1954, B.F. Skinner va plus loin en créant la machine « Glider », qui ne se contente pas d'évaluer, mais guide réellement l'apprentissage, donnant aux écoles un outil innovant pour structurer l'enseignement. [2]

3.3 Utilisation des médias de diffusion (1925-1930s)

Dans les années 1920 et 1930, l'éducation à distance prend un nouvel élan grâce aux technologies de diffusion. En 1925, l'Université de l'Iowa lance le tout premier « cours radio » de cinq crédits, utilisant la radio pour toucher un large public, ce qui marque une étape importante. Dans les années 1930, cette même université adopte le téléphone et la télévision pour enrichir ses cours à distance, rendant l'apprentissage plus accessible. Après les années 1930, la Fondation Annenberg et la Corporation for Public Broadcasting créent des « Tele-Courses » pour les universités, surtout dans des zones isolées comme l'Alaska, intégrant ces méthodes dans les pratiques éducatives courantes. [2]

3.4 Ère des ordinateurs et premiers systèmes e-learning (1960s-1970s)

Dans les années 1960 et 1970, l'e-learning prend forme avec l'arrivée des technologies informatiques. Au début des années 1960, Don Bitzer lance PLATO, un système d'instruction par ordinateur qui révolutionne l'apprentissage avec des forums, courriels, chats et jeux multi-joueurs, posant les bases des communautés en ligne d'aujourd'hui. Pendant cette décennie, l'Apprentissage Assisté par Ordinateur pour les Langues (CALL), ou « Behaviorist CALL », voit le jour, utilisant des exercices répétitifs, des tests de traduction et des explications grammaticales. En 1967, Patrick Suppes crée la Computer Curriculum Corporation, qui développe un système de gestion de l'instruction (CMI) pour enseigner les maths aux enfants. En 1971, Murray Turoff invente le système de conférence Delphi (EMISARI), considéré comme le premier du genre. Enfin, en 1974, le New Jersey Institute of Technology introduit la « Virtual Classroom » et le système EIES, renforçant les communications éducatives et scientifiques. [2]

3.5 L'essor de l'e-learning : des ordinateurs personnels au web (1980-1990)

Dans les années 1980 et 1990, l'e-learning connaît une transformation majeure grâce aux ordinateurs personnels et à l'avènement du web. En 1980, l'introduction des PC, comme le MAC d'Apple, rend l'apprentissage à domicile plus accessible, renforçant les compétences informatiques et donnant une forme concrète à l'e-learning. Dès le début des années 1980, BitNet connecte les universités mondialement, ouvrant la voie à des réseaux éducatifs collaboratifs pour des cours en ligne, des salles de classe virtuelles et des diplômes. En 1985,

Nova Southeastern University propose des programmes de diplôme en ligne pour les études supérieures, suivie en 1989 par l'Université de Phoenix, qui lance le premier diplôme universitaire en ligne, et par Mindweave Communications, qui publie Lotus Notes, une plateforme commerciale d'apprentissage. Pendant cette période, le CMI évolue vers la Formation (CBT) et l'Apprentissage Basé sur Ordinateur (CBL), tandis que le « Communicative CALL » utilise les ordinateurs pour des activités comme la correction orthographique. Dans les années 1990, les formations sur CD permettent de stocker textes, vidéos et animations, et l'«Integrative CALL» intègre le multimédia pour enseigner les compétences linguistiques. L'e-learning s'intègre aux curriculums avec des activités comme les correspondances électroniques et les forums. En 1998, le World Wide Web (WWW) révolutionne l'accès aux contenus interactifs, remplaçant les CD et popularisant la Formation Basée sur le Web (WBT) via les navigateurs. [2]

3.6 L'e-learning à l'ère des systèmes modernes et du mobile (2000-2020)

Dans les années 2000 et 2010, l'e-learning entre dans une phase moderne avec des technologies avancées et une portée mondiale. En 1996, Cecil devient le premier système de gestion de l'apprentissage (LMS) basé sur le web, permettant d'organiser et de diffuser des cours en ligne. En 2001, Martin Dougiamas lance Moodle, un LMS open-source qui facilite la création de cours interactifs. En 2005, Jawed Karim, Steve Chen et Chad Hurley fondent YouTube, révolutionnant le partage de vidéos éducatives et démocratisant la diffusion de contenu. En 2008, George Siemens et Stephen Downes introduisent les MOOCs avec leur cours « Connectivism and Connective Knowledge » à l'Université du Manitoba, attirant 2300 participants du monde entier. Après 2008, des plateformes comme Udacity, Coursera, FutureLearn et EdX émergent, collaborant avec des universités pour offrir des cours en ligne à des millions d'étudiants. Dans les années 2010, le mobile learning (m-learning) prend son essor, utilisant smartphones, tablettes et ordinateurs portables pour un apprentissage mobile, collaboratif et spontané. En 2020, la pandémie de COVID-19 booste l'e-learning, avec la fermeture des écoles et le passage au virtuel, touchant 1,5 milliard d'apprenants et 63 millions d'éducateurs selon l'UNESCO, tandis que le marché mondial de l'e-learning est estimé à plus de 243 milliards de dollars d'ici 2022. [2]

4. Plateformes éducatives

Dans le domaine de l'apprentissage en ligne, les plateformes éducatives jouent un rôle crucial pour rendre l'éducation accessible à tous. Ces plateformes ont transformé l'apprentissage en ligne en proposant des outils numériques sur mesure pour les apprenants et les professeurs. Parmi ces outils, les systèmes de gestion de l'apprentissage (SGA), en anglais c'est Learning management system (LMS) offrent la possibilité de gérer et de dispenser des cours, d'assurer le suivi des progrès des apprenants, ainsi que favoriser les interactions éducatives.

Les systèmes de gestion de contenus d'apprentissage (LCMS) pour leur part se focalisent sur la conception, l'administration et la réutilisation de contenus pédagogiques.

4.1 Types des plateformes éducatives

Lorsque l'on aborde le sujet des plateformes éducatives, on peut les classer en plusieurs grandes catégories, toutes présentant une légère variation dans leur fonction.

4.1.1 Learning Management System (LMS)

Également connue sous le nom de système de gestion de l'apprentissage (SGA) est un logiciel qui vous aide à créer, gérer, organiser et fournir du matériel d'apprentissage en ligne aux apprenants. [3]

Exemple: Moodle

4.1.2 Learning Content Management Systems (LCMS):

En français est les systèmes de gestion de contenus d'apprentissage, c'est un système de gestion de contenu d'apprentissage ou de formation. Il permet aux formateurs de créer des ressources éducatives personnalisées et d'organiser ces réserves d'informations dans des modules, cours ou parcours. [4]

Exemple : Xyleme LCMS

4.1.3 Massive open online courses (MOOC) :

Le **MOOC** signifie Massive Open Online Course, autrement dit, c'est un nouveau format pédagogique qui consiste à mettre en libre accès des formations en ligne pour les apprenants. [5] Les MOOCs sont généralement gratuits ou d'un coût modique, proposées par des experts ou des universités.

Exemple : Coursera.

4. Tableau comparative entres les plateformes

Le tableau ci-dessous compare cinq plateformes éducatives en ligne : Moodle, Open Edx, Chamilo, Canvas et Coursera. Il analyse plusieurs critères techniques et fonctionnels tels que le type de plateforme, le langage de programmation utilisé, la facilité d'installation, ainsi que la scalabilité.

Tableau 1.1 - Tableau comparative entre les plateformes

Critère	Moodle	Open Edx	Chamilo	Canvas	Coursera
Définition	Un LMS Utilisé pour gérer, dispenser et évaluer la formation et l'apprentissage en ligne. Parfait pour écoles et entreprises. [6]	Un LMS en ligne open source pour créer et diffuser des cours massifs, idéale pour universités et grandes organisations . [7]	Un LMS à distance sous forme de logiciel open - source, simple pour gérer des cours et formations, pensé pour petites structures. [8]	Un lms open source qui sert de plateforme centrale pour les cours en ligne. Solution moderne, Souvent utilisée dans les écoles. [9]	Un MOOC propose une variété de sujets de formats d'apprentissage, tel que des cours, des diplômes. [10]
Type	LMS	LMS	LMS	LMS	MOOC
Langage	PHP	Python	PHP	Ruby	Java, Scala, Python
Facilité d'installation	Facile, nécessite un peu technique au début	Complexe, demande des compétences techniques	Facile et compatible	Facile et compatible	Non installable localement (solution hébergée)
Scalabilité	Bonne, supporte des dizaines de milliers d'utilisateurs	Excellente, conçue pour des millions (MOOCs)	Moyenne, adaptée à des milliers d'utilisateurs	Bonne, scalable avec support Instructeur	Excellente, utilisée par des millions d'étudiants

5. Conclusion

Le premier chapitre a introduit le concept d'e-learning, son histoire et présenté les plateformes éducatives, leurs différents types, aussi qu'une comparaison entre ces plateformes. En conclusion pour la réalisation de notre projet (plateforme éducative interactive basée sur l'optimisation des apprentissages grâce aux résumés linguistiques), nous avons optés **pour Moodle**. Le choix de Moodle s'est imposé en raison : sa souplesse grâce aux plugins et sa capacité à accueillir un grand nombre d'utilisateurs, ce qui le rend idéal pour divers contextes éducatifs. De plus, son aptitude à intégrer des outils d'intelligence artificielle via des plugins en phase expérimentale ouvre la voie à intégration des fonctionnalités sophistiquées telles que le résumé linguistique automatique pour l'apprentissage.

Chapitre 2 Traitement Automatique du Langage Naturel (TALN)

1. Introduction

Avec la diffusion de la technologie numérique et de l'apprentissage à distance, les étudiants trouvent tous les cours voulus via des plateformes d'apprentissage affiliée à leur université comme **MOODLE**, et des divers documents (format PDF) qu'ils possèdent. Mais parfois, les étudiants souhaitent se concentrer uniquement sur les points essentiels de ces cours et ces documents afin de mieux comprendre les points essentiels avant dans commencer une étude profonde.

Dans ce contexte, le résumé automatique s'impose comme une solution efficace grâce à traitement de langage naturel (NLP).

Le traitement automatique du langage naturel (NLP) est un sous-domaine de l'informatique et de l'intelligence artificielle (IA) qui utilise le machine learning pour permettre aux machines de comprendre, interpréter, générer et manipuler le langage humain de manière naturelle et efficace [11]. Le TALN englobe des applications telles que la traduction automatique, la reconnaissance vocale, la synthèse de textes, l'analyse de sentiments, les chatbots ou encore les réponses automatiques aux questions.

2. Différentes applications de TALN

TALN (traitement automatique de langage naturel) est une discipline qui trouve des applications dans de nombreux domaines, en particulier dans l'éducation, les technologies, les affaires et la communication, voici une présentation des principales applications du TALN.

2.1 Traduction automatique

C'est le processus qui permet à une machine de traduire un texte d'une langue à une autre, comme passer du français à l'anglais comme par exemple, Google Translate qui utilise la technique transformer.

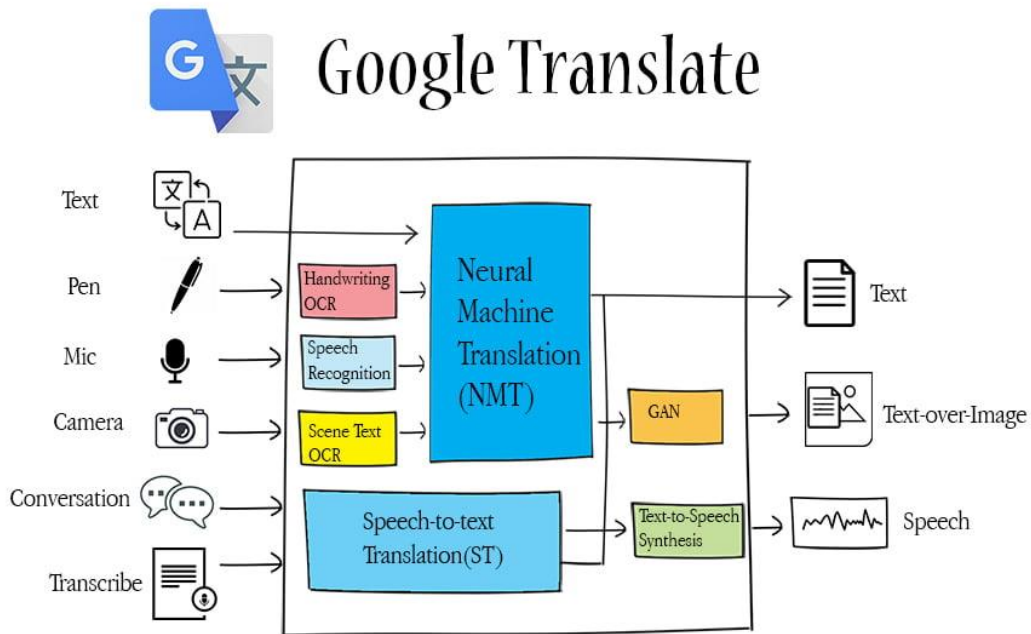


Figure 2.1 - Architecture et fonctionnement de google traduction [12]

La figure 2.1 illustre le fonctionnement de Google Translate, une application de traduction basée sur le TALN. Elle montre comment des entrées comme le texte, l'écriture manuscrite (OCR), la parole (reconnaissance vocale), les images (OCR de scène) ou les conversations sont traitées par un moteur de traduction neuronale (NMT). Ce dernier, aidé par des modules comme GAN (pour les images traduites) et Speech-to-text Translation (pour la parole), génère des sorties sous forme de texte, d'images annotées ou de parole synthétique.

2.2 Reconnaissance vocale

La reconnaissance de la parole consiste à transcrire automatiquement un contenu parlé afin d'obtenir la séquence de mots correspondante. [13]

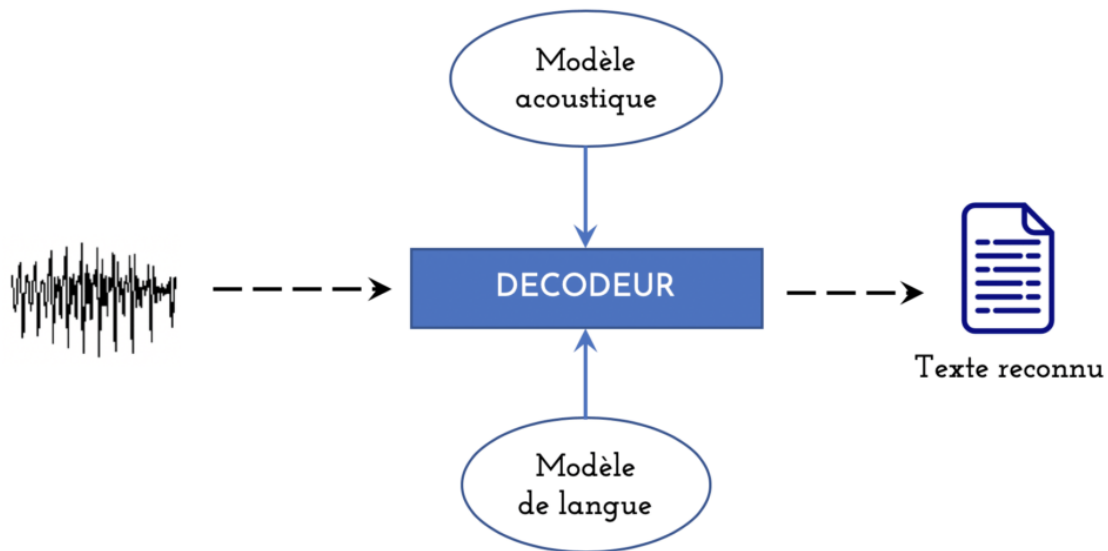


Figure 2.2: Architecture classique (et hybride) d'un Système de Reconnaissance Automatique de la Parole (SRAP) [13]

La figure 2.2 illustre le processus de reconnaissance automatique de la parole, qui transforme un signal audio en texte reconnaissable. À gauche, une onde sonore représentant la parole est introduite dans le système, puis traitée par un décodeur central qui repose sur deux modèles complémentaires : le modèle acoustique, qui analyse les caractéristiques sonores pour identifier les phonèmes, et le modèle de langue, qui utilise des règles grammaticales et statistiques pour assembler ces sons en mots cohérents. Ce décodeur, en combinant ces deux approches, produit à droite un texte reconnu, comme un document transcrit à partir d'une conférence ou d'une conversation.

2.3 Classification de textes

La classification de texte consiste à attribuer automatiquement une ou plusieurs catégories prédéfinies à un texte en fonction de son contenu ou de son sens.

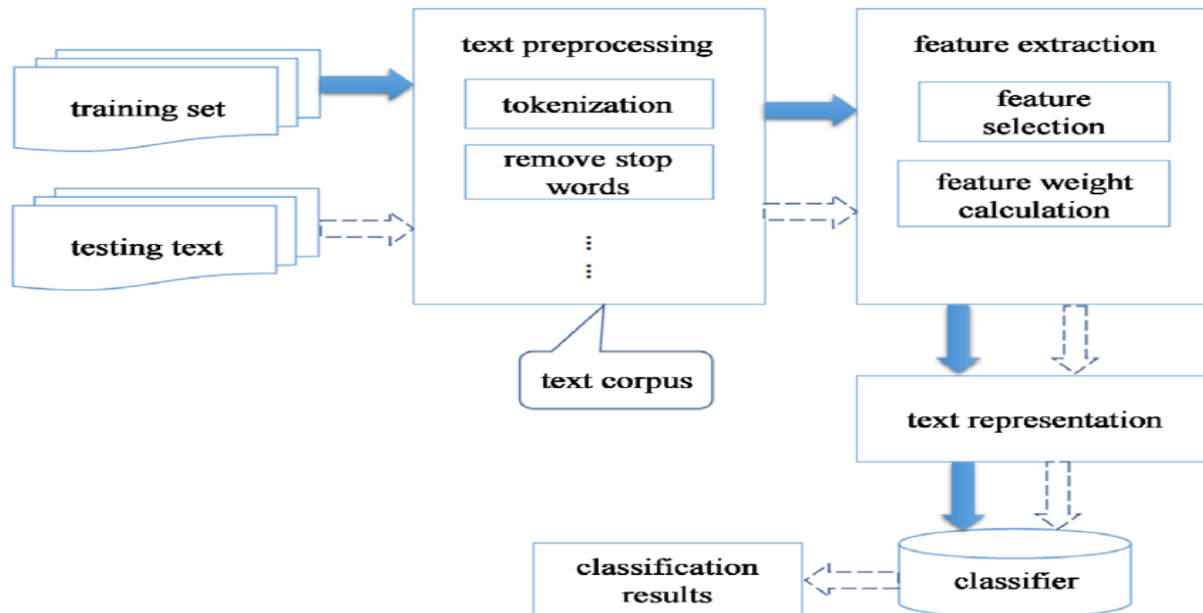


Figure 2.3: Architecture de classification de texte [38]

Elle se décompose en plusieurs étapes interconnectées. Tout d'abord, les données d'entrée incluent un training set (ensemble d'entraînement) et un testing text (texte à tester), qui sont transmis à la phase de prétraitement du texte. Cette étape comprend la tokenisation (division du texte en unités comme les mots) et la suppression des mots vides (élimination des mots non significatifs comme « le » ou « et »), créant un texte corpus (corpus de texte traité). Ensuite, le texte corpus est envoyé à la phase d'extraction des caractéristiques, où la sélection des caractéristiques et le calcul du poids des caractéristiques (comme avec TF-IDF) déterminent les éléments pertinents du texte. Ces caractéristiques sont ensuite transformées en une représentation de texte (ex. : vecteurs), qui est utilisée par un classificateur pour produire des résultats de classification (ex. : catégorisation en « positif » ou « négatif »). Le classificateur s'appuie sur les données d'entraînement pour apprendre et tester sur les données non vues, optimisant ainsi la précision.

2.4 Génération du résumés automatiques

Le résumé de texte consiste à réduire un texte à son contenu essentiel sans intervention humaine directe Cette tâche repose sur des algorithmes capables d'analyser, comprendre et synthétiser les informations, utile pour résumer des articles, des textes et des cours. [14]

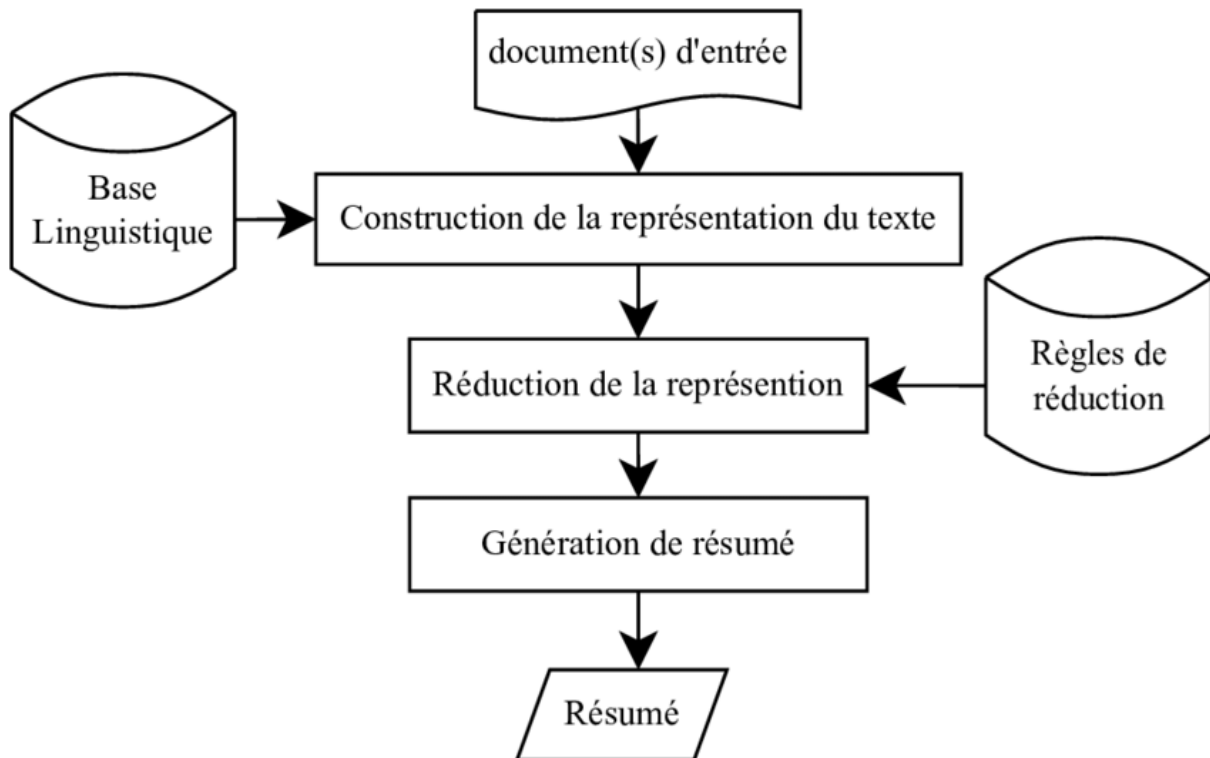


Figure 2.4 - Modèle général pour l'approche linguistique de résumé automatique [15]

La figure 2.4 illustre le processus de résumé automatique de texte, une application du traitement automatique du langage naturel (TALN). Le processus commence avec un ou plusieurs documents(s) d'entrée, qui sont transmis à l'étape de construction de la représentation du texte. Cette étape transforme le texte brut en une structure compréhensible, en s'appuyant sur une base linguistique (connaissances grammaticales, lexicales, etc.). Ensuite, la réduction de la représentation simplifie cette structure en éliminant les informations non essentielles, en utilisant des règles de réduction spécifiques. La phase finale, génération de résumé, produit un résumé concis qui conserve l'essentiel du contenu initial.

