

Université Cadi Ayyad
École Supérieure De Technologie-Safi
Département : Informatique
Filière : Génie Informatique

Rapport de stage

Développement d'une application mobile éducative basée sur l'intelligence artificielle

Réalisé par :

Mlle. RGUIBI Farah
Mlle. ASSAF Aya

Encadré par :

M. EL ABDELLAOUI Said
Mme. ELKOURCHI Asmaa

ANNÉE UNIVERSITAIRE : 2025/2026

Introduction

Dans un contexte marqué par l'essor des technologies numériques et l'évolution constante des méthodes d'apprentissage, le recours à des solutions éducatives intelligentes devient de plus en plus essentiel. L'intégration de l'intelligence artificielle (IA) dans le domaine de l'éducation offre de nouvelles perspectives pour accompagner les étudiants dans leur parcours d'apprentissage, de manière personnalisée, interactive et efficace.

C'est dans cette optique que s'inscrit le projet réalisé durant ce stage, qui a consisté à développer une application web/mobile éducative intelligente. Cette plateforme vise à offrir aux étudiants un environnement d'apprentissage complet, combinant plusieurs fonctionnalités innovantes telles qu'un chatbot conversationnel, un générateur automatique de QCM à partir de documents, un outil de traduction multilingue, et un module de génération de résumés.

Le développement de ce projet a mobilisé des compétences techniques variées, allant de la conception d'une interface utilisateur fluide avec **React Native**, à la mise en place d'une architecture backend robuste et modulaire en **Flask** avec gestion de base de données via **SQLAlchemy**. L'accent a été mis sur la modularité, l'ergonomie, la fiabilité et l'évolutivité du système afin de garantir une expérience utilisateur optimale.

Ce rapport présente, dans un premier temps, le contexte et les objectifs du projet. Il décrit ensuite les différentes étapes de sa réalisation, les choix technologiques adoptés, ainsi que les défis rencontrés. Enfin, une réflexion sur les perspectives d'amélioration et les apports personnels de ce stage viendra clore ce travail.

Dédicace

Je dédie ce travail, en premier lieu, à mes parents, dont l'amour inconditionnel, le soutien sans faille et les innombrables sacrifices ont été la base solide de tout ce que j'ai pu entreprendre jusqu'à aujourd'hui. Merci pour votre patience, vos encouragements et votre confiance en moi, même dans les moments de doute. Votre présence a été ma plus grande force, et c'est grâce à vous que j'ai pu avancer avec détermination.

J'adresse également ma profonde gratitude à mes amis, pour leur présence rassurante, leur écoute, leurs conseils et leur bonne humeur. Vous avez su m'apporter le soutien moral nécessaire pour garder le cap, et rendre ce parcours plus doux et plus motivant.

Enfin, je me dédie aussi ce travail. Pour ma persévérance, pour les nuits blanches, pour les moments de remise en question, et pour ne pas avoir abandonné. Ce stage représente bien plus qu'une simple étape académique : c'est une preuve que je suis capable d'aller au bout de mes objectifs. Et ce n'est que le début.

Remerciements

Je tiens à exprimer ma gratitude à toutes les personnes qui ont contribué à la réussite de ce stage.

Je tiens à exprimer ma gratitude à toutes les personnes qui ont contribué à la réussite de ce stage. Je tiens à exprimer avec un grand plaisir et un grand respect toute ma reconnaissance à mon encadrant **Monsieur EL ABDELLAOUI Said** et **Madame ELKOURCHI Asmaa** pour leur attention, leur orientation et leur aide pendant la réalisation de ce travail, ainsi que pour être une source d'information et de communication, n'hésitant jamais à consacrer une part de leur temps précieux.

Je souhaite également remercier ma collègue **Farah RGUIBI**, avec qui j'ai partagé ce stage. Merci pour ton engagement, ton sérieux, ton esprit d'équipe et ta bonne humeur. Travailler à tes côtés a été à la fois enrichissant et motivant.

Enfin, j'exprime toute ma gratitude à **mes parents et mes proches** pour leur soutien moral et matériel inestimable, et pour avoir toujours cru en moi.

Table des matières

Introduction	1
Dédicace	2
Remerciements	3
1 Présentation general du projet	8
1 Introduction	9
2 Contexte général	9
2.1 Problématique	9
2.2 Objectifs du projet	9
2.3 Solution Proposée	9
3 Cahier des charges	10
3.1 Besoins Fonctionnels	10
3.1.1 Application Éducative Intelligente	10
3.1.2 Fonctionnalités par Profil Utilisateur	10
3.2 Besoins Techniques	11
3.2.1 Architecture MVC	11
3.2.2 Frontend (interface utilisateur)	11
3.2.3 Backend (logique de l'application)	12
3.2.4 Base de données	12
3.2.5 Sécurité & gestion des utilisateurs	12
3.2.6 Extensibilité et maintenance	12
4 Choix des modèles IA	13
2 l'Intelligence Artificielle (IA)	15
1 Introduction	16
2 l'Intelligence Artificielle (IA)	16
3 Signification et objectifs de l'IA	16
4 Vue d'ensemble des techniques d'IA	17
4.1 Machine Learning (Apprentissage automatique)	17
4.2 Deep Learning (Apprentissage profond)	18
4.3 Traitement Automatique du Langage Naturel (NLP)	19
5 Conclusion	20
3 Analyse et Conception	21

1	Introduction	22
2	Diagramme de Cas d'Utilisation	22
2.1	Enseignant	22
2.2	Étudiant	22
4	Architecture utilisée	24
1	Introduction	25
2	Structure du backend	25
3	Fonctionnement de l'API	26
4	Structure du frontend	27
5	Conclusion	27
5	Outils & Environnement de Travail	28
1	Introduction	29
2	Langages de Programmation	29
2.1	JavaScript	29
2.2	Python	29
3	Frameworks et bibliothèques	30
3.1	Flask	30
3.2	Expo	30
3.3	React Native	31
3.4	Gradio	31
3.5	Hugging Face	31
4	Environnements de Développement	32
4.1	Visual Studio Code	32
4.2	Astah	32
4.3	XAMPP	32
4.4	MySQL Workbench	33
4.5	Google colab	33
5	Conclusion	33
6	Réalisation	34
1	Introduction	35
2	Implementation	35
2.1	Installation des bibliothèques nécessaires	35
2.2	Chargement et préparation des ressources linguistiques	35
2.3	Intégration de l'API Bard	35
2.4	Extraction de texte depuis un PDF	35
2.5	Nettoyage et segmentation du texte	36
2.6	Recherche d'information par mot-clé	36
2.7	Traduction automatique du texte	36
2.8	Résumé automatique du contenu PDF	36
2.9	Génération de QCM basés sur le contenu	37
	Perspectives	39
	Conclusion	40

