République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene
Faculté d'Informatique
Département Intelligence Artificielle et Sciences des Données
Mémoire de Licence
Spécialité : Informatique Académique

Thème :
Utilisation des LLMs pour développer un outil d'apprentissage de la culture générale en Arabe

---

Encadré par :
Pr. GUESSOUM AHMED
Dr. BERKANI LAMIA

Présenté par :
Mlle BENTAIBA Selma
Mlle HADJ-YAHIA Nesrine

Membres du Jury :
Dr. SAADI ABDELFETTAH
Dr. ABDELLAHOUM HAMZA

Soutenu le : 04/06/2024

Projet : ACAD 122 / 2024

REMERCIEMENTS

Tout d'abord, nous tenons à exprimer notre profonde gratitude à Allah, le Tout-Puissant, de nous avoir donné la force et la persévérance nécessaires pour mener à bien ce travail de thèse. Nous adressons ensuite nos sincères remerciements à nos encadrants, Professeur Guessoum et Dr. Berkani Lamia, pour leur investissement complet dans notre projet.
Le Professeur Guessoum, nous a guidés avec sagesse et expertise tout au long de ce parcours. Sa capacité d'analyse de nos idées, ses conseils avisés et son encadrement constant ont été d'un soutien inestimable. Docteur Berkani Lamia, notre encadrante, nous a apporté un soutien constant et précieux. Son implication sans faille, sa disponibilité, y compris pour répondre à nos emails tardifs, et sa parfaite gestion des séances de travail ont été d'une grande aide. Ce fut un honneur de travailler à vos côtés. Vous nous avez apporté un soutien qui a dépassé le cadre de vos obligations et nous a permis d'apprendre énormément.
Nos remerciements s'adressent également au Docteur Mohamed Hadj Ameur et à Madame Asma Aouichat, qui ont enrichi nos travaux par leur présence aux séances de travail et leur partage de connaissances.
Enfin, nous remercions les membres du jury qui ont accepté d'évaluer notre travail et toutes les personnes qui, de près ou de loin, ont contribué à sa réalisation.

DÉDICACES

Je dédie au peuple Palestinien résilient, dont le courage et la persévérance inébranlables ont profondément marqué l'histoire. Que ce travail témoigne de votre indomptable esprit et porte l'espoir d'un avenir plus radieux.

À ma chère mère et mon cher père, piliers indéfectibles de mon parcours académique. Votre guidance aimante et vos encouragements ont nourri ma détermination et façonné mon chemin.

À ma sœur et mon frère, êtres chers et réconfortants. Votre affection sincère et votre compagnie ont illuminé même les jours les plus sombres. Je vous en suis éternellement reconnaissante.

À tous les membres de ma famille bien-aimée, ancrages précieux de ma vie. Votre bienveillance et votre soutien indéfectible ont été la lumière guidant chaque défi et triomphe.

À mes merveilleux amis, dont le rire et la camaraderie ont rempli ces années d'études de joie et de souvenirs impérissables. Votre amitié fut réconfort et inspiration, enrichissant ma vie au centuple.

Ce travail vous est dédié à tous, car votre affection, votre soutien et votre sollicitude ont rendu ce parcours véritablement significatif et gratifiant.

Bentaiba Selma.

DÉDICACES

Je dédie ce travail, avant tout, à mes parents. Leur soutien et leurs encouragements ont été un pilier essentiel tout au long de ce parcours. Je ne serais pas là sans eux.

Ensuite, à ma sœur et à mes frères, sources de joie et de fierté. À toute ma famille, pour leur présence constante.

À Selma, mon binôme et amie précieuse. Travailler à tes côtés a été une expérience enrichissante. Ton ambition est une source d'inspiration quotidienne.

À mes amis Wiam, Maroua, Dyhia, Amani, Djinane, Yasmine et Karima. Votre amitié a rendu ce chemin plus facile et plus agréable.

Hadj-Yahia Nesrine.

ARABIC ABSTRACT

الهدف من هذا المشروع هو تحسين مهارات المتعلمين في اللغة العربية، وخاصة قراءة وفهم الثقافة العامة العربية، وذلك من خلال تطوير أداة خاصة بمعالجة اللغة الطبيعية. تهدف هذه الأداة إلى تقديم مجموعة متنوعة من النصوص العربية التي تغطي موضوعات مختلفة مثل التاريخ والجغرافيا والأدب والعلوم والفنون. من خلال التفاعل مع هذه النصوص، يمكن للمتعلمين إثراء مفرداتهم وتعزيز إتقانهم للغة في سياقات ثقافية مختلفة

يعتمد نهجنا على إنشاء منصة اختبار تفاعلي مدعومة بالذكاء الاصطناعي وباستخدام نماذج اللغة الضخمة. لقد استخدمنا نموذجًا مدربًا مسبقًا وقمنا بضبطه باستخدام أسئلة ثقافية عربية محددة في خمس فئات مختلفة. لتحسين قدرة نموذج اللغة على فهم الثقافة العربية بشكل أفضل، استخدمنا تقنيات ضبط دقيقة متقدمة لتحسين أداء النموذج وتكييفه مع مجال تطبيقه. وتم تطوير تطبيق ويب للسماح للمنصة بإنشاء اختبارات مخصصة بناءً على مستوى مهارة كل مستخدم، وفقًا لما يحدده نموذج اللغة الضخمة

كان التحدي الرئيسي يكمن في قلة نماذج اللغة الضخمة العربية، وخاصة تلك التي تم ضبطها بعناية للمحتوى الثقافي. صُمم تطبيقنا ليتكيف مع مستوى تعلم المستخدم ويعالج الفجوات الموجودة في الفئات الخمس. يهدف مشروعنا إلى المساهمة في مجتمع نماذج اللغة الضخمة العربية من خلال سد هذه الفجوة وإظهار فائدة نماذج اللغة الضخمة العربية في التطبيقات التعليمية. ومن خلال القيام بذلك، نأمل في تسهيل الوصول إلى أدوات التعلم الثقافي وتشجيع التطورات الجديدة في مجال معالجة اللغة الطبيعية العربية

الكلمات المفتاحية: معالجة اللغة الطبيعية، تعليم اللغة، اللغة العربية، نماذج اللغة الضخمة، التعليم التكيفي، موارد التعلم الثقافي، واجهة برمجة تطبيقات

RÉSUMÉ

L'objectif de ce projet est d'améliorer les compétences des apprenants en Arabe, en particulier leur lecture et leur compréhension de la culture générale arabe, à travers le développement d'un outil de traitement automatique du langage naturel (TALN). Cet outil vise à fournir une variété de textes arabes couvrant divers sujets tels que l'histoire, la géographie, la littérature, les sciences et les arts. En interagissant avec ces textes, les apprenants peuvent enrichir leur vocabulaire et approfondir leur maîtrise de la langue dans différents contextes culturels.

Notre approche repose sur la création d'une plateforme de quiz adaptative alimentée par l'IA en utilisant des grands modèles de langage (Large Language Models ,LLM). Nous avons utilisé le modèle pré-entraîné GEMMA et l'avons affiné en utilisant des questions culturelles arabes spécifiques dans cinq catégories différentes. Pour optimiser le modèle de langage afin de mieux comprendre la culture arabe, nous avons utilisé des techniques d'affinement avancées. Ces techniques permettent d'améliorer la performance du modèle et de l'adapter à son domaine d'application. Une application web a été développée pour permettre à la plateforme de générer des quiz personnalisés en fonction du niveau de compétence de chaque utilisateur, tel que déterminé par le LLM.

Un défi majeur rencontré était le manque de LLMs bien entrainés pour l'Arabe, en particulier ceux affinés pour des contextes culturels. Notre application est conçue pour s'adapter au niveau d'apprentissage de l'utilisateur et couvrir ses lacunes dans les cinq catégories. Notre projet vise à contribuer à la communauté des LLMs arabes en comblant cette lacune et en démontrant leur utilité dans les applications éducatives. Ce faisant, nous espérons faciliter un meilleur accès aux outils d'apprentissage culturel et encourager de nouveaux développements dans le domaine du TALN arabe.

Mots-clés : Traitement Automatique du Langage Naturel , Apprentissage des Langues, Langue Arabe, Grnads modèles de Langage , Éducation Adaptative (Adaptive Learning), Ressources d'Apprentissage Culturel, GEMMA, LoRA, PEFT, Streamlit, Langchain, API Gemini.

ABSTRACT

The objective of this project is to improve learners' skills in Arabic, particularly their reading and comprehension of Arabic general culture, through the development of a Natural Language Processing (NLP) tool. This tool aims to provide a variety of Arabic texts covering various subjects such as history, geography, literature, science, and the arts. By interacting with these texts, learners can enrich their vocabulary and deepen their mastery of the language in different cultural contexts.

Our approach relies on the creation of an adaptive quiz platform powered by AI using large language models (LLMs). We used the pre-trained GEMMA model and fine-tuned it using specific Arabic cultural questions in five different categories. To optimize the language model to better understand Arabic culture, we used advanced Fine Tuning techniques to improve the performance of the model and to adapt it to its field of application. A web application was developed to allow the platform to generate custom quizzes based on each user's skill level, as determined by the LLM.

A major challenge encountered was the lack of Arabic LLMs, particularly those fine-tuned for cultural contexts. Our application is designed to adapt to the user's learning level and cover their gaps in the five categories. Our project aims to contribute to the Arabic LLM community by bridging this gap and demonstrating the usefulness of Arabic LLMs in educational applications. In doing so, we hope to facilitate better access to cultural learning tools and encourage new developments in the field of Arabic NLP.

Keywords : Natural Language Processing (NLP), Language Learning, Arabic Language, Large Language Models (LLMs), Adaptive Education, Cultural Learning Resources, GEMMA, LoRA, PEFT, Streamlit, Langchain, Gemini API.

TABLE DES MATIÈRES

LISTE DES FIGURES

ACRONYMES

Acronymes	**Signification**
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
NLP	Natural Language Processing
TALN	Traitement Automatique du Langage Naturel
LLM	Large Language Model
NN	Neural Network
GA	Genetic Algorithm
EA	Evolutionary Algorithm
TS	Tabu Search
QCM	Questionnaire à Choix Multiples
CPU	Central Processing Unit
GPU	Graphics Processing Unit
DDP	Distributed Data Parallel
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interface
LoRA	Low-Rank Adaptation
PEFT	Parameter-Efficient Fine-Tuning
RAG	Retrieval-Augmented Generation
VSCode	Visual Studio Code
MOOC	Massive Open Online Courses
GPT	Generative Pre-trained Transformer
ITS	Systèmes Intelligent de Tutorat -Intelligent Tutoring Systems
URL	Uniform Resource Locator
BERT	Bidirectional Encoder Representations from Transformers
NSP	Next Sentence Prediction
TLN	Traitement du Langage Naturel
TII	Technology Innovation Institute
NLU	Natural Language Understanding
TAL	Traitement Automatique des Langues
FTI	Feature Training Inference

INTRODUCTION GÉNÉRALE

La langue arabe, malgré sa richesse et son utilisation étendue, se heurte à des défis majeurs dans le domaine de l'intelligence artificielle. Des efforts considérables ont été déployés par certains chercheurs, mais un manque crucial d'outils, de données et de recherches entrave encore le développement de projets de traitement automatique de la langue arabe à la pointe du progrès. Cependant, les récentes avancées technologiques, notamment dans les domaines des chatbots et des modèles de langage à grande échelle (LLMs), offrent de nouvelles opportunités pour l'intelligence artificielle appliquée à la langue arabe. La puissance et la rapidité de ces outils encouragent la communauté des chercheurs à développer des modèles, des ressources et des outils plus efficaces pour l'apprentissage des apprenants. Ces avancées technologiques offrent des perspectives prometteuses pour surmonter les défis actuels et permettre le développement de solutions innovantes adaptées aux besoins spécifiques de la langue arabe dans le domaine de l'intelligence artificielle.

Ce projet de fin d'études de licence vise à améliorer les compétences en langue Arabe des apprenants, notamment en lecture et en compréhension de la culture générale arabe, à travers AraQuiz, un outil de traitement automatique du langage naturel. Notre plateforme de quiz adaptative utilise des LLMs pour proposer des questions culturelles spécifiques, adaptées au niveau de l'utilisateur. En affinant le modèle GEMMA avec des techniques avancées et en intégrant l'API Gemini, nous avons surmonté le défi du manque de modèles de langage bien entraînés pour l'arabe. Nous espérons ainsi enrichir les ressources éducatives en arabe et encourager le développement du TALN dans cette langue.

Dans le chapitre 1, nous présenterons l'état de l'art du domaine, mettant en lumière les avancées récentes et les défis à relever dans le TALN et les chatbots. Nous aborderons les tâches fondamentales du TALN, les systèmes d'apprentissage sous-jacents ainsi que les éléments constitutifs des modèles de langage.

Le chapitre 2 se concentrera sur la conception de notre projet. Nous décrirons notre approche pour développer un outil d'apprentissage dédié à la culture générale, en mettant l'accent sur la génération de questions et le fine-tuning des modèles de langage.