3. Résumé Automatique

Le résumé automatique consiste à prendre un long texte, ou même un ensemble de textes, et à générer de façon automatique un texte court qui contient la majorité des informations [16]. Cette méthode permet d'extraire ou de reformuler les informations principales, que ce soit pour des articles scientifiques, des rapports ou des contenus éducatifs, le résumé automatique gagne en importance, aidant les utilisateurs à gérer la surcharge informationnelle tout en personnalisant l'apprentissage.

Le résumé automatique fonctionne en analysant un texte pour identifier les phrases ou concepts les plus pertinents, souvent grâce à des algorithmes d'apprentissage automatique ou des modèles de deep learning. Il existe deux grandes approches : Résumé extractif et Résumé abstrait. En commençant par expliquer ces deux approches.

Dans le contexte éducatif, cela permet par exemple de résumer un cours long en quelques points clés pour les étudiants, réduisant leur temps de lecture tout en préservant l'essentiel.

3.1 Types de résumé automatique

Le résumé automatique se divise en deux grandes approches Résumé extractif et Résumé abstrait. Ces techniques répondent à des besoins différents, l'extractif est rapide et précis pour des textes factuels, tandis que l'abstrait est plus naturel et adapté à des synthèses complexes.

3.1.1 Résumé Extractif

Le résumé extractif est une technique de résumé automatique qui consiste à extraire les phrases les plus importantes d'un texte sans les modifier. On peut dire aussi consiste à extraire telles quelles des informations avant de les recoller pour créer un résumé. [16]

Exemple 1

- **Texte source** : "Le traitement automatique du langage naturel (TALN) est un domaine de l'intelligence artificielle. Il permet aux machines de comprendre et de générer du langage humain. Le TALN est utilisé dans les assistants vocaux, les systèmes de traduction et les moteurs de recherche."
- **Résumé extractif** : "Le traitement automatique du langage naturel (TALN) est un domaine de l'intelligence artificielle. Il permet aux machines de comprendre et de générer du langage humain."

Exemple 2

- **Texte original** : "Les progrès récents dans le traitement du langage naturel ont révolutionné la recherche d'informations. Les chercheurs ont développé des modèles puissants comme BERT et GPT-4."
- **Résumé extractif** : "Les progrès dans le traitement du langage naturel incluent des modèles tels que BERT et GPT-4."

Les phrases extraites ont été choisies car elles sont centrales pour la compréhension du sujet.

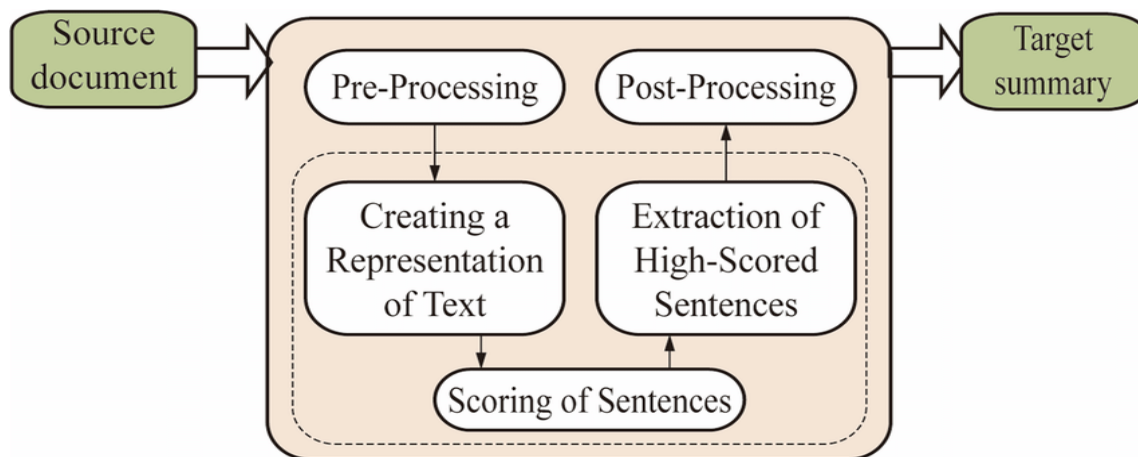


Figure 2.5 - Architecture d'un système de résumé extractif

La figure 2.5 illustre l'architecture d'un système de résumé de texte extractif, une méthode de traitement automatique du langage naturel (TALN). Le processus commence avec un source document (document source), qui est d'abord soumis à une étape de pré-traitement pour nettoyer et structurer le texte (ex. : tokenisation, suppression des mots vides). Ensuite, une représentation du texte est créée, suivie d'une étape de scoring des phrases, où chaque phrase est évaluée selon des critères comme sa pertinence ou sa position (via des algorithmes comme TextRank). Les phrases les mieux notées sont alors sélectionnées lors de l'extraction des phrases à haut score. Enfin, un post-traitement ajuste le résumé pour assurer sa cohérence, produisant un target summary (résumé cible).

3.1.2 Résumé abstractive

Le résumé abstrait est une technique avancée de résumé automatique qui ne se contente pas d'extraire des phrases du texte original (comme le fait un résumé extractif), mais génère un nouveau texte à partir de la compréhension globale du contenu. Il reformule le contenu de manière concise, tout en préservant les idées principales. D'une autre façon le résumé abstrait consiste à créer de nouvelles phrases, qui n'apparaissent pas à l'origine dans les documents, afin d'avoir un résumé plus fluide et plus libre [16].

Exemple 1

- **Texte source** : "Le développement de la technologie de l'information a profondément modifié la manière dont les entreprises interagissent avec leurs clients."
- **Résumé abstractive** : "Les technologies de l'information ont révolutionné les relations entre entreprises et clients."

Exemple 2

- **Texte original** : "L'essor de l'apprentissage profond a transformé divers domaines. Les réseaux de neurones excellent dans la reconnaissance d'images, la synthèse vocale et la traduction linguistique."
- **Résumé abstrait** : "L'apprentissage profond, en particulier les réseaux de neurones, a révolutionné la reconnaissance d'images, la synthèse vocale et la traduction."

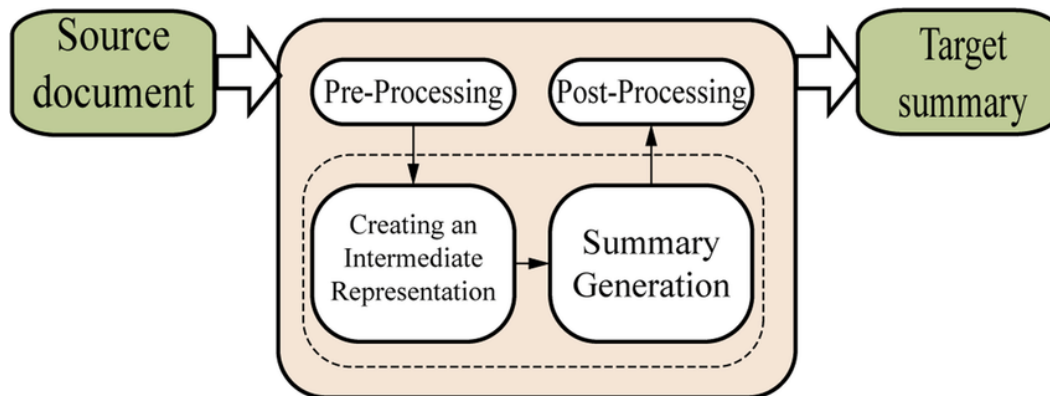


Figure 2.6 - Architecture d'un système de résumé abstrait

La figure 2.6 illustre l'architecture d'un système de résumé de texte abstrait, une méthode du traitement automatique du langage naturel (TALN). Le processus commence avec un source document (document source), qui est d'abord traité par une étape de pré-traitement pour nettoyer et structurer le texte (ex. : tokenisation, suppression des mots inutiles). Ensuite, une représentation intermédiaire est créée, souvent à l'aide de modèles comme les transformateurs (ex. : BERT), pour capturer le sens et le contexte du texte. Cette représentation est ensuite utilisée dans la phase de génération de résumé, où un décodeur génère un texte reformulé original. Enfin, un post-traitement ajuste le résumé pour assurer sa clarté et sa cohérence, produisant un target summary (résumé cible).

3.2 Algorithmes de Résumé Automatique

Les techniques de résumé de texte sont des procédés informatiques destinés à générer des condensés pertinents et logiques de textes tout en conservant les éléments clés et le sens du contenu initial. Ces algorithmes occupent une place centrale dans le secteur du traitement automatique du langage naturel (TALN), facilitant la réduction de vastes quantités de texte en formats plus simples à comprendre.

Dans un monde inondé d'informations et de contenu digital en constante augmentation, ces algorithmes se sont avérés essentiels. Ils offrent aux utilisateurs la possibilité d'extraire rapidement les points principaux d'un document sans devoir lire l'ensemble du texte, ce qui les rend indispensables dans des domaines tels que l'éducation.

L'importance des algorithmes de résumé linguistique se trouve dans leur potentiel à rendre les informations plus accessibles et compréhensibles. Toutefois, leur développement ne se fait pas sans difficultés. Il s'agit de surmonter des défis considérables tels que la compréhension du contexte, le maintien de la cohérence du résumé et l'assurance de sa précision face aux nuances du langage humain. Les évolutions récentes en matière de machine Learning, en particulier grâce à des modèles de langage de grande taille tels que BART, T5 ou GPT, ont conduit à des améliorations majeures en affinant la capacité des algorithmes à saisir les relations sémantiques et à générer des résumés plus fluides.

3.2.1 TextRank

L'algorithme de résumé automatique TextRank, basé sur la théorie des graphes, a été présenté par Rada Mihalcea et Paul Tarau en 2004 [17]. Il permet d'identifier les phrases les plus importantes dans un texte. Le concept de base de TextRank consiste à attribuer un score à chaque phrase en fonction de son importance. La première phrase affichée est considérée comme l'idée principale du texte et peut également être interprétée comme son résumé. [18]

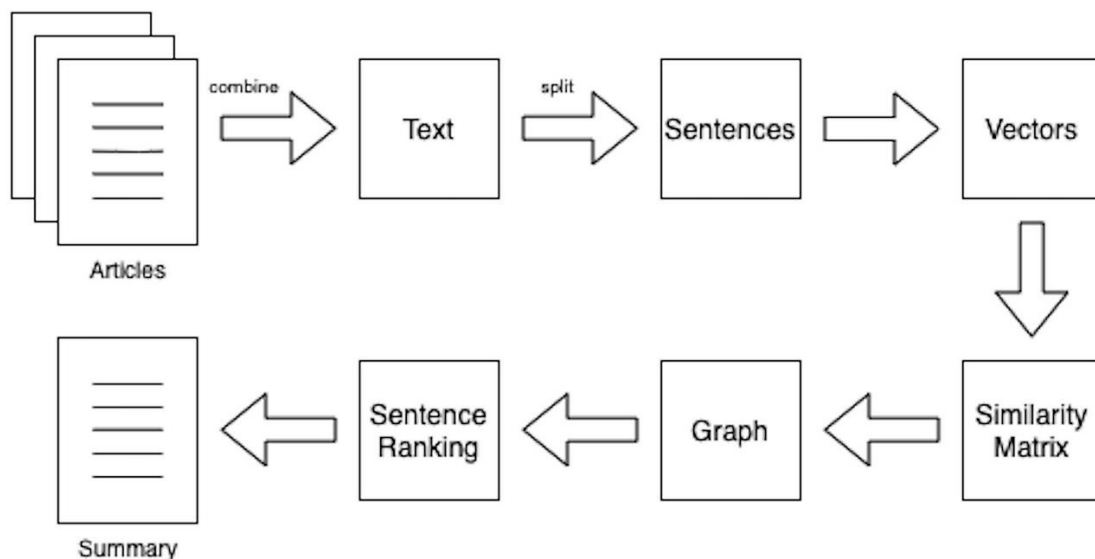


Figure 2.7 - Résumé de texte utilisant l'algorithme TextRank (Logique de base) [18]

La figure 2.7 illustre le processus de résumé de texte basé sur un algorithme de classement de graphes TextRank, utilisé dans le traitement automatique du langage naturel (TALN). Le processus commence avec plusieurs articles, qui sont combinés pour former un texte unique. Ce texte est ensuite divisé en phrases individuelles, qui sont converties en nœuds pour une représentation numérique (par exemple, via des embeddings). Parallèlement, à partir d'un

résumé ou d'un texte cible, une étape de classement des phrases évalue leur importance en s'appuyant sur un graphe de relations, construit à partir d'une matrice de similarité qui mesure les liens entre phrases (basée sur leur contenu sémantique). Le graphe oriente le classement, permettant de sélectionner les phrases les plus pertinentes pour générer le résumé final.

Exemple

Texte source: *'He is a nice guy. He has a lot of friends. Raj is his best friend'.*

Le texte ci-dessus comporte 3 phrases séparées par un point. Créons d'abord une matrice de similarité M.

	He is a nice guy	He has a lot of friends	Raj is his best friend
He is a nice guy	0	0.53	0.2
He has a lot of friends	0.53	0	0.9
Raj is his best friend	0.2	0.9	0

Figure 2.8 - Matrice de similarité

Ce résultat est obtenu après l'application de l'algorithme PageRank, ce qui montre que le score le plus élevé se situe entre « He has a lot of friends » et « Raj is his best friend ». Ici, nous observons que la similarité entre les mots « amis » et « ami » augmente le score par rapport à la première phrase. Ainsi, en conclusion, le résumé du texte source peut être « He has a lot of friends, Raj is his best friend » ou « Raj is his best friend, He has a lot of friends ». Cependant, comme TextRank respecte le sens et l'ordre des phrases, le résultat final est « Il a beaucoup d'amis, Raj est son meilleur ami ».

3.2.2 Transformers

Transformer est le premier modèle de transduction qui repose entièrement sur l'auto-attention pour calculer les représentations de son entrée et de sa sortie, sans utiliser de RNN alignés séquentiellement ni de convolutions, Introduits en 2017 par Vaswani et al, dans leur article « Attention Is All You Need » [19]. Les Transformers ont radicalement modifié le paysage du traitement automatique du langage naturel (TALN). Grâce à leur aptitude à gérer efficacement les dépendances de longue durée dans les séquences de données et à offrir la possibilité de calculs parallèles, ils se distinguent des modèles récurrents plus anciens.

3.2.2.1 Composants principaux des Transformers

Les Transformers reposent sur plusieurs éléments clés qui les rendent uniques et performants Figure 2.9 :

3.2.2.1.1 Mécanisme d'attention auto-régressive (Self-Attention)

Ce mécanisme permet au modèle d'évaluer l'importance de chaque mot dans une phrase par rapport aux autres. Par exemple, dans "Le chat mange le poisson", le modèle peut déterminer que "mange" est lié à "chat" et "poisson". Cela améliore la compréhension du contexte global.

3.2.2.1.2 Attention multi-têtes masquée (Masked Multi-Head Attention)

Une variante de l'attention qui masque les positions futures dans la séquence de sortie pour assurer que la prédiction ne dépend que des mots précédents (auto-régression). Cela empêche de "tricher" en regardant les mots à venir.

3.2.2.1.3 Attention multi-têtes (Multi-Head Attention)

Plusieurs "têtes" d'attention fonctionnent en parallèle, capturant différentes relations dans les données. Les résultats sont ensuite combinés pour obtenir une représentation plus complète et nuancée de la séquence.

3.2.2.1.4 Encodage positionnel

Les Transformers ne traitent pas les données de manière séquentielle comme les réseaux récurrents. Pour indiquer l'ordre des mots, des encodages positionnels sont ajoutés aux embeddings d'entrée, informant le modèle de la position de chaque élément.

3.2.2.1.5 Structure encodeur-décodeur

- **Encodeur** : Transforme la séquence d'entrée en une représentation riche en une seule passe.
- **Décodeur** : Génère la séquence de sortie, un mot à la fois, en s'appuyant sur la représentation de l'encodeur. Cette structure est idéale pour des tâches comme la traduction.

3.2.2.1.6 Réseaux Feed-Forward

Chaque couche inclut un réseau feed-forward appliqué à chaque position, enrichissant les représentations à chaque étape du traitement.

3.2.2.1.7 Normalisation des couches et connexions résiduelles

Ces techniques stabilisent l'entraînement et permettent de construire des modèles plus profonds en facilitant la propagation des gradients.

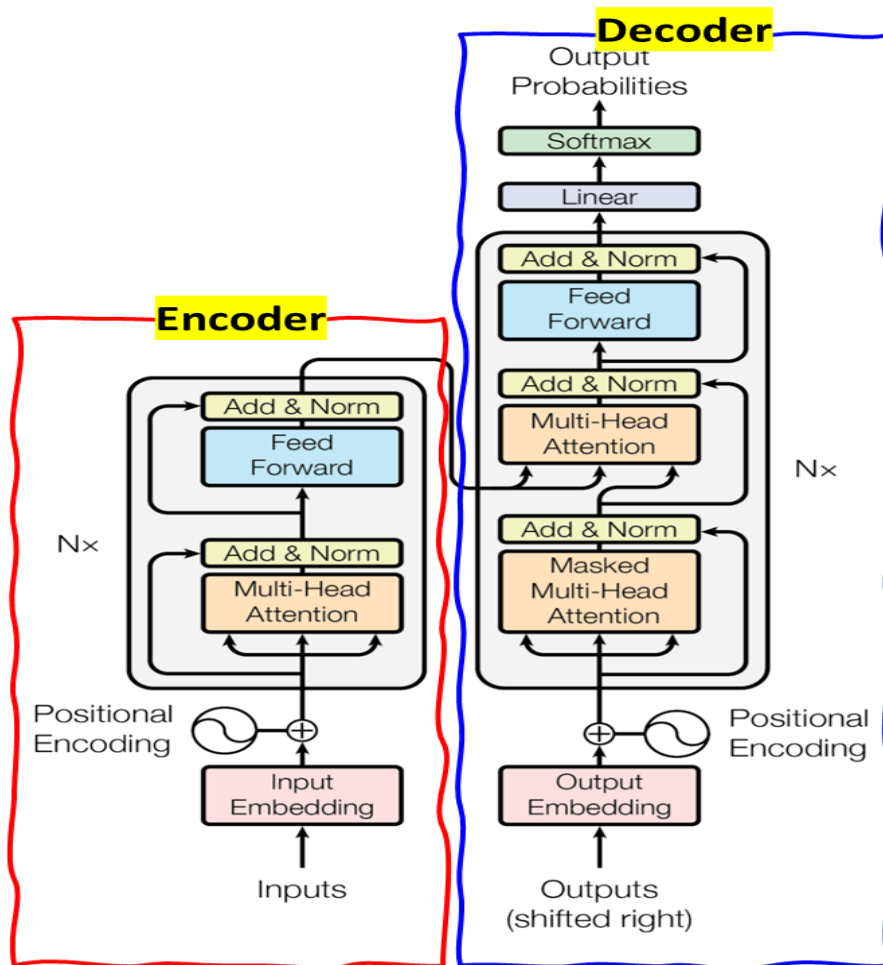


Figure 2.9 - Architecture du transformer

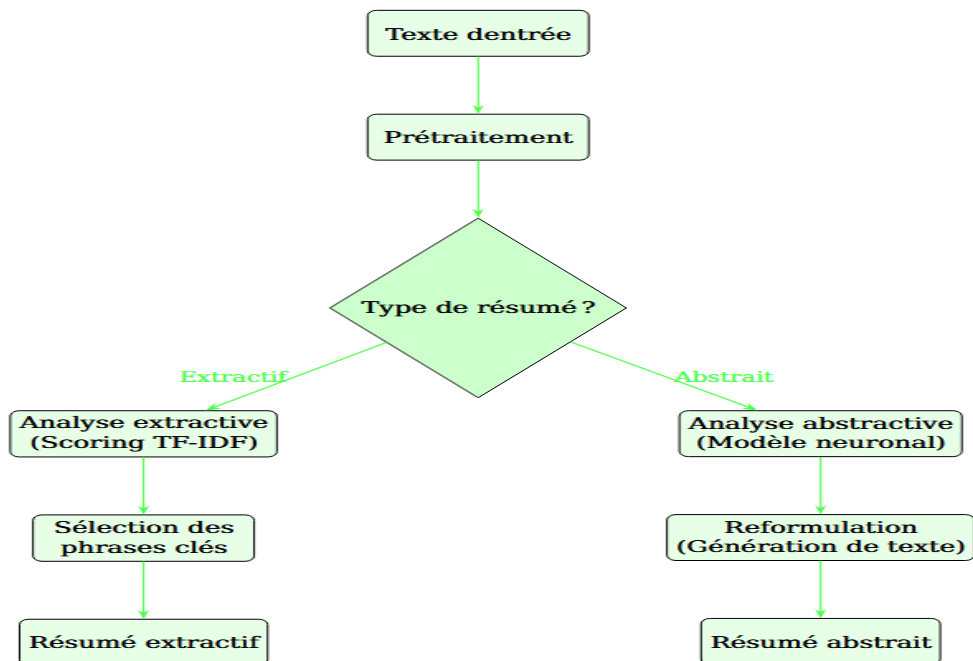


Figure 2.10 - Différence entre les approches extractives et abstractive

4. Large Language Models (LLMs)

Les modèles de langage à grande échelle (LLMs), qui ont la capacité de générer des textes cohérents, répondre à des interrogations complexes et aider dans diverses activités, sont au centre de cette révolution.

Ces progrès ont des conséquences significatives dans des domaines tels que la technologie, l'éducation et la santé, rendant les échanges entre l'homme et la machine plus naturels et instinctifs.

4.1 Définition

Les modèles de langage de grande taille, appelés LLM (Large Language Models), sont des systèmes d'intelligence artificielle capables de comprendre, générer et interagir en langage naturel, fondés sur des architectures de type transformer [20]. Il est capable de produire du texte de manière similaire à un humain.

4.2 Fonctionnement des LLMs

Le fonctionnement des LLMs se base sur un processus en deux phases majeurs : le pré-entraînement et le fine-tuning.

4.2.1 Pré-entraînement de LLMs

Durant la phase de pré-entraînement, le modèle est exposé à un corpus de texte non étiqueté et apprend à prédire des mots ou des segments de texte masqués. Par exemple :

- Dans le cas de **BERT**, le modèle est entraîné à prédire des mots masqués dans une phrase tout en tenant compte du contexte bidirectionnel.
- Pour **GPT**, le modèle apprend à prédire le mot suivant dans une séquence, ce qui lui permet de générer du texte de manière cohérente.

4.2.2 Fine-tuning de LLMs

Après le pré-entraînement, le modèle peut être affiné (*fine-tuned*) sur des données spécifiques à une tâche particulière, comme la traduction ou le résumé de texte. Cette étape permet d'ajuster les paramètres du modèle pour optimiser ses performances sur des applications concrètes, souvent avec des ensembles de données beaucoup plus petits.