Table des figures

2.1	l'Intelligence Artificielle	17
2.2	L'ensemble des techniques d'IA	17
2.3	Apprentissage automatique	18
2.4	Apprentissage profond	19
2.5	Traitement Automatique du Langage Naturel	20
3.1	Diagramme de Cas d'Utilisation	23
4.1	Structure du backend	25
4.2	Fonctionnement de l'API	26
4.3	Structure du frontend	27
5.1	JavaScript	29
5.2	Python	29
5.3	Flask	30
5.4	Expo	30
5.5	React Native	31
5.6	Gradio	31
5.7	Hugging Face	31
5.8	Visual Studio Code	32
5.9	Astah	32
5.10	XAMPP	32
5.11	MySQL Workbench	33
5.12	Google colab	33
6.1	Processus d'interaction utilisateur avec le fichier PDF	35
6.2	Interface de traduction du fichier PDF téléversé	36
6.3	Résumé du fichier PDF téléversé.	37
6.4	Génération des questions QCM à partir du fichier PDF	37
6.5	Interaction avec l'utilisateur pour poser des questions générales	37

Chapitre 1

Présentation general du projet

1 Introduction

Ce premier chapitre a pour objectif de présenter le cadre général dans lequel s'inscrit le projet de stage. Il permet de contextualiser la problématique abordée, de définir les objectifs poursuivis, ainsi que de justifier l'importance du sujet dans un environnement éducatif en pleine transformation numérique.

Face aux nouveaux besoins des apprenants et aux défis posés par l'évolution rapide des technologies, il devient essentiel d'explorer des approches innovantes capables de rendre l'apprentissage plus accessible, interactif et adapté à chacun. Ce chapitre se propose donc de poser les bases nécessaires à la compréhension du projet, en mettant en lumière les enjeux, les motivations et la démarche adoptée.

2 Contexte général

2.1 Problématique

Malgré la diversité des ressources pédagogiques disponibles en ligne, de nombreux étudiants rencontrent encore des difficultés à organiser efficacement leur apprentissage, à comprendre certains contenus complexes, ou à interagir avec les documents de cours de manière active. De plus, les outils existants sont souvent fragmentés : un pour résumer, un autre pour traduire, un troisième pour générer des quiz, etc., ce qui nuit à l'efficacité et à l'expérience utilisateur.

2.2 Objectifs du projet

- Générer automatiquement des supports éducatifs à partir de documents PDF.
- Intégrer un chatbot éducatif interactif basé sur des modèles NLP.
- Proposer des QCM adaptés et des fiches de révision.
- Mettre en place un système de répétition espacée pour faciliter la mémorisation.
- Déployer une interface web et mobile intuitive.
- Suivre la progression des utilisateurs (enseignants et apprenants).

2.3 Solution Proposée

Pour répondre aux besoins identifiés dans le contexte de l'apprentissage moderne, une solution éducative intelligente a été conçue sous la forme d'une application web/mobile. Cette plateforme vise à centraliser et simplifier l'expérience d'apprentissage des étudiants en regroupant plusieurs fonctionnalités utiles dans un seul outil interactif.

La solution se compose principalement de quatre modules :

- **Chatbot IA**
- **Génération de QCM**
- **Traduction multilingue**
- **Résumé de documents**

3 Cahier des charges

3.1 Besoins Fonctionnels

3.1.1 Application Éducative Intelligente

La solution proposée prend la forme d'une application web/mobile éducative, interactive et intelligente offrant :

- Une interface responsive, fluide et compatible avec tous les types d'appareils (smartphone, tablette, ordinateur).
- Une navigation modulaire facilitant l'accès rapide aux fonctionnalités principales (Chatbot, QCM, Traduction, Résumé).
- Un historique centralisé pour chaque module, permettant aux utilisateurs de retrouver facilement leurs contenus passés.
- Des interactions dynamiques avec un assistant IA capable de comprendre et de traiter du texte ou des documents.
- Un système de suivi pédagogique, permettant aux enseignants de consulter les QCM générés par les élèves et d'analyser leur progression.
- Une personnalisation de l'expérience selon le profil de l'utilisateur (étudiant ou enseignant).

3.1.2 Fonctionnalités par Profil Utilisateur

Chatbot IA

L'utilisateur (principalement l'étudiant) peut :

- Démarrer une nouvelle session de discussion avec l'assistant intelligent.
- Envoyer des questions sous forme de texte ou transmettre des documents PDF.
- Recevoir des réponses contextualisées en fonction du contenu transmis.
- Accéder à l'historique complet de ses conversations.
- Reprendre une conversation précédente avec conservation de l'historique.
- Supprimer une ou plusieurs conversations de son espace personnel.
- Visualiser les échanges sous forme de messages clairs et ordonnés chronologiquement.

Module QCM (Quiz à Choix Multiples)

L'utilisateur (principalement l'étudiant) peut :

▷ Côté étudiant :

- Générer un QCM automatiquement à partir d'un fichier PDF.
- Choisir le nombre de questions et nommer le quiz.
- Visualiser le QCM avec les questions et les propositions de réponses.
- Identifier la bonne réponse sans surbrillance pour un apprentissage actif.
- Consulter l'historique des QCM déjà générés.
- Relancer un QCM pour le revoir ou le refaire.
- Supprimer des QCM de son historique.

▷ **Côté enseignant :**

- Importer des cours (en PDF ou texte) pour créer des QCM destinés aux élèves.
- Gérer une base de QCM par classe, par niveau ou par thème.
- Assigner des QCM à un groupe d'étudiants.
- Suivre les résultats et la progression des élèves (statistiques, réponses correctes/incorrectes...).

Module de Traduction

L'utilisateur peut :

- Traduire du texte brut ou un document PDF en plusieurs langues.
- Choisir la langue source et la langue cible.
- Afficher côte à côte le texte original et sa traduction.
- Consulter l'historique de toutes ses traductions.
- Recharger une traduction pour la revoir ou la réutiliser.
- Supprimer une ou plusieurs traductions.

Module de Résumé

L'utilisateur (principalement l'étudiant) peut :

- Démarrer une nouvelle session de discussion avec l'assistant intelligent.
- Envoyer des questions sous forme de texte ou transmettre des documents PDF.
- Recevoir des réponses contextualisées en fonction du contenu transmis.
- Accéder à l'historique complet de ses conversations.
- Reprendre une conversation précédente avec conservation de l'historique.
- Supprimer une ou plusieurs conversations de son espace personnel.
- Visualiser les échanges sous forme de messages clairs et ordonnés chronologiquement.

3.2 Besoins Techniques

Pour réaliser cette application éducative intelligente, plusieurs outils et technologies sont nécessaires afin d'assurer un bon fonctionnement, une bonne organisation du code et une facilité d'évolution.