Enfin, dans le chapitre 3, nous détaillerons l'implémentation de notre solution. Nous présenterons l'environnement de développement, les bibliothèques utilisées et les fonctionnalités de l'application finale. Nous conclurons en surlignant les contributions majeures de ce travail, tout en discutant des perspectives d'avenir et des améliorations potentielles de notre application.

CHAPITRE 1

ÉTAT DE L'ART

1.1 Introduction

Ce chapitre s'articulera autour de deux axes principaux : le traitement automatique du langage naturel (TALN) et ses applications, et la découverte des systèmes d'apprentissage, en particulier ceux basés sur l'intelligence artificielle (IA).

Tout d'abord, nous établirons le contexte du TALN, soulignant son importance croissante et les défis qu'il présente. Ensuite, nous explorerons ses différentes applications, telles que la traduction automatique et la génération de texte.

Par la suite, nous nous pencherons sur les systèmes d'apprentissage. Nous commencerons par examiner les approches traditionnelles avant de nous plonger dans le monde de l'IA et de ses applications dans l'apprentissage. Enfin, nous comparerons les deux types de systèmes pour en cerner les forces et les faiblesses.détail dans les chapitres suivants.

1.2 Traitement automatique du langage naturel (TALN) et ses applications

Le traitement automatique du langage naturel a connu une évolution spectaculaire ces dernières années, tant sur le plan méthodologique que sur celui des applications prises en charge. Les avancées méthodologiques vont de nouvelles façons de représenter les documents à de nouvelles techniques de synthèse du langage. Ces progrès ont permis de développer de nouvelles applications, allant des systèmes conversationnels ouverts aux techniques utilisant le langage naturel pour l'interprétation des modèles.

Le traitement automatique du langage naturel a connu une évolution spectaculaire ces dernières années, tant sur le plan méthodologique que sur celui des applications prises en charge. Les avancées méthodologiques vont de nouvelles façons de représenter les documents à de nouvelles techniques de synthèse du langage. Ces progrès ont permis de développer de nouvelles applications, allant des systèmes conversationnels ouverts aux techniques utilisant le langage naturel pour l'interprétation des modèles.

Le TALN est un domaine situé au carrefour de l'informatique, de l'intelligence artificielle et de la linguistique. Il s'agit de construire des systèmes capables de traiter et de comprendre le langage humain. Depuis ses débuts dans les années 1950 et jusqu'à très récemment, le TALN était principalement le « domaine privé » des universités et des laboratoires de recherche, nécessitant une longue formation académique. Les percées de la dernière décennie ont conduit à une utilisation accrue du TALN dans des domaines très variés tels que la vente au détail, la santé, la finance, le droit, le marketing, les ressources

humaines, et bien d'autres encore.

1.2.1 Tâches fondamentales fréquemment rencontrées dans les projets de TALN

Le domaine du traitement automatique du langage naturel (TALN) englobe un large éventail de tâches, chacune visant à améliorer l'interaction entre les ordinateurs et le langage humain [1] .

- **Modélisation du langage :** Cette tâche consiste à prédire le mot suivant dans une phrase en se basant sur l'historique des mots précédents. L'objectif est d'apprendre la probabilité d'une séquence de mots apparaissant dans une langue donnée. La modélisation du langage est utile pour construire des solutions à des problèmes variés, tels que la reconnaissance vocale, la reconnaissance optique de caractères, la reconnaissance de l'écriture manuscrite, la traduction automatique et la correction orthographique.

- **Classification de textes :** Il s'agit de catégoriser le texte en fonction de son contenu dans un ensemble de classes prédéfinies. La classification de texte est de loin la tâche la plus courante en TALN et est utilisée dans de nombreux outils, allant de l'identification des spams dans les emails à l'analyse de sentiments sur les réseaux sociaux (et autres).

- **Extraction d'information :** Comme son nom l'indique, il s'agit d'extraire des informations pertinentes d'un texte, tels que les événements du calendrier à partir d'e-mails ou les noms de personnes mentionnées dans une publication comme sur les réseaux sociaux.

- **Recherche d'information :** Cette tâche consiste à trouver des documents pertinents à une requête d'utilisateur dans une large collection. Des applications comme Google Search sont des exemples bien connus de recherche d'information.

- **Agent conversationnel :** Il s'agit de construire des systèmes de dialogue capables de converser dans des langues humaines. Alexa, Siri, etc., sont des applications courantes de cette tâche.

- **Résumé de texte :** Cette tâche vise à créer des résumés courts de documents longs tout en conservant le contenu principal et en préservant le sens global du texte.

- **Réponse aux questions :** Il s'agit de construire des systèmes capables de répondre automatiquement à des questions posées en langage naturel.

- **Traduction automatique :** Cette tâche consiste à convertir un texte d'une langue à une autre. Des outils comme Google Translate sont des applications courantes de cette tâche.

- **Modélisation de sujets :** Cette tâche consiste à découvrir la structure thématique d'une grande collection de documents. La modélisation de sujets est un outil courant d'exploration de texte et est utilisée dans de nombreux domaines, de la littérature à la bio-informatique

1.3 Systèmes d'apprentissage

Le domaine de l'éducation connait une transformation majeure grâce à l'émergence de l'intelligence artificielle. Cette technologie révolutionnaire ouvre de nouvelles perspectives pour l'apprentissage et l'enseignement, en offrant des outils et des ressources innovantes qui ont le potentiel de transformer radicalement l'expérience éducative.

1.3.1 Systèmes d'apprentissage traditionnels

Au cours de la dernière décennie, les systèmes d'apprentissage en ligne se sont largement inspirés de la pédagogie traditionnelle, en proposant des cours magistraux, des tutoriels et d'autres formats similaires. Cette approche présente des avantages et des inconvénients. Prenons l'exemple des MOOC (Massive Open Online Courses) pour illustrer ce propos.

Les MOOC adoptent une structure hebdomadaire rythmée par des vidéoconférences, des forums de discussion et des quiz visant à stimuler l'interaction. Toutefois, ces formations sont conçues pour un large public et n'offrent pas d'encadrement pédagogique personnalisé.

D'une part, le contenu des MOOC se concentre principalement sur la formation continue, le développement professionnel et l'acquisition de compétences répondant aux besoins des adultes. D'autre part, l'évaluation d'un grand nombre d'apprenants et la stimulation de l'interaction restent des défis majeurs. De plus, ces formats tendent à exclure d'autres publics cibles, tels que les enfants et les adolescents [2] .

1.3.2 Systèmes d'apprentissage basés sur l'IA

L'IA a commencé à influencer l'éducation de manière visible depuis la sortie de différents modèles d'IA génératifs, et le fait qu'ils deviennent accessibles au public (ChatGPT étant le premier). Cette influence change complètement la vision sur les systèmes d'apprentissage, offrant des applications prometteuses et plusieurs avantages.

Les applications les plus importantes incluent l'apprentissage personnalisé, qui s'adapte aux besoins, au niveau et au rythme de chaque apprenant. Cette personnalisation permet aussi d'offrir un tutorat intelligent, comme le font les ITS (systèmes intelligents de tutorat), proposant une interaction engageante et un feedback correspondant à l'utilisateur. Une autre application est l'automatisation des tâches, telle que l'évaluation, ayant pour résultat une optimisation notable du temps.
Cet avancement a conduit à un accès global à une éducation de qualité, à de meilleurs résultats d'apprentissage pour les élèves défavorisés et à un gain d'efficacité pour les enseignants et les institutions. Cependant, il reste des défis à relever, en particulier la sécurité des données et le biais [3].

1.3.3 Comparaison entre les systèmes d'apprentissage traditionnels et ceux basés sur l'IA

En comparant les systèmes d'apprentissage traditionnels et ceux basés sur l'IA, on constate que chacun a ses limites. Cependant, l'IA apporte des éléments manquants à la pédagogie traditionnelle. Contrairement à l'apprentissage passif dans les systèmes traditionnels où l'utilisateur reçoit l'information sans interagir, l'IA favorise la participation active avec un retour d'information personnalisé tout au long du parcours. Une étude [4] illustre parfaitement ce point en comparant QuizBot, un chatbot basé sur l'IA, à une application de flashcards. L'étude montre que l'IA encourage les utilisateurs à consacrer plus de temps à l'apprentissage. Plus de 68% des participants ont préféré QuizBot à l'application, leurs performances se sont améliorées de 20% et il a été jugé plus divertissant et plus efficace que l'application de flashcards, même si le processus était plus long.
Cela démontre que les vastes ressources de l'IA lui confèrent un net avantage et que son intégration dans les systèmes d'apprentissage les porterait à un niveau supérieur. Néanmoins, il est crucial d'aborder cette intégration avec prudence.

1.4 Les éléments constitutifs des LLMs

Les LLMs sont des systèmes capables de traiter et de générer du langage naturel de manière sophistiquée. Ils surpassent les modèles de langage traditionnels en termes de performance et d'applicabilité, ils reposent sur une combinaison de technologies puissantes.

1.4.1 Apprentissage automatique (Machine Learning)

L'apprentissage automatique (ML) est généralement considéré comme un sous-domaine de l'intelligence artificielle qui vise à permettre aux ordinateurs d'apprendre un modèle sous-jacent dans les données par eux-mêmes, sans être programmé avec la connaissance préalable de ces données. Le ML utilise divers algorithmes, qui sont des séquences d'étapes prédéfinies permettant de produire une réponse [5].En règle générale, l'apprentissage automatique s'attaque à des problèmes que les humains ne savent pas décomposer en étapes, mais qu'ils résolvent naturellement. C'est le cas, par exemple, de la reconnaissance de visages dans des images ou de certains mots dans une discussion orale [6].
De manière informelle, le ML est l'application d'un algorithme mathématique qui nécessite des données d'entrée dans un certain format, produit une réponse dans un format de sortie prédéfini, et ne fournit aucune autre garantie si ce n'est qu'il minimise un certain nombre que l'on appelle la métrique d'évaluation [5]. Cela inclut l'apprentissage d'une fonction générale qui mappe les entrées aux sorties en fonction de l'expérience passée (apprentissage supervisé), l'extraction de modèles et de structures cachés à partir des données (apprentissage non supervisé), et l'apprentissage de la façon d'agir dans un environnement dynamique en fonction de récompenses indirectes (apprentissage par renforcement) [7] .

1.4.1.1 Apprentissage automatique pour le traitement automatique du langage naturel

Les techniques d'apprentissage automatique sont appliquées aux données textuelles tout comme elles le sont à d'autres formes de données, telles que les images, la parole et les données structurées. Les techniques d'apprentissage automatique supervisé, telles que la classification et la régression, sont largement utilisées pour diverses tâches de TALN.

Toute approche d'apprentissage automatique pour le TALN, supervisée ou non supervisée, peut être décrite comme comprenant trois étapes courantes : l'extraction de caractéristiques du texte, l'utilisation de la représentation des caractéristiques pour apprendre un modèle, et l'évaluation et l'amélioration du modèle [1].

Figure 1.1 – Les relations entre différents domaines tels que : IA, ML, DL,NLP [7].

1.4.2 Réseaux de neurones (Neural Networks)

Un réseau de neurones (Neural Network, NN) est un programme d'apprentissage automatique qui imite (très schématiquement) le fonctionnement du cerveau humain (neurones biologiques) pour prendre des décisions. Chaque réseau de neurones est composé de couches de nœuds (une couche d'entrée, de une ou plusieurs cachées, et une de sortie). Ces nœuds sont connectés entre eux et possèdent chacun

un poids. Chaque connexion multiplie l'entrée par un poids et ajoute un biais, la somme pondérée des entrées passe par une fonction d'activation qui est appliquée au nœud et le résultat de cette opération est envoyé à chaque nœud de la couche suivante du réseau.

Ces poids sont déterminés plus tard par rétropropagation, leur optimisation est ce que l'on appelle l'apprentissage il permet au neurones d'améliorer leur précision au fil du temps grâce à des données d'entraînement. Une fois optimisés pour la précision, ils deviennent des outils informatiques puissants en intelligence artificielle. Ils permettent de classer et de regrouper des données à grande vitesse ou de développer des modèles pour des applications très diverses, couvrant pratiquement tous les champs de l'activité humaine [8].

1.4.3 Apprentissage profond (Deep Learning)

L'apprentissage profond est un sous-domaine de l'apprentissage automatique qui se concentre sur l'utilisation de modèles larges et complexes avec de nombreuses couches pour apprendre automatiquement des représentations hiérarchiques des données. Cette approche exploite de vastes ensembles de données et une puissance de calcul significative pour découvrir des motifs et des structures complexes, permettant des avancées dans des domaines tels que la reconnaissance d'images et de la parole, le traitement du langage naturel et les systèmes autonomes. Le terme "profond" reflète la profondeur de ces modèles, qui contiennent de multiples couches de traitement extrayant progressivement des caractéristiques de plus en plus abstraites à partir des données brutes [9] .

L'évolution du terme "apprentissage profond" : Le terme "apprentissage profond" a émergé pour différencier ces modèles avancés et multilayered des réseaux neuronaux artificiels plus simples et antérieurs. Le travail pionnier de Geoffrey Hinton au milieu des années 2000, notamment sur les réseaux de croyance profonde et les machines de Boltzmann profondes, a démontré l'efficacité de la formation d'architectures profondes couche par couche. Ces développements ont mis en évidence le potentiel de l'apprentissage profond pour gérer des tâches complexes que les techniques précédentes de réseaux neuronaux avaient du mal à résoudre, principalement en raison de limitations de puissance de calcul et de disponibilité des données. À mesure que ces limitations ont diminué, l'apprentissage profond est devenu un paradigme distinct et puissant au sein de l'intelligence artificielle [10] .