4.3 Architecture des LLMs

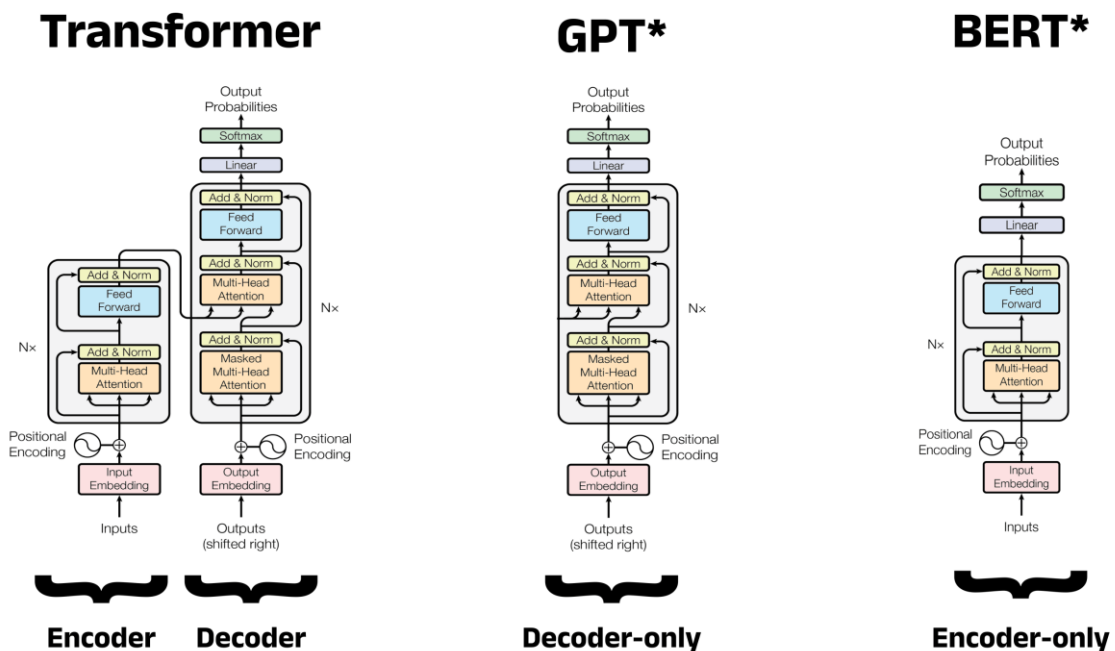
L'architecture des LLMs est basée sur le modèle *Transformer*, qui se compose de deux parties principales : l'**encodeur** et le **décodeur**.

- **Encodeur** : L'encodeur traite la séquence d'entrée et génère une représentation contextuelle riche de chaque mot ou token, en tenant compte de son contexte dans la phrase.

- **Décodeur** : Le décodeur utilise cette représentation pour générer la sortie, souvent de manière autorégressive, c'est-à-dire en prédisant un token à la fois en fonction des tokens précédents.

Les LLMs peuvent être configurés de différentes manières :

- **Modèles encodeur-only** (ex. : BERT) : Excellents pour les tâches de compréhension, telles que la classification de texte ou l'extraction d'entités.
- **Modèles décodeur-only** (ex. : GPT) : Spécialisés dans la génération de texte, comme la complétion de phrases ou la création de contenu.
- **Modèles encodeur-décodeur** (ex. : T5) : Polyvalents, utilisés pour des tâches nécessitant à la fois compréhension et génération, comme la traduction.



*Illustrative example, exact model architecture may vary slightly

Figure 2.11 : Architecture de Encoder-Decoder, Decoder et Encoder

1. Modèles les plus connus de LLM

5.1 Generative Pre-trained Transformer (GPT)

Est un modèle d'intelligence artificielle basée sur un modèle de réseau neuronal profond appelé « transformer », qui a révolutionné le domaine du traitement du langage naturel [21].

GPT est spécialisé dans la génération : il peut créer des phrases, des paragraphes, voire des articles entiers, en se basant sur un simple point de départ.

Architecture

GPT est basé sur l'architecture Transformer, une structure novatrice présentée en 2017 par Vaswani et ses collaborateurs. À l'inverse des modèles plus anciens tels que les réseaux de neurones récurrents (RNN), le Transformer traite le texte en parallèle, ce qui lui confère une rapidité et une efficacité supérieures.

Décodeur uniquement (Decodeur-Only)

GPT utilise exclusivement le décodeur du Transformer, ce qui le différencie de modèles tels que BERT, qui s'appuie sur l'encodeur. GPT fonctionne comme un décodeur autorégressif, générant du texte en anticipant chaque mot basé sur ceux qui le précèdent. C'est comme si le modèle composait un récit, mot après mot, en s'appuyant sur ce qu'il a déjà produit.

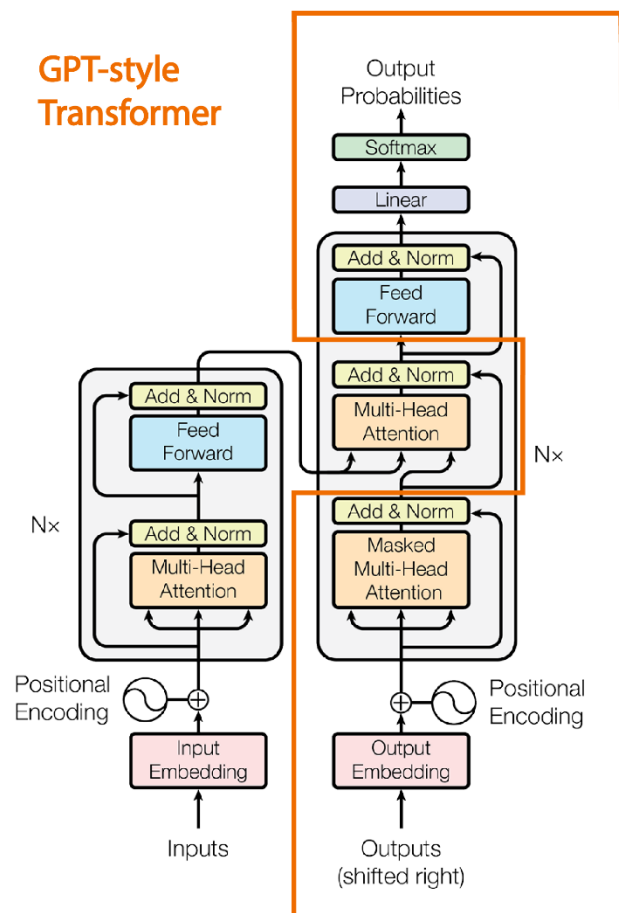


Figure 2.12 - Architecture du GPT

5.2 Bidirectional and Auto-Regressive Transformers (BART)

BART, ou Bidirectional and Auto-Regressive Transformers, est un modèle de TALN et un auto-encodeur de débruitage qui mappe un document corrompu vers le document original dont il est dérivé [22]. C'est un modèle Transformer encodeur-décodeur, ce qui signifie qu'il a deux parties : une pour lire et comprendre le texte (l'encodeur), et une autre pour produire du texte (le décodeur).

Architecture

L'architecture de BART est basée sur le Transformer, mais avec une touche spécifique :

- **Encodeur bidirectionnel** : Comme BERT, l'encodeur de BART lit le texte dans les deux sens, ce qui lui permet de comprendre le contexte complet d'un mot, avant et après.
- **Décodeur autorégressif** : Comme GPT, le décodeur génère du texte mot par mot, en se basant sur ce qui a été écrit avant, ce qui assure une cohérence dans la production.

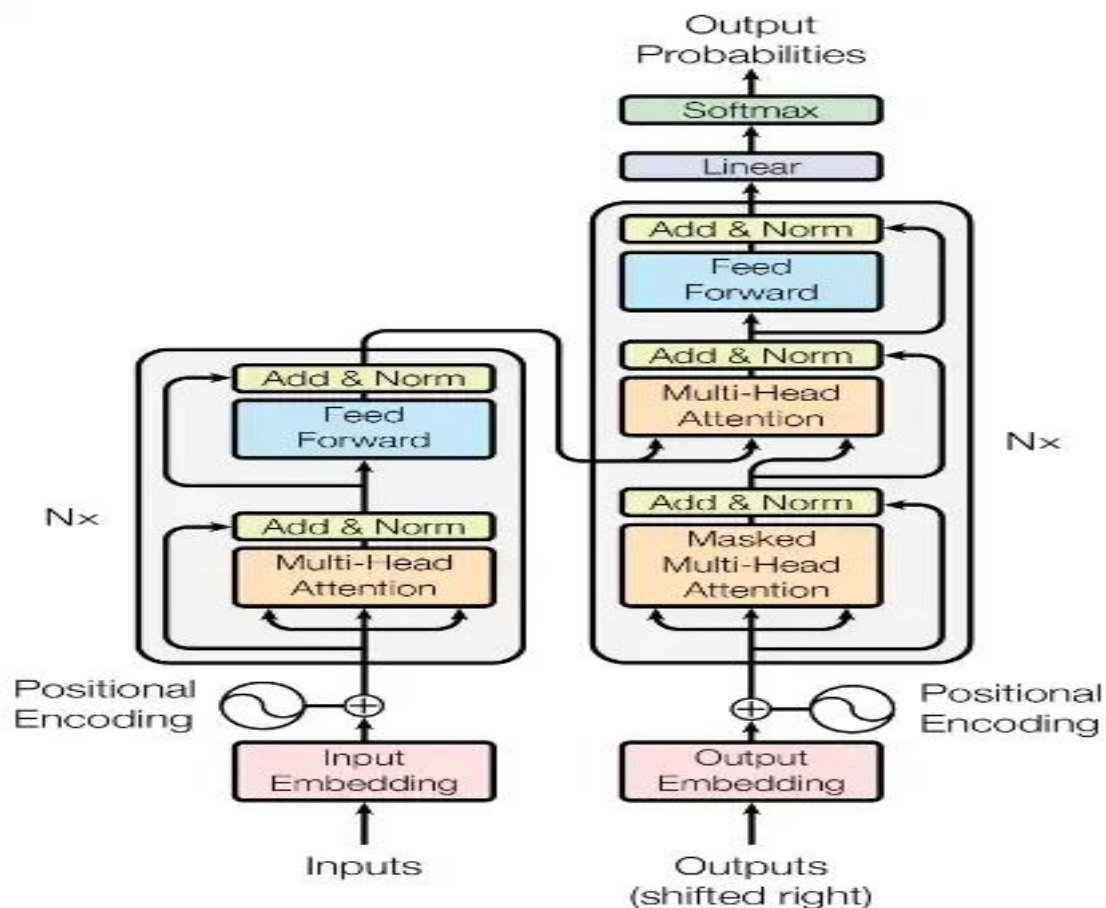


Figure 2.13 - Architecture du BART [23]

Les entrées de l'encodeur n'ont pas besoin d'être alignées avec les sorties du décodeur, ce qui permet des transformations bruitées arbitraires.

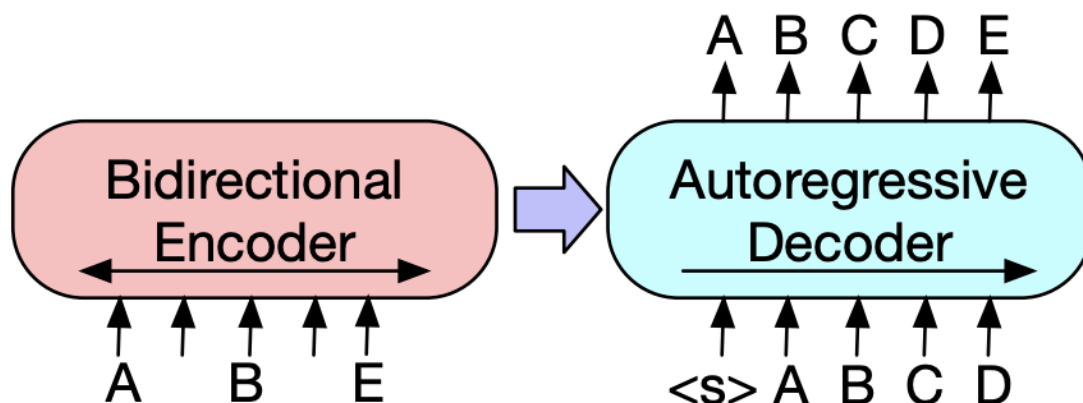


Figure 2.14 -Schématique de BART [22]

Encodage bidirectionnel (Bidirectional Encoder) – à gauche

- Le bloc rose représente l’encodeur de BART.
- Ce composant lit toute la séquence d’entrée dans les deux directions (gauche → droite et droite → gauche).
- Dans l’image :
 - Les lettres **A _ B _ E** montrent que certaines parties du texte ont été masquées ou corrompues (technique de “noising”).
 - L’encodeur essaie de comprendre le contexte global, même autour des trous (tokens masqués).

Décodage autoregressif (Autoregressive Decoder) – à droite

- Le bloc bleu est le **décodeur** de BART.
- Il fonctionne **de manière autoregressive** :
 - Il génère le texte mot par mot, de gauche à droite.
 - Chaque mot généré est utilisé comme entrée pour prédire le mot suivant.
- Sur le schéma :
 - L’entrée du décodeur est la séquence <s> A B C D.
 - Le modèle prédit ensuite **A, B, C, D, E**, un à un.

Fonctionnement

Le fonctionnement de BART se divise en deux étapes clés : le pré-entraînement et le fine-tuning.

Pré-entraînement

Pendant le pré-entraînement, BART est exposé à des textes corrompus de différentes manières :

- **Masquage de mots** : Certains mots sont cachés, et BART doit les deviner.
- **Permutation de phrases** : Les phrases sont mélangées, et BART doit les remettre dans l'ordre.
- **Remplacement de tokens** : Des mots sont remplacés par d'autres, et BART doit restaurer le texte original.

Ces techniques forcent le modèle à comprendre non seulement les mots, mais aussi la structure et le sens global du texte. C'est comme si on lui apprenait à réparer des puzzles textuels, ce qui le rend très doué pour reconstruire ou générer du texte cohérent.

Fine-tuning

Une fois pré-entraîné, BART peut être affiné (fine-tuned) pour des tâches spécifiques. Par exemple, pour faire du résumé de texte, on lui donne des articles et leurs résumés, et il ajuste ses paramètres pour exceller dans cette tâche. Cette flexibilité est l'un de ses grands atouts : un seul modèle peut être adapté à une multitude d'applications avec relativement peu de données supplémentaires.

5.3 Bidirectional Encoder Representations from Transformers (BERT)

Modèle de langage introduit en octobre 2018 par des chercheurs de Google. Il apprend à représenter le texte sous forme d'une séquence de vecteurs en utilisant un apprentissage auto-supervisé. Il repose sur une architecture de transformeur à encodeur uniquement. BERT a considérablement amélioré l'état de l'art pour les grands modèles de langage. À partir de 2020, BERT est devenu une référence universelle dans les expériences de traitement automatique du langage naturel (TALN) [24].

Architecture

Sous le capot, BERT utilise l'encodeur du Transformer, une architecture qui a fait ses preuves pour traiter les séquences de texte. Concrètement, ça veut dire qu'il est bourré de couches d'*attention multi-têtes*, qui lui permettent de se concentrer sur différentes parties de la phrase en même temps. C'est un peu comme avoir plusieurs paires d'yeux qui scrutent le texte sous tous les angles. BERT existe en deux tailles principales : BERT-base, avec 110 millions de paramètres, et BERT-large, qui en a 340 millions. Plus il y a de paramètres, plus le modèle peut capter de subtilités, mais ça demande aussi plus de puissance de calcul. Chaque couche du modèle affine la représentation des mots, en tenant compte de leur contexte dans la phrase. Et grâce à l'attention, BERT peut même repérer des liens entre des mots très éloignés, ce qui est crucial pour les phrases complexes.

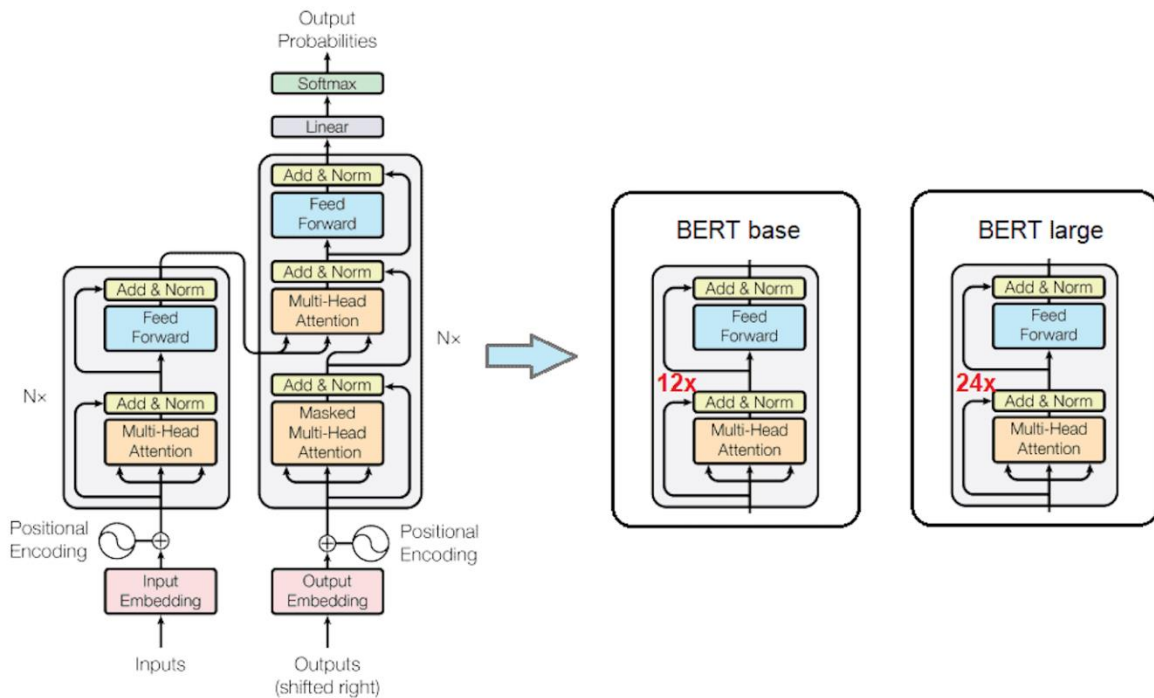


Figure 2.15 - Architecture de BERT

Fonctionnement

Le fonctionnement de BERT se divise en deux grandes étapes : le pré-entraînement et le fine-tuning.

Pré-entraînement

Pendant le pré-entraînement, BERT apprend sur des tonnes de texte brut, comme des articles de Wikipédia ou des livres. Mais il ne se contente pas de lire bêtement. Il s'entraîne sur deux tâches clés :

- **Modélisation de langage masquée (Masked Language Modeling MLM)** : On cache certains mots dans une phrase, et BERT doit deviner ce qu'ils sont en se basant sur le contexte. Par exemple, dans "Le chat [MASQUE] sur le tapis", il doit trouver que le mot caché est probablement "dort" ou "joue". Ça lui apprend à comprendre les relations entre les mots.
- **Prédiction de la phrase suivante (Next Sentence Prediction NSP)** : BERT doit aussi deviner si deux phrases se suivent logiquement. Ça lui permet de saisir les liens entre les idées, pas juste entre les mots.

Ces deux tâches font que BERT développe une compréhension profonde du langage, sans avoir besoin de données annotées pour des tâches spécifiques.

Fine-tuning

Une fois pré-entraîné, BERT peut être affiné (fine-tuned) pour des tâches précises, comme répondre à des questions ou classifier des textes. Pour ça, on lui donne un petit jeu de données spécifiques à la tâche, et on ajuste ses paramètres. C'est un peu comme prendre un expert généraliste et lui apprendre un métier précis. Cette flexibilité est l'un des grands atouts de BERT : un seul modèle peut servir à plein de choses différentes.

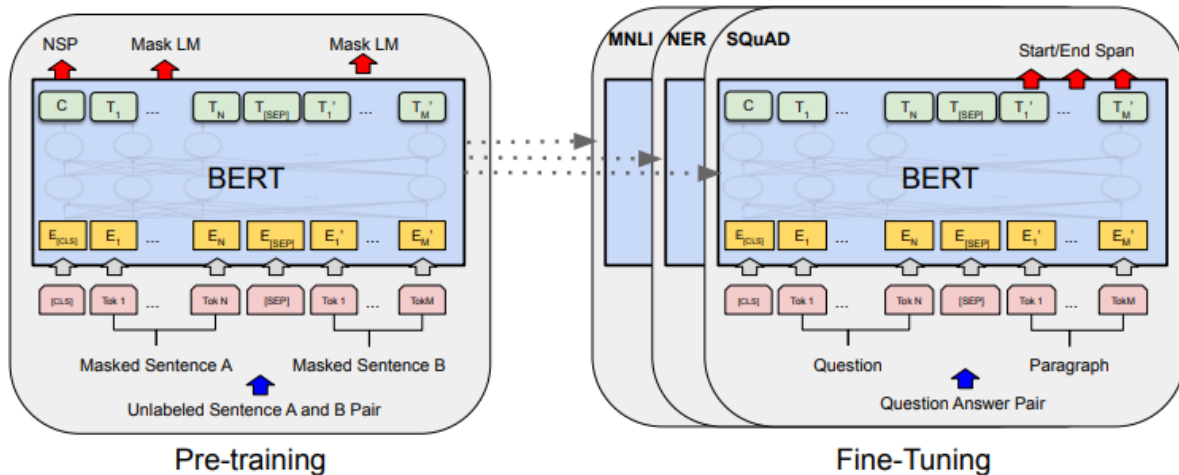


Figure 2.16 - Pré-entraînement et Ajustement fin de BERT

5.4 Large Language Model Meta AI (LLAMA)

LLaMA est une famille de grands modèles de langage (LLM) publiés par Meta AI à partir de février 2023. La dernière version est Llama 4, publiée en avril 2025. Ces modèles sont élaborés pour comprendre et générer du langage naturel, favorisant ainsi une interaction plus fluide entre les humains et les machines [25].

Les versions les plus connues incluent :

- **LLAMA 1 (2023)** : première version lancée en open-source.
- **LLAMA 2 (juillet 2023)** : version améliorée avec plus de données et de robustesse.
- **LLAMA 3 (avril 2024)** : modèle de troisième génération, incluant des variantes comme LLaMA 3-8B, 70B, et des modèles instruct comme "Scout" et "Versatile".

LLaMA Model Evolution Timeline



LLaMA 1 (Feb 2023)

- Meta's first large language model suite.
- Released for research use only (with access approval).
- Variants: 7B, 13, 33B



LLaMA 2 (Jul 2023)

- Openly released for both research and commercial use.
- Variants: 7B, 13B, 70B
- Pretrained and instruction/chat-tuned versions included
- Widely adopted by open-source community and integrated into many platforms



Code LLaMA (Aug 2023)

- Code-specialized version of LLaMA 2
- Variants: 7B, 13B, 34B
- Trained on a curated code dataset from GitHub
- Included both base and instruction-tuned models



LLaMA 3 (April 2024)

- Meta's most powerful and capable model to date.
- Built with a Mixture of Experts (MoE) architecture.
- Focused on multimodal capabilities (text + image).
- High-performance on reasoning, agentic tasks, and embedded assistant use.



Figure 2.17 - Evolution de LLaMA

Architecture

L'architecture de LLaMA suit globalement la structure des Transformers autorégressifs introduits par Vaswani et al. (2017), avec certaines optimisations :

Couche d'entrée

- **Embeddings** : Les mots ou tokens d'entrée sont convertis en vecteurs numériques (embeddings) qui capturent leur signification.
- **RMS Norm** : Une normalisation de la racine moyenne des carrés est appliquée aux embeddings pour stabiliser l'entraînement et normaliser les activations.