3.2.1 Architecture MVC

Le projet est réalisé suivant l'architecture MVC (Modèle-Vue-Contrôleur), il nous permet de bien séparer la logique de notre application (backend), la partie présentation (frontend) et l'accès et la gestion de nos données (base de données). Cela nous facilite la tâche pour toute évolution et la maintenance du projet.

3.2.2 Frontend (interface utilisateur)

Le projet site est réalisé avec React Native, il est entièrement responsive et adaptable à tous les supports (ordinateur, tablette, smartphone). Il utilise des hooks personnalisés pour gérer les fonctionnalités principale comme par exemple le chatbot, les QCM, la recherche de traduction ou encore tout simplement l'affichage de résumé.

3.2.3 Backend (logique de l'application)

Le serveur est créé à partir de Flask. Ce serveur expose bien des service REST. Chaque fonctionnalité comme le chatbot, QCM, traduction... a sa petite place et est bien divisée. Le code suit aussi une organisation en couches : routes, contrôleurs, cas d'usage (use cases), et accès aux données (repositories).

3.2.4 Base de données

On utilise le module SQLAlchemy pour gérer la base de donnée. Cette base contiendra bien évidemment les utilisateurs, les conversations, les exercices, les traductions et les résumés. Les relations entre nos entités (tables) sont bien définies, et la suppression en cascade est aussi prise en charge.

3.2.5 Sécurité & gestion des utilisateurs

La gestion des utilisateurs est propre (compte utilisateur, authentification, profil...), les données sont bien sûres isolées entre les utilisateurs, et bien entendu ont n'a pas oublié de sécuriser chaque accès à n'importe qu'elle fonctionnalité.

3.2.6 Extensibilité et maintenance

Grâce à son architecture MVC, les évolutions sur ce projet sont facilitées. Le code est bien agencé, très commenté et prêt à recevoir de nouveaux modules dans le futur.

4 Choix des modèles IA

Nous avons opté pour plusieurs modèles de traitement automatique du langage naturel (NLP), chacun spécialisé dans une tâche précise.

BART (Facebook AI)

Utilisation : Résumé automatique de documents.

Avantages :

- Excellente qualité de résumé grâce à l'architecture encoder-decoder.
- Open-source, pré-entraîné sur de grands corpus.
- Bon compromis entre précision et vitesse.

Inconvénients :

- Relativement lourd en ressources.
- Moins rapide que les modèles distillés.

DistilBERT (Hugging Face)

Utilisation : Questions-réponses (QéR), extraction d'informations.

Avantages :

- Plus rapide et plus léger que BERT.
- Performant pour les tâches de QéR.
- Facile à intégrer dans une API Flask.

Inconvénients :

- Moins précis que BERT sur certains jeux de données.
- Limitation sur la taille des textes d'entrée.

MarianMT (Helsinki-NLP)

Utilisation : Traduction automatique multilingue.

Avantages :

- Supporte de nombreuses paires de langues.
- Modèle optimisé pour la traduction.
- Léger et rapide.

Inconvénients :

- Moins fluide que certains modèles commerciaux (DeepL, Google).
- Moins adapté aux textes techniques ou spécialisés.

Bard (Google)

Utilisation : Génération avancée de questions pour QCM.

Avantages :

- Capacité à générer des questions logiques et pertinentes.

- Bonne compréhension contextuelle.
- Utile pour des QCM d'un niveau plus élevé ou personnalisé.

Inconvénients :

- Non open-source.
- Dépendance à une API externe.

Flan-T5 (Google)

Utilisation : Génération de texte, résumés, dialogue conversationnel.

Avantages :

- Entraîné sur de nombreuses consignes (instruction tuning).
- Bon générateur de texte polyvalent.
- Bonne compatibilité avec Hugging Face.

Inconvénients :

- Consomme beaucoup de mémoire.
- Moins spécialisé pour certaines tâches (ex : QéR).

Mistral (via OpenRouter + LangChain)

Utilisation : Réponses intelligentes à partir de documents PDF, génération de texte cohérente, conversation contextuelle avancée.

Avantages :

- Modèle léger et performant pour les tâches de génération et de raisonnement.
- Compatible avec LangChain pour les chaînes de traitement complexes.
- Très utile avec des embeddings pour répondre à partir de contexte PDF.
- Peut être utilisé via OpenRouter sans serveur local.

Inconvénients :

- Nécessite une API key tierce (OpenRouter).
- Moins précis que GPT-4 pour des questions très ouvertes.
- La qualité dépend fortement de la qualité des chunks extraits.

Chapitre 2

l'Intelligence Artificielle (IA)

1 Introduction

Ce chapitre présente les bases de l'intelligence artificielle (IA), ses objectifs et ses domaines d'application. Il met en lumière les techniques fondamentales telles que le machine learning, le deep learning et le traitement automatique du langage naturel (NLP). L'IA y est abordée comme une technologie capable d'imiter certaines fonctions humaines, ouvrant la voie à des systèmes plus autonomes, intelligents et adaptatifs.

2 L'Intelligence Artificielle (IA)

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à créer des systèmes capables d'imiter les capacités humaines telles que la compréhension du langage, l'apprentissage, le raisonnement, la résolution de problèmes ou encore la perception visuelle. Elle regroupe un ensemble de technologies permettant aux machines de prendre des décisions de manière autonome ou semi-autonome.

L'IA repose sur des données, des algorithmes et une puissance de calcul pour apprendre de l'expérience et s'améliorer au fil du temps. Aujourd'hui, elle est intégrée dans de nombreux domaines comme la santé, la finance, l'éducation, le commerce ou encore la sécurité.

3 Signification et objectifs de l'IA

Le terme '**intelligence artificielle**' désigne la capacité d'une machine, généralement un système informatique, à simuler certaines fonctions de l'intelligence humaine telles que l'apprentissage, le raisonnement, la perception ou la compréhension du langage. Il ne s'agit pas simplement d'automatiser des actions, mais de permettre aux machines de prendre des décisions, de s'adapter et d'apprendre de l'expérience.

Son objectif principal est de **résoudre efficacement des problèmes** en automatisant des tâches répétitives, prédictives ou nécessitant une analyse avancée. L'IA vise à améliorer la productivité, à optimiser les processus décisionnels, et à offrir des services intelligents dans de nombreux domaines.

Plus précisément, l'IA permet de :

- **Automatiser les tâches répétitives ou cognitives** : Elle peut remplacer ou assister l'humain dans des tâches chronophages telles que la saisie de données, la gestion d'inventaire ou l'analyse de documents.
- **Détecter des modèles ou des anomalies** : Grâce à l'analyse de grands volumes de données, l'IA peut identifier des tendances, corriger des erreurs, ou signaler des situations inhabituelles.
- **Fournir des recommandations personnalisées** : Les systèmes intelligents analysent les comportements des utilisateurs pour proposer des suggestions adaptées à leurs besoins ou préférences (comme sur les plateformes de streaming ou d'e-commerce).
- **Faciliter l'interaction homme-machine** : À travers la voix, le texte ou l'image, l'IA permet une communication plus naturelle entre l'utilisateur et la machine (ex. : assistants vocaux, chatbots, traduction automatique).
- **Apprendre et s'adapter à de nouvelles données ou environnements** : Contrairement aux systèmes traditionnels, les modèles d'IA peuvent évoluer, améliorer leur performance et s'ajuster à des contextes changeants.