1.4.4 Transformers

Le Transformer est un modèle de réseau neuronal séquence à séquence initialement conçu pour la traduction de langues, il vise à générer des séquences cibles à partir de séquences sources, par exemple une des phrases en anglais en ayant comme entrée des phrases en espagnol [11].Cette méthode fait partie de la classe des modèles “encodeur-décodeur”, elle est maintenant utilisée pour la modélisation générale du langage. La plupart des LLMs modernes dont ChatGpt, Bard et LLaMA, adoptent cette architecture.

Architecture Encodeur-Décodeur

- L'encodeur prend en charge la séquence d'entrée et génère des représentations contextuelles riches pour chaque élément. Il s'agit de capturer les relations entre les mots et leur importance dans la séquence.

- Le décodeur utilise ensuite ces représentations contextuelles générées par l'encodeur pour produire la séquence de sortie. Il traduit les informations codées en une séquence intelligible, que ce soit une traduction, un texte généré ou une réponse à une question [3] .

Le passage par ces deux modules va permettre au modèle d'apprendre à capturer les différences détaillées entre la séquence d'entrée et celle de sortie. Cela pour une bonne compréhension du contexte et précision

dans la génération. L'illustration 1.2 représente le fonctionnement des Transformers de manière plus claire [12] .

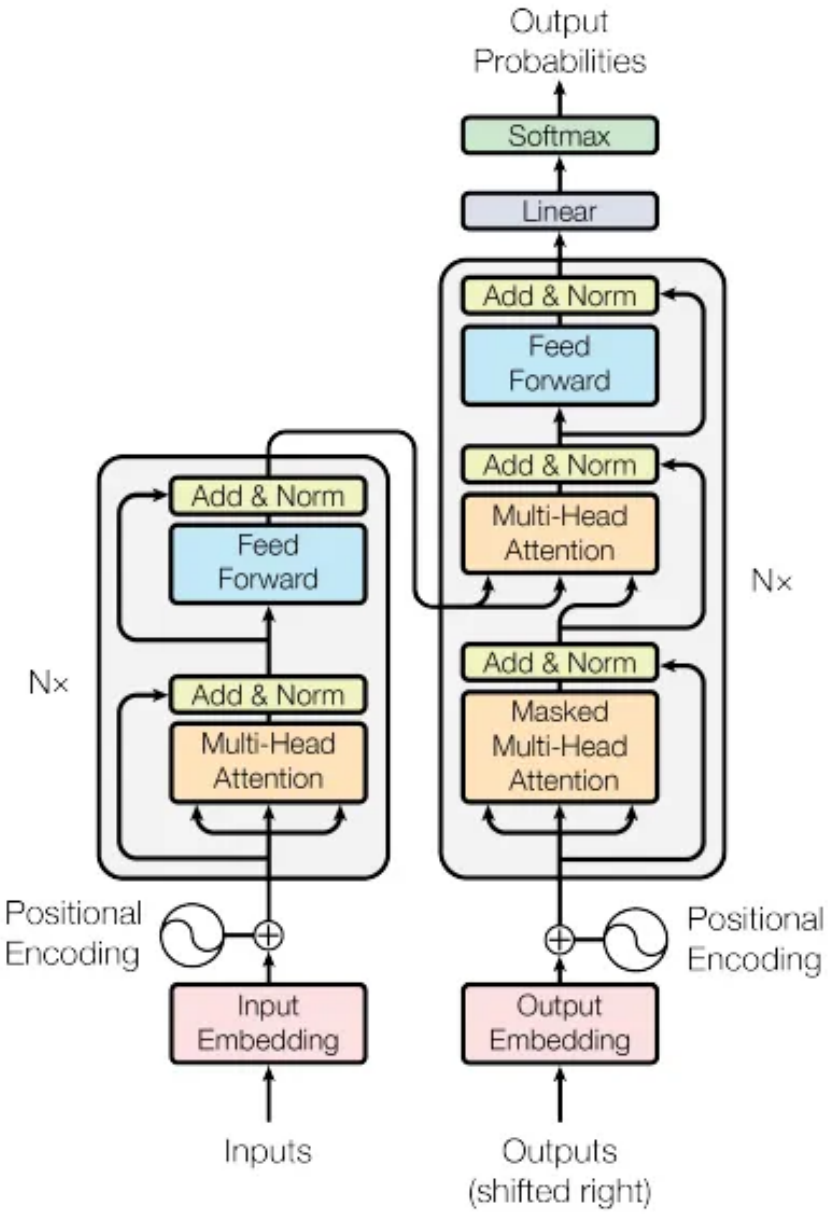

Figure 1.2 – Achitecture de Transformers
[12]

Mécanisme d'Attention

L'un des aspects clés du Transformer réside dans son mécanisme d'attention. Ce mécanisme permet au modèle de se concentrer sur les parties les plus pertinentes de la séquence d'entrée lors du traitement. Le modèle apprend à attribuer des poids à chaque élément de la séquence d'entrée, en fonction de son importance et de son contexte. Cela permet au modèle de comprendre les relations entre les mots et de générer une sortie plus précise et cohérente.

Couches principale

- **Intégration de mots ("Word Embedding") :** Cette méthode permet de convertir des mots en nombres. L'idée principale est d'utiliser un réseau de neurones simple avec une entrée pour

chaque mot/symbole du vocabulaire (ces mots et symboles sont appelés "Tokens"). Les jetons sont connectés à des fonctions d'activation.

- **Codage de position ("Positional Encoding") :** Il s'agit du mécanisme clé que nous avons évoqué plus tôt.
- **Mécanisme d'attention à soi-même ("Self-attention") :** C'est le mécanisme crucial que nous avons présenté plus haut.
- **Connexions résiduelles ("Residual Connection") :** Ces connexions permettent aux informations de sauter directement entre les couches, évitant ainsi le problème du gradient qui s'évanouit où les signaux des couches précédentes s'affaiblissent en se propageant dans le réseau. Cela garantit que les informations initiales atteignent la sortie finale, permettant au réseau d'apprendre des caractéristiques plus complexes et de s'entraîner plus efficacement. Elles sont obtenues en ajoutant les valeurs codées par le mot et la position aux valeurs d'attention personnelle afin de créer des raccourcis au sein de l'architecture.

1.4.5 Compréhension générale des LLMs

La modélisation du langage est un sous-domaine du TALN qui implique la création de modèles statistiques ou de modèles d'apprentissage profonds pour prédire la probabilité d'une séquence de mots (dits jetons ou tokens) dans un vocabulaire spécifique (un ensemble limité et connu de jetons) [13].

L'historique du traitement automatique du langage naturel est passé de la modélisation du langage statistique à la modélisation du langage neuronal, puis aux modèles de langage pré-entraînés (Pre-trained Language Models) aux LLM (Large Language Models) [14].

Un LLM est un modèle de réseau neuronal profond qui a été entraîné sur de vastes quantités de données textuelles, telles que des livres, du code source, des articles et des contenus de sites Web tels que Wikipedia. Grâce à cet entraînement, le modèle apprend les modèles et les relations sous-jacentes dans la langue pour laquelle il a été entrainé. Ainsi, le LLM est capable de générer du contenu cohérent, comme des phrases et des paragraphes grammaticalement corrects imitant le langage humain ou des extraits de code syntaxiquement valides [13].

Ces modèles sont entraînés sur des volumes massifs de données textuelles, ce qui leur permet de capturer les complexités et les nuances du langage humain. Les LLM peuvent effectuer une large gamme de tâches linguistiques, de la classification simple de texte à la génération de texte, avec une grande précision, fluidité et style [14].

1.5 Entraînement des LLMs

Le processus d'entraînement joue un rôle fondamental dans le développement et la performance des LLM. Il s'agit d'un processus itératif qui permet au modèle d'apprendre et de s'améliorer en permanence. Et il implique plusieurs étapes clés

1.5.1 Pré-entraînement

Le pré-entraînement d'un modèle de langage implique l'entraînement du modèle sur un large corpus de données textuelles, telles que des articles, des livres, des sites Web ou même un ensemble de données soigneusement sélectionnées. Au cours de cette phase, le modèle apprend à générer du texte pour un corpus général ou au service d'une tâche spécifique. Ce processus aide le modèle à apprendre la grammaire, la syntaxe et un certain niveau de sémantique à partir des données textuelles.

La fonction d'objectif utilisée pendant le pré-entraînement est généralement la perte d'entropie croisée (Cross-Entropy Loss), qui mesure la différence entre les probabilités de jetons prédites et les probabilités

de jetons réelles. Le pré-entraînement permet au modèle d'acquérir une compréhension fondamentale du langage, qui peut ensuite être affiné pour des tâches spécifiques.

Voici la formule de la perte par entropie croisée :

$$H(P, Q) = -\sum_{x} P(x) \log Q(x) \tag{1.1}$$

Comme montré dans l'Équation 2.1, la perte par entropie croisée est une métrique importante en apprentissage automatique.

Le processus de pré-entraînement d'un LLM (Large Language Model - modèle de langage large) peut évoluer au fil du temps à mesure que les chercheurs trouvent de meilleures façons d'entraîner les LLM et abandonnent les méthodes qui ne sont pas aussi efficaces.

Par exemple, moins d'un an après la sortie du modèle BERT original de Google qui utilisait la tâche de pré-entraînement de prédiction de la phrase suivante (NSP), une variante de BERT appelée RoBERTa par Facebook AI a montré qu'elle n'avait pas besoin de la tâche NSP pour égaler et même surpasser les performances du modèle BERT original dans plusieurs domaines [15].

1.5.2 Les modèles multilingues les plus puissants récemment développés

- **GPT-4 par OpenAI**

  GPT-4 se démarque par ses capacités créatives et collaboratives accrues. Il peut générer, éditer et interagir avec les utilisateurs sur des tâches d'écriture créative et technique, allant de la composition de chansons à la rédaction de scénarios, en passant par l'apprentissage du style d'écriture d'un utilisateur. OpenAI a également mis l'accent sur la sécurité et la fiabilité de GPT-4.

  Le modèle est 82% moins susceptible de répondre aux demandes de contenu interdit et 40% plus susceptible de produire des réponses factuelles par rapport à son prédécesseur GPT-3.5, selon leurs évaluations internes [16] .

- **PaLM 2 par Google** PaLM 2 est la nouvelle génération de modèles de langage large de Google. Il s'appuie sur l'héritage de Google en matière de recherche de pointe en apprentissage automatique et en intelligence artificielle responsable. PaLM 2 excelle dans des tâches complexes comme le raisonnement avancé, la traduction et la génération de code. Son architecture innovante lui permet de surpasser son prédécesseur PaLM en unifiant trois avancées distinctes dans la recherche sur les modèles de langage volumineux. PaLM 2 atteint des résultats remarquables sur les tests de référence de raisonnement. Il démontre également des capacités de traduction multilingue supérieures à celles de PaLM et de Google Traduction dans des langues comme le portugais et le chinois [17].

- **Llama 3 par Meta**

  L'intégration de Llama 3 dans Meta AI, l'assistant intelligent de Meta, élargit les possibilités pour les utilisateurs de réaliser des tâches, de créer et de se connecter avec Meta AI. Il est possible de constater par soi-même les performances de Llama 3 en utilisant Meta AI pour des tâches de codage et de résolution de problèmes. Que ce soit pour le développement d'agents ou d'autres applications alimentées par l'IA, Llama 3, disponible en versions 8B et 70B, offre les capacités et la flexibilité nécessaires pour concrétiser des idées [18].

- **Claude 2 par Anthropic**

  Claude est une suite de modèles d'IA fondamentaux polyvalents. Vous pouvez interagir directement avec Claude sur leur site web pour brainstorming d'idées, analyse d'images et traitement de

longs documents. Anthropic propose également un accès API aux développeurs et aux entreprises, leur permettant de construire directement sur leur infrastructure d'IA. Claude se distingue par sa robustesse face aux contournements et aux utilisations abusives, ainsi que par ses faibles taux d'hallucination (fourniture d'informations incorrectes). Il est capable de traiter des documents très longs avec une grande précision [19].

- **Falcon LLM**

  Falcon LLM est un modèle de langage large génératif conçu pour faire progresser les applications et les cas d'utilisation de l'IA. Falcon propose une gamme de modèles allant de 1.3B à 180B de paramètres, ainsi qu'un ensemble de données de haute qualité nommé REFINEDWEB [20].

1.5.3 Modèles de langage arabe récents et performants

– **JAIS**

JAIS est un modèle de langage large (LLM) puissant, basé sur une architecture de pré-entraînement générative de type GPT-3. Il se concentre spécifiquement sur l'arabe et ne possède qu'un décodeur (decoder-only). Pour pallier la quantité limitée de données de pré-entraînement en arabe, les chercheurs ont eu recours à un ingénieux procédé. Ils ont entraîné JAIS sur un corpus colossal de 395 milliards de jetons, comprenant 72 milliards de jetons en arabe (dupliqués 1,6 fois pour atteindre un total effectif de 116 milliards de jetons arabes), 232 milliards de jetons en anglais et le reste étant du code dans divers langages de programmation. De plus, ils ont conçu un pipeline spécialisé pour le traitement du texte arabe, incluant un filtrage et un nettoyage minutieux des données afin de garantir une qualité optimale. L'équipe de JAIS a également développé une version de leur modèle spécialement adaptée aux instructions, nommée Jais-chat. Ce dernier a été entraîné sur plus de 3,6 millions de paires instruction-réponse en arabe et 6 millions en anglais [21].