Blocs répétés (N x)

Chaque couche répétée comprend plusieurs sous-composants :

- **Self-Attention (Grouped-Query Multi-Query Attention with KV Cache) :**
 - Utilise une attention multi-tête avec une optimisation appelée **Grouped-Query Multi-Query Attention**, qui regroupe les requêtes pour améliorer l'efficacité.
 - Inclut un **KV Cache** (cache des clés et valeurs) pour accélérer les inférences en réutilisant les calculs précédents.
 - Les entrées sont divisées en **Q** (requêtes), **K** (clés), et **V** (valeurs), qui interagissent pour calculer les poids d'attention.
 - Cette attention est enrichie par un **Rotary Positional Encoding** (encodage positionnel rotatif), qui ajoute des informations de position aux embeddings sous forme de rotations, améliorant la compréhension de l'ordre des mots par rapport aux encodages positionnels classiques.
- **Add** : Une connexion résiduelle ajoute les résultats de l'attention à l'entrée précédente pour faciliter la propagation des gradients.
- **RMS Norm** : Une autre normalisation RMS est appliquée après l'attention pour stabiliser les sorties.
- **Feed Forward (SwishLU)** : Une couche d'alimentation directe avec une fonction d'activation **SwishLU** (une variante de Swish combinée avec une unité linéaire) introduit une non-linéarité pour capturer des relations complexes dans les données.
- **Add** : Une autre connexion résiduelle combine les résultats du feed-forward avec l'entrée précédente.

Couche de sortie

- **RMS Norm** : Une normalisation finale ajuste les activations avant la couche de sortie.
- **Linear** : Une couche linéaire transforme les vecteurs en logits (scores bruts pour chaque token possible).
- **Softmax** : Convertit les logits en **Output Probabilities**, donnant la probabilité de chaque token dans la séquence générée.

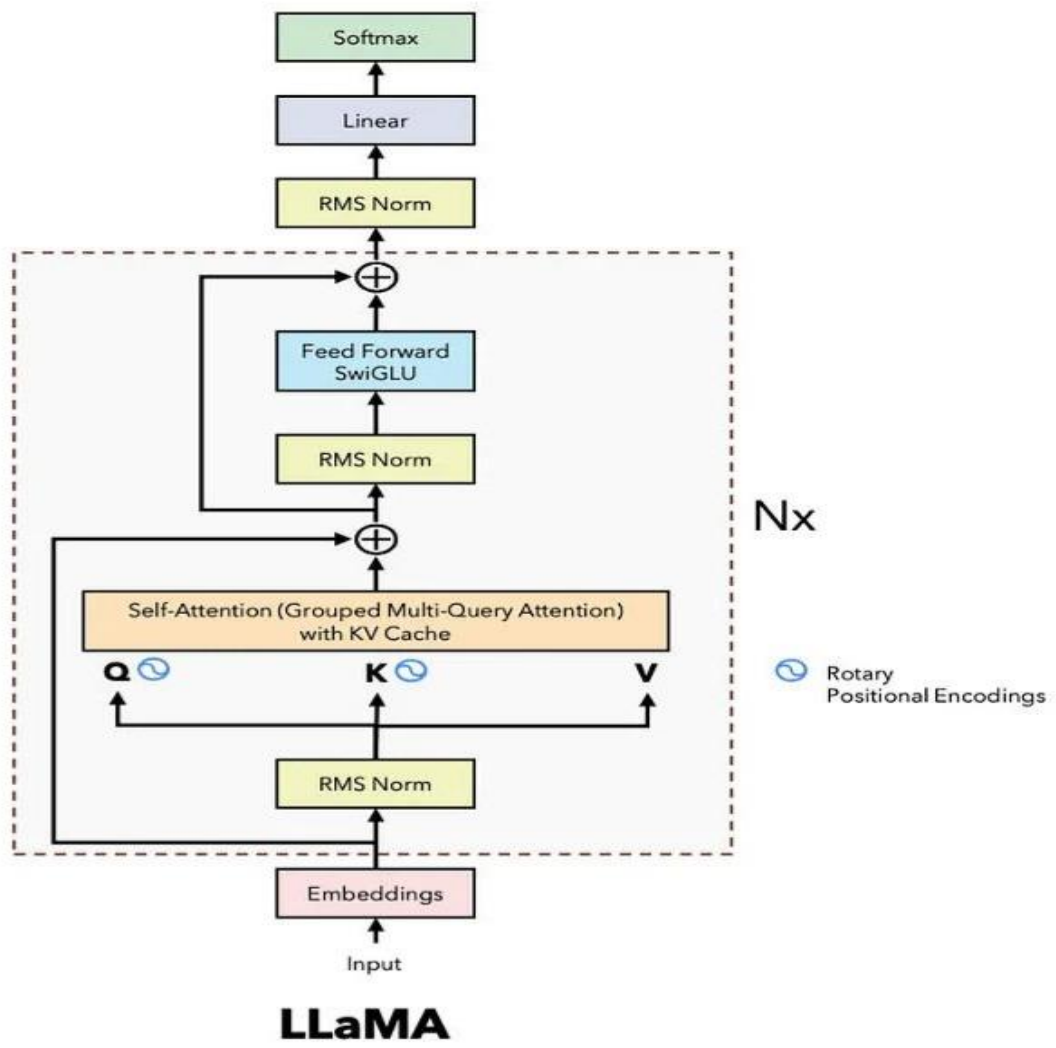


Figure 2.18 - Architecture du LLaMA

6. Conclusion

Dans ce chapitre, nous avons présenté une introduction de traitement automatique du langage naturelle (TALN), à travers l'étude des principales méthodes de la traduction automatique à la reconnaissance vocale, en passant par la classification de textes et le résumé automatique nous avons mis en évidence les architectures, les fonctionnements de chaque méthode.

Le cœur de ce chapitre, c'est sans doute la plongée dans le résumé automatique, avec ses deux grandes approches : l'extractif, qui pioche les idées clés sans les retoucher, et l'abstrait, qui recrée un texte plus fluide et naturel.

Et puis, il y a les grands modèles de langage (LLMs) comme BERT, GPT, BART et LLaMA, qui, avec leur architecture sophistiquée et leur entraînement sur des montagnes de données, apportent une touche de magie en comprenant et générant du langage presque comme un humain.

Chapitre 3 expérimentation et la conception

1. Introduction

Après avoir exploré les fondements théoriques du Traitement Automatique du Langage Naturel (TALN) et les avancées des grands modèles de langage (LLMs) dans le chapitre précédent, il est temps de passer de la théorie à la pratique.

Ce chapitre vise à détailler le processus de conception de la plateforme **EduBrief**, en mettant l'accent sur les choix techniques, les méthodologies d'expérimentation et les étapes de développement qui permettront de transformer les concepts en une solution fonctionnelle. Ainsi que l'environnement de développement de ce projet.

2. Environnement et outils de développement

2.1 Environnement matériel

Pour le développement de notre application et les modèles que nous avons traité dans ce chapitre, nous avons utilisé un ordinateur portable de marque **LENOVO**, ayant les caractéristiques techniques suivantes :

- **Nom de l'appareil** : LAPTOP-Q4O5JA99
- **Processeur**: AMD Ryzen 7 PRO 5850U with Radeon Graphics 1.90 GHz
- **Mémoire RAM installée** : 16,0 Go (14,8 Go utilisable)
- **Architecture** : Système d'exploitation 64 bits, processeur x64
- **Ecran** : Non tactile

2.2 Environnement logiciel

➤ Visual Studio Code (VS Code)

VS Code est un éditeur de code libre, qui fonctionne sur les systèmes d'exploitation macOS, Linux et Windows. Il prend en charge une vaste gamme de langages de programmation (comme Python, JavaScript, C++, etc.). VS Code combine une interface utilisateur avec des fonctionnalités avancées telles que l'auto-complétion intelligente, le débogage intégré, le contrôle de version (Git) [26].



Figure 3.1 - Logo de vs code

➤ **WAMP**

WampServer est une plate-forme de développement Web sous Windows pour des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement vos bases de données [27].



Figure 3.2 - Logo de WAMP server

➤ **MariaDB**

MariaDB est un système de gestion de bases de données relationnelles (SGBDR) open-source, il est fabriqué par les développeurs originaux de MySQL et garanti pour rester open source. Il est distribué sous la licence GPL et est largement adopté pour des applications web, notamment dans des piles comme WAMP ou LAMP [28].



Figure 3.3 - Logo de MariaDB

➤ **Modelio**

Modelio est un outil de modélisation open-source développé par Modeliosoft, c'est une solution de modélisation offrant une large gamme de fonctionnalités basées sur des normes couramment utilisées pour l'architecture d'entreprise, la modélisation de processus, le développement de logiciels et l'ingénierie des systèmes. Le soutien à la

modélisation et aux processus d'entreprise (BPMN¹) est intégré dans Modelio. Ces modèles BPMN peuvent être reliés à d'autres normes soutenues par l'outil, par exemple UML, afin d'intégrer ces processus métier dans un contexte plus large tel qu'une architecture d'entreprise [29].

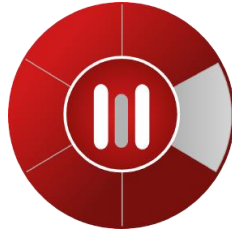


Figure 3.4 - Logo de modelio

2.3 bibliothèques utilisées

Dans cette section, nous avons présenté les bibliothèques utilisées dans les modèles et l'évaluation.

- **PdfPlumber** : Utilisée pour extraire le texte des fichiers PDF de manière structurée (par exemple, texte par page).
- **Fitz** : Alternative à pdfplumber pour extraire le texte ou manipuler des PDF, souvent utilisée pour des rendus ou annotations.
- **Nltk**: Fournit des outils pour le traitement du langage naturel, comme la tokenisation par phrases (sent_tokenize) ou la lemmatisation.
- **Re (regular Expressions)**: Bibliothèque standard Python pour les expressions régulières, utile pour nettoyer ou extraire des motifs dans le texte.
- **Networkx**: Création et analyse de graphes, souvent pour des algorithmes comme TextRank.
- **sklearn.feature_extraction.text.TfidfVectorizer**: Conversion de texte en vecteurs TF-IDF pour mesurer l'importance des mots.

¹ **The Business Process Modeling Notation (BPMN)** : est une norme pour la modélisation des processus d'entreprise qui fournit une notation graphique pour spécifier les processus métier dans un diagramme de processus d'entreprise.

- **sklearn.metrics.pairwise.cosine_similarity**: Calcule la similarité cosinus entre des vecteurs (par exemple, entre phrases représentées par TF-IDF), souvent utilisée pour déterminer l'importance relative dans un résumé.
- **Transformers** : Fournit les modèles comme BART (BartTokenizer, BartForConditionalGeneration) et T5 (T5Tokenizer, T5ForConditionalGeneration), ainsi que Seq2SeqTrainer et TrainingArguments pour l'entraînement.
- **Accelerate**: Accélère l'entraînement en optimisant l'utilisation des GPU/CPU, compatible avec Seq2SeqTrainer.
- **Pipeline** : Interface simplifiée pour utiliser des modèles pré-entraînés (par exemple, pour le résumé).
- **AutoTokenizer, AutoModel**: Classes génériques pour charger dynamiquement des tokenizers et modèles.
- **Sumy**: Bibliothèque de résumé automatique basée sur des algorithmes comme TextRank (TextRankSummarizer), utile pour des approches extractives.
- **Streamlit**: Permet de créer des applications web interactives en Python, idéale pour une interface utilisateur où tu pourrais uploader des PDF et afficher des résumés générés.
- **Os** : Bibliothèque standard Python pour interagir avec le système d'exploitation (gestion des fichiers, chemins, etc.), probablement pour lire ou écrire des fichiers PDF/textes.
- **Fpdf** : Permet de générer des fichiers PDF à partir de texte ou d'images, utile pour sauvegarder les résumés générés.

- **Groq** : Client Python pour l'API Groq, qui fournit un accès à des modèles d'IA (probablement pour générer des résumés ou répondre à des questions sur le texte extrait). Groq est une plateforme d'IA concurrente à xAI, mais compatible avec des workflows similaires à ceux de Hugging Face.
- **from bert_score import score** : Évaluer la qualité sémantique d'un texte généré (comme un résumé) en le comparant à un texte de référence à l'aide de modèles BERT.
- **import matplotlib.pyplot as plt** : Créer des graphiques courbes, histogrammes, barres, etc.
- **import pandas as pd** : Gérer des tableaux de données (DataFrames), un peu comme un tableau Excel en Python.
- **import textstat** : Mesurer la lisibilité d'un texte avec différents indices.

2.4 Langage de programmation

Dans cette section, nous présentons les langages de programmation utilisés pour le développement de notre modèle d'intelligence artificielle, ainsi que les langages utilisés par la plateforme MOODLE.

➤ Python

C'est un langage de programmation open source et orienté objet, puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet. Python se distingue par sa polyvalence. Python est un langage idéal pour l'écriture de scripts et le développement rapide d'applications dans de nombreux domaines et sur la plupart des plateformes [30].



Figure 3.5 - Logo de python

➤ PHP

Est un langage de scripts généraliste et Open Source, spécialement conçu pour le développement d'applications web, conçu pour être intégré facilement au HTML. Il est exécuté côté serveur, permettant de générer dynamiquement du contenu web en fonction de données, sa flexibilité et sa capacité à être embarqué dans des pages HTML en font un outil puissant pour créer des sites interactifs et des applications web [31].



Figure 3.6 - Logo de php

➤ HTML

Le langage HTML (Hyper Text Markup Language) est utilisé afin de mettre en œuvre des pages web contenant des liens et des éléments graphiques. Ces pages sont conçues pour être affichées dans des applications appelées par un navigateur web [32].



Figure 3.7 - Logo de HTML

➤ CSS

Le langage CSS (Cascading Style Sheets) adresse les problématiques d’affichage des pages HTML. Son objectif est de permettre la séparation claire entre la structure d’une page, par le biais du langage HTML, et sa présentation avec CSS. Fondé sur les applications en cascade de styles sur les éléments contenus dans les pages HTML, il permet d’appliquer des styles en fonction de la position des balises [32].



Figure 3.8 - Logo de CSS

2.5 Ensemble de données

Dans cette section nous présenterons l’ensemble de données que nous avons utilisés dans le modèle, l’ensemble des PDF qui ont des documents et le corpus que nous avons déjà préparés.

- **Dataset_text**

Cet ensemble de données est composé d’un ensemble de textes et ses résumés, il est de 100 textes et leurs résumés, Les textes couvrent une variété de sujets, principalement dans les domaines de la chimie, de la physique, des mathématiques, et de l’informatique (par exemple, intelligence artificielle, systèmes distribués, technologies web), inclus des thèmes comme chimie des matériaux, réactions redox, géométrie euclidienne, réseaux de neurones profonds, et architectures web modernes. Ils sont écrits dans un style éducatif ou scientifique, adaptés à un public d’étudiants. Les résumés sont cohérents avec les textes sources.

	text	resume
0	La chimie des matériaux est une branche de la...	La chimie des matériaux explore la relation en...
1	L'eau est une molécule simple composée de deux...	L'eau est un solvant universel essentiel pour ...
2	Les acides et bases sont des concepts centraux...	Les acides et bases, définis par différentes t...
3	La catalyse est un processus chimique qui perm...	La catalyse accélère les réactions chimiques e...
4	La chimie organique est l'étude des composés c...	La chimie organique traite des composés carbon...
5	Les réactions d'oxydo-réduction, ou réactions ...	Les réactions redox sont des processus chimiqu...
6	Les polymères sont de grandes molécules consti...	Les polymères sont des molécules essentielles ...
7	La chimie des solutions s'intéresse aux intera...	La chimie des solutions étudie l'interaction e...
8	L'azote, composant principal de l'atmosphère, ...	L'azote est vital pour l'agriculture et l'indu...
9	La chimie des gaz concerne les propriétés et c...	La chimie des gaz analyse leurs propriétés et ...
10	Les acides nucléiques, comme l'ADN et l'ARN, s...	Les acides nucléiques, responsables de l'infor...
11	La catalyse hétérogène est un processus où le ...	La catalyse hétérogène, utilisée dans l'indust...
12	Les polymères biodégradables sont des matériau...	Les polymères biodégradables offrent une alter...
13	Les éléments rares, ou terres rares, sont un g...	Les éléments rares sont essentiels pour les te...

Figure 3.9 - Ensemble de données des textes

- **Dataset CNN/DailyMail**

Est un ensemble de données en anglais contenant 311 971 articles de presse uniques, rédigés par des journalistes des sites CNN et Daily Mail. Initialement conçu en 2015 par des chercheurs notamment Hermann et al, pour des tâches de lecture automatique et de compréhension (machine reading comprehension) ainsi que de réponse à des questions abstraites (abstractive question answering), il a évolué pour supporter également la génération de résumés (summarization).

[61]:	id	article	highlights
0	0001d1afc246a7964130f43ae940af6bc6c57f01	By . Associated Press . PUBLISHED: . 14:11 EST...	Bishop John Folda, of North Dakota, is taking ...
1	0002095e55fcb3a2f366d9bf92a95433dc305ef	(CNN) -- Ralph Mata was an internal affairs li...	Criminal complaint: Cop used his role to help ...
2	00027e965c8264c35cc1bc55556db388da82b07f	A drunk driver who killed a young woman in a h...	Craig Eccleston-Todd, 27, had drunk at least t...
3	0002c17436637c4fe1837c935c04de47adb18e9a	(CNN) -- With a breezy sweep of his pen Presid...	Nina dos Santos says Europe must be ready to a...
4	0003ad6ef0c37534f80b55b4235108024b407f0b	Fleetwood are the only team still to have a 10...	Fleetwood top of League One after 2-0 win at S...
5	0004306354494f090ee2d7bc5ddb80b63e80de6	He's been accused of making many a fashion fau...	Prime Minister and his family are enjoying an ...
6	0005d61497d21ff37a17751829bd7e3b6e4a7c5c	By . Daily Mail Reporter . PUBLISHED: . 01:15 ...	NBA star calls for black and Hispanic communit...
7	0006021f772fad0aa78a977ce4a31b3faa6e6fe5	By . Daily Mail Reporter . This is the moment ...	London Midland service had been pulling into T...
8	00083697263e215e5e7eda753070f08aa374dd45	There are a number of job descriptions waiting...	Tony Pulis believes Saido Berahino should look...
9	000940f2bb357ac04a236a232156d8b9b18d1667	Canberra, Australia (CNN) -- At first glance, ...	Black box data from Flight 370 could be analyz...
10	0009ebb1967511741629926ef9f5faea2bb6be24	By . Ellie Zolfagharifard . Take a look at a m...	The distortion is the result of the Mercator m...
11	000c83555db62e319854d9f8912061cdca1893e	Two lawyers representing a woman who . claims ...	Alan Dershowitz has filed defamation suits aga...
12	000ca3fc9d877f8d4bb2ebd1d6858c69be571fd8	It's the moment every pet owner dreads - when ...	Sarah Ernhart, the owner of Sarah Beth Photogr...
13	000cd1ee0098c4d510a03ddc97d11764448ebac2	Louis van Gaal said he had no option but to su...	Manchester United beat Southampton 2-1 at St M...
14	000e009f6b1d954d827c9a550f3f24a5474ee82b	(CNN) -- One can hardly read the news these da...	U.S. corporations merge with foreign companies...

Figure 3.10 - Les 15 premières lignes de l'ensemble de données CNN/DailyMail

- **Dossier PDF**

Ce dossier contient des fichiers PDF provenant de notre université plus précisément, ce sont des documents des cours de master 1 spécialité SIC. Nous avons utilisé ces documents pour tester les performances de modèle avant et après le **fine-tuning**.

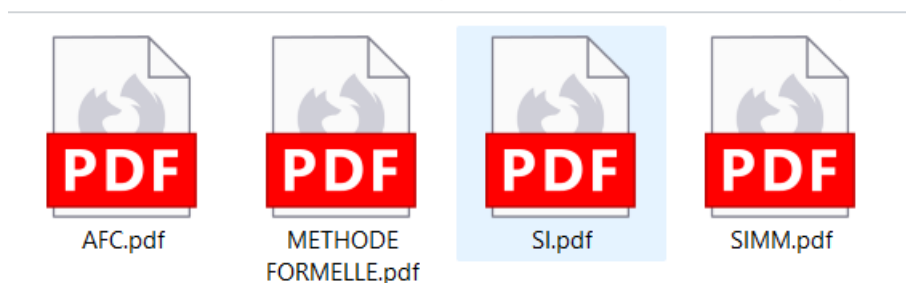


Figure 3.11 - Ensemble de PDF

De l'ACP à l'AFC

- ACP = étude des liaisons contenues dans un tableau individus *variables, lorsque toutes les variables sont quantitatives .
- AFC (analyse factorielle des correspondances) et l'ACM (analyse des Correspondances multiples) = étude des liaisons continues dans un tableau individus *variables, lorsque toutes les variables sont qualitatives.
- L'AFC est l'étude des correspondances entre les modalités de deux variables qualitatives .
- L'ACM est une généralisation de l'AFC à plus de deux variables qualitatives.

2

Analyse Factorielle des correspondance (AFC)

- L'Analyse Factorielle des Correspondances (A.F.C.) est une méthode factorielle de Statistique Descriptive Multidimensionnelle.
- Son objectif est d'analyser la liaison existant entre deux variables qualitatives (si on dispose de plus de deux variables qualitatives, on aura recours à l'Analyse des Correspondances Multiples). Ainsi, avant de mettre en œuvre une A.F.C., il faut s'assurer que cette liaison existe bien.

1. Systèmes d'informations coopératifs

1.1. Définition

1.1.1. Collaboration et coopération

Collaboration c'est le fait de participer à un travail réalisé en commun par une ou plusieurs personnes, coopération c'est soutenir réciproquement la réalisation d'un but commun.

1.1.2. Systèmes d'informations coopératifs

C'est un système d'information composé à son tour d'un ensemble de systèmes d'informations autonomes disposant chacun d'une capacité de coexistence dans un environnement commun pour accomplir une tâche afin d'atteindre le but du système d'informations coopératif (SIC). Chaque système d'information possède sa propre identité et il est indépendant par rapport à la tâche qu'il réalise. Cette dernière est en but commun avec les autres systèmes d'informations dont l'accomplissement nécessite plus de ressources que l'en possède chaque système séparément. Ou encore que l'accomplissement de ce but ne peut se faire efficacement qu'avec la coopération de plusieurs systèmes.

1.1.3. Exemples de systèmes coopératifs

Les premiers systèmes coopératifs existants et qui sont à l'origine des systèmes multi agents actuels, sont les communautés des fourmis et des abeilles. Il existe d'autres systèmes coopératifs tels que l'univers, le système solaire et le corps humain. Mais leur fonctionnement est plus complexe à imiter.

1.2. Coopération entre système d'information

C'est la possibilité d'échanger l'information et de répartir les tâches (pour l'accomplissement d'un but commun) de manière à rendre une fonctionnalité d'un système d'information disponible aux autres systèmes d'informations.

1.3. Paradigmes des systèmes d'informations coopératifs (SICo)

Il repose principalement sur trois éléments :

- Systèmes répartis et distribués (déjà défini.)
- Echange (gestion et accès) de l'information et de l'expertise pour garantir l'interopérabilité du système d'informations coopératif.
- Coordination des activités (planification et ordonnancement) et parfois même la négociation pour la résolution d'un problème ou l'accomplissement d'une tâche avec un but commun.

1.4. Les avantages des systèmes d'informations coopératifs

- Performances améliorées et réponses relatives au coût.