Ainsi, l'IA ne se limite pas à exécuter des instructions ; elle apprend, s'améliore et devient capable de généraliser ses connaissances à de nouveaux contextes.

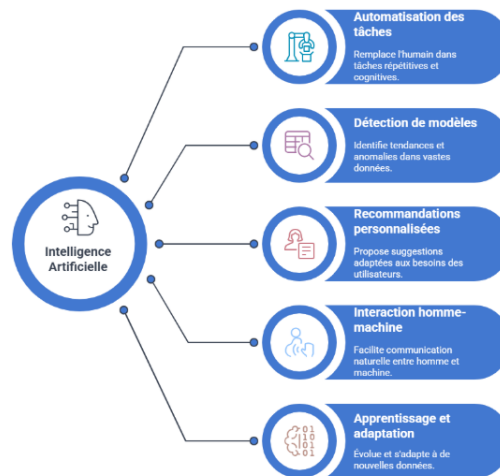


FIGURE 2.1 – l’Intelligence Artificielle

4 Vue d’ensemble des techniques d’IA

Les techniques d’IA se basent sur différents sous-domaines, parmi lesquels le Machine Learning, le Deep Learning et le NLP. Ces approches permettent de traiter des données complexes, de reconnaître des motifs et de prendre des décisions automatisées ou assistées.

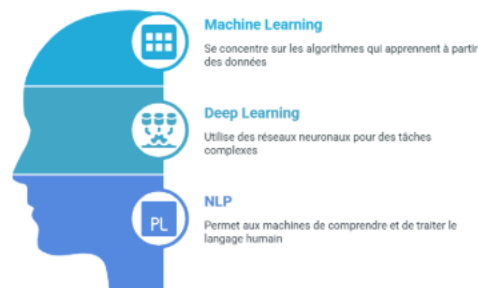


FIGURE 2.2 – L’ensemble des techniques d’IA

4.1 Machine Learning (Apprentissage automatique)

Le Machine Learning est une branche de l’IA qui permet à un système d’apprendre à partir de données sans être explicitement programmé. Le système identifie des patterns dans les données d’entraînement et les utilise pour faire des prédictions ou prendre des décisions sur de nouvelles données.

▷ Apprentissage supervisé

Dans ce type d’apprentissage, le modèle est entraîné sur un jeu de données étiquetées. Chaque entrée est associée à une sortie connue (label). Exemples : détection de spam, reconnaissance d’images, diagnostic médical assisté.

▷ Apprentissage non supervisé

Ici, les données ne sont pas étiquetées. Le système essaie de découvrir seul des structures ou des regroupements naturels dans les données. Exemples : regroupement de clients (clustering), réduction de dimensions, détection d'anomalies.

▷ Apprentissage par renforcement

Le système apprend par essais et erreurs. À chaque action, il reçoit une récompense ou une pénalité, ce qui lui permet d'optimiser son comportement au fil du temps. Exemples : jeux vidéo, robotique, agents autonomes.

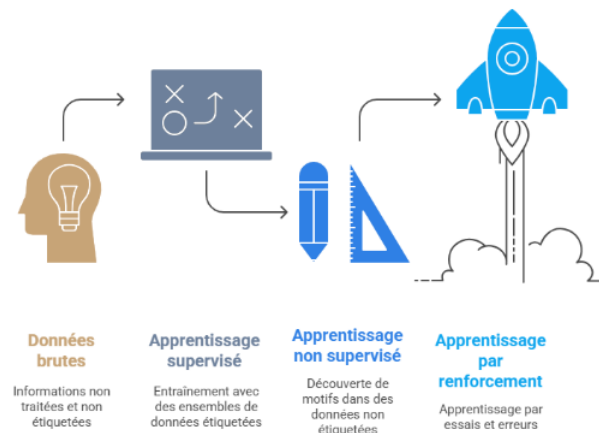


FIGURE 2.3 – Apprentissage automatique

4.2 Deep Learning (Apprentissage profond)

Le Deep Learning est une forme avancée de Machine Learning, utilisant des réseaux de neurones artificiels à plusieurs couches pour apprendre des représentations complexes à partir de grandes quantités de données. Il est particulièrement performant sur les données non structurées.

▷ Réseaux de neurones convolutifs (CNN)

Utilisés principalement dans le traitement d'images, ces réseaux détectent automatiquement les motifs visuels (formes, textures, objets) et sont à la base de nombreuses applications en vision par ordinateur.

▷ Réseaux récurrents (RNN) et LSTM

Adaptés à l'analyse de séquences temporelles (texte, audio, séries temporelles), les RNN permettent de mémoriser des informations passées, tandis que les LSTM améliorent la mémoire à long terme pour un meilleur traitement contextuel.

▷ Transformers

Cette architecture récente est au cœur des modèles de traitement du langage comme GPT, BERT ou T5. Elle repose sur des mécanismes d'attention pour capturer efficacement les dépendances dans les séquences de texte.

▷ GANs (Generative Adversarial Networks)

Les GANs opposent deux réseaux (générateur et discriminateur) pour créer des contenus réalistes, comme des images, des musiques, ou des vidéos. Ils sont utilisés dans la création artistique, la simulation, et la génération de données.

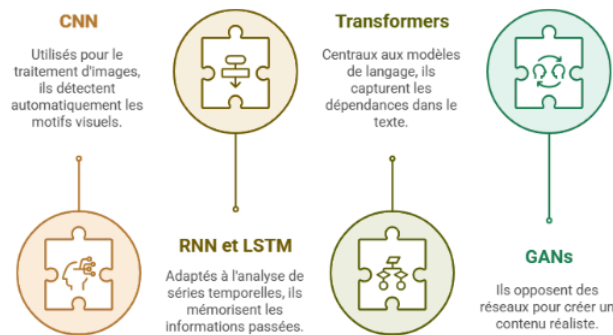


FIGURE 2.4 – Apprentissage profond

4.3 Traitement Automatique du Langage Naturel (NLP)

Le NLP permet aux machines de comprendre, analyser, générer et manipuler le langage humain. Il combine des techniques linguistiques, statistiques et neuronales pour rendre les interactions homme-machine plus naturelles et efficaces.

▷ Tokenisation et analyse grammaticale

La tokenisation découpe le texte en unités compréhensibles (mots, phrases). Ensuite, l'analyse grammaticale identifie les rôles syntaxiques (verbe, sujet, complément) pour permettre aux modèles de comprendre la structure linguistique.