Le Technology Innovation Institute (TII), un centre de recherche mondial basé à Abu Dhabi, a récemment annoncé le lancement de NOOR, le plus grand modèle de traitement du langage naturel (TLN) en arabe à ce jour. NOOR est un modèle à décodeur uniquement, similaire dans sa structure à GPT-3. Il est basé sur l'architecture Transformer et a été programmé pour exceller dans les tâches génératives. Son architecture intègre les derniers développements du domaine de l'apprentissage automatique, notamment des améliorations au niveau des plongements de position (positional embeddings). Pour construire NOOR, les chercheurs du TII ont conçu un pipeline complet de collecte de données de haute qualité. Ce processus comprend l'exploration (crawling), le filtrage et la curation à grande échelle. Le modèle a été entraîné sur un ensemble de données arabe exceptionnellement volumineux, fruit de mois de travail de collecte, de filtrage et de curation de sources variées [22].
AraBERT est un modèle de représentation du langage arabe conçu pour améliorer les performances de pointe dans plusieurs tâches de compréhension du langage naturel (NLU) en arabe. Il est basé sur le modèle BERT, un encodeur Transformer bidirectionnel empilé. AraBERT établit un nouvel état de l'art pour plusieurs tâches en aval pour la langue arabe. De plus, il est 300 Mo plus léger que le modèle BERT multilingue, ce qui le rend plus efficace pour les applications embarquées [23].

1.5.4 Fine tuning

Dans le domaine de l'intelligence artificielle, fine-tuning est le processus d'ajustement d'un modèle d'IA pré-entrainé afin d'améliorer ses performances sur des tâches ou des ensembles de données spécifiques. Cela implique d'entraîner le modèle sur de nouvelles données, lui permettant ainsi de s'adapter et d'améliorer ses capacités pour mieux répondre aux applications ciblées [24] .

Parameter-Efficient Fine-Tuning "PEFT" Le fine-tuning des modèles pré-entraînés est une technique efficace pour de nombreuses tâches de (TAL). Cependant, cette méthode est gourmande en paramètres car elle crée un nouveau modèle pour chaque tâche. Récemment, des recherches proposent de ne régler qu'une petite partie des paramètres, gardant le reste partagé entre les tâches. Ces méthodes appelées PEFT, surprennent par leur performance et leur stabilité accrue par rapport au réglage complete [25].

1.5.5 Prompting

Le "prompting" fait référence au processus qui consiste à fournir des instructions ou des indices spécifiques à un système d'IA afin de guider son comportement ou de générer les résultats souhaités. Un "prompt" d'IA est un élément de texte ou d'information qui sert d'instruction ou de commande pour le modèle d'IA. Il nous permet de communiquer nos intentions à la machine [26] .

Prompt engineering : C'est le processus de création d'instructions ("prompts") efficaces permettant aux modèles d'IA de générer des réponses basées sur des entrées données. En substance, l'ingénierie de prompt consiste à écrire intelligemment des instructions pour des tâches d'Intelligence Artificielle basées sur du texte, plus spécifiquement des tâches de traitement automatique du langage naturel. Dans le cadre de telles tâches textuelles, ces instructions aident l'utilisateur et le modèle à générer un résultat particulier en fonction des besoins [27] .

1.6 Conclusion

Les LLMs ont le potentiel de révolutionner l'apprentissage de l'arabe en offrant des expériences d'apprentissage plus personnalisées, immersives et interactives. Cependant, leur développement est encore freiné par le manque de données de qualité. La collecte et l'annotation de données en arabe sont en effet essentielles pour le développement de LLMs plus performants.

En conclusion, les LLMs ont le potentiel de transformer l'apprentissage de l'arabe et de le rendre plus accessible, efficace et agréable pour tous. Il est important de continuer à investir dans la recherche et le développement de ces technologies pour en tirer pleinement parti.

CHAPITRE 2

CONCEPTION

2.1 Introduction

Malgré la richesse de la langue Arabe, l'état de l'art montre un manque de ressources, que ce soit en termes de travaux de recherches, de modèles ou de données, comparé à d'autres langues. L'objet de ce chapitre est de proposer notre contribution qui vise à développer **AraQuiz**, un outil d'aide à l'apprentissage de langue arabe avec une attention particulière sur la culture générale. Il sera structuré en trois sections (1) une présentation de l'architecture générale de notre approche ; (2)une description détaillée de notre approche ; et (3) la phase d'entraînement de notre modèle.

2.2 Un outil d'apprentissage dédiée à la culture générale

Avant de passer au développement, il était indispensable de cerner les besoins auxquels l'outil devait répondre.

2.2.1 Objectifs de recherche

Ce projet vise à concevoir un outil dédié à des apprenants qui souhaitent approfondir leur connaissance en langue arabe. **AraQuiz** est une application intelligente capable de générer des questions de culture générale de type QCM offrant des choix multiples. Les questions seront générées automatiquement selon la catégorie et le niveau de difficultés choisis préalablement.
Afin d'atteindre cet objectif, nous allons explorer l'utilisation des LLMs qui sont avérés capables de générer des textes appropriés selon les prompts des utilisateurs. Une comparaison de plusieurs LLMs sera effectuée avant d'en choisir le plus approprié à notre cas d'étude.

2.2.2 Description de l'approche proposée

Afin de construire notre outil, nous avons choisi d'utiliser une approche basée sur l'architecture Feature/Training/Inference (FTI), qui est structurée de pipelines d'apprentissage automatique. Elle divise le processus en trois étapes :

- Pipeline de fonctionnalités : qui transforme les données brutes en un format adapté au LLM et à la tâche désirée,
- Pipeline d'entraînement : qui construit et forme le modèle en utilisant les fonctionnalités créées à l'étape précédente,

– Pipeline d'inférence : qui utilise le modèle entraîné pour prédire de nouvelles données.

La figure 2.1 illustre cette approche.

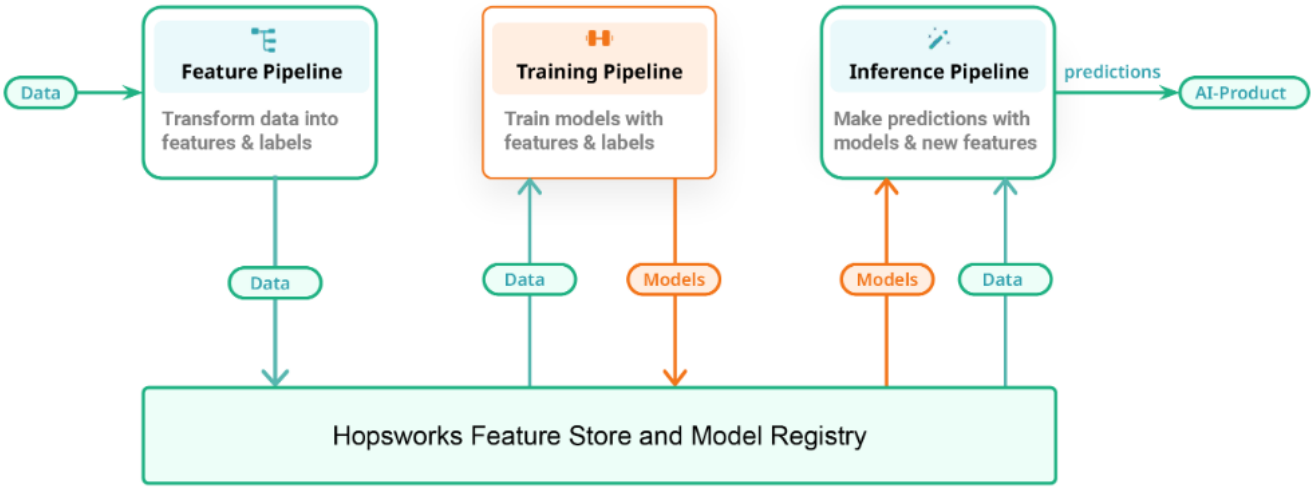

FIGURE 2.1 – Architecture Feature/Training/Inference (FTI)
[28] .

La figure 2.2 montre une vue globale de l'architecture que nous avons adoptée dans ce travail, en nous basant sur l'approche des pipelines, pour le développement de notre système dédiée à la génération de questions de culture générale en langue arabe.

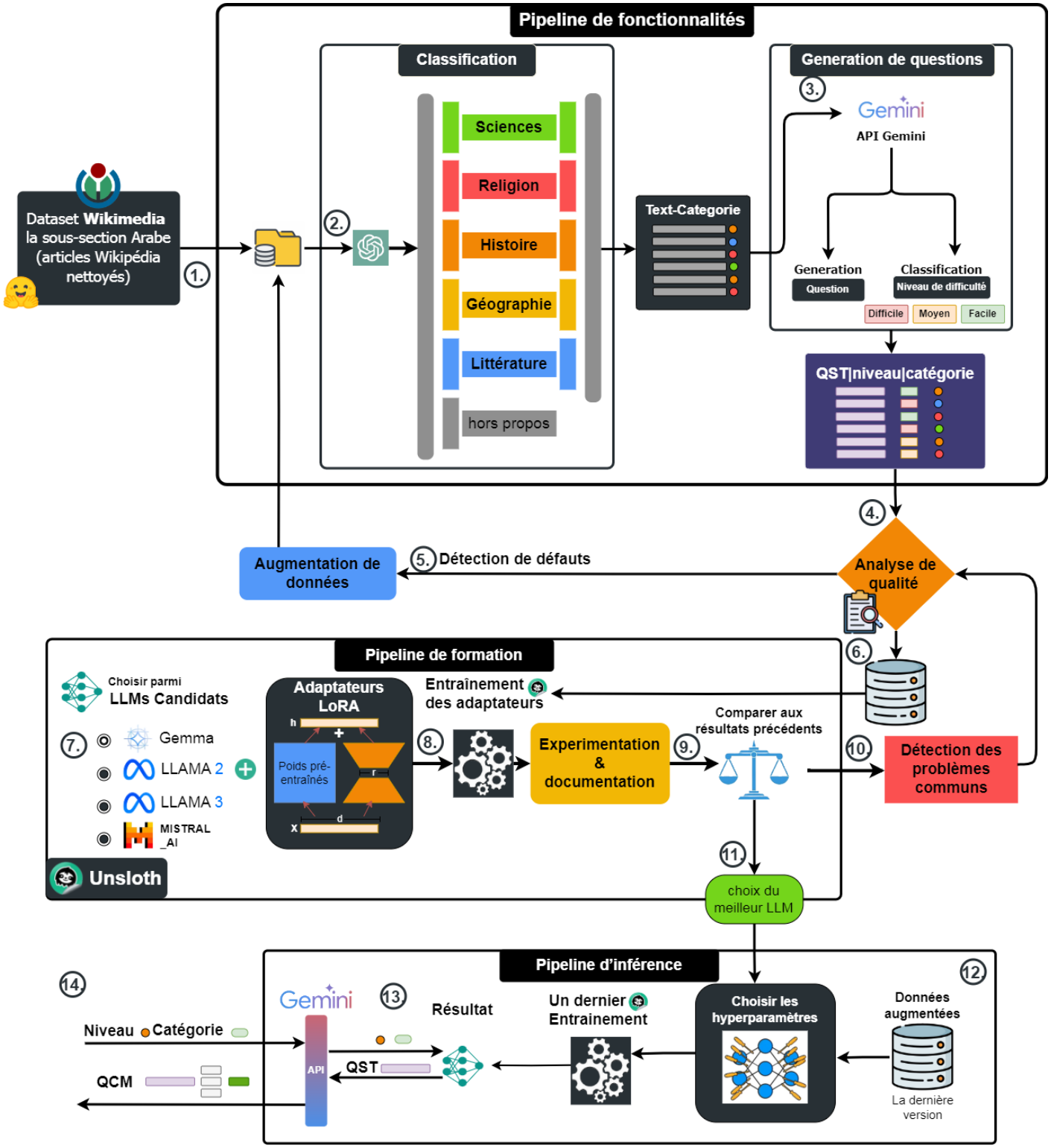

FIGURE 2.2 – La phase de conception : architecture globale de l'application proposée

1. Choix du corpus de base Wikimedia
2. Filtrage par catégorie avec l'API ChatGPT
3. Génération de questions et leur filtrage par niveau avec l'API Gemini
4. Analyser la qualité de données
5. Augmentation de données en cas de détection de défauts
6. Structurer les données dans un format utilisable dans l'entraînement
7. Choisir un des **LLMs** à utiliser et préparer les adaptateurs **LoRA**
8. Fine Tuning en utilisant les données préparées, documentation des résultats
9. Comparer les résultats obtenus pour chaque **LLM**
10. En cas de détection de problèmes communs : essayer de faire la relation avec la qualité de données et ainsi revenir à l'étape (4.)
11. Choisir le **LLM** le plus performant
12. Utiliser la dernière version de Data pour un dernier entraînement avec la concentration sur les hyperparamètres
13. Exploiter l'API Gemini encore une fois pour transformer les questions en QCMs
14. Atteindre l'objectif : catégorie et niveau en entrée QCM en sortie

2.3 Description détaillée de l'approche proposée

Nous allons plonger dans les détails de la conception et des choix méthodologiques qui ont guidé le développement de notre outil.