Figure 3.12 - PDFs, AFC.pdf(a gauche) SI.pdf(a droite)

3. Flux du travail propose

3.1 Schéma de résumé un document s

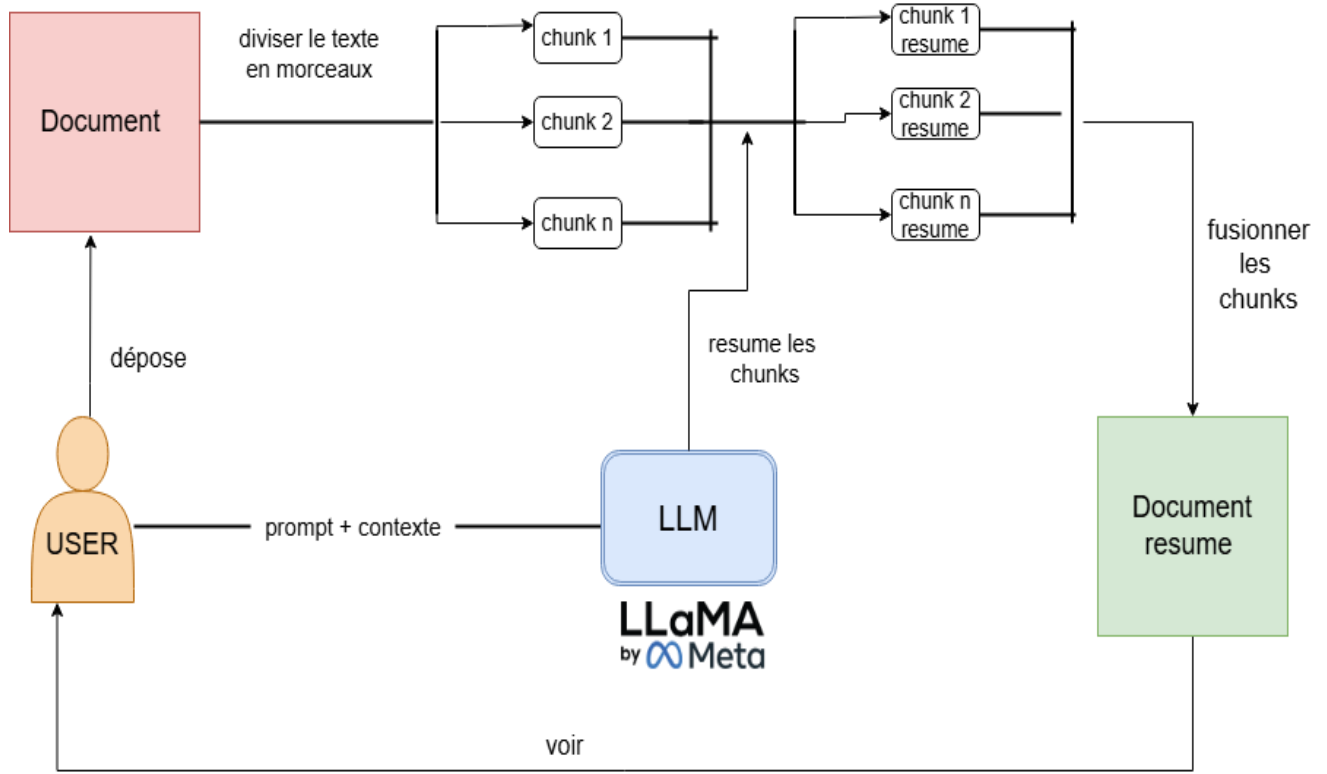


Figure 3.13 - Schéma partie de résumé un document pdf

La figure 3.13 présente le processus de résumé automatique d'un document Pdf, le processus commence par le dépôt du document PDF par l'utilisateur. Le document en entré est divisé en plusieurs segment (**chunk1, chunk2, ... , chunk n**).

Chaque chunk est transmis au modèle **LLAMA (Llama inclus la tokenization)** avec un prompt et un contexte. Le modele LLM généré un résumé individuel pour chaque chunk (**résumé de chunk1, résumé de chunk2 , ..., résumé chunk n**). Les résumés individuels des chunks sont combines pour produire un **résumé final** de document initial.

En conclusion ce schéma montre un processus de summarization par segmentation. Le document est découpé en segments gérables (chunks) pour respecter la limite de contexte du LLM. Chaque segment est résumé indépendamment, puis les résumés sont fusionnés pour créer un résumé cohérent du document entier. Cela est utile pour traiter des documents longs, comme dans votre plateforme éducative, où un manuel ou un article est résumé en étapes.

3.2 Schéma de génération des questions

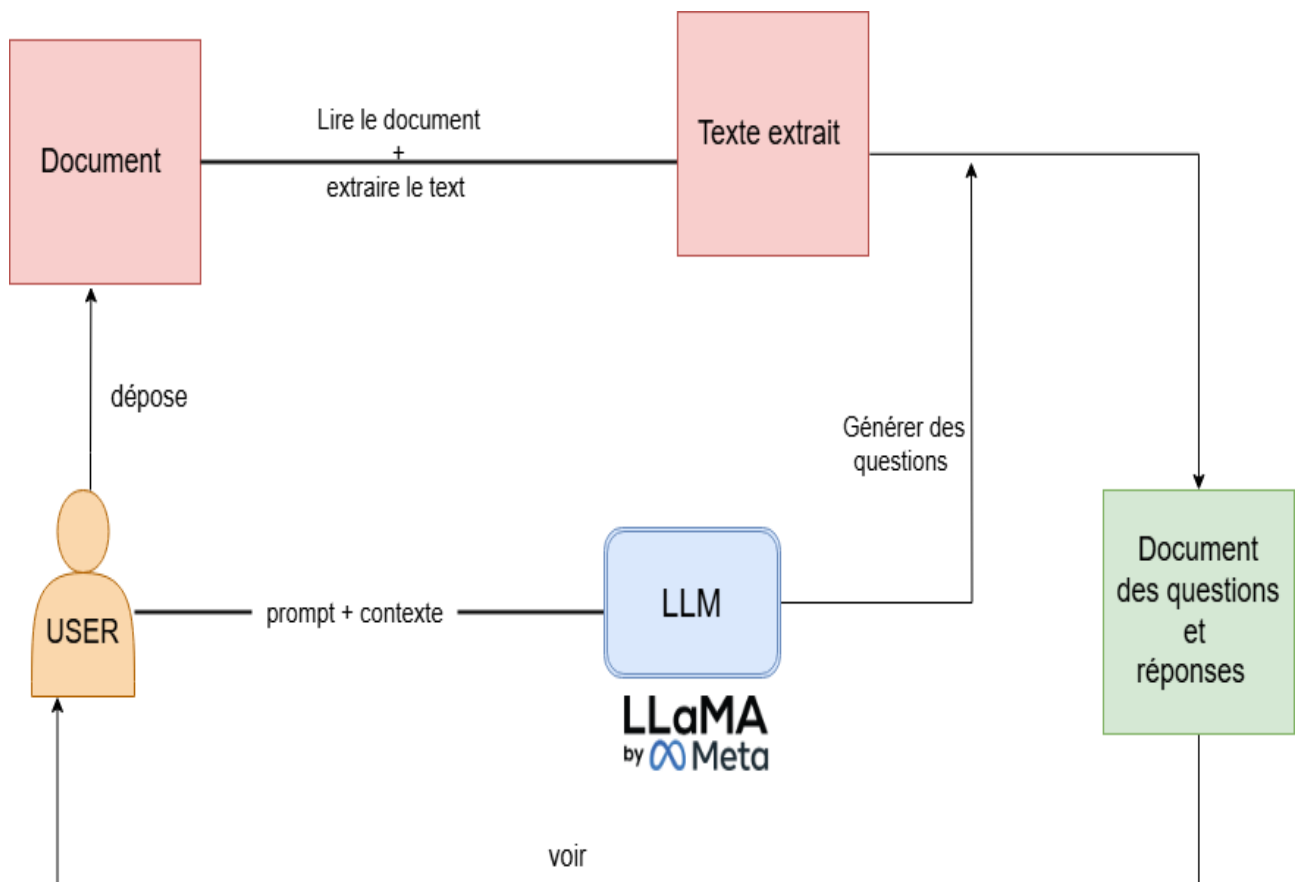


Figure 3.14 - Schéma partie de génération des questions

La figure 3.14 présente le processus de génération de questions - réponses d'un document PDF via prompt adapté pour le **modèle Llama**.

Premièrement, L'utilisateur dépose le document PDF, le texte qui est déjà extrait, puis transmis ce texte extrait au modèle **LLM (Llama)** avec un prompt, pour but de générer des **questions** avec leurs **réponses**. En sortie, Le système produit un **document PDF** qui contient les **questions générées** ainsi que leurs **réponses potentielles**.

En conclut, ce schéma illustre un processus de génération de questions et réponses (**Q&A**). Après avoir extrait le texte du document, le LLM est utilisé pour la création des questions basées sur ce contenu, avec des réponses associées.

3.3 Schéma de chatbot

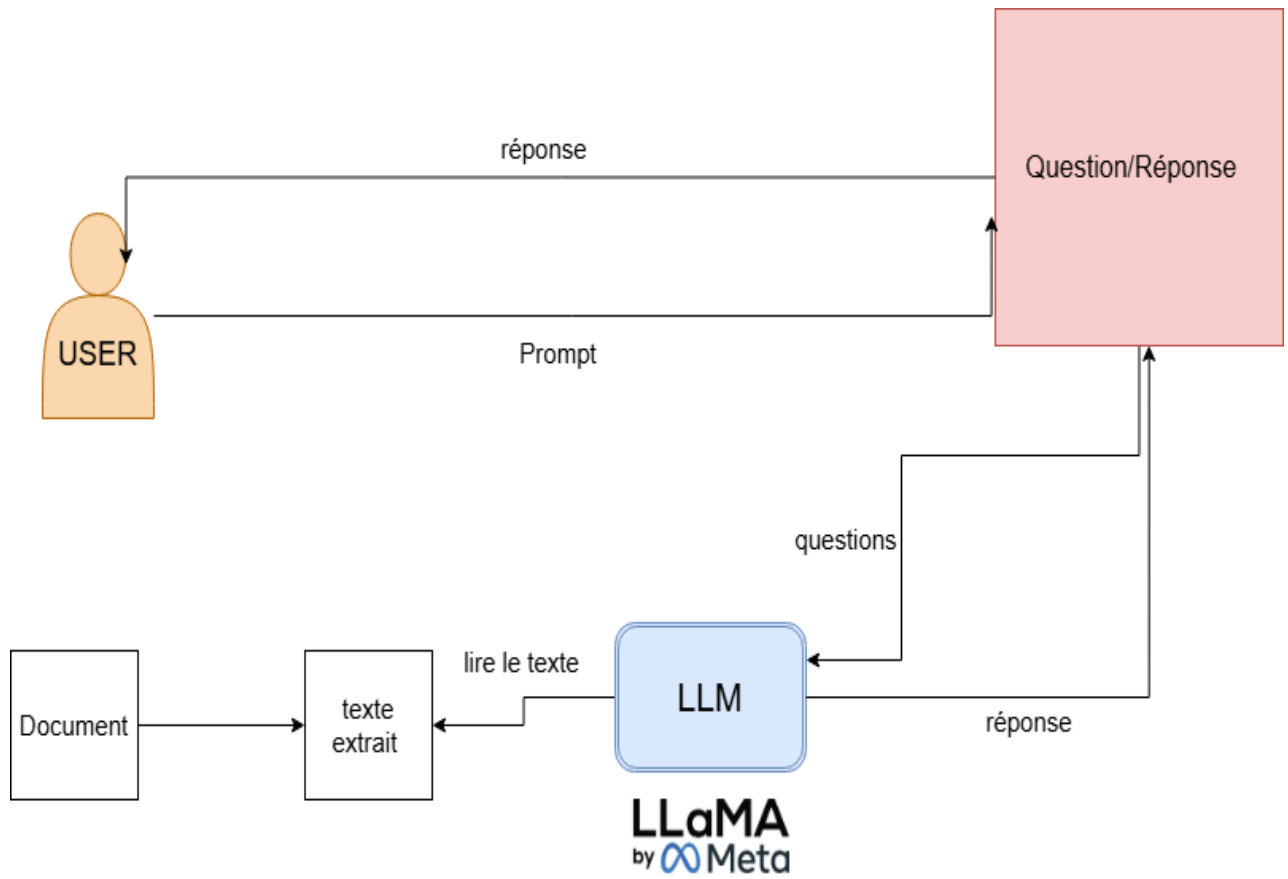


Figure 3.15 - Schéma partie de chatbot

La figure 3.15 présente le processus **question – réponse interactive via LLMA**.

Une fois texte extrait du document PDF, celui est envoyé immédiatement au LLM (modèle **Llama**). L'utilisateur fournit une question qui est en lien avec le contenu du document.

Le modèle utilise son processus (indiquer dans la section Large Language Model Meta AI (LLAMA) page 30, 31, 32, 33). La réponse est retournée à l'utilisateur.

En conclusion ce schéma représente un système de chatbot ou Q&A interactif. Contrairement au schéma précédent, où les questions sont générées automatiquement, ici l'utilisateur pose une question spécifique. Le LLM utilise le texte extrait du document pour fournir une réponse contextuelle.

4. LLAMA

Parmi les étapes du projet, Nous avons choisir le modèle LLaMA (LLM) qui sera le noyau de ce système. Le modèle que nous avons choisi est : **“meta-llama/llama-4-scout-17b-16e-instruct”**,

ce modèle est pré-entraîné + fine-tuné sur des instructions générales comme résumé un texte, traduire, répondre à des questions ..., via un prompt comme **“Résume ce texte de manière claire et concise”**, notre système est de réalisé des résumés clairs pour la compréhension des apprenants, donc pour cette raison nous avons choisi ce modèle [33].

Partie 1

Dans cette partie, nous présentons les étapes de l’intégration du model Llama qui base sur PDF.

Etape 1 : commençons avec le modèle qui est connecté via un API de GROQ

```
GROQ_API_KEY = "gsk_5nEbGrewf [REDACTED]"  
MODEL_NAME_SUMMARY="meta-llama/llama-4-scout-17b-16e-instruct"
```

Figure 3.16 - Choix du modèle Llama via api groq

Etape 2 : la fonction **extract_text_chunks** sert à lire tout le texte de fichier PDF et extraire le texte de chaque page du PDF, découper ce texte en morceaux (chunks) de taille fixe, elle retourne une liste des morceaux et le texte complet.

```
def extract_text_chunks(pdf_path, chunk_size=CHUNK_SIZE):  
    doc = fitz.open(pdf_path)  
    full_text = "\n".join([page.get_text() for page in doc])  
    doc.close()  
    return [full_text[i:i+chunk_size] for i in range(0, len(full_text), chunk_size)], full_text
```

Figure 3.17 - Fonction d'extraction du texte du PDF

Etape 3 : La fonction **summarize_text** utilise un modèle LLM (comme LLaMA) via une API pour générer un résumé intelligent et informatif d’un document texte complet via une instruction.

```
51  
52  
53 def summarize_text(text, client):  
54     response = client.chat.completions.create(  
55         model=MODEL_NAME_SUMMARY,  
56         messages=[{"role": "system", "content": "Tu es un assistant qui résume de manière claire et riche."},  
57                 {"role": "user", "content": f"lire tous le document et fais un résumé court et riche d'informations :\n\n{text}"}]  
58     )  
59     return response.choices[0].message.content.strip()  
60
```

Figure 3.18- Fonction de résumé du texte extrait

Etape 4 : la fonction `generate_questions` utilise le même modèle de langage (LLaMA) via un client (Groq) pour générer automatiquement deux types de questions pédagogiques à partir d'un texte des question ouvertes avec réponses et des QCM.

```
71
72 def generate_questions(text, client):
73     prompt = (
74         "À partir de ce texte, génère :\n"
75         "- questions ouvertes avec réponses\n"
76         "- QCM (4 choix + bonne réponse indiquée)\n\n"
77         f"Texte :\n{text}"
78     )
79     response = client.chat.completions.create(
80         model=MODEL_NAME_SUMMARY,
81         messages=[{"role": "system", "content": "Tu es un assistant expert en pédagogie."},
82                   {"role": "user", "content": prompt}]
83     )
84     return response.choices[0].message.content.strip()
85
```

Figure 3.19 - Fonction de génération des questions et réponses

Etape 5 : La fonction `chat_with_pdf` permet de poser une question à un modèle de langage (LLaMA) et d'obtenir une réponse en s'appuyant sur le contenu complet d'un document PDF (déjà extrait et stocké dans `st.session_state.full_text` via Streamlit).

```
def chat_with_pdf(question, client):
    full_text = st.session_state.full_text
    response = client.chat.completions.create(
        model=MODEL_NAME_SUMMARY,
        messages=[{"role": "system", "content": "Tu es un assistant qui répond aux questions generales et des questions sur le document"},
                  {"role": "user", "content": f"Réponds à cette question : {question}\n\n : {full_text}"}]
    )
    return response.choices[0].message.content.strip()
```

Figure 3.20 - Fonction de question/réponse

Partie 2

Dans cette partie nous présentons les résultats de ce modèle LLaMA pour traitement d'un document PDF.

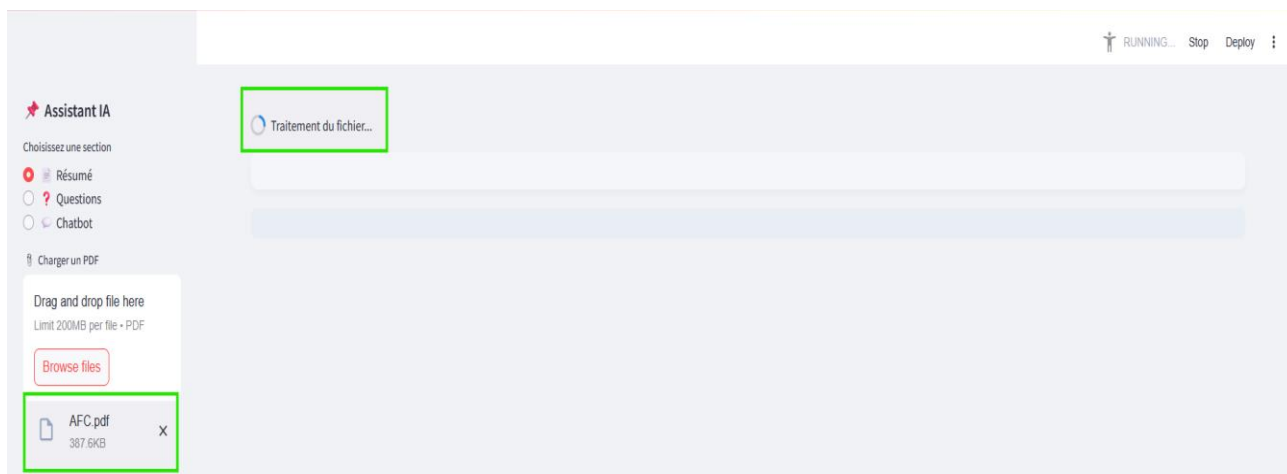


Figure 3.21 - Traitement document PDF

Cette figure présente le traitement du document AFC.pdf, le traitement est de lire les pages de ce document extraire le texte de chaque page avant de commence l'étape 3 de la partie 1 (page 51), la prochaine figure c'est le résultat de cette figure.



Figure 3.22 - Résumé du document AFC.pdf

Cette figure présente le résumé du document AFC.pdf après l'étape 3 (page 51).

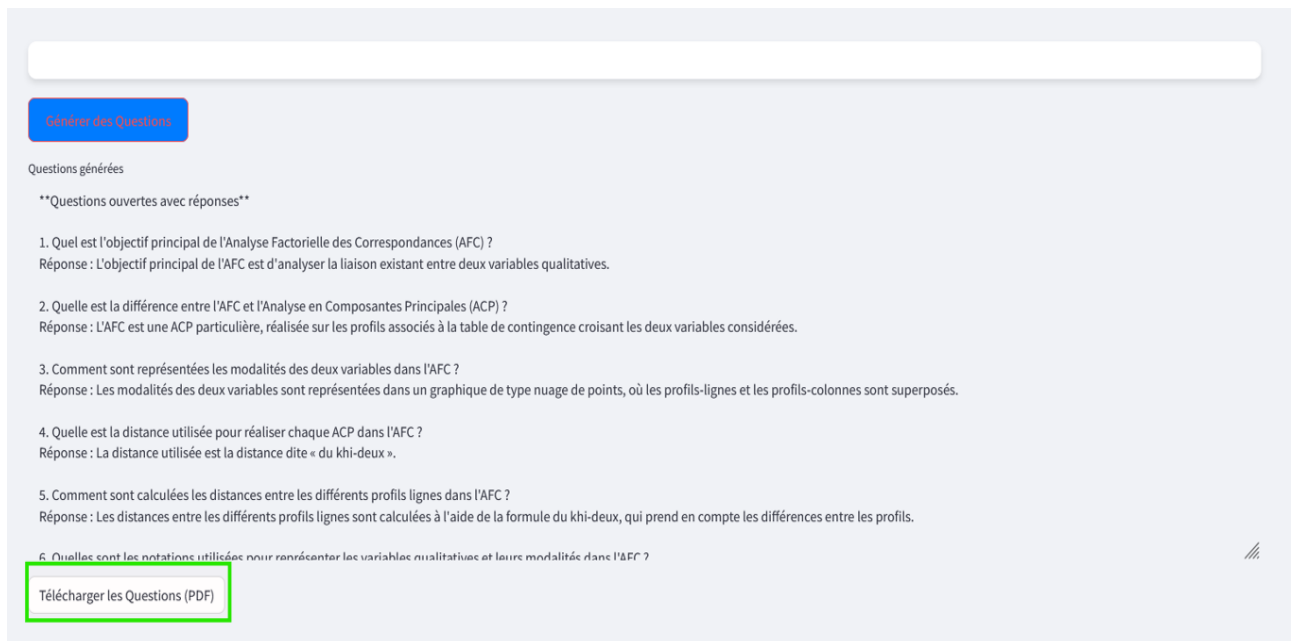


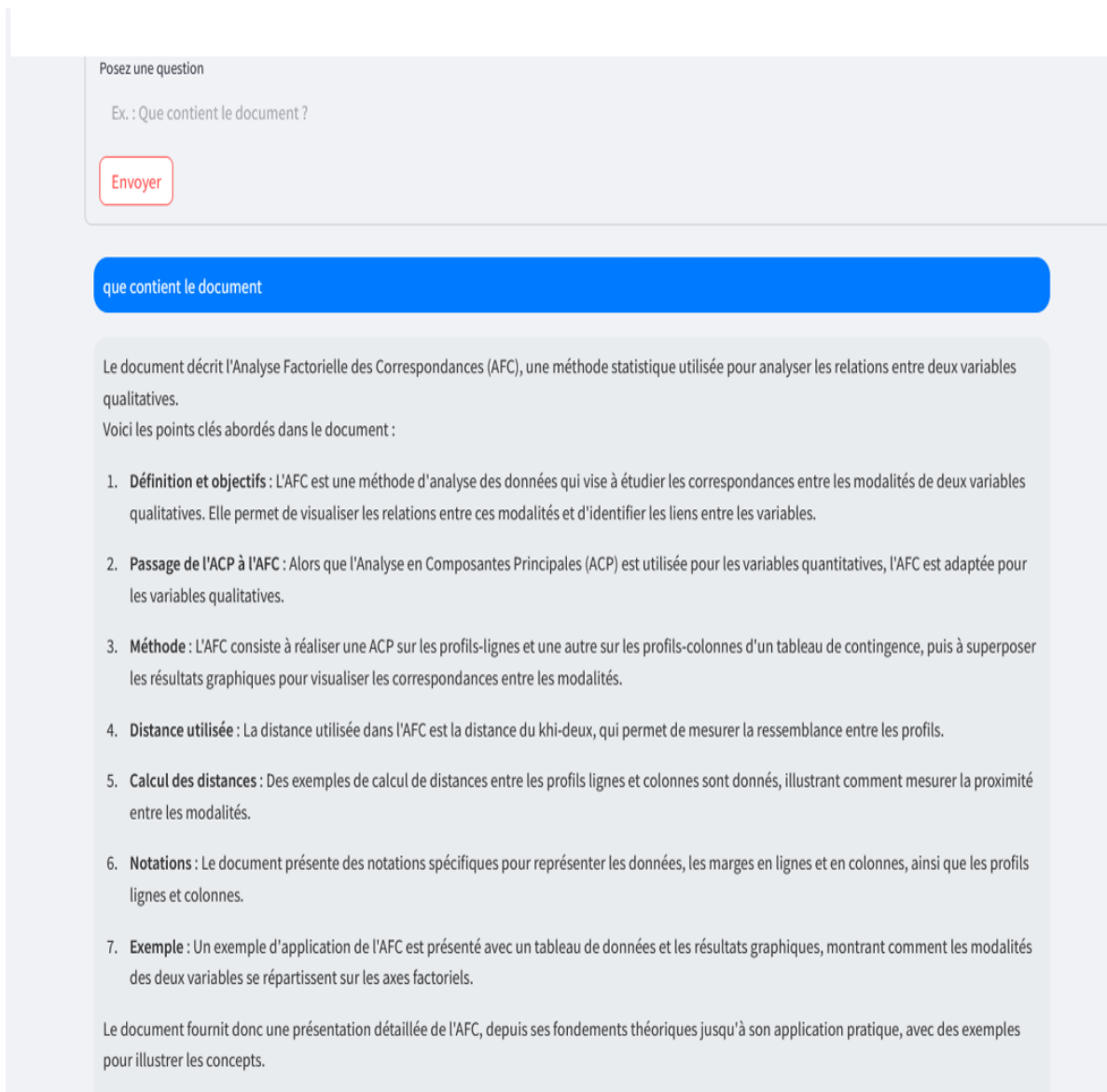
Figure 3.23 - Génération des questions

Après l'étape 4 (page 52) le résultat attendu est cette figure, on voit qu'il y a des questions directes avec réponses, vous pouvez télécharger les questions comme indiqué dans le cadre vert dans la figure 3.24.