▷ Résumé automatique de texte

Il permet de condenser un document tout en conservant ses idées clés. Les modèles de NLP peuvent extraire ou reformuler les phrases essentielles pour créer un résumé ciblé et cohérent.

▷ Traduction automatique

Grâce à des modèles multilingues puissants, le NLP peut traduire des phrases ou des documents d'une langue à une autre tout en préservant le sens et le ton. Cela facilite l'accès aux ressources éducatives dans différentes langues.

▷ Génération de texte

Les modèles de génération peuvent produire des textes entiers en réponse à une consigne, une question ou un sujet. C'est la technologie utilisée dans les assistants virtuels, les rédacteurs automatiques ou les chatbots.

▷ Représentations vectorielles de mots

Les mots sont transformés en vecteurs numériques permettant aux algorithmes de manipuler le langage mathématiquement. Word2Vec, GloVe, BERT et d'autres techniques permettent de capturer les relations sémantiques entre les mots.

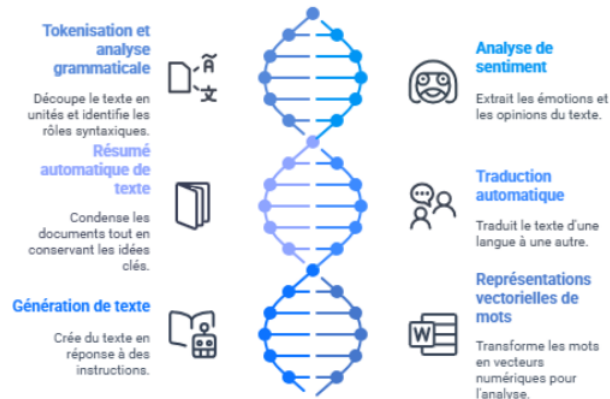


FIGURE 2.5 – Traitement Automatique du Langage Naturel

5 Conclusion

Ce chapitre a permis de découvrir les principaux concepts et techniques liés à l'intelligence artificielle. Il a montré comment l'IA peut analyser des données, apprendre de l'expérience, interagir avec le langage humain et s'adapter à divers contextes. Ces avancées témoignent de son rôle croissant dans l'innovation technologique et dans la transformation de nombreux secteurs.

Chapitre 3

Analyse et Conception

1 Introduction

2 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation présente les interactions clés entre les deux profils d'utilisateurs de l'application mobile éducative – l'Enseignant et l'Étudiant. Il met en évidence les fonctionnalités attribuées à chacun selon son rôle, tout en soulignant l'importance de l'authentification préalable. Cette dernière conditionne l'accès aux différentes fonctionnalités afin d'assurer la sécurité des données et la traçabilité des actions réalisées

2.1 Enseignant

L'Enseignant a pour rôle principal la création de contenus pédagogiques et le suivi de la progression des apprenants. Il interagit avec le système principalement à travers les fonctionnalités suivantes :

- **Créer du contenu pédagogique** : Ajouter des documents, ressources ou supports de cours à la plateforme.
- **Suivre la progression des élèves** : Accéder aux statistiques de réussite, de participation et aux résultats des évaluations des étudiants.
- **Soumettre un PDF de cours** : Télécharger un fichier PDF contenant un cours, afin qu'il soit consultable par les étudiants.

Toutes ces actions nécessitent que l'enseignant soit authentifié dans le système.

2.2 Étudiant

L'Étudiant utilise la plateforme pour accéder aux ressources pédagogiques, interagir avec un chatbot, et s'autoévaluer à travers des quiz. Les principales fonctionnalités auxquelles il a accès sont :

- **Rechercher dans les anciens cours** : Parcourir les cours déjà disponibles dans la base de données du système.
- **Passer un quiz chronométré** : Tester ses connaissances via des évaluations à temps limité.
- **Poser des questions au chatbot** : Obtenir des réponses rapides sur le contenu pédagogique via une IA conversationnelle.
- **Générer un résumé** : Résumer automatiquement un contenu pédagogique, qu'il s'agisse d'un fichier PDF ou d'un simple texte.
- **Traduire un contenu** : Traduire un texte ou un cours, qu'il soit saisi manuellement ou extrait d'un document, vers une autre langue pour en faciliter la compréhension.
- **Faire des exercices générés** : Accéder à des exercices créés automatiquement à partir des cours disponibles.

Toutes ces fonctionnalités sont conditionnées par l'authentification de l'étudiant.