2.3.1 Construction d'un jeu de données

L'objectif final est d'obtenir un ensemble de données où chaque point de données contient une paire : catégorie-difficulté et une question appropriée. Le modèle sera entraîné sur de telles questions afin de produire de nouvelles questions en fonction du niveau de difficulté et de la catégorie choisie (Sciences, Histoire, Géographie, Littérature et Religion). Nous notons que les questions générées ne concernent pas le cas de questions ouvertes, mais doivent être adéquates pour l'utilisation dans un QCM à réponses multiples.

2.3.2 Corpus de base - Wikimedia

La qualité de données utilisée lors de l'entraînement détermine en grande partie la performance du modèle, c'est pour cela qu'on a pris le temps d'explorer les datasets en langue Arabe avant de choisir.

Les datasets de Wikimedia sont des ensembles de données liées aux projets de Wikimedia, tels que Wikipedia, Wikivoyage et Wiktionary. Dans la carte de jeu de données de hugging face il est expliqué que : "chaque exemple contient le contenu d'un article complet de Wikipédia avec un nettoyage pour supprimer le markdown et les sections indésirables (références, etc.)."

Nous avons opté pour l'utilisation de ce dataset en raison de sa taille, sa variété (contient des textes dans toutes les catégories qui nous intéressent) et la qualité de ses données[29].

La Figure 2.3 présente le dataset utilisé. Un extrait de ce dataset est présenté dans la Figure 2.4, illustrant sa structure et quelques exemples de texte inclus.

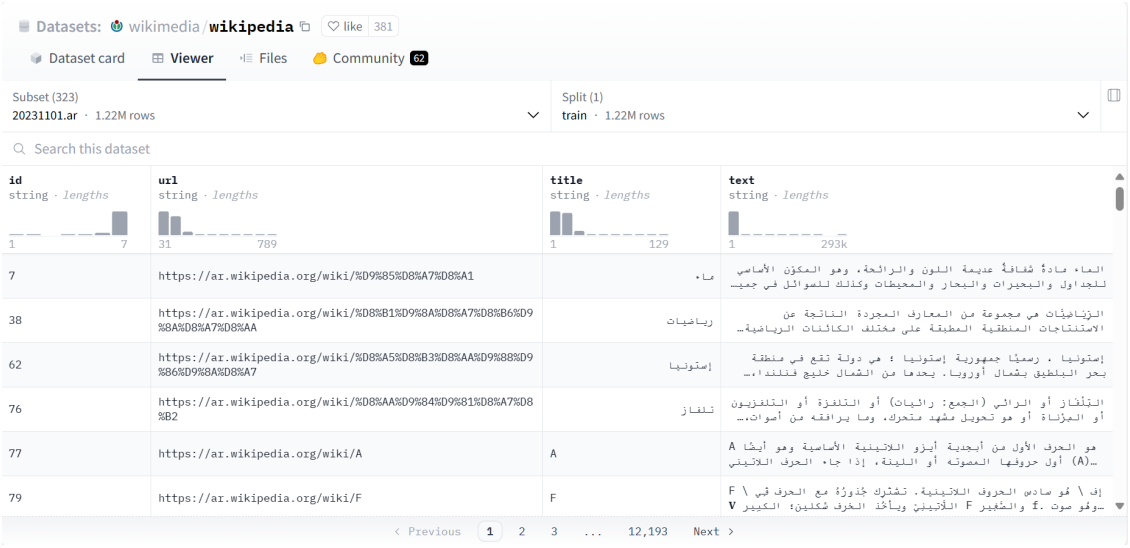

FIGURE 2.3 – DataSet Wikimedia via HugginFace

ماء	الماء مادةٌ شفافةٌ عديمة اللون والرائحة، وهو المكوّن الأساسي للجداول والبحيرات والبحار والمحيطات وكذلك للسوائل في جميـ…
رياضيات	الرِّيَاضِيَّات هي مجموعة من المعارف المجردة الناتجة عن الاستنتاجات المنطقية المطبقة على مختلف الكائنات الرياضية…
إستونيا	إستونيا ، رسميًا جمهورية إستونيا ؛ هي دولة تقع في منطقة بحر البلطيق بشمال أوروبا. يحدها من الشمال خليج فنلندا،…
تلفاز	التِّلْفَاز أو الرائي (الجمع: رائيات) أو التلفزة أو التلفزيون أو المِرْنَاة أو هو تحويل مشهد متحرك، وما يرافقه من أصوات،…
	هو الحرف الأول من أبجدية أيزو اللاتينية الأساسية وهو أيضًا A (A)… أول حروفها المصوته أو اللينة، إذا جاء الحرف اللاتيني
	إف \ F هُو سادس الحروف اللاتينية. تشترك جُذورُهُ مع الحرف قِي V وهُو صوت f. والصَّغِير F اللَّاتِينِيّ ويأخُذ الخَرف شكلين؛ الكبير…

FIGURE 2.4 – Extrait du DataSet Wikimedia

2.3.3 Filtrage par catégorie

Afin de générer des questions dans cinq catégories culturelles ciblées, nous avons procédé à un filtrage des textes que nous voulons manipuler. Les données complètes contiennent certainement des textes n'appartenant à aucune catégorie, qu'on doit exclure. Le choix de l'API ChatGPT pour le filtrage par catégorie n'est pas seulement effectué en raison de ses performances robustes en classification de textes, mais aussi pour le contrôle qu'elle offre en sorties, où nous pouvons par exemple choisir le nombre de mots en sortie.
En utilisant la variable « max tokens » (nombre maximum jetons) limitée à 1 pour éviter la génération d'un texte ou un mot au lieu d'une seule valeure numérique, avec du prompting invitant le modèle à renvoyer un nombre relatif à la catégorie concernée, associant les indexes de textes à leurs catégories.

2.3.4 Génération de questions

La génération de questions dépend d'un apprentissage efficace, qui nécessite d'avoir un nombre suffisant de textes classifiés. Nous avons choisi d'effectuer un apprentissage de type "parameter efficient fine tuning", sachant qu'au début on avait imposé la contrainte de disposer au minimum de 1000 textes avec 200 textes par catégorie (un nombre dans la plage recommandée). Nous avons utilisé l'API Gemini à cause de sa performance et efficacité pour le cas de tâches linguistiques par rapport à d'autres APIs. Nous avons considéré en entrée le texte classifié avec sa catégorie, le résultat attendu serait une question en arabe avec le niveau de difficulté correspondant à cette question. Pour plus de précision à propos du niveau de difficulté, nous avons fourni des exemples pour chaque niveau lors de la génération des questions, à raison d'une catégorie à la fois.

Cette méthode étant la plus simple et directe permettant de procéder avec une classification initiale par rapport à la difficulté. Nous notons aussi qu'au début, nous avions même créé les options et la réponse afin de garantir que la question ait le format souhaité.

Voici un extrait montrant quelques exemples et le format catégorie, niveau, question, options, bonne réponse du dictionnaire généré :

```
{'mcq0': {'category': 'Science',
  'level': 'hard',
  'question': 'أي مما يلي يعتبر تجريدًا أول في الرياضيات؟',
  'options': ['العدد', 'الهندسة', 'الجبر', 'القيمة الموضعية'],
  'answer': ['العدد']},
 'mcq1': {'category': 'Science',
  'level': 'medium',
  'question': 'ما العنصر الكيميائي الذي يرمز له حرف K ؟',
  'options': ['البوتاسيوم', 'الصوديوم', 'الكلور', 'الأكسجين'],
  'answer': ['البوتاسيوم']},
```

Figure 2.5 – Extrait du dictionnaire généré

2.3.5 Structuration des données

Dans le cadre du fine tuning, la structure des données est un détail crucial à ne pas négliger. En effet, l'efficacité de l'apprentissage peut être considérablement affectée par une structure inadéquate ou mal organisée. Après l'exploration de quelques structures existantes, la structure Alpaca s'est montrée la plus efficace. Cette structure est principalement utilisée avec les modèles Llama, mais on peut aussi l'utiliser avec d'autres LLMs, construite de trois champs : entrée (instruction), contexte et sortie. Ce format s'est adapté à nos données facilement. Il est simple à traiter et compatible avec un large éventail de LLMs durant l'entraînement et sa flexibilité nous permettra de l'utiliser pour la comparaison de divers modèles.

Dans la figure 2.6 ci-dessous, nous pouvons voir la structure plus clairement après le chargement du dataset sur le site de Hugging Face, l'instruction est commune, la catégorie et le niveau comme contexte et la question en sortie.

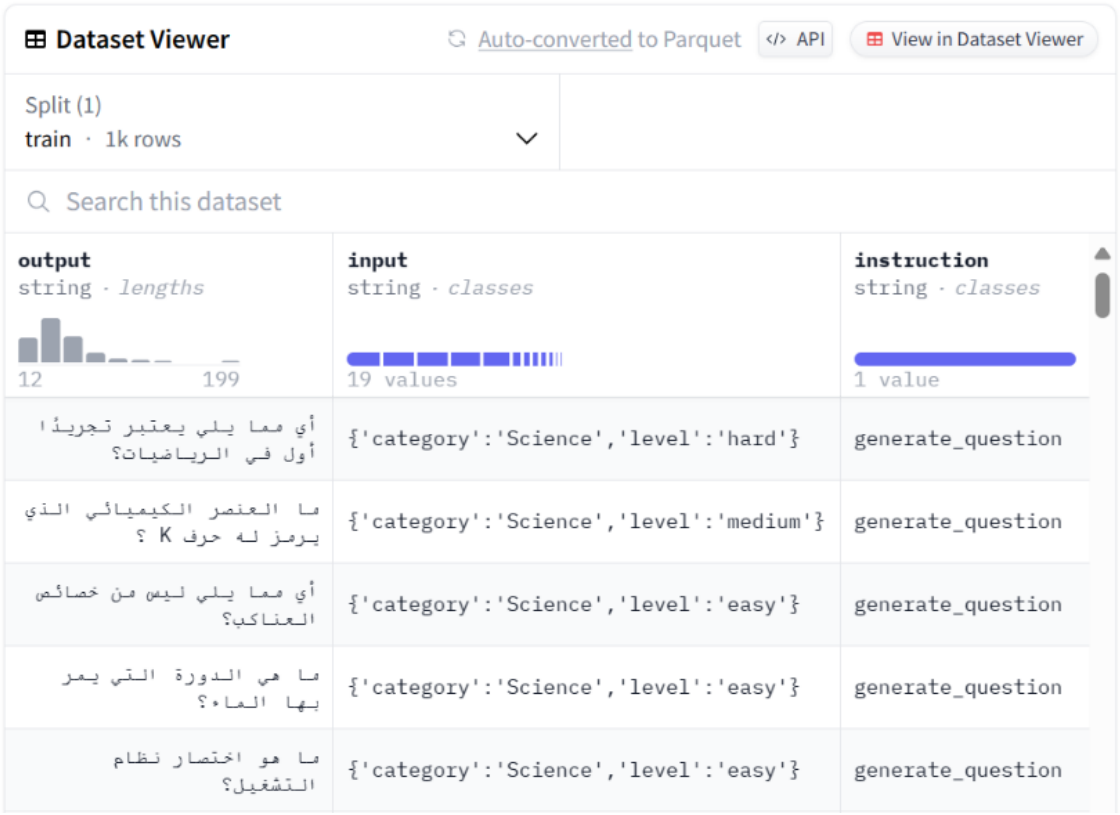

FIGURE 2.6 – La structure du dataset

2.3.6 Augmentation des données

Dans le cadre du fine tuning, la structure des données est un détail crucial à ne pas négliger. En effet, l'efficacité de l'apprentissage peut être considérablement affectée par une structure inadéquate ou mal organisée. Après l'exploration de quelques structures existantes, la structure Alpaca s'est montrée la plus efficace. Cette structure est principalement utilisée avec les modèles Llama, mais on peut aussi l'utiliser avec d'autres LLMs, construite de trois champs : entrée (instruction), contexte et sortie. Ce format s'est adapté à nos données facilement. Il est simple à traiter et compatible avec un large éventail de LLMs durant l'entraînement et sa flexibilité nous permettra de l'utiliser pour la comparaison de divers modèles.