<p>**Questions ouvertes avec réponses**</p> <p>1. Quel est l'objectif principal de l'Analyse Factorielle des Correspondances (AFC) ? Réponse : L'objectif principal de l'AFC est d'analyser la liaison existant entre deux variables qualitatives.</p> <p>2. Quelle est la différence entre l'AFC et l'Analyse en Composantes Principales (ACP) ? Réponse : L'AFC est une ACP particulière, réalisée sur les profils associés à la table de contingence croisant les deux variables considérées.</p> <p>3. Comment sont représentées les modalités des deux variables dans l'AFC ? Réponse : Les modalités des deux variables sont représentées dans un graphique de type nuage de points, où les profils-lignes et les profils-colonnes sont superposés.</p> <p>4. Quelle est la distance utilisée pour réaliser chaque ACP dans l'AFC ? Réponse : La distance utilisée est la distance dite « du khi-deux ».</p>	<p>**QCM**</p> <p>1. Quel est l'objectif principal de l'AFC ?</p> <p>a) Analyser la liaison entre deux variables quantitatives</p> <p>b) Analyser la liaison entre deux variables qualitatives</p> <p>c) Réduire la dimension d'un tableau de données</p> <p>d) Classer les individus en fonction de leurs caractéristiques</p> <p>Bonne réponse : b) Analyser la liaison entre deux variables qualitatives</p>
---	--

Figure 3.24 - Partie des questions directes avec réponses et partie des QCM

La dernière étape (étape 5 page 52) était la question / réponse avec le document pdf qu'on a déjà trait, La figure 3.25 présente l'étape 5 question en bleu.



Posez une question

Ex. : Que contient le document ?

Envoyer

que contient le document

Le document décrit l'Analyse Factorielle des Correspondances (AFC), une méthode statistique utilisée pour analyser les relations entre deux variables qualitatives.

Voici les points clés abordés dans le document :

1. **Définition et objectifs** : L'AFC est une méthode d'analyse des données qui vise à étudier les correspondances entre les modalités de deux variables qualitatives. Elle permet de visualiser les relations entre ces modalités et d'identifier les liens entre les variables.
2. **Passage de l'ACP à l'AFC** : Alors que l'Analyse en Composantes Principales (ACP) est utilisée pour les variables quantitatives, l'AFC est adaptée pour les variables qualitatives.
3. **Méthode** : L'AFC consiste à réaliser une ACP sur les profils-lignes et une autre sur les profils-colonnes d'un tableau de contingence, puis à superposer les résultats graphiques pour visualiser les correspondances entre les modalités.
4. **Distance utilisée** : La distance utilisée dans l'AFC est la distance du khi-deux, qui permet de mesurer la ressemblance entre les profils.
5. **Calcul des distances** : Des exemples de calcul de distances entre les profils lignes et colonnes sont donnés, illustrant comment mesurer la proximité entre les modalités.
6. **Notations** : Le document présente des notations spécifiques pour représenter les données, les marges en lignes et en colonnes, ainsi que les profils lignes et colonnes.
7. **Exemple** : Un exemple d'application de l'AFC est présenté avec un tableau de données et les résultats graphiques, montrant comment les modalités des deux variables se répartissent sur les axes factoriels.

Le document fournit donc une présentation détaillée de l'AFC, depuis ses fondements théoriques jusqu'à son application pratique, avec des exemples pour illustrer les concepts.

Figure 3.25 - Interaction avec chatbot

5. Évaluation

Dans cette section, nous présentons les modèle BART et TextRank, et le test de performance entre le modèle LLaMA qu'on a utilisés et ces modelés.

5.1 Modèle BART

- **Chargement des bibliothèques**

```

2 import pdfplumber
3
4 from transformers import BartTokenizer, BartForConditionalGeneration, Seq2SeqTrainingArguments, Seq2SeqTrainer
5

```

Figure 3.26 - Bibliothèques utilisées

- **Import BartTokenizer:** Permet de tokeniser (transformer le texte en tokens) pour l'entrée du modèle BART.
- **BartForConditionalGeneration :** Le modèle BART pré-entraîné pour les tâches de génération conditionnelle (résumé, traduction, réponse à des questions...).
- **Seq2SeqTrainingArguments :** Permet de définir tous les paramètres d'entraînement du modèle nombre d'époques, batch size, learning rate, chemin de sauvegarde, etc.
- **Seq2SeqTrainer :** Le gestionnaire d'entraînement.

Nous avons utilisé pdfplumber pour extraire le texte de chaque page du fichier PDF "AFC.pdf" et l'assembler en un seul bloc.

Nous avons utilisé le modèle BART-large-CNN, pré-entraîné sur CNN/DailyMail, pour générer des résumés reformulés en chargeant un tokenizer et le modèle BART. Il transforme le texte en une forme compréhensible et produit automatiquement des résumés à partir du contenu fourni.

Le texte transmis au modèle BART pour produire des résumés grâce à la méthode beam search.

- **Résumé finale**

Résumé final généré : L'Analyse Factorielle des Correspondances (A.F.C.) est une Analyse en Composantes Principales particulière, réalisée sur les p profils associés à la table de contingence croisant les deux variables considérées. L'AFC est l'étude des correspondances entre les modalités de deux variables qualitatives, ainsi, avant de mettre en œuvre une liaison existe bien. Les profils-colonnes consiste en réaliser une A.C.P. sur deux profils lignes, une autre sur les profils colonnes. réalise l'A.C.P. du tableau des profils-lignes du table de contingence, c'est-à-dire les modalités de X et Y, plusieurs méthodes de type nuage de points. En donne un seul graphique si on ne conserve que deux dimensions, plusieurs dans le cas contraire, réalise la représentation graphique des individus sur un exemple. En réalisant une A.C.P. particulière, nous ne donnons pas pas trop de détails techniques sur cette méthode. Ce texte fait superposer les deux représentations graphiques du khi-reprehensibilité du A.C.P., dite la distance usuelle entre les différents profils lignes ou colonnes. L'AFC fait calculs pour toutes les lignes entre toutes les colonnes entre mis elles. L'AFC utilise les khi-deux pour rendre compte de la ressemblance entre deux lignes ou deux colonnes, mais la distance dite du khi deux dite un peu particulièrement du AFC. Summarize: + 32/12 (5/11-4/14) 2 notations: X et Y : deux variables qualitatives à respectivement p et q modalités, avec i pour X et j pour Y, marge en ligne, colonnes, factorielle des correspondance, analyse Factorielle des correspondances (AFC), données in French, English, French, Italian, Spanish, German, Italian and French. Exemple AFC profils lignes et profils colonnes sur axe 1 et axe 2 (92%) researched in French, English and German. Exercises included: Axe 1 (71%) --> axe 1 (71%) --> enthusiastically14. Axe 1 and axe 2 were examined over a period of 28 days in France, Belgium, France, France and the U.S. The results were published in the journal Biochemistry and Physiology, volume 28, issue number 28.

Figure 3.27 - : Résumé du document AFC.pdf

5.2 Modèle basée sur TextRank

- Chargement des bibliothèques

```
2
3 import fitz # PyMuPDF
4 from nltk.tokenize import sent_tokenize
5 from sumy.parsers.plaintext import PlaintextParser
6 from sumy.nlp.tokenizers import Tokenizer
7 from sumy.summarizers.text_rank import TextRankSummarizer
8 from fpdf import FPDF
9 import os
```

Figure 3.28 - Bibliothèque du TextRank

- **Import Fitz:** Lire et manipuler des fichiers PDF.
- **from nltk.tokenize import sent_tokenize:** Découper un texte en phrases.
- **from sumy.parsers.plaintext import PlaintextParser:** Préparer un texte brut pour le résumé avec la bibliothèque Sumy.
- **from sumy.nlp.tokenizers import Tokenize:** Tokeniseur (découpeur) utilisé avec Sumy pour découper le texte en phrases
- **from sumy.summarizers.text_rank import TextRankSummarizer:** Implémente TextRank, un algorithme de résumé extractif.
- **import os:** Manipuler les chemins de fichiers et dossiers du système d'exploitation.
- **from fpdf import FPDF:** Créer des fichiers PDF à partir de texte.

- Résumé extractif

On va générer un résumé extractif à partir d'un texte, en s'appuyant sur l'algorithme TextRank. Cet algorithme repose sur un graphe de similarité entre phrases pour identifier les idées principales.

- Résumé finale

```

--- Résumé Extractif (TextRank) ---
De l'ACP à l'AFC BACP = étude des liaisons contenues dans un tableau individus *variables, lorsque toutes les variables sont quantitatives .
BACF (analyse factorielle des correspondances) et l'ACM (analyse des correspondances multiples) = étude des liaisons continues dans un tableau individus *variables, lorsque toutes les variables s
ont qualitatives.
L'AFC est l'étude des correspondances entre les modalités de deux variables qualitatives .
Son objectif est d'analyser la liaison existant entre deux variables qualitatives (si on dispose de plus de deux variables qualitatives, on aura recours à l'Analyse des Correspondances Multiples
).
particulière, réalisée sur les profils associés à la table de contingence croisant les deux variables considérées.
Les résultats graphiques de ces deux analyses sont ensuite superposés pour produire un graphique (éventuellement plusieurs) de type nuage de points, dans lequel sont réunies les modalités des deu
x variables considérées, ce qui permet d'étudier les correspondances entre ces modalités, autrement dit la liaison entre les deux variables.
sont les lignes de la table de contingence, c'est-à-dire les modalités de X) et l'on fait la représentation graphique des individus, donc des modalités de X (dans cette A.C.P.
sont maintenant les colonnes de la table de contingence, c'est-à-dire les modalités de Y) et l'on fait la représentation graphique des individus, donc des modalités de Y.
se correspondent (ce qui est normal, puisque leurs données sont extraites de la même table de contingence) et qu'il est donc légitime de superposer les deux représentations graphiques.
On obtient ainsi un graphique de type nuage de points, représentant à la fois les modalités de X et celles de Y.
Analyse Factorielle des correspondance Méthode Bsignalons que la distance entre profils (lignes ou colonnes), utilisée pour réaliser chaque A.C.P., est un peu particulière : ce n'est pas la dista
nce usuelle, mais la distance dite « du khi- deux ».
Analyse Factorielle des correspondance Bcalculez les distances entre les différents profils lignes ?
Analyse Factorielle des correspondance Méthode Bexemple de base de données et du calcul des distances khi-deux entre les lignes (ABC) et les colonnes (1,2,3) : Bentre A et B = 32/8(1/11-2/7)2+32/
12 (5/11-2/7)2+32/12 (5/11-3/7)2 Bentre B et C = 32/8 (2/7-5/14)2+ 32/12 (2/7-5/14)2+32/12 (3/7-4/14)2 Bentre A et C = 32/8 (1/11-5/14)2+ 32/12 (5/11-5/14)2+ 32/12 (5/11-4/14)2 Bquels sont les l
ignes les plus proches entre elles ?
Notations B X et Y : deux variables qualitatives à respectivement p et q modalités B=(nij), avec i=1,...,p et j=1,...,q : nombre d'individus ayant la modalité i pour X et j pour Y ai=1 p oj
=1 q nij= n BMarge en ligne : (ni.) BMarge en colonne : (n.j) = ai=1pmij(effectifs des modalités de Y) BDIA et DY matrice diagonale contenant les marges en lignes et en colonnes sur la dia
gonale BProfils lignes : nij ni .

```

Figure 3.29 - Résumé de TextRank

5.3 Evaluation final

Dans cette section, on va faire une étude comparative entre Llama, BART et TextRank, avec des algorithmes d'évaluation basé sur le résumé linguistique. On va voir aussi les critères d'évaluation, ainsi que leur fonction pour l'évaluation de résumés.

5.3.1 BERTScore

BERTScore est une métrique d'évaluation de similarité sémantique entre un résumé généré et un texte de référence, basée sur les embeddings contextuels pré-entraînés de BERT. BERTScore calcule la similarité entre deux phrases en sommant les similarités cosinus entre les embeddings de leurs jetons [34].

BERTScore renvoie 3 scores sont précision, rappel, F1-score

Tableau 3.1 - Normes d'évaluation du BERTScore

Score	Signification
Précision P	À quel point les mots du résumé généré ont un sens proche de ceux du texte de référence
Rappel R	À quel point les mots du texte de référence sont bien capturés dans le résumé
F1-Score	Moyenne harmonique de P et R : score final de qualité

Plus les scores sont proches de **1.0 (ou 100%)**, meilleur est le résumé.

Donc pour l'évaluation avec BERTScore qui utilise **bert-base-multilingual-cased**². Ce modèle développé par google pré-entraîné sur 104 langues (dont le français), Ainsi sur OSCAR³, Il est largement utilisé dans des tâches comme la classification, la recherche d'information, ou les mesures de similarité sémantique [35].

Elle donne ces résultats (voir figure 3.35)

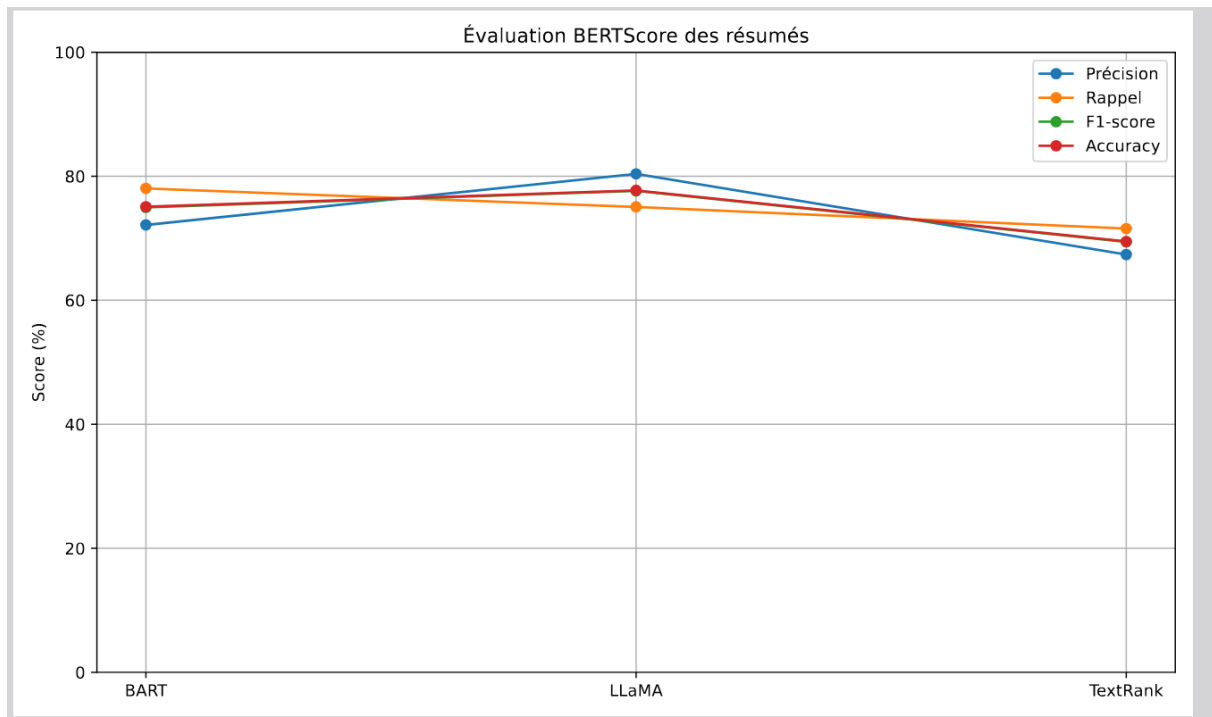


Figure 3.30 - Graphe d'évaluation selon la mesure BERTScore

La figure 3.35 présente une évaluation des performances de trois modèles de génération de résumés (BART, LLaMA, et TextRank) à l'aide de la métrique BERTScore, qui mesure la qualité sémantique des résumés par rapport au texte source. Les métriques évaluées incluent la précision (bleu), le rappel (orange), le F1-score (vert), et l'exactitude (rouge), exprimées en pourcentage sur l'axe vertical. L'axe horizontal indique les trois modèles.

La légère baisse des scores pour TextRank pourrait indiquer qu'il repose davantage sur une extraction simple (basée sur les graphes) plutôt qu'une génération sémantique riche comme BART ou LLaMA.

Nous avons présenté ces points indique dans le graphe d'une façon plus lisible sous forme un tableau des scores figure 3.36.

² **Bert-base-multilingual-cased** : <https://huggingface.co/google-bert/bert-base-multilingual-cased>

³ **OSCAR**: (Open Super-large Crawled ALMANaCH coRpus) : un corpus web massif multilingue, extrait de Common Crawl.

Tableau des scores BERTScore

Modèle	Précision (%)	Rappel (%)	F1-score (%)	Accuracy (%)
BART	72.15	78.04	74.98	75.09
LLaMA	80.38	75.06	77.63	77.72
TextRank	67.38	71.56	69.41	69.47

Figure 3.31 - Tableau des scores

5.3.2 FREScore

Est une version francophone de la célèbre métrique BERTScore (French BERTScore). Il permet d'évaluer automatiquement la qualité sémantique d'un texte résumé en français en le comparant à un texte de référence qui le texte d'origine. Il utilise un modèle BERT optimisé pour le français (camembert-base [36], flaubert-base).

La figure 3.37 présente un tableau de scores entre LLaMA, BART et TextRank

Évaluation de lisibilité (FREScore)

Modèle	FREScore	Lisibilité
BART	55.53	Assez difficile
LLaMA	65.57	Moyenne
TextRank	49.43	Difficile

Figure 3.32 - Scores des modèles avec FREScore

Le FREScore, qui évalue la lisibilité des résumés, varie de 49,43 (TextRank) à 65,57 (LLaMA), révélant une différence notable entre les modèles. LLaMA se démarque en produisant les résumés les plus lisibles avec un score de 65,57, ce qui le rend idéal pour un public général grâce à sa clarté et sa simplicité. BART, avec un score intermédiaire de 55,53, offre une lisibilité convenable, adaptée à un public ayant une certaine familiarité avec le sujet, tandis que TextRank, avec 49,43, est le moins adapté à une lecture facile, probablement en raison de son approche basée sur l'extraction de phrases clés sans optimisation pour la fluidité.

En conclusion LLaMA apparaît comme le meilleur modèle global. Il combine une excellente performance sémantique (meilleur F1-score et accuracy avec BERTScore) avec une lisibilité supérieure (FRESScore de 65,57), le rendant adapté à un étudiant, notamment dans un contexte éducatif.

6. Conception par UML

Dans cette section, nous verrons l'analyse des besoins et les conceptions nécessaires pour réaliser notre système, en utilisant des modèles de diagrammes. Ces diagrammes sont modélisés en utilisant le langage UML (langage de modélisation standardisé).

6.1 Définition des acteurs

Dans notre système, on a trois acteurs principaux qui sont :

- **Administrateur** : l'admin gère les cours, les catégories de cours et les utilisateurs.
- **Enseignant** : il gère les ressources et les activités, il communique avec les autres utilisateurs, il peut utiliser l'assistant IA.
- **Etudiant** : il consulte les cours, participe aux activités, il peut communiquer avec les utilisateurs, ainsi qu'il peut utiliser l'assistant IA.

6.2 Besoins fonctionnels

Administrateur :

- L'authentification.
- Création et gère les cours et les catégories.
- Création et Gérer les utilisateurs et leurs rôles dans un cours.
- Gérer les permissions.
- Envoie et recevoir des messages.

Enseignant :

- L'authentification.
- Ajouter des étudiants au cours.
- Gérer les activités et les ressources.
- Communication avec les utilisateurs.
- Utilisation de l'assistant IA.

Etudiant :

- L'authentification.
- Participations aux activités.
- Communication avec les utilisateurs.
- Consultation des cours.
- Utilisation de l'assistant IA.

6.3 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation UML résume les informations des acteurs d'un système et leurs interactions. Les diagrammes de cas d'utilisation sont parfaits pour organiser les fonctionnalités de système.

6.3.1 Acteur Administrateur

Ce diagramme de cas d'utilisation décrit les différentes actions de l'administrateur.