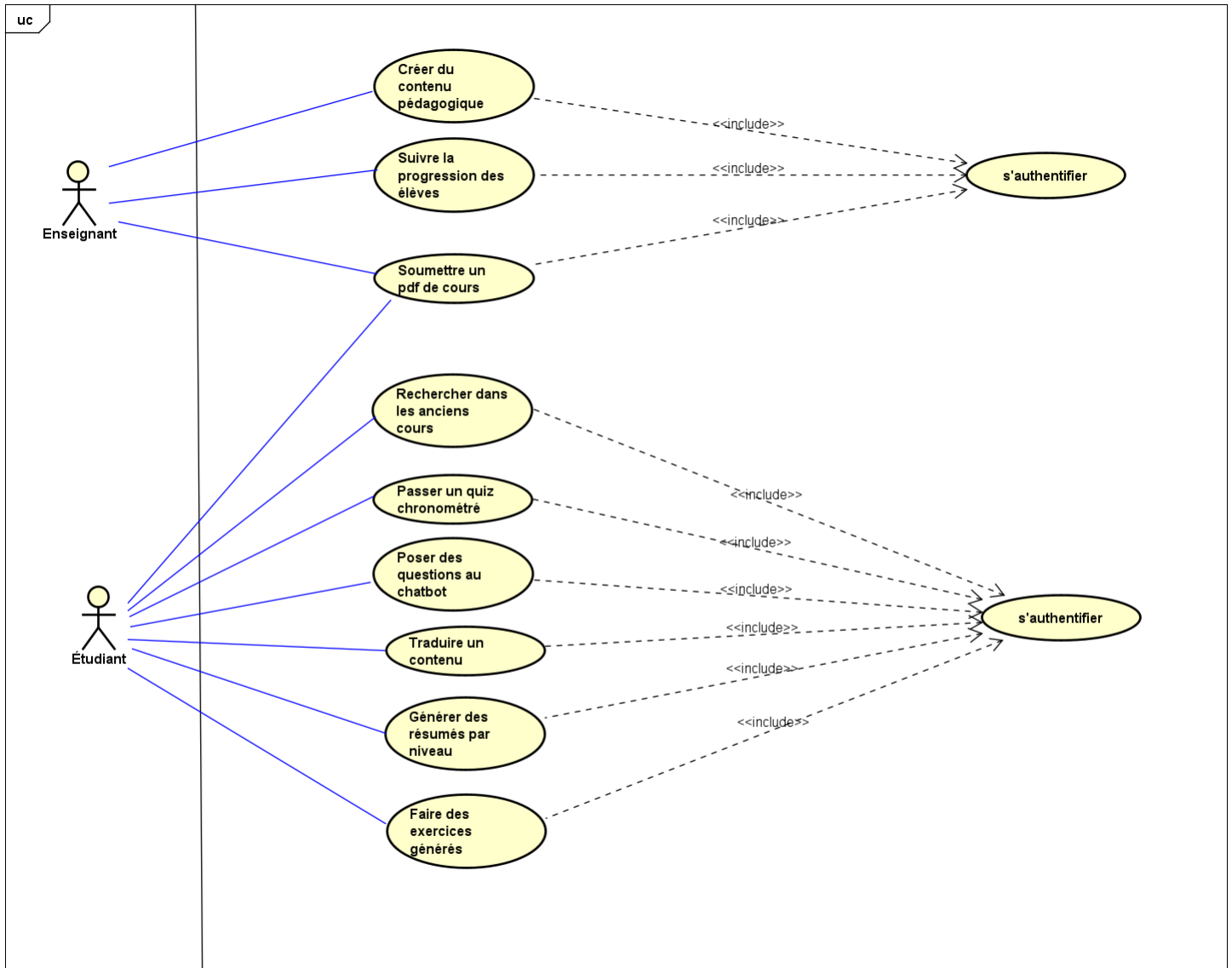


FIGURE 3.1 – Diagramme de Cas d'Utilisation

Chapitre 4

Architecture utilisée

1 Introduction

Cette section décrit l'architecture technique adoptée pour le développement de notre application mobile éducative basée sur l'intelligence artificielle. Elle couvre à la fois la structure du backend, qui assure le traitement des données et la logique métier, et celle du frontend, responsable de l'interface utilisateur et des interactions. L'objectif est de garantir une communication fluide et efficace entre les deux parties grâce à une API bien structurée.

2 Structure du backend

Dans le développement de notre application, nous avons adopté une architecture modulaire, en suivant une structure bien organisée composée des éléments suivants :

- **Modèles** : Représentent la structure des données manipulées par l'application.
- **Repositories** : Responsables des interactions avec la base de données.
- **Routes** : Permettent de définir les différentes entrées de l'API et de les lier aux contrôleurs correspondants.
- **Middlewares** : Utilisés pour intercepter et traiter les requêtes HTTP avant qu'elles n'atteignent les contrôleurs (par exemple, pour l'authentification ou la gestion des erreurs).
- **Contrôleurs** : Contiennent la logique de traitement des requêtes et appellent les services.
- **Services** : Regroupent la logique métier de l'application.
- **Cas d'utilisation** : Décrit des scénarios d'interaction entre l'utilisateur et l'application, orchestrant les services nécessaires pour répondre aux besoins fonctionnels.

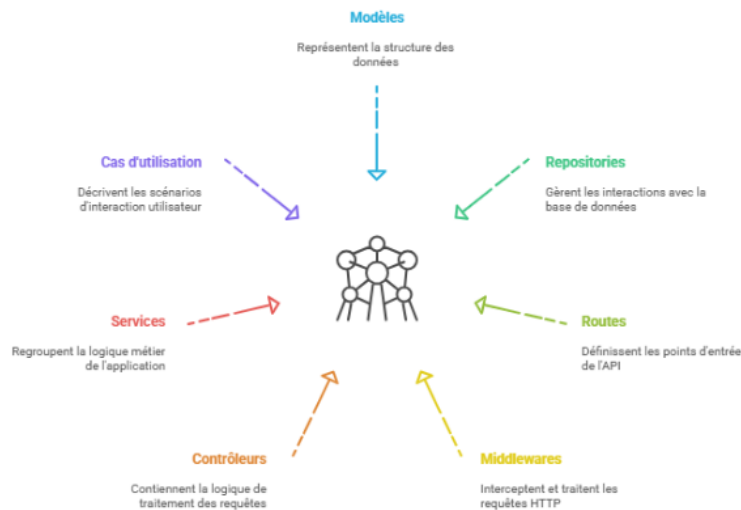


FIGURE 4.1 – Structure du backend

3 Fonctionnement de l'API

L'API de l'application permet la communication entre le frontend et le backend, en facilitant l'échange de données à travers des requêtes HTTP. Voici les étapes principales de son fonctionnement :

1. **Requête HTTP** : Le client (frontend) envoie une requête HTTP vers l'API, en spécifiant une méthode (GET, POST, PUT, DELETE) et une route (URL).
2. **Traitement de la requête** : La requête est capturée par les routes définies dans l'API. Chaque route est associée à un contrôleur qui va traiter la demande.
3. **Contrôleur** : Le contrôleur exécute la logique métier nécessaire, comme récupérer des données, les mettre à jour ou en créer de nouvelles.
4. **Réponse HTTP** : Après le traitement de la requête, l'API renvoie une réponse HTTP au frontend, généralement sous forme de données JSON, contenant les informations demandées ou un message de statut (succès, erreur, etc.).
5. **Gestion des erreurs** : Si une erreur se produit à n'importe quelle étape, l'API renvoie une réponse d'erreur appropriée, souvent avec un code d'état HTTP et un message descriptif.



FIGURE 4.2 – Fonctionnement de l'API

Dans notre projet, nous avons lié le front-end et le back-end à travers des appels API réalisés depuis le front-end avec React Native vers le back-end qui utilise Flask. Le front-end envoie des requêtes HTTP aux points de terminaison définis dans le back-end pour récupérer ou envoyer des données. Ces requêtes sont gérées par des routes dans Flask qui exécutent la logique de traitement nécessaire et renvoient les données sous forme de réponses JSON au front-end.

Les composants du front-end, tels que les formulaires ou les tableaux, utilisent des fonctions comme 'fetch' ou des bibliothèques comme 'Axios' pour effectuer ces appels API. Une fois les données reçues, elles sont ensuite utilisées pour mettre à jour l'interface utilisateur de manière dynamique, offrant ainsi une interaction fluide entre l'utilisateur et le système.

Cette intégration entre le front-end et le back-end garantit une communication efficace et rapide, facilitant la gestion des données et la mise à jour en temps réel de l'interface utilisateur.

4 Structure du frontend

L'architecture du frontend a été conçue pour respecter une organisation modulaire et claire, facilitant le développement, la maintenance et l'évolutivité. Le projet est structuré autour de plusieurs dossiers principaux :

- **App** : contient les différentes pages ou écrans de l'application, comme l'accueil, le détail des traductions, etc.
- **Components** : regroupe tous les composants réutilisables (boutons, entêtes, cartes, etc.).
- **Assets** : contient les ressources statiques telles que les images, icônes et polices.
- **Hooks** : stocke les hooks personnalisés permettant de centraliser certaines logiques métiers réutilisables.
- **Styles** : regroupe les fichiers de styles globaux pour assurer une cohérence visuelle à travers toute l'application.