Dans la figure 2.7 ci-dessous, nous pouvons voir la structure plus clairement après le chargement du dataset sur le site de Hugging Face, l'instruction est commune, la catégorie et le niveau comme contexte et la question en sortie.

```
Entrée [572]: qsts['input'].value_counts()

   Out[572]: input
             {'category':'History','level':'easy'}          118
             {'category':'Geography','level':'hard'}        102
             {'category':'Science','level':'hard'}          100
             {'category':'Religion','level':'hard'}         100
             {'category':'Literature','level':'hard'}        99
             {'category':'Science','level':'easy'}           97
             {'category':'Geography','level':'easy'}         95
             {'category':'History','level':'hard'}           93
             {'category':'Literature','level':'medium'}      92
             {'category':'Geography','level':'medium'}       83
             {'category':'Religion','level':'medium'}        80
             {'category':'Religion','level':'easy'}          78
             {'category':'Literature','level':'easy'}        75
             {'category':'Science','level':'medium'}         43
             {'category':'History','level':'medium'}         21
             Name: count, dtype: int64
```

Figure 2.7 – Dataset avant l'augmentation

Au cours de cette étape, en vérifiant les questions manuellement, nous avons aussi constaté la présence de questions mal classées par rapport à leurs niveaux respectifs, Par conséquent, nous avons procédé à un nettoyage manuel et reclassé ces données. La figure 2.8 décrit un exemple d'une question mal classée (la question 551) où nous pouvons voir que son niveau de difficulté est défini comme difficile alors que cette dernière est une question facile.

```
549
ما هو مصطلح السنة كما ورد في النص؟
551
ما هي اللغة التي كُتب بها الجزء الأكبر من العهد الجديد في الكتاب المقدس المسيحي؟
552
ما هو المصطلح الإسلامي الذي يشير إلى حياة الأمم قبل الإسلام؟
553
ما هو الاسم الذي كانت تُعرف به الإلهة إنانا عند البابليين والآشوريين؟
```

Figure 2.8 – Traitement des questions dans notre dataset

Voici le nombre de questions pour chaque contexte après l'augmentation et le nettoyage :

```
Entrée [641]: qsts['input'].value_counts()

Out[641]: input
          {'category':'Geography','level':'hard'}       102
          {'category':'Science','level':'medium'}       101
          {'category':'Literature','level':'easy'}      101
          {'category':'History','level':'medium'}       101
          {'category':'Science','level':'hard'}         100
          {'category':'Religion','level':'easy'}        100
          {'category':'Religion','level':'hard'}        100
          {'category':'Geography','level':'medium'}     100
          {'category':'Literature','level':'hard'}       99
          {'category':'Religion','level':'medium'}       99
          {'category':'Science','level':'easy'}          97
          {'category':'History','level':'easy'}          97
          {'category':'Geography','level':'easy'}        95
          {'category':'History','level':'hard'}          93
          {'category':'Literature','level':'medium'}     92
          Name: count, dtype: int64
```

FIGURE 2.9 – Dataset après l'augmentation

2.4 Fine Tuning : Entraînement du modèle

Avec les données prêtes, on peut passer à l'entraînement des modèles pour générer des questions pertinentes.

2.4.1 LoRA : Une approche efficace pour le fine-tuning

L'approche du fine-tuning doit être rapide pour nous permettre d'en faire plusieurs pour la comparaison des modèles pré-entraînés plus tard, on a besoin dans cette partie de balancer entre la rapidité et l'efficacité. Comme présenté dans le premier chapitre, un fine-tuning complet exige des ressources qu'une machine ou un environnement en ligne ne peut pas toujours offrir. L'approche alternative, le Parameter-Efficient Fine-Tuning (PEFT), permet de pallier ce manque de ressources.Parmi les différentes méthodes de PEFT, telles que Prompt Tuning, LoRA et Prefix Tuning, nous avons choisi LoRA pour plusieurs raisons. LoRA est simple à implémenter, se démarque en termes de vitesse d'entraînement et d'utilisation de la mémoire ,cette méthode peut souvent atteindre des performances similaires, voire surpasser le fine-tuning complet, car le surajustement ou l'oubli des informations précédemment apprises est évité. Pour plus de détails, voir l'annexe. a
.

2.4.2 Exploiter Unsloth pour le fine-tuning

Le problème rencontré même avec une méthode PEFT est l'utilisation de la mémoire, le chargement d'un LLM avec les méthodes traditionnelles et la sauvegarde après la formation nécessitent beaucoup de mémoire et le réglage fin nécessite un ou plusieurs GPU performant. Unsloth est une plateforme qui permet un réglage fin des LLM sur un seul GPU, il prétend être 30 fois plus rapide qu'un réglage fin habituel, c'est la raison pour laquelle nous avons décidé de l'utiliser comme auxiliaire pour notre fine tuning. Pour plus de détails, voir l'annexe. b .

2.4.3 Analyse comparative des LLM

Expérimenter avec différents LLMs est une phase importante de l'etude. En effet, le choix du LLM approprié pour une tâche donnée peut avoir un impact significatif sur la qualité et la pertinence des questions générées. Les modèles que nous avons choisis pour analyser leur performance pour la génération de questions sont : Llama2 7b, Llama3 7b, Mistral 7b et Gemma 7b. Ces modèles sont les plus connus offerts par la Bibliothèque Unsloth. Dans le chapitre qui suit, nous allons établir une base de comparaison équitable en choisissant des conditions favorables pour les LLMs, c'est-à- dire les hyper paramètres avec lesquels le modèle donne les meilleurs résultats. Pour l'évaluation des performances de LLMs formés, il est relativement difficile de l'automatiser. La solution appliquée pour remédier au manque de métriques consiste à s'appuyer sur les connaissances humaines pour avoir une idée de la qualité des questions. Nous allons examiner dans le chapitre suivant les questions produites par chaque LLM pour les 15 combinaisons possibles de catégorie-difficulté (5 catégories de question * 3 niveaux de difficultés), en tenant compte de la clarté de la question, l'exactitude factuelle (réponses correctes), la performance par rapport à la catégorie et au niveau. Nous allons également considérer la perte (loss) finale atteinte par chaque modèle après le fine-tuning, Unsloth a optimisé le calcul de la perte d'entropie croisée pour réduire considérablement la consommation de mémoire . Voici la formule de la perte par entropie croisée :

$$H(P, Q) = -\sum_{x} P(x) \log Q(x) \tag{2.1}$$

2.5 Approfondissement de l'adaptabilité et de l'expérience utilisateur

2.5.1 Notre approche de l'apprentissage adaptatif

AraQuiz utilise des techniques d'apprentissage adaptatif pour répondre aux rythmes d'apprentissage et préférences de ses utilisateurs. Cette adaptabilité est facilitée par un mécanisme de score qui ajuste dynamiquement la difficulté des questions en fonction des performances et de l'engagement de l'utilisateur dans une catégorie spécifique.

2.5.2 Ajustement de la difficulté en fonction des séries de bonnes réponses

L'algorithme ajuste dynamiquement le niveau de difficulté des questions en fonction de la série de bonnes réponses de l'utilisateur. Si l'utilisateur démontre une capacité constante à répondre correctement aux questions (série positive), le niveau de difficulté augmente progressivement pour offrir des questions plus stimulantes. Inversement, si l'utilisateur rencontre des difficultés (série négative), le niveau de difficulté diminue pour proposer des questions plus accessibles. Ce mécanisme adaptatif garantit que l'expérience du quiz reste attrayante et adaptée au niveau de l'utilisateur, favorisant l'apprentissage continu et la progression.

2.5.3 Système de score

L'algorithme utilise un système de score pour suivre les performances de l'utilisateur tout au long du quiz. Les bonnes réponses contribuent au score de l'utilisateur, ce qui encourage la précision et récompense la maîtrise. De plus, le système de score facilite la rétroaction en présentant le score final de l'utilisateur à la fin du quiz, fournissant une mesure de réussite et encourageant l'amélioration. En quantifiant le succès de l'utilisateur, le système de score renforce la motivation et encourage la participation active au processus d'apprentissage.

L'algorithme de score est le suivant :

Algorithm 1 Algorithme d'apprentissage adaptatif

Entrée : Aucune
Sortie : Score final sur 15, Niveau de complexité final

```
Score ← 0  Niveau actuel ← "facile"  Streak ← 0  Question précédente correcte ← faux
for i = 1 à 15 do
    SélectionnerQuestion(Niveau actuel)
     Afficher(question)  Réponse de l'utilisateur ← ObtenirRéponse()
     if VérifierRéponse(Réponse de l'utilisateur) then
        Score ← Score + 1  if Question précédente correcte then
            Streak ← Streak + 1
        end
        else
            Streak ← 1
        end
        Question précédente correcte ← vrai
    end
    else
        if Streak > 0 then
            Streak ← 0
        end
        else
            Streak ← Streak- 1
        end
        Question précédente correcte ← faux
    end
    AjusterNiveauDifficulté(Streak, Niveau actuel)
end
Afficher(Score final : Score sur 15)  Afficher(Niveau de complexité final : Niveau actuel)
```

L'algorithme de score vise à trouver un équilibre entre défi et soutien en adaptant dynamiquement le niveau de difficulté tout en maintenant l'engagement de l'utilisateur. En surveillant la série de bonnes réponses de l'utilisateur, l'algorithme garantit que le quiz reste suffisamment stimulant pour stimuler l'apprentissage et l'engagement cognitif sans surcharger l'utilisateur. Simultanément, il fournit un soutien en ajustant le niveau de difficulté en réponse aux compétences de l'utilisateur, garantissant une expérience d'apprentissage aussi optimale que possible et adaptée aux niveaux de compétence individuels. Cet équilibre favorise un environnement d'apprentissage positif où les utilisateurs sont stimulés et soutenus de manière appropriée, favorisant le développement continu des compétences et la rétention des connaissances.

2.6 Conlusion

Ce chapitre nous a permis de présenter les étapes de création de notre dataset en utilisant différentes APIs, les choix conceptuels en termes d'entraînement, et la démarche à suivre lors de la comparaison des LLMs. Le chapitre suivant sera consacré aux aspects techniques liés au développement et à l'expérimentation de notre solution.

CHAPITRE 3

IMPLEMENTATION

3.1 Introduction

Notre projet visait à créer un outil d'apprentissage de la culture générale arabe en utilisant des LLMs. **AraQuiz**, l'objectif principal, était d'améliorer la compréhension des apprenants en leur fournissant une variété de textes arabes couvrant des sujets divers tels que l'histoire, la géographie, la littérature, les sciences et les arts. Ce projet a permis aux utilisateurs d'approfondir leur compréhension des textes arabes et de développer leur connaissance du contexte culturel.

Dans ce chapitre, nous avons exploré l'aspect technique de notre projet, en détaillant le processus de programmation, les fonctionnalités développées, et l'implémentation de l'adaptabilité pour offrir une expérience utilisateur optimale. La réussite de **AraQuiz** repose sur des technologies de pointe, des frameworks robustes et des méthodologies innovantes. Nous avons créé une application web sophistiquée en utilisant Google Colab pour l'entraînement des LLMs et VSCode pour le développement local.

3.2 L'environnement de développement

Notre projet se compose de deux parties : l'entraînement du LLM et la construction de l'application fonctionnelle en reliant le LLM aux fonctionnalités de l'application web. Nous utilisons deux environnements différents, présentés ci-dessous.

3.2.1 Google Colab Notebook

Nous avons utilisé la plateforme cloud de Google en ligne pour entraîner notre LLM car elle offre de meilleures conditions que l'environnement local, telles que le GPU, la RAM et la puissance du CPU. De plus, nous avons utilisé Colab pour tester les différents LLMs que nous avons affinés afin de choisir le meilleur en fonction de ses performances.

3.2.2 VsCode

Nous avons utilisé l'environnement local pour coder l'intégralité du front-end et du back-end de notre application web et pour construire les principaux composants de l'application.

3.3 Entraînement des LLM

Dans cette étape, nous rassemblons les modèles candidats avec les données pour les entraîner et sélectionner le meilleur.

3.3.1 Analyse comparative des LLM

Nous avons débuté en effectuant une recherche par grille (Grid Search) pour optimiser les performances de Gemma, en ajustant certains hyperparamètres clés. Pour ce faire, nous avons exploré différentes valeurs pour chaque hyperparamètre afin de déterminer les combinaisons les plus efficaces. Voici les hyperparamètres que nous avons ajustés, ainsi que les plages testées pour chaque hyperparamètre et les valeurs optimales sélectionnées :

Batch size : 1-3 Meilleure valeur sélectionnée : 3.

Steps number : 60,92,100 Meilleure valeur sélectionnée : 92 .

Learning rates 1e-3,1e-4,2e-5

Meilleure valeur sélectionnée : 1e-4 .

Ces ajustements se sont avérés bénéfiques pour chaque modèle, contribuant à des performances optimales pour une comparaison plus juste. Il est essentiel de noter que l'ensemble de données initialement utilisé pour la comparaison contenait l'ensemble des QCM avec des options et une réponse correcte. Cette inclusion nous a aidés à évaluer la capacité factuelle des modèles, facilitant ainsi la sélection des meilleures configurations pour Gemma.