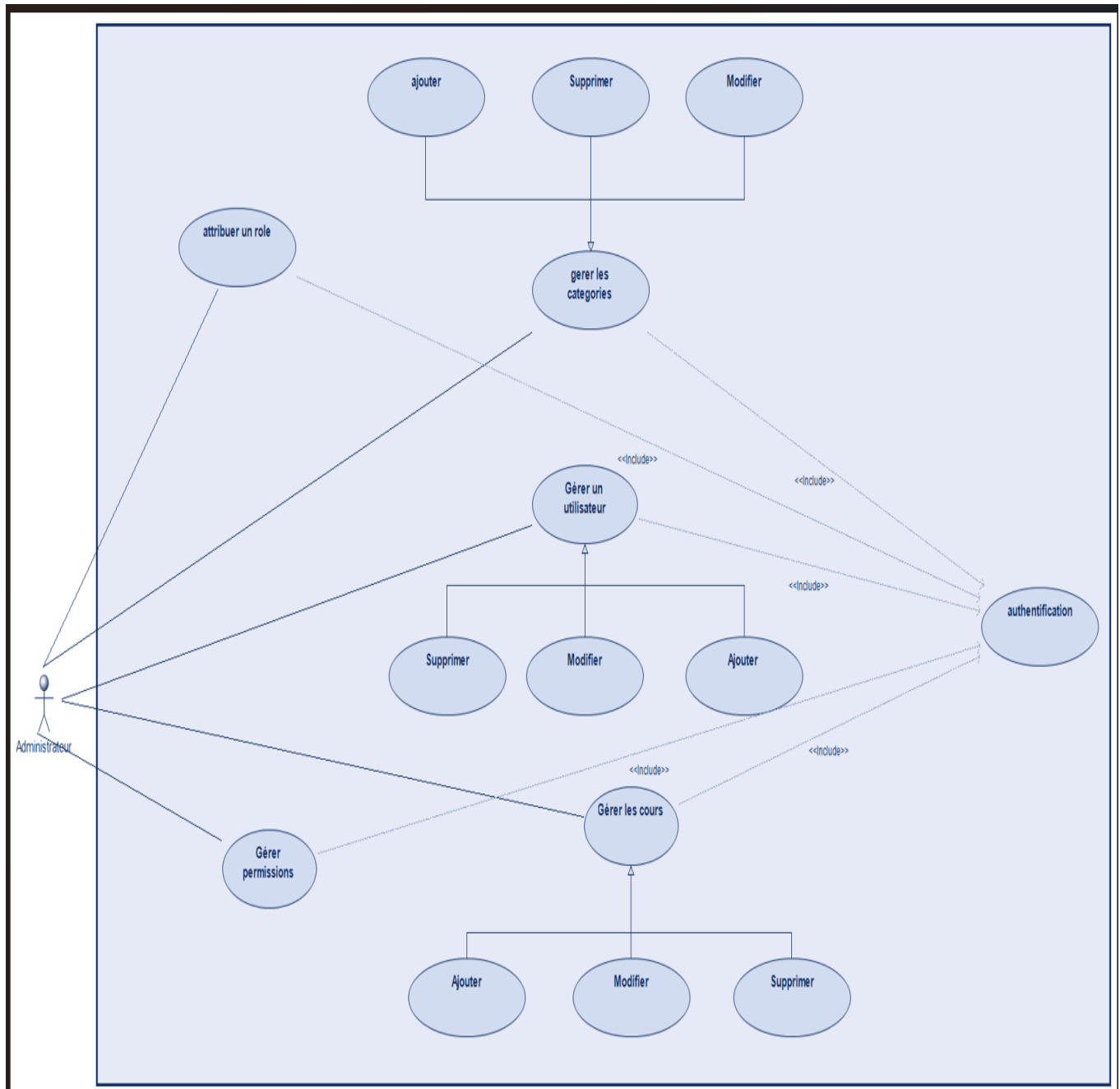


Figure 3.33 - Diagramme de cas d'utilisation « Client »

6.3.2 Acteur Enseignant

Ce diagramme de cas d'utilisation décrit les différentes actions de l'enseignant.

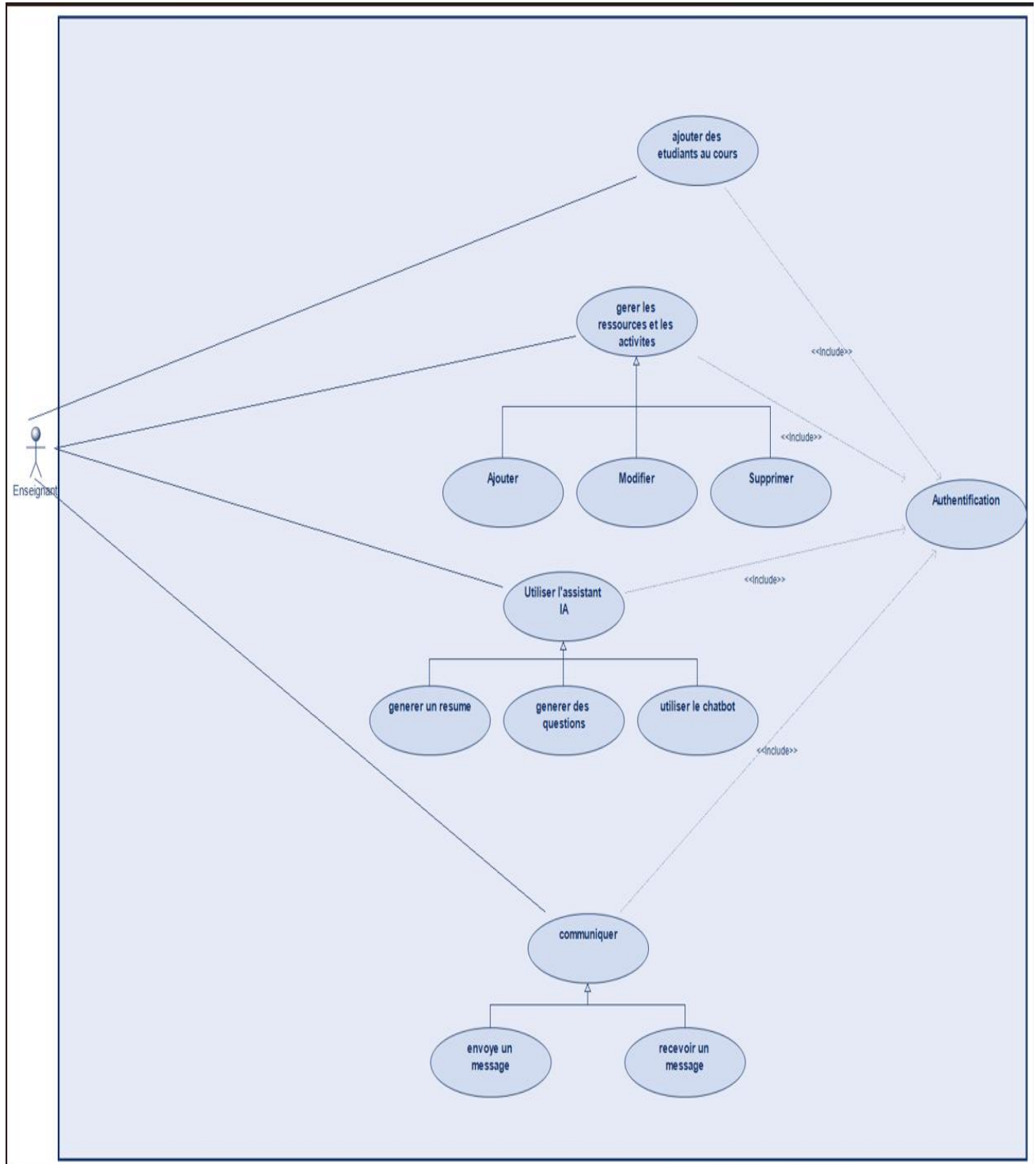


Figure 3.34 – Diagramme de de cas d'utilisation « Enseignant »

6.3.3 Acteur Etudiant

Ce diagramme de cas d'utilisation décrit les différentes actions de l'étudiant.

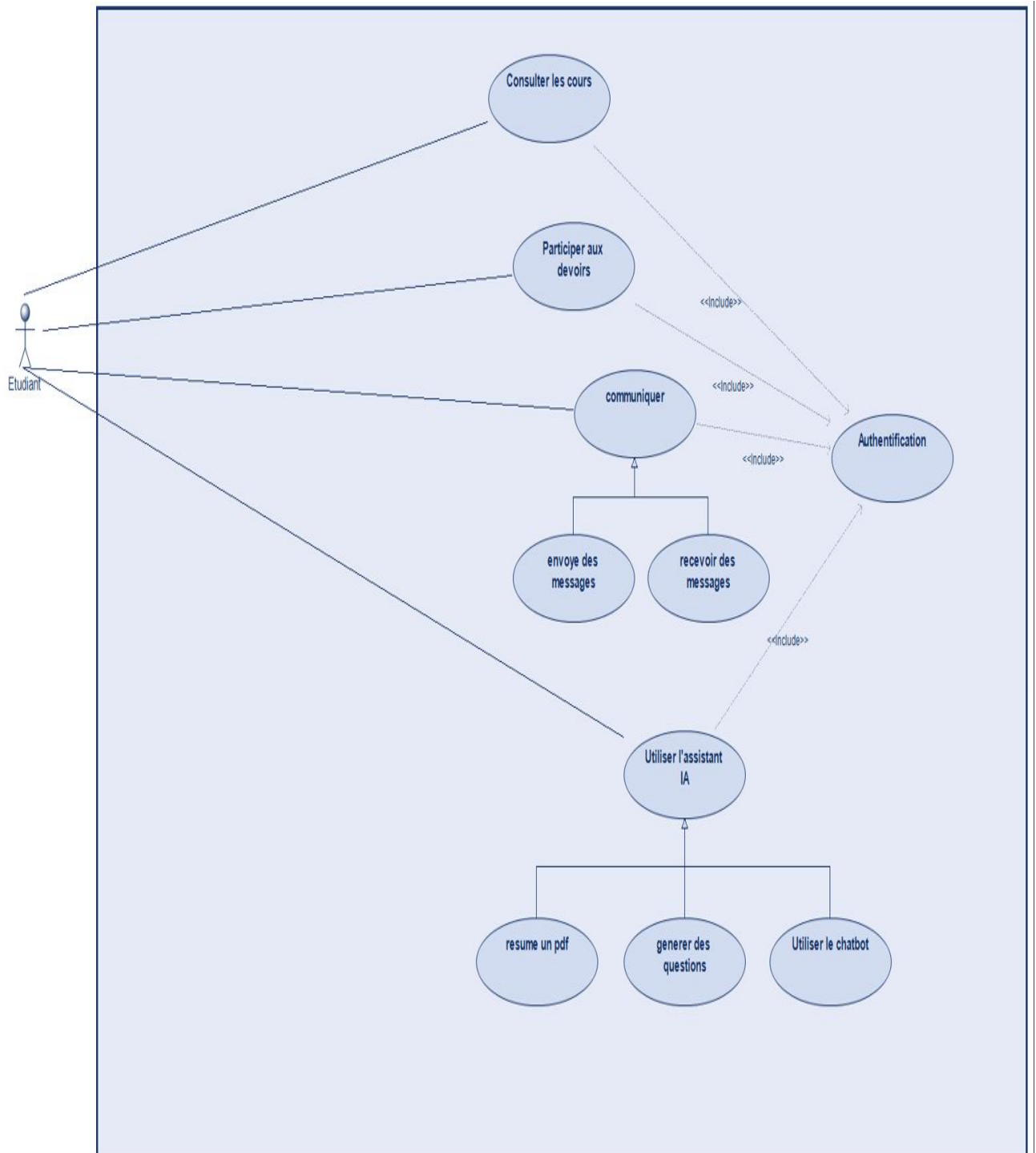


Figure 3.35 - Diagramme de cas d'utilisation « Etudiant »

6.4 Diagramme de classe

Les diagrammes de classe utilisent pour documenter l'architecture des logiciels, il illustre des modèles de données pour des systèmes d'information quel que soit leur degré de complexité. [37]

Ce diagramme représente la structure et les relations entre les différentes classes de notre plateforme EtudeBrief.

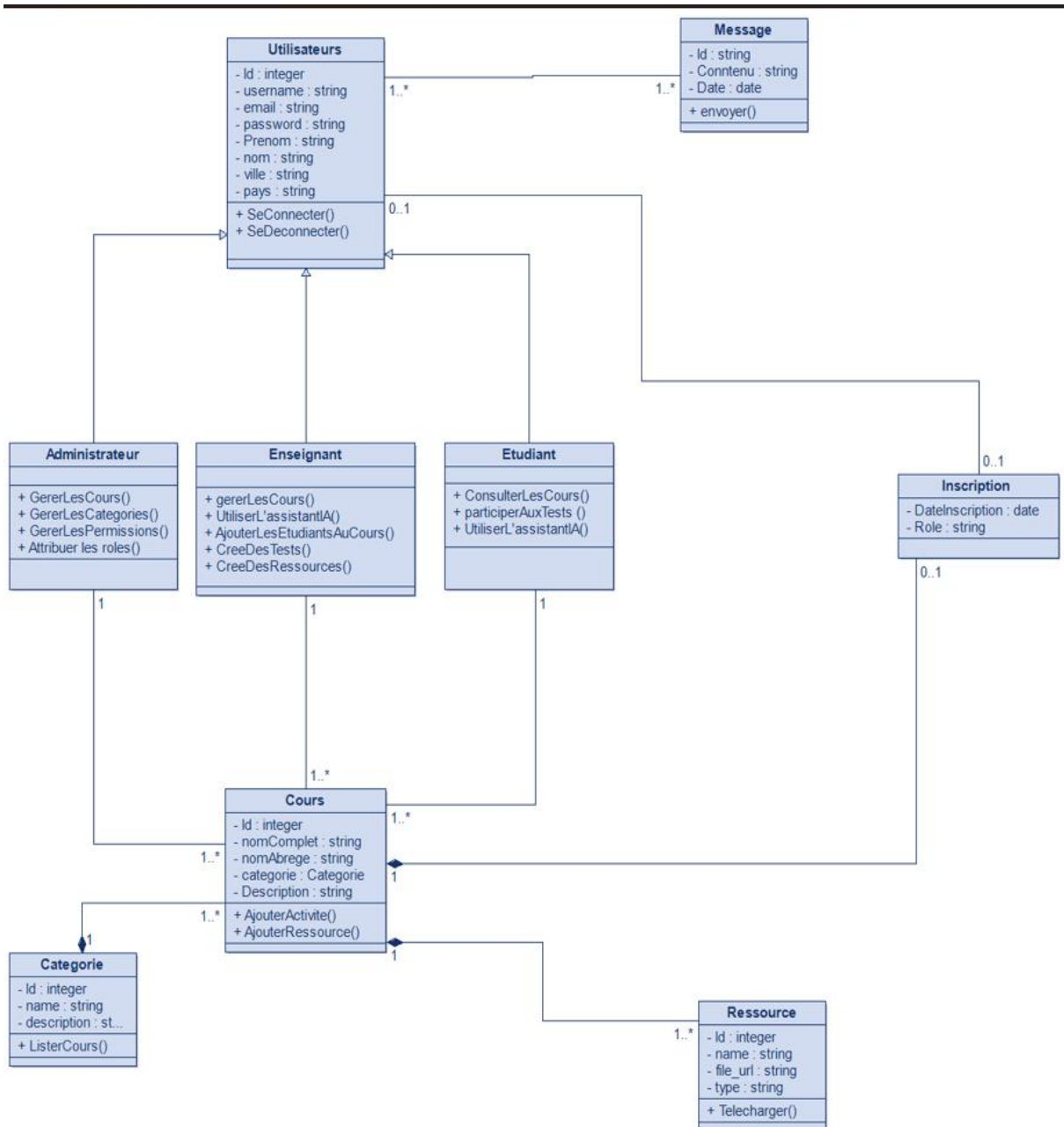


Figure 3.36 - Diagramme de classe

7. Présentation des interfaces

Nous présenterons les interfaces principales de cette plateforme "EtudeBrief" de moodle, on commence par l'interface de connexion.

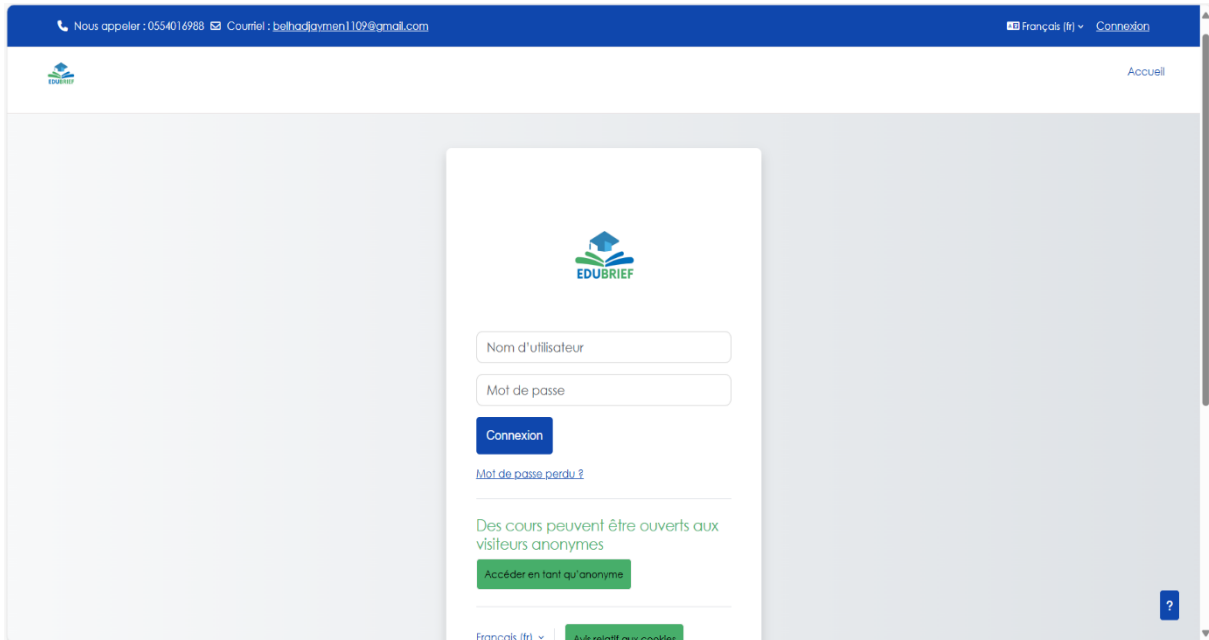


Figure 3.37 – Interface de connexion

La prochaine interface est l'accueil dans cette interface on présente les cours qui existent

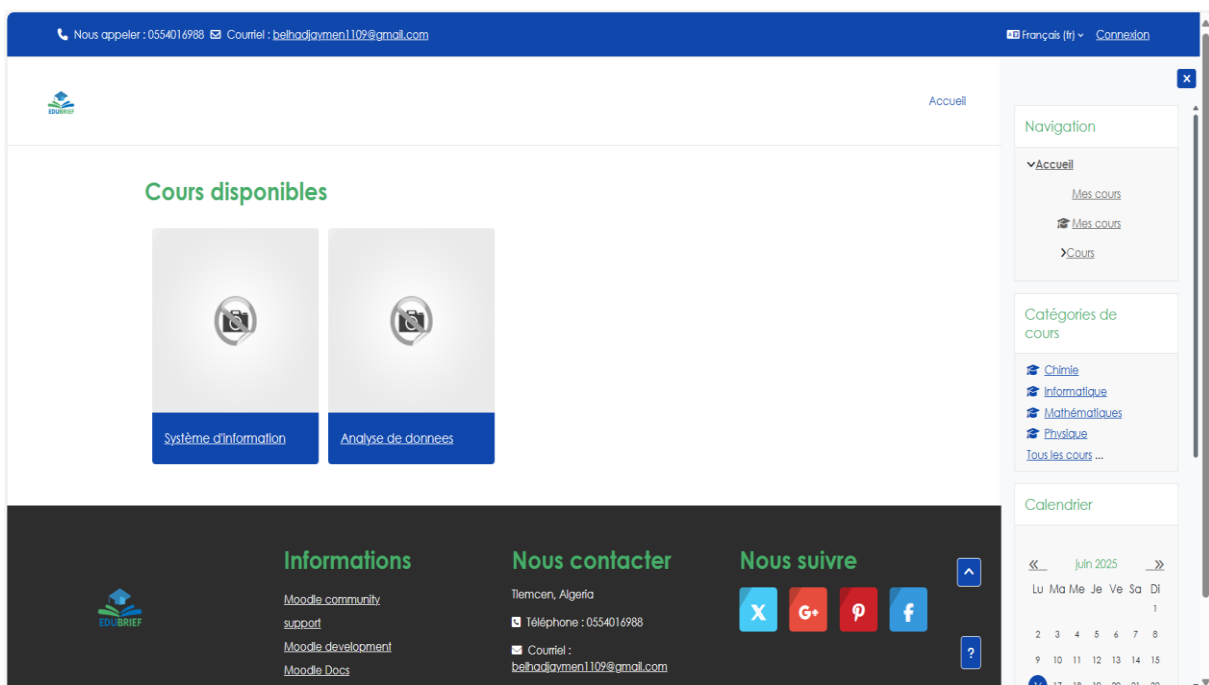


Figure 3.38 – Interface d'accueil

L'interface de figure 3.44 présente la création d'un utilisateur via le compte de l'administrateur.

The screenshot shows the 'EduBrief' user creation form. The top navigation bar includes contact information and a search bar. The main menu has tabs for 'Général', 'Utilisateurs', 'Cours', 'Notes', 'Plugins', 'Présentation', 'Serveur', 'Rapports', and 'Développement'. The 'Général' section is expanded, showing fields for 'Nom d'utilisateur' (filled with 'etudiant2'), 'Choisir une méthode d'authentification' (set to 'Comptes manuels'), 'Nouveau mot de passe' (masked with dots), 'Prénom' (filled with 'aymen'), 'Nom de famille' (filled with 'mah'), 'Adresse de courriel' (filled with 'aymenmah13@gmail.com'), and 'Visibilité de l'adresse de courriel' (set to 'Visible pour tout le monde'). There are also checkboxes for 'Compte suspendu' and 'Générer le mot de passe et informer l'utilisateur'. A 'tout réinitialiser' link is visible in the top right.

Figure 3.39 – Interface d'ajouter un utilisateur

La figure 3.45 illustre le processus de création d'une catégorie.

The screenshot shows the 'EduBrief' course category creation form. The top navigation bar is similar to the previous figure. The main menu has tabs for 'Général', 'Utilisateurs', 'Cours', 'Notes', 'Plugins', 'Présentation', 'Serveur', 'Rapports', and 'Développement'. The 'Cours' tab is selected, and the page title is 'Ajouter une catégorie de cours'. The form includes fields for 'Catégorie supérieure' (set to 'Niveau supérieur'), 'Nom de catégorie' (filled with 'Informatique'), and 'Numéro d'identifiant de catégorie' (filled with '14'). A rich text editor is provided for the 'Description' field, with a toolbar containing options like 'Modifier', 'Afficher', 'Insérer', 'Format', 'Outils', 'Tableau', and 'Aide'. At the bottom, there are 'Créer catégorie' and 'Annuler' buttons.

Figure 3.40 – Interface d'ajouter une catégorie

Après la création de la catégorie on va créer le cour (course en anglais) sous forme un module, la figure 3.46 illustre ce processus.

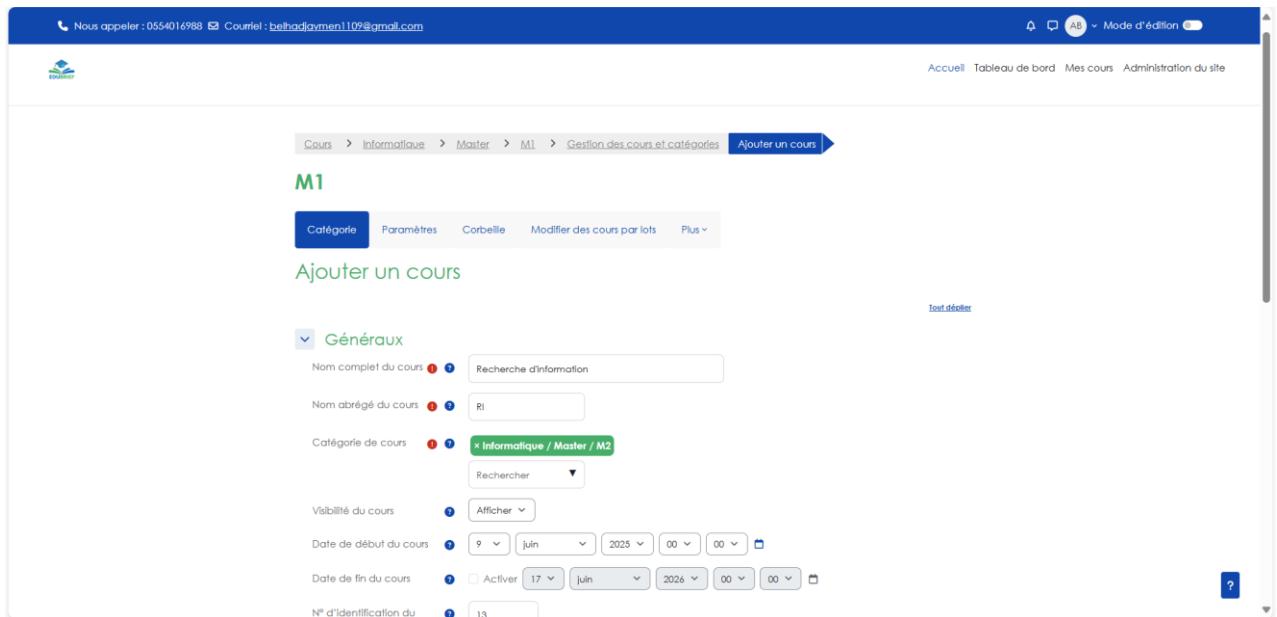


Figure 3.41 - Interface d'ajouter un cours

Après avoir la création du cours, l'enseignant peut ajouter de ressource comme document pdf. La figure illustre le processus d'ajouter une ressource.