FIGURE 4.3 – Structure du frontend

5 Conclusion

En résumé, l'architecture de l'application repose sur une séparation claire entre le frontend et le backend, assurant ainsi modularité, évolutivité et maintenabilité. Grâce à l'utilisation de Flask pour le backend et React Native pour le frontend, combinés à des échanges de données via une API, l'application offre une expérience utilisateur fluide et dynamique, tout en garantissant un traitement efficace des données.

Chapitre 5

Outils & Environnement de Travail

1 Introduction

Cette section présente les outils et technologies utilisés pour le développement d'une application mobile éducative basée sur l'intelligence artificielle. Ces éléments sont essentiels à la mise en œuvre des différentes fonctionnalités et contribuent à assurer le bon fonctionnement de l'application.

2 Langages de Programmation

2.1 JavaScript

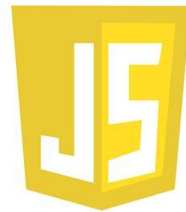


FIGURE 5.1 – JavaScript

Langage de programmation interprété, principalement utilisé pour le développement web côté client.

2.2 Python

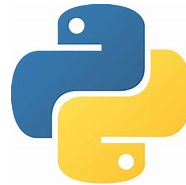


FIGURE 5.2 – Python

Python est un langage polyvalent, facile à apprendre, utilisé dans le développement web, l'IA et l'analyse de données. Sa syntaxe claire et ses bibliothèques permettent un développement rapide.

3 Frameworks et bibliothèques

3.1 Flask



FIGURE 5.3 – Flask

Flask est un micro-framework Python pour le développement web, léger et flexible, offrant les bases essentielles et permettant d'ajouter facilement des extensions selon les besoins du projet.

3.2 Expo

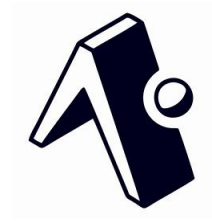


FIGURE 5.4 – Expo

Expo est un framework open-source pour React Native, simplifiant le développement, le déploiement et les tests d'applications mobiles.

3.3 React Native



FIGURE 5.5 – React Native

React Native est un framework open-source permettant de développer des applications mobiles en utilisant JavaScript et React, offrant des performances natives tout en partageant une grande partie du code entre les plateformes iOS et Android.

3.4 Gradio



FIGURE 5.6 – Gradio

Gradio est un framework open-source qui permet de créer facilement des interfaces web interactives pour tester ces modèles en temps réel.

3.5 Hugging Face



FIGURE 5.7 – Hugging Face

Hugging Face est une plateforme open-source pour le traitement du langage naturel, offrant des bibliothèques et des modèles pré-entraînés pour des tâches comme la génération et la traduction de texte.

4 Environnements de Développement

4.1 Visual Studio Code



FIGURE 5.8 – Visual Studio Code

Un éditeur de code source léger, open-source, et extensible, utilisé pour écrire et déboguer du code dans de nombreux langages de programmation, dont JavaScript, Python, et C#.

4.2 Astah



FIGURE 5.9 – Astah

Astah est un outil de modélisation UML qui permet de créer des diagrammes de conception logicielle et de générer du code.

4.3 XAMPP



FIGURE 5.10 – XAMPP

Un package open-source qui permet d'installer Apache, MySQL, PHP et Perl sur une machine locale. Il est utilisé principalement pour créer des environnements de développement local pour les applications web.

4.4 MySQL Workbench



FIGURE 5.11 – MySQL Workbench

Un outil graphique utilisé pour gérer les bases de données MySQL. Il permet la conception, le développement et l'administration des bases de données, ainsi que l'exécution de requêtes SQL.

4.5 Google colab



FIGURE 5.12 – Google colab

Google Colab est un environnement de développement gratuit basé sur le cloud, permettant d'exécuter du code Python directement dans le navigateur. Il est particulièrement utilisé pour le machine learning, l'analyse de données et l'exécution de notebooks Jupyter.

5 Conclusion

En résumé, les outils et environnements choisis ont permis d'assurer une coordination efficace entre les différentes couches de l'application, facilitant son développement, son déploiement et sa maintenance. Grâce à une combinaison cohérente de solutions modernes, nous avons pu répondre aux exigences fonctionnelles et techniques du projet tout en garantissant une expérience utilisateur fluide et intuitive.

Chapitre 6

Réalisation

1 Introduction

2 Implementation

2.1 Installation des bibliothèques nécessaires

Pour mettre en place notre système interactif d'analyse de documents PDF, nous avons commencé par installer un ensemble de bibliothèques Python essentielles à l'aide de la commande `!pip install`. Ces bibliothèques incluent `gradio` pour l'interface utilisateur, `PyPDF2` et `pymupdf` pour l'extraction de texte depuis les fichiers PDF, `nltk` pour le traitement du langage naturel, `transformers` et `torch` pour l'utilisation de modèles préentraînés, `python-dotenv` pour la gestion des variables d'environnement, et `bardapi` pour interagir avec l'API Bard de Google.

2.2 Chargement et préparation des ressources linguistiques

Afin de traiter le contenu textuel des documents PDF en français, nous avons utilisé la bibliothèque NLTK pour charger les *stopwords* (mots fréquents inutiles à l'analyse). Cette étape nous a permis de filtrer les mots non significatifs lors de l'analyse sémantique ou la recherche par mot-clé. Un mécanisme de téléchargement automatique est prévu si les ressources nécessaires ne sont pas encore installées.

2.3 Intégration de l'API Bard

Nous avons intégré l'API Bard afin d'enrichir les fonctionnalités de notre système, notamment pour la génération de réponses cohérentes, de résumés optimisés, ou encore de QCM. Une clé d'API est chargée dans les variables d'environnement pour activer cette fonctionnalité, tout en prévoyant un mode de repli si l'API n'est pas disponible.

2.4 Extraction de texte depuis un PDF

Nous avons développé une fonction robuste pour extraire le texte d'un fichier PDF. Elle prend en compte différents formats d'entrée selon le contexte d'utilisation (fichier local, téléchargement via interface, etc.). L'extraction utilise en priorité la bibliothèque `PyMuPDF` pour sa précision, avec un recours à `PyPDF2` si nécessaire. Le texte brut est ensuite nettoyé pour supprimer les caractères non pertinents.