Dans la section suivante, nous présenterons les résultats détaillés de cette analyse comparative, en mettant en lumière les forces et les faiblesses de chaque LLM :

LLama 2 :

- **Clarté :**
  - Ambiguïté
  - Incohérence
- **Exactitude factuelle :**
  - Des réponses fausses
- **Performance par rapport à la catégorie :**
  - Plutôt correcte

LLama 3 : performances globales bien meilleures que celles de Llama 2

- **Clarté :**
  - Grammatiquement bonne
  - Claire et bien formulée
- **Exactitude factuelle :**
  - La plupart des réponses sont incorrectes
- **Performance par rapport à la catégorie :**
  - Questions adéquates aux catégories

mauvaise réponse :

{'category': 'Geography', 'level': 'easy',
'question':'ما هي أكبر جزيرة في العالم؟',
'options':['جزر الأنتيل الصغرى و الكبرى','جزر الأنتيل','جزر الأنتيل الكبرى','جزر الأنتيل الصغرى'],
'answer':[جزر الأنتيل الصغرى]}

bonne réponse :

{'category': 'Geography', 'level': 'hard',
'question':'ما هي العاصمة الإقليمية لولاية اورينتال في كاليفورنيا؟',
'options':...],
'answer':[سان فرانسيسكو']}

Mistral 7b :

- **Clarté :**
  - Mauvaises performances en langue arabe
- **Exactitude factuelle :**
  - Toutes les options et réponses sont fausses
  - Les options répétitives et pas claires
- **Performance par rapport à la catégorie :**
  - Plus ou moins correcte

examples :

{'category': 'Science', 'level': 'easy',
'question':'ما هو العنصر الكيميائي الذي يتكون من عنصر كيميائي و يتم استخدامه في صناعة الاسمدة؟',
'options':['النيتروجين','الاكسجين','الكلور','الفلور'],
'answer':['النيتروجين']}

{'category': 'Science', 'level': 'medium',
'question':'ما هي علامة المعادلة المستقبلية للعد المستقبلي؟',
'options':['n^2','n^3','n^4','n^5"],
'answer':['n^2']}

{'category': 'History', 'level': 'hard',
'question':'répétition d'une question incompréhensible..

Gemma 7b :

- **Clarté :**
  - Langage simple et compréhensible
  - Questions claires
- **Exactitude factuelle :**
  - Beaucoup de réponses sont fausses mais c'est les meilleurs résultats par rapport aux autres modèles
- **Performance par rapport à la catégorie :**
  - Bonne performances

exemples : bonne réponse :

{'category': 'Science', 'level': 'hard',
'question':'X ما هي الوظيفة الرئيسية للكروموسوم',
'options':[...],
'answer':[تخزين المعلومات الجينية و تنظيمها و تعبئتها و تخزينها]}

mauvaise réponse :

{'category': 'Geography', 'level': 'medium',
'question':'ما هي أكبر مدينة في المغرب؟',
'options':['طنجة','فاس','الدار البيضاء','الرباط'],
'answer':['الرباط']}

Les problèmes commun entre tous les modèles et leurs solutions :

Tous les modèles ont mal performé en ce qui concerne le niveau de difficulté. C'est la raison pour laquelle nous avons fait l'augmentation de données. Cependant, les résultats de cette étude comparative restent tout de même valides, et en les observant nous avons pu décider que le meilleur modèle à utiliser est Gemma.
Le deuxième point commun entre les LLMs analysés consiste on la non véracité des réponses générées ce qui fait qu'entrainer le modèle sur la génération des questions seulement et utiliser une API pour les options et réponses sera la meilleure solution afin d'assurer la génération de QCM de qualité. L'API Gemini était bien adaptée à cette tâche et donnait des réponses précises la plupart du temps.
La figure suivante montre une invite de discussion demandant à générer les options pour une question que le modèle Gemma a générée avec une mauvaise réponse, après le fine tuning :

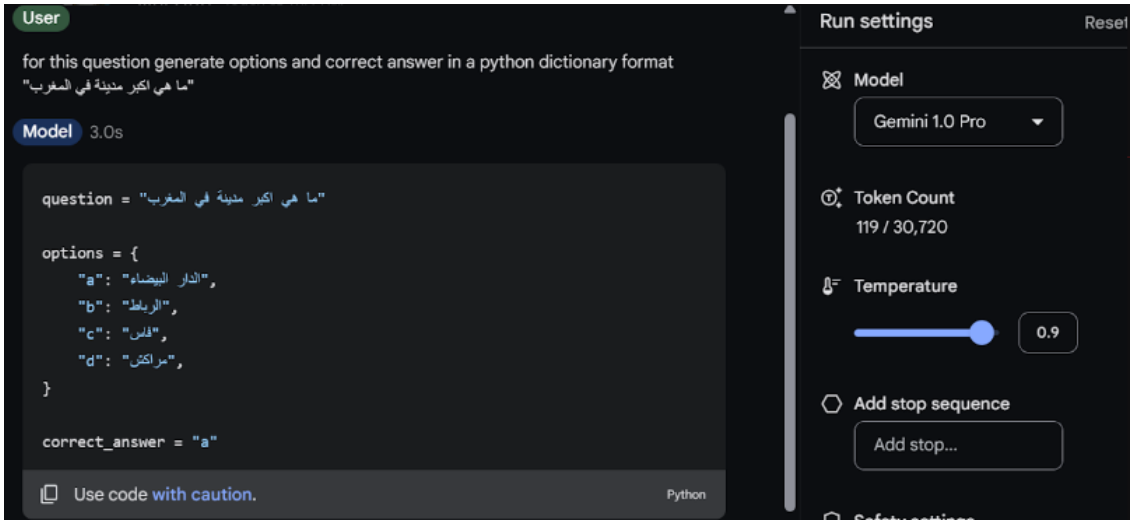

FIGURE 3.1 – API de Gemini

L'API a généré une réponse correcte et des options de qualité

3.4 Optimisation des données et des hyperparamètres

Dans ce fine-tuning, nous avons utilisé le dataset augmenté et nettoyé au format Alpaca, puis procédé à l'optimisation des hyperparamètres.

Hyperparamètres :

Taille du lot d'entraînement par apparei : 4. Nombre d'étapes d'accumulation du gradient : 4. Nombre de pas : 92. Taux d'apprentissage : 1×10^{-4} .

Le choix du taux d'apprentissage s'est basé sur une série de tests utilisant une approche logique dichotomique. En ce qui concerne les autres paramètres, leur sélection visait à garantir que le modèle puisse parcourir un maximum de points de données. Notamment, cela est d'autant plus crucial étant donné que la taille du jeu de données est de 1477 entrées. Pour avoir une idée du dimensionnement du modèle par rapport à la taille des données, nous avons calculé le produit de la taille du lot (batch size), du nombre d'étapes (steps), et du nombre de pas (epochs), qui donne 4 $\times$ 4 $\times$ 92 = 1472. Cette approche vise à optimiser l'efficacité de l'apprentissage tout en exploitant au mieux les informations disponibles dans le jeu de données.

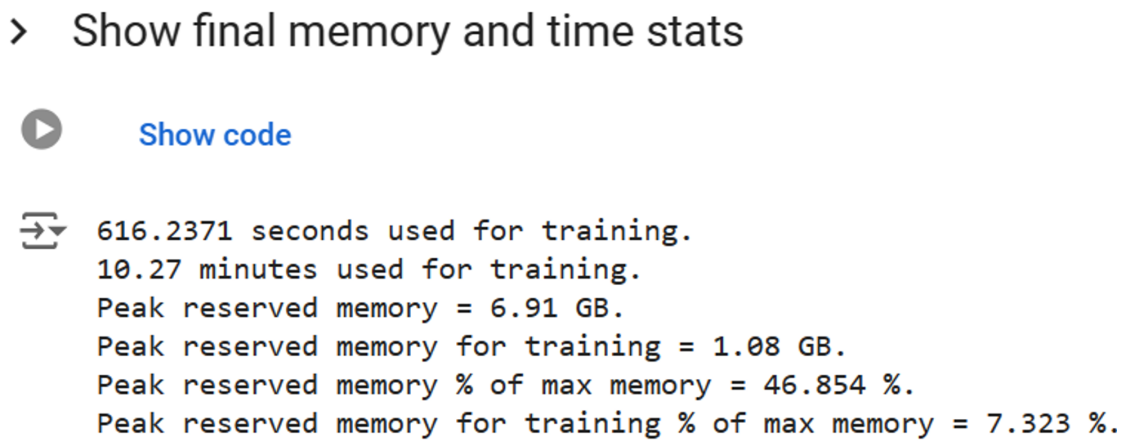

FIGURE 3.2 – Stats de Temps et Mémoires d'entrainement

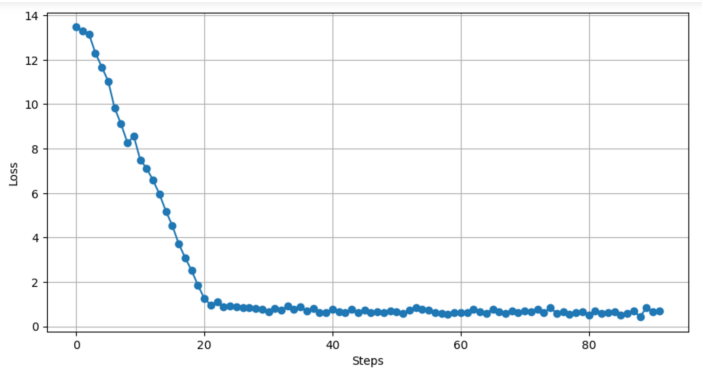

FIGURE 3.3 – la fonction erreur (Loss Function)

Exemples de bonnes questions :

{'category': 'Science', 'level': 'medium',
'question':'ما هي العلاقة بين الكتلة والكتلة الحرارية'}

{'category': 'History', 'level': 'easy',
'question':'ما هي أهم الأحداث التي شهدتها أوروبا في القرن الثامن عشر'}

Exemples de mauvaises questions :

{'category': 'Geography', 'level': 'medium',
'question':'ما هي عاصمة ولاية الجزائر'}

{'category': 'Literature', 'level': 'hard',
'question':'ما هو الاسم الذي أطلق على كتاب الجامعي الذي ألفه ابن خلدون'}

Un livre qui n'existe pas (hallucinations)

3.5 Bibliothèques, frameworks et outils

3.5.1 Intégration d'API externes

L'API Gemini a été integrée comme le montre le code suivant pour générer les options et les réponses.

3.5.2 Streamlit

Streamlit est un framework Python que nous utilisons pour construire l'interface utilisateur web dynamique de notre application LLM. Nos utilisations de Streamlit :

– **Gestion de la mise en page** : Les fonctions `st.write` et autres fonctions de mise en page de Streamlit aident à organiser le contenu sur la page web.

- **Widgets interactifs** : Nous utilisons des widgets pour les "boutons" et les "listes déroulantes" afin de capturer la saisie de l'utilisateur pour la catégorie et de déclencher la génération de QCM.
- **Gestion de l'état de session** : `st.session_state`
  Cette fonction permet de maintenir et de suivre les entrées, les donnes et les sélections de l'utilisateur sur différentes interactions. Ceci est important en particulier lorsque chaque interaction de l'utilisateur implique généralement un aller-retour vers le serveur, ce qui pourrait entraîner une perte d'état sans une gestion appropriée.
  * Exemple d'utilisation dans le projet :
    1. Stockage des questions générées et des sélections de l'utilisateur .
    2. Gestion des sélections de l'utilisateur.
- **Style personnalisé**
  L'utilisation d'un style personnalisé dans notre application Streamlit améliore l'expérience utilisateur en fournissant un retour visuel et en rendant l'interface plus attrayante. Le CSS personnalisé est injecté dans une application Streamlit à l'aide de la fonction `st.markdown` avec le paramètre `unsafe_allow_html=True`. Cela permet à l'application d'inclure du HTML et du CSS pour le style des composants.

Unsloth

Comme nos ressources de calcul sont limitées, nous utilisons Unsloth, une bibliothèque conçue pour optimiser les performances des modèles volumineux pendant l'adaptation par adaptateur. Nous avons utilisé Unsloth à la fois dans la phase d'affinage et dans la phase de chargement au sein de l'application elle-même. Cela facilite à la fois le processus d'affinage et la génération de questions à choix multiples par la suite.

Les principaux avantages de l'utilisation d'Unsloth sont la réduction du temps de chargement et la diminution de la puissance de calcul requise par rapport à d'autres méthodes, ce qui la rend viable sur du matériel aux capacités limitées.

Dans notre projet, nous exploitons Unsloth pour charger notre modèle affiné, Gemma, et son tokenizer en utilisant l'adaptation par adaptateur et la quantification 4 bits. Lorsque l'utilisateur clique sur le bouton "Générer une question", l'application construit une invite pour le modèle. Le modèle génère alors une question en fonction de la catégorie et du niveau de difficulté sélectionnés. L'utilisation de ‘use cache=True‘ garantit que les résultats intermédiaires sont mis en cache, ce qui accélère le processus de génération.

3.5.3 Ngrok

Dans notre projet, nous avons initialement utilisé Ngrok pour créer des tunnels sécurisés afin de tester notre application Streamlit directement depuis Google Colab. Ngrok est un outil qui expose des serveurs locaux à Internet via des tunnels sécurisés, ce qui est particulièrement utile pour le développement et les tests. Cette configuration nous permet de générer une URL publique qui peut être utilisée pour accéder à l'application Streamlit exécutée dans Colab. Ceci est très pratique pour les tests en temps réel et le partage de l'application avec les membres de l'équipe pour un développement collaboratif.

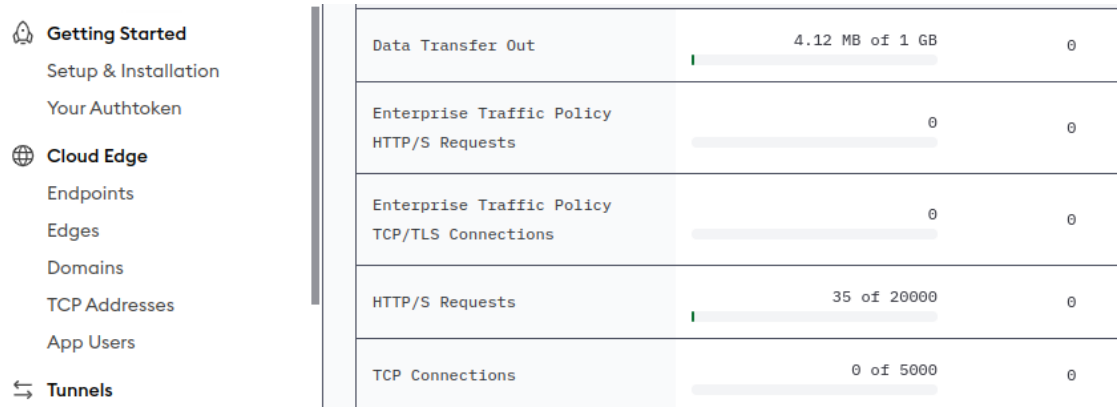

FIGURE 3.4 – Configuration de ngrok pour l'hébergement local

Malgré ses avantages, l'utilisation de Ngrok en combinaison avec Colab a posé plusieurs défis, tels que :

- **Durée de session limitée** : Les sessions Google Colab peuvent être interrompues de manière inattendue, et les tunnels Ngrok sont également soumis à des délais, ce qui les rend impraticables pour une utilisation à long terme.
- **Bande passante et vitesse** : Les versions gratuites de Ngrok ont des limitations de bande passante et peuvent être plus lentes par rapport aux déploiements locaux.

En raison de ces limitations, nous avons partiellement abandonné l'utilisation de Ngrok pour le développement continu. En revanche, à des fins de déploiement, nous utilisons toujours Ngrok si nécessaire pour partager l'application sans avoir à mettre en place un environnement de production complet.

FIGURE 3.5 – Premières étapes de la création de notre application avec ngrok

3.5.4 sqlite3

Dans notre application, nous intégrons SQLite3 pour gérer l'authentification des utilisateurs et suivre leurs performances. SQLite3 est une base de données légère basée sur disque qui ne nécessite pas de processus serveur distinct, ce qui en fait un choix idéal pour les applications de petite à moyenne taille.Nous l'utilisons principalement pour trois tâches :

- **Gestion de l'authentification des utilisateurs** : connexion et authentification des utilisateurs.
- **Suivi des performances des utilisateurs** : stocker les données de performance, telles que l'identifiant de l'utilisateur, l'identifiant de la question, la réponse sélectionnée et la justesse.
- **Enregistrement des performances** : chaque fois qu'un utilisateur interagit avec les questions générées par le LLM, ses performances sont enregistrées.

3.5.5 Langchain

Langchain est un framework puissant qui simplifie le travail avec les grands modèles, en fournissant des outils performants pour la gestion des invites, le suivi de la mémoire et la maintenance de l'historique des utilisateurs. Son intégration a non seulement rationalisé notre processus de développement, mais a également considérablement amélioré la fonctionnalité et l'expérience utilisateur de notre application.

Nous avons integré Langchain pour les fonctionnalités suivants : :

- **Modèles d'invite de formatage (ingénierie d'invite)** : permet de tirer parti des capacités du LLM pour générer des questions de haute qualité.
- **Suivi de la mémoire** : il permet de suivre les interactions de l'utilisateur, car le modèle doit se souvenir des entrées et des sorties précédentes pour fonctionner correctement.
- **Gestion de l'historique utilisateur** : cela aide à fournir une expérience personnalisée, car LangChain facilite la journalisation des interactions des utilisateurs, qui peuvent être utilisées pour suivre les performances des utilisateurs au fil du temps et adapter les prochaines questions en fonction des interactions passées, ceci qui contribue finalement à rendre notre application plus adaptative à l'utilisateur, et crée donc une meilleure expérience d'apprentissage.

3.6 Fonctionnalités de l'application AraQuiz

Nous couvrons dans cette section les fonctionnalités de l'application du point de vue de l'utilisateur :

3.6.1 Sélection de la catégorie

Comme illustré dans la figure 3.22, l'utilisateur commence par sélectionner une catégorie d'intérêt parmi une liste prédéfinie, telle que la littérature, les sciences, la religion, la géographie ou l'histoire. Cette sélection est le principal moteur de l'expérience d'apprentissage, guidant le système pour adapter le contenu et les questions pertinents à la catégorie choisie.

3.6.2 Génération de questions

Une fois que l'utilisateur a sélectionné une catégorie, **AraQuiz** utilise le LLM pour générer dynamiquement un ensemble de questions liées à cette catégorie. Ces questions couvrent une gamme de sujets au sein de la catégorie sélectionnée, offrant une expérience d'apprentissage complète à l'utilisateur.

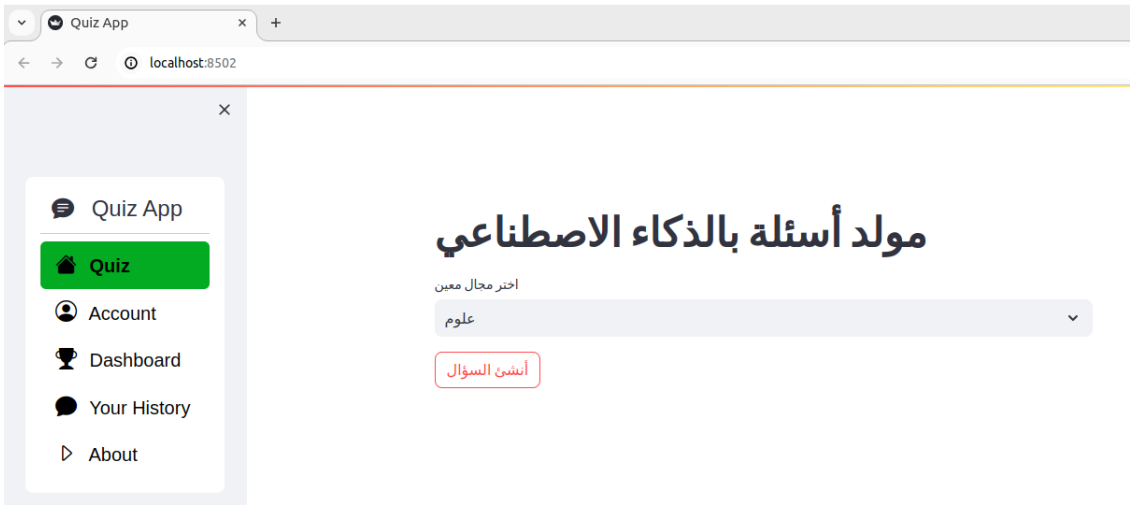

FIGURE 3.6 – Page d'accueil de l'application de quiz - Générateur de questions par IA

3.6.3 Options de réponse et génération de réponse

À côté de chaque question, **AraQuiz** propose plusieurs options de réponse générées via l'API Gemini. L'utilisateur sélectionne une réponse parmi les options proposées, indiquant sa réponse à la question. Lors de la sélection, le système utilise le LLM pour générer une réponse, qui peut inclure des commentaires sur la justesse de la réponse ou offrir des explications et des informations supplémentaires liées à la question.

FIGURE 3.7 – Exemple de Quiz avant de choisir la réponse

FIGURE 3.8 – Exemple de Quiz après saisie de la reponse par l'utilisateur

Après chaque réponse, le résultat révélera si la réponse est correcte ou incorrecte, accompagnée d'une explication supplémentaire pour comprendre la raison derrière la réponse. Cela vise à améliorer la compréhension des questions et des réponses.
Chaque question dispose d'un délai d'une minute. Si le minuteur expire avant que l'apprenant ne choisisse une réponse, le système passera automatiquement à la question suivante, et aucun point ne sera accordé pour la question non répondue.

3.6.4 Apprentissage adaptatif et suivi des progrès

Tout au long de l'interaction, AraQuiz s'adapte aux performances et à l'engagement de l'apprenant. Les réponses correctes et incorrectes entraînent des ajustements adaptatifs, tels que l'ajustement du niveau de difficulté des questions suivantes ou la fourniture d'explications ciblées pour renforcer l'apprentissage. L'application suit les progrès de l'apprenant dans la catégorie sélectionnée, fournissant des informations sur son parcours d'apprentissage et les points à améliorer.

3.6.5 Interface utilisateur et interaction

L'application AraQuiz propose une interface conviviale via le framework web Streamlit. Les utilisateurs interagissent avec l'application en sélectionnant des catégories, en répondant à des questions et en accédant aux commentaires et aux explications. L'interface est conçue pour être intuitive et accessible, facilitant une expérience d'apprentissage harmonieuse pour les utilisateurs de niveaux et d'horizons différents.
Voici deux exemples de questions générées dans les catégories histoire et religion, respectivement.

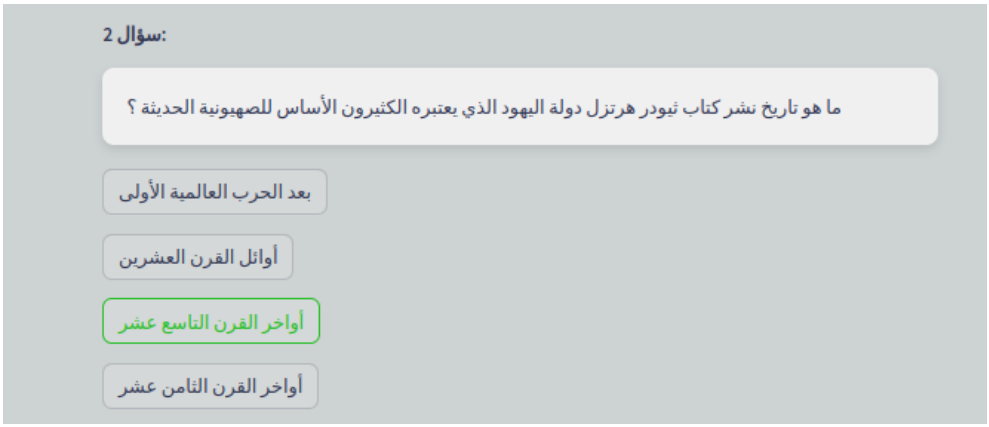

FIGURE 3.9 – Question de la catégorie Histoire

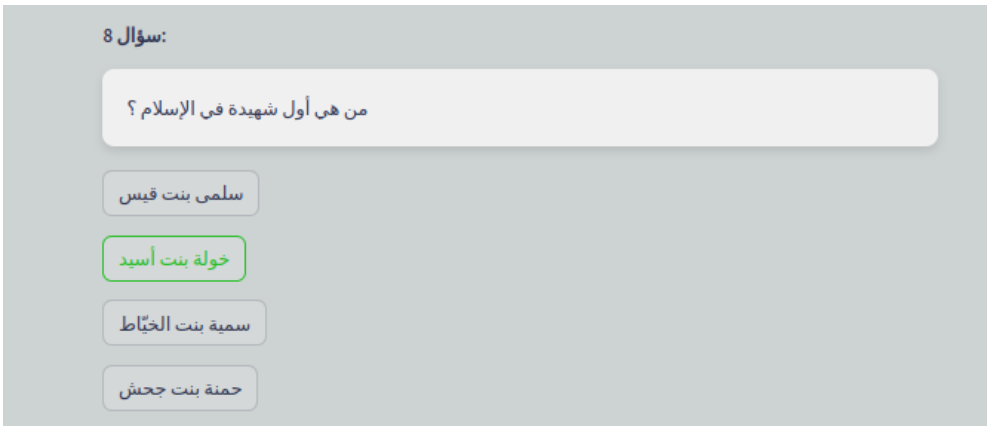

FIGURE 3.10 – Question de la catégorie Religion

3.7 Conclusion

En conclusion, notre projet a réussi à démontrer la puissance des grands modèles de langage pour l'apprentissage de la culture générale arabe. En utilisant des LLMs capables de comprendre et de générer du texte en Arabe, nous avons créé un outil qui enrichit l'expérience d'apprentissage des utilisateurs. Nous avons relevé plusieurs défis techniques et conceptuels, et grâce à l'utilisation de technologies de pointe et de méthodologies innovantes, nous avons développé AraQuiz, une application web performante et évolutive. Cette application permet aux utilisateurs de développer leur connaissance de la culture arabe dans un environnement interactif et engageant. Les futures améliorations de l'application continueront à enrichir l'expérience utilisateur et à promouvoir une compréhension plus profonde de la culture générale arabe.

CONCLUSION GÉNÉRALE

Ce travail a porté sur le développement d'un outil d'apprentissage de la culture générale en langue arabe en exploitant la puissance des LLMs et leur fine-tuning. Premièrement, l'API de ChatGPT a été exploitée pour la classification de textes et celle de Gemini pour la génération de questions, dans le but de construire notre jeu de données, contenant des questions dans cinq catégories ; Histoire, Géographie, Science, Littérature et