Système d'information

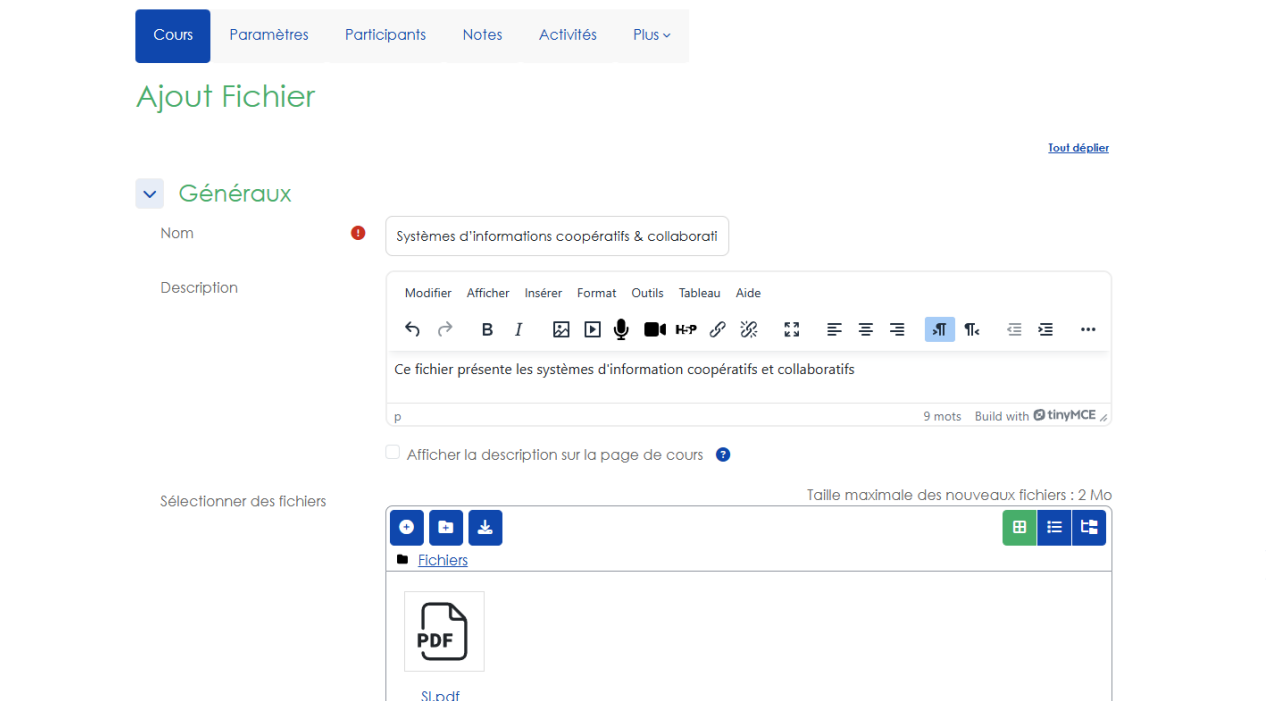


Figure 3.42 - Interface d'ajouter une ressource

L'étudiant peut utiliser l'assistant IA pour générer un résumé, des questions pour la révision et de discuter avec le document via un chatbot, Les figure illustrent ces processus

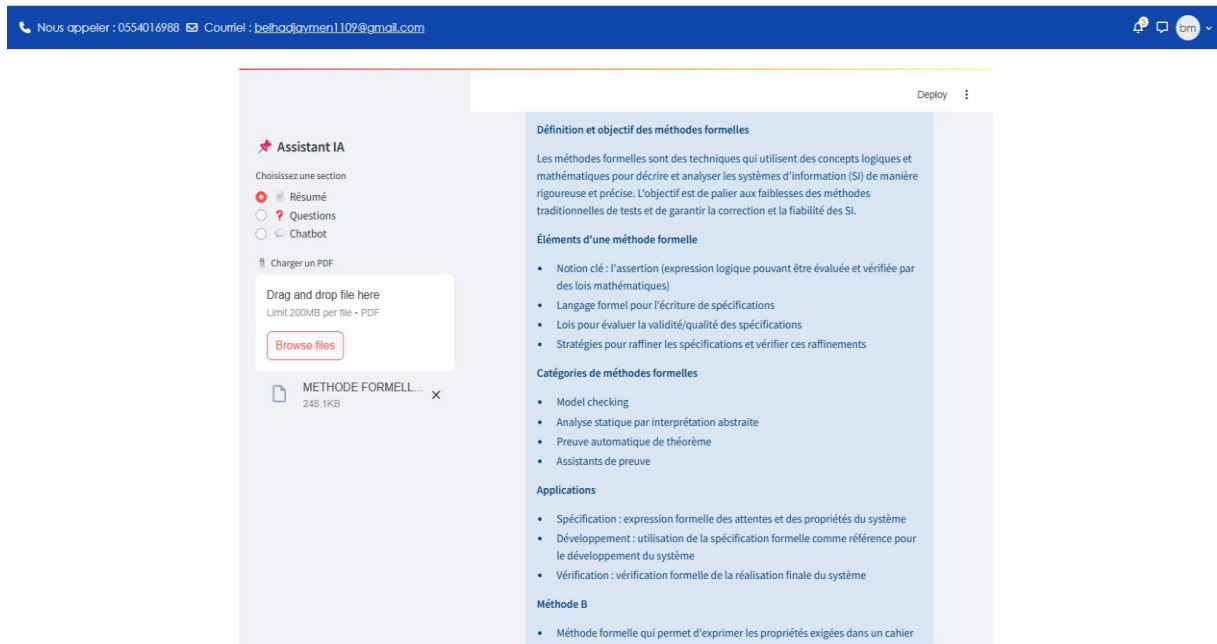


Figure 3.43 – Interface du résumé un document PDF

Il peut générer des questions sur les PDF

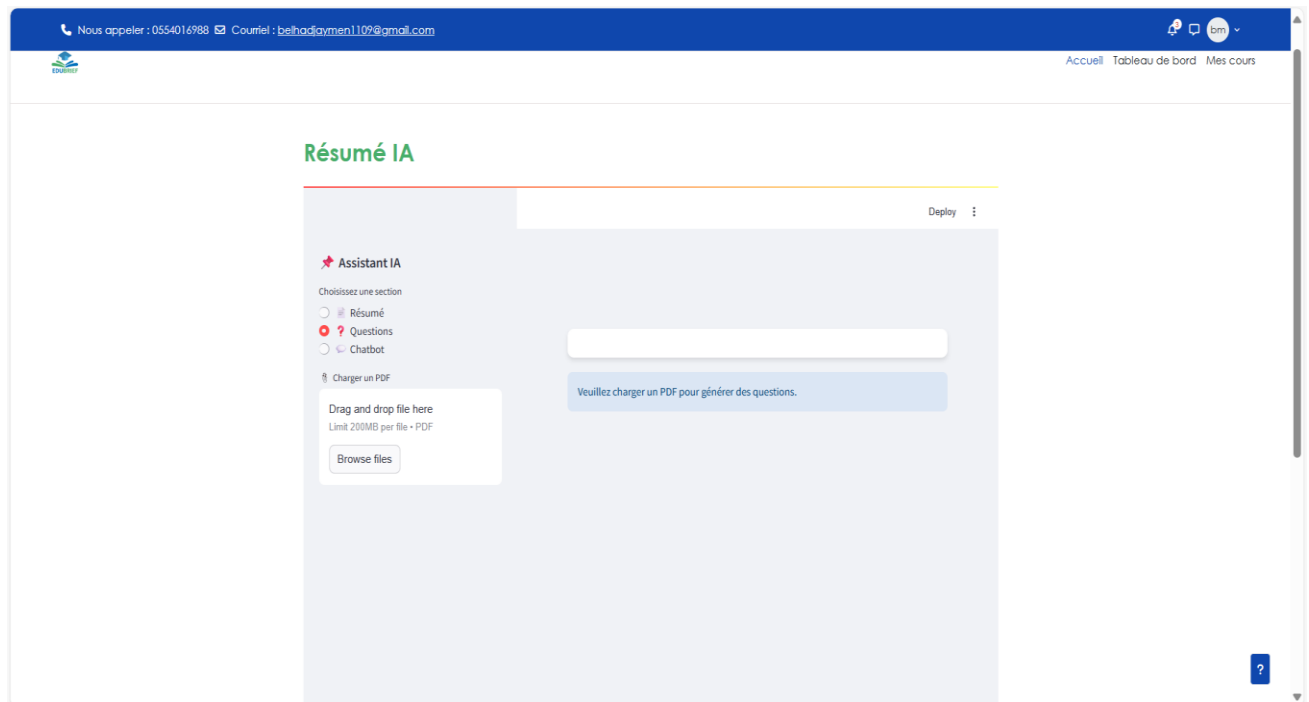


Figure 3.44 – Interface de génération des questions

Après la génération des question, l'étudiant peut discuter avec le document question /réponse.

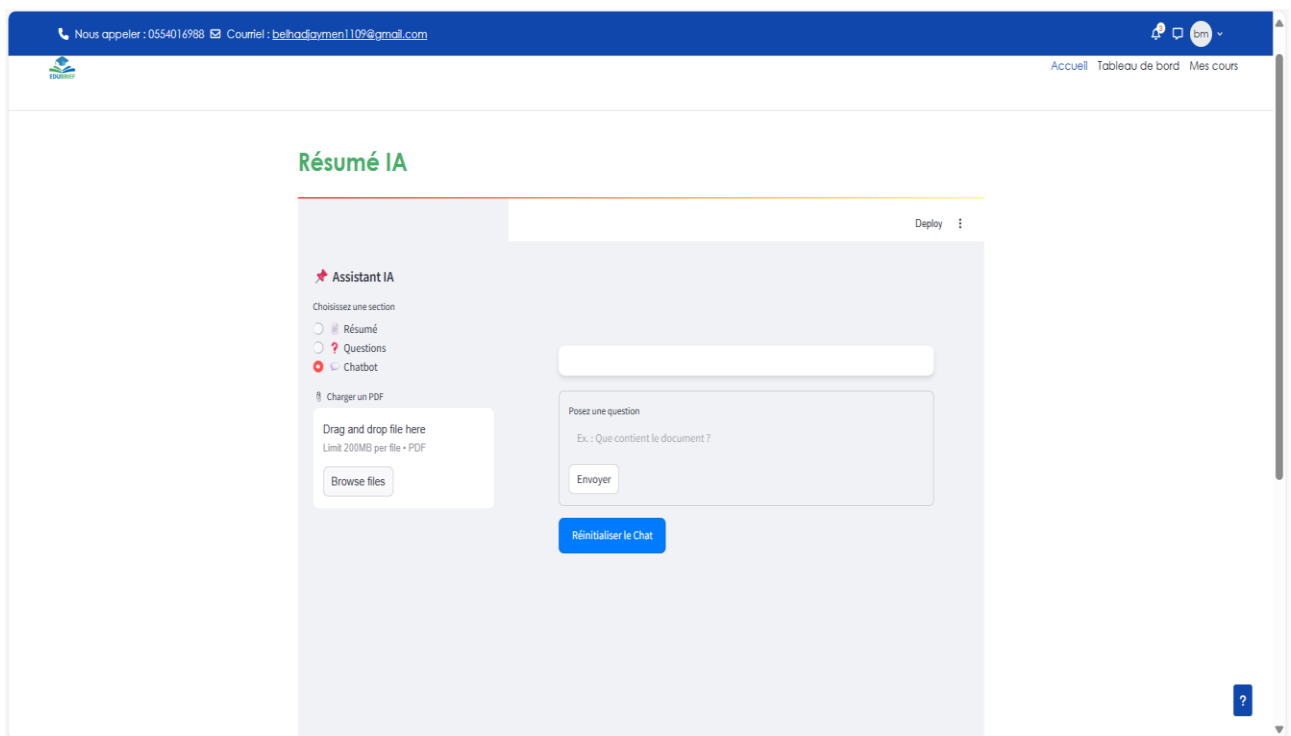


Figure 3.45 - Interface de chatbot

8. Conclusion

Dans ce chapitre nous avons parlé du flux du travail proposé, L'expérimentation avec le modèle LLaMA, intégré via l'API Groq, a démontré sa capacité à générer des résumés clairs. Les évaluations comparatives avec BART et TextRank, basées sur les métriques BERTScore et FRESScore, ont révélé la supériorité de LLaMA en termes de qualité sémantique et de lisibilité, le rendant idéal pour un contexte éducatif. Les diagrammes UML (cas d'utilisation et classe) ont structuré les besoins fonctionnels des acteurs (administrateur, enseignant, étudiant), tandis que les interfaces conçues sur Moodle illustrent une intégration fluide et intuitive.

Conclusion générale et perspectives

En conclusion de ce mémoire consacré à développer une plateforme éducative interactive (moodle) basée sur l'optimisation des apprentissages grâce aux résumés linguistiques.

Dans ce projet, nous avons proposé une méthode pour résoudre le problème du résumé d'un document en faisant un système de résumé automatisé via API groq qui permet à l'étudiant de résumer un document (cours pdf) et d'obtenir un résumé court et riche des informations de ce document, cette méthode améliore la compréhension des étudiants.

De plus, nous avons proposé deux méthodes différentes de LLM : LLaMA qui est entraînée et affinée sur des datasets de résumé et des instructions, la deuxième méthode est BART support LLM mais elle est entraînée et affinée que sur les datasets. Les instructions ou bien les prompts dans l'évolution de la technologie et de traitement automatique du langage naturel sont importants. Nous considérons que nous avons réussi à atteindre la grande partie de l'objectif de ce projet, cette solution pour optimiser l'apprentissage.

Perspectives

- **Intégration d'un système de classification automatique des documents** : Développer une fonctionnalité dans EtudeBrief pour classer automatiquement les documents PDF (ex. : par sujet, niveau scolaire, ou type de contenu).
- **Déploiement cloud** : Migrer la plateforme vers un environnement cloud pour dépasser les contraintes matérielles actuelles.
- **Evaluation continue** : Mettre en place un système de feedback utilisateur pour optimiser la qualité des sorties.
- **Gestion améliorée des PDF de longue taille** : Améliorer EtudeBrief pour traiter efficacement des documents PDF volumineux.
- **Utilisation d'un matériel plus performant** : Équiper avec un serveur performant (AMD Ryzen 9/Intel Xeon, 32-64 Go RAM, GPU NVIDIA RTX 3090) pour accélérer le traitement des modèles IA comme LLaMA.

Bibliographie

- [1] C. européenne, «Réaliser un espace européen de l'apprentissage tout au long de la vie,» 21 11 2001. [En ligne]. Available: <https://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=COM:2001:0172:FIN:FR:PDF>. [Accès le 12 05 2025].
- [2] S. Niaz, S. Memon et S. Khokhar, «Development of E-learning: A Historical Review with Global Perspective,» 10 2021. [En ligne]. Available: https://www.researchgate.net/publication/355737261_Development_of_E-learning_A_Historical_Review_with_Global_Perspective.
- [3] Moodle, «Qu'est-ce qu'un LMS (Learning Management System) ?,» 22 01 2022. [En ligne]. Available: <https://moodle.com/fr/nouvelles/quest-ce-quun-lms-learning-management-systems-explique/>.
- [4] «Tout savoir sur la différence entre LMS vs LCMS,» [En ligne]. Available: <https://www.didask.com/post/tout-savoir-sur-la-difference-entre-lms-vs-lcms>. [Accès le 24 4 2025].
- [5] «MOOC : Définition,» [En ligne]. Available: <https://www.digiforma.com/definition/mooc/>. [Accès le 14 03 2025].
- [6] «What is Moodle? The complete guide to Moodle LMS,» [En ligne]. Available: <https://www.hubkengroup.com/resources/what-is-moodle-lms-guide#:~:text=Put%20simply%2C%20Moodle%20is%20a,measure%20training%20and%20learning%20online..>
- [7] «Ce qu'est Open edX et quels sont les avantages de Open edX ?,» [En ligne]. Available: <https://aulasneo.com/fr/ce-qui-est-open-edx-et-quels-sont-les-avantages-de-open-edx/>.
- [8] «Chamilo,» [En ligne]. Available: <https://www.ludiscape.com/chamilo/>.
- [9] «What is a learning management system (LMS)?,» [En ligne]. Available: <https://www.instructure.com/lms-learning-management-system>.
- [10] «What Is Coursera?,» [En ligne]. Available: <https://www.coursera.org/articles/what-is-coursera>.

- [11] C. S. Jim Holdsworth, «Qu'est-ce que le NLP ?», [En ligne]. Available: <https://www.ibm.com/fr-fr/think/topics/natural-language-processing#:~:text=Le%20traitement%20automatique%20du%20langage,de%20communiquer%20en%20langage%20humain..>
- [12] L. Post, «Google Translate Architecture Illustrated: comment Google Translate utilise le modèle Transformer,» 29 12 2024. [En ligne]. Available: <https://www.lavivienpost.com/fr/google-translate-and-transformer-model/>.
- [13] «[Commande vocale] Reconnaissance Automatique de la Parole,» [En ligne]. Available: <https://www.aquiladata.fr/insights/commande-vocale-reconnaissance-automatique-de-la-parole/>.
- [14] J.-M. Torres-Moreno, «Automatic Text Summarization,» [En ligne]. Available: https://www.researchgate.net/publication/269930486_Automatic_Text_Summarization.
- [15] A. Aries, *Résumé automatique de textes*, 2013, p. 21.
- [16] R. Bois, «Introduction au résumé automatique,» 19 10 2021. [En ligne]. Available: <https://ledatablog.com/ia/nlp/python/2021/10/19/introduction-au-resume-automatique.html>.
- [17] R. Mihalcea et P. Tarau, *TextRank: Bringing Order into Text*, Association for Computational Linguistics éd., D. Wu et D. Lin, Éd., 2004, p. 404–411.
- [18] A. C. Özmen, *Extractive text summarization using TextRank algorithm (Basic Logic)*, 2021.
- [19] Ashish Vaswani, Niki Parmar, Jakob Uszkoreit, Llion Jones et A. Gomez, *Attention Is All You Need*, 2017.
- [20] MICROSOFT, *Que sont les modèles de langage de grande taille (LLM) ?*.
- [21] *GPT : un outil révolutionnaire pour le monde du BtoB*.
- [22] M. LEWIS, Y. LIU, N. Goyal, M. Ghazvininejad, A. Mohamed et L. Zettlemoyer, *BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension*, 2019.
- [23] A. VIDHYA, *BART Model – A Complete Guide to BART for Text Generation*, 2024.

- [24] A. Rogers, O. Kovaleva et A. Rumshisky, *A Primer in BERTology: What we know about how BERT works*, 2020.
- [25] YOUTIPS, *LLAMA (Large Language Model - Meta AI)*.
- [26] MICROSOFT, *Visual Studio Code – Documentation*.
- [27] WAMPSEVER, *Serveur de développement Web*.
- [28] M. FOUNDATION, *Site officiel de MariaDB*.
- [29] Modelio, *Modelio Open Source Modeling Environment*.
- [30] P. S. FOUNDATION, *Tutoriel Python 3 – Documentation officielle*.
- [31] P. D. GROUP, *Introduction au langage PHP – Manuel officiel*.
- [32] T. TEMPLIER et A. GOUGEON, *JavaScript pour le Web 2.0 : Programmation objet, DOM, Ajax, Prototype, Dojo, Script.aculo.us, Rialto...*, Eyrolles éd., 2007.
- [33] M. AI, *LLaMA 4 : Intelligence multimodale*.
- [34] T. ZHANG, *BERTScore: Evaluating Text Generation with BERT*, 2019.
- [35] G. Research, *BERT base multilingual cased*, 2018.
- [36] I. ALMANACH, *CamemBERT Base pour le français*, 2020.
- [37] Lucidchart, *Diagramme de classes UML*.
- [38] L. Zhu et D. Luo, «A Novel Efficient and Effective Preprocessing Algorithm for Text Classification,» 13 03 2023. [En ligne]. Available: https://www.researchgate.net/publication/369247298_A_Novel_Efficient_and_Effective_Preprocessing_Algorithm_for_Text_Classification.

Résumé

L'évolution technologique dans le domaine éducatif surtout avec L'IA est devenue crucial, notamment face à la quantité massive d'informations que les étudiants accumulent à partir des cours, surtout lorsqu'il y a plusieurs cours. Les étudiants recherchent constamment des résumés de ces cours afin de les comprendre avant de les aborder en détail. Dans ce projet nous avons proposé un modèle qui la base de traitement automatique de langage naturel TALN, visant à faciliter le résumé des cours, ce modèle est basé sur LLaMA qui est déjà affiné et entraîné sur les instructions humaines via Groq (API). L'objectif de ce projet est de permettre à l'étudiant de faire des résumés automatiques pour ces cours pour mieux comprendre les idées principales.

Mot clés : Technologie, Educatif, IA, Modèle, TALN, Groq, API, résumé automatique.

Abstract

Technological evolution in the educational field, particularly with AI, has become crucial, especially given the massive amount of information students accumulate from courses, particularly when they are enrolled in multiple courses. Students constantly seek summaries of these courses to understand them before delving into the details. In this project, we propose a model based on Natural Language Processing (NLP), aimed at facilitating course summarization. This model is based on LLaMA, which has already been fine-tuned and trained on human instructions via Groq (API). The objective of this project is to enable students to generate automatic summaries of their courses to better grasp the main ideas.

Keywords: Technology, Education, AI, Model, NLP, Groq, API, Automatic Summarization.

ملخص

تطورت التكنولوجيا في المجال التعليمي، وبالأخص مع الذكاء الاصطناعي، لتصبح أمرًا حاسمًا، خاصة مع الكم الهائل من المعلومات التي يتراكمها الطلاب من الدروس، وخاصة عندما يكونون مسجلين في عدة دروس. يبحث الطلاب باستمرار عن ملخصات لهذه الدروس لفهمها قبل الخوض في تفاصيلها. في هذا المشروع، نقترح نموذجًا يعتمد على معالجة اللغة الطبيعية (NLP)، بهدف تسهيل تلخيص الدروس. يعتمد هذا النموذج على LLaMA، الذي تم تهيئته وتدريبه مسبقًا على التعليمات البشرية عبر Groq (API). الهدف من هذا المشروع هو تمكين الطلاب من إنشاء ملخصات آلية لدروسهم لفهم الأفكار الرئيسية بشكل أفضل.

الكلمات المفتاحية: التكنولوجيا، التعليم، الذكاء الاصطناعي، النموذج، معالجة اللغة الطبيعية، NLP، Groq، API، التلخيص الآلي.