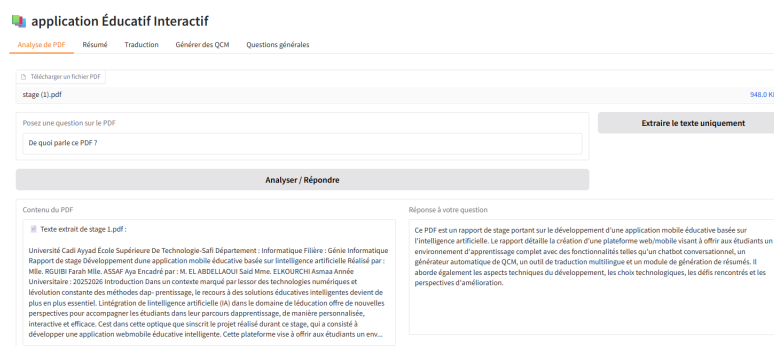


FIGURE 6.1 – Processus d'interaction utilisateur avec le fichier PDF

Dans cette partie, l'utilisateur téléverse un fichier PDF, le système en extrait le contenu, puis l'utilisateur saisit une question relative au document, à laquelle le modèle Bard fournit une réponse.

2.5 Nettoyage et segmentation du texte

Une fois le texte extrait, nous l'avons nettoyé à l'aide d'expressions régulières afin de supprimer les espaces inutiles et les caractères spéciaux. Nous avons ensuite segmenté ce texte en phrases pour faciliter l'analyse, la recherche par mot-clé et la génération de questions.

2.6 Recherche d'information par mot-clé

Lorsque l'API Bard n'est pas disponible, nous avons prévu une méthode alternative pour retrouver les informations pertinentes dans le PDF. Cette méthode consiste à extraire les mots-clés d'une question en ignorant les mots vides, puis à rechercher dans le texte les phrases contenant ces mots-clés. Les résultats les plus pertinents sont ensuite présentés à l'utilisateur.

2.7 Traduction automatique du texte

Nous avons ajouté une fonction de traduction pour convertir le texte entre différentes langues, en utilisant les modèles Helsinki-NLP proposés par Hugging Face. Le texte est divisé en phrases, traduites individuellement, afin de garantir la lisibilité et la fluidité du résultat final.

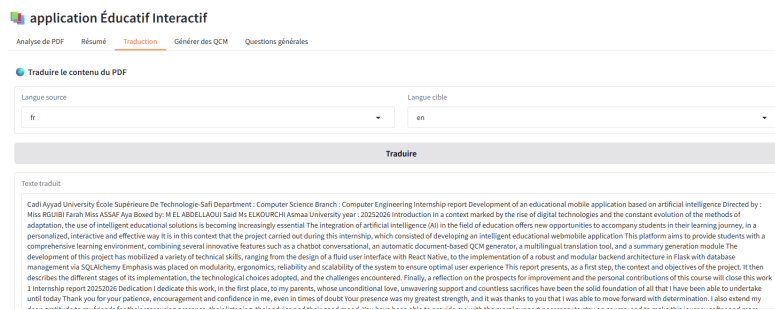


FIGURE 6.2 – Interface de traduction du fichier PDF téléversé

Dans cette partie, l'utilisateur traduit le fichier PDF qu'il a téléversé dans la première étape, en choisissant la langue souhaitée pour la traduction.

2.8 Résumé automatique du contenu PDF

Pour faciliter la compréhension rapide d'un document, nous avons mis en place une fonctionnalité de résumé automatique à l'aide du modèle facebook/bart-large-cnn. Le texte est découpé en segments, chacun étant résumé individuellement. Le résumé final peut être reformulé et amélioré via l'API Bard pour plus de fluidité et de cohérence.

application Éducatif Interactif

Analyse de PDF **Résumé** Traduction Générer des QCM Questions générales

Résumer le contenu du PDF

Longueur du résumé
Court (50-100 mots), Moyen (100-150 mots), Long (150-200 mots)

☐ short ☐ medium ☒ long

Générer un résumé

Résumé du document

Absolument l'ici une réorganisation du résumé pour une meilleure fluidité et cohérence :

Le traitement du langage naturel (NLP) est un domaine de l'intelligence artificielle qui permet aux machines de comprendre, d'analyser, de générer et d'interagir avec le langage humain. Cette technologie trouve des applications concrètes dans des domaines sérieux tels que la sécurité, l'éducation et la cybersécurité.

Par exemple, le NLP peut être utilisé pour :

- * Aider à l'analyse de messages suspects ou de menaces.
- * Générer des rapports de police à partir de données brutes.
- * Créer de faux messages de phishing à des fins de cybersécurité.
- * Développer des générateurs de dialogues pour l'entraînement de policiers.
- * Mettre en place des simulations d'abus pour détecter les incohérences dans les récits.

Comprendre le fonctionnement du NLP ouvre ainsi la voie à de nombreuses applications, y compris dans des contextes aussi importants que la sécurité et la protection.

FIGURE 6.3 – Résumé du fichier PDF téléversé.

Dans cette partie, l'utilisateur choisit la longueur du résumé (court, moyen, long) du fichier déjà téléversé, puis la réponse est affichée.

2.9 Génération de QCM basés sur le contenu

L'une des fonctionnalités principales de notre projet est la génération automatique de QCM à partir du contenu des fichiers PDF. En utilisant l'API Bard, notre système extrait des phrases significatives et demande à l'IA de formuler des questions pertinentes avec trois options de réponse. Nous avons également prévu une méthode simplifiée de génération en l'absence de Bard.

application Éducatif Interactif

Analyse de PDF Résumé Traduction **Générer des QCM** Questions générales

Générer des questions à choix multiples basées sur le contenu du PDF

Nombre de questions QCM

5

Générer des QCM

Questions générées

Question 1 :
Bien sûr, voici une question à choix multiple basée sur "22 4" :
Quelle opération mathématique pourrait relier le mieux les nombres 22 et 4 ?

A) Addition
B) Soustraction
C) Division

Réponse correcte : C) Division

Question 2 :
Bien sûr, voici une question à choix multiple basée sur la phrase : "31 Rapport de stage 2025/2026 4" :
Question : "

FIGURE 6.4 – Génération des questions QCM à partir du fichier PDF

Dans cette partie, l'utilisateur entre le nombre de questions QCM qu'il souhaite, puis les réponses sont affichées.

application Éducatif Interactif

Analyse de PDF Résumé Traduction Générer des QCM **Questions générales**

Poser une question générale

Qu'est-ce que l'intelligence artificielle ?

Poser la question

Réponse

L'intelligence artificielle (IA) est un domaine de l'informatique qui vise à créer des machines capables de simuler l'intelligence humaine. Cela inclut des capacités telles que l'apprentissage, le raisonnement, la résolution de problèmes, la perception et la prise de décision. En termes simples, l'IA permet aux ordinateurs d'effectuer des tâches qui nécessitent normalement l'intelligence humaine.

FIGURE 6.5 – Interaction avec l'utilisateur pour poser des questions générales

Dans cette partie, l'utilisateur peut entrer des questions générales (non obligatoires) concernant le fichier, puis la réponse sera affichée.

Perspectives

Conclusion générale

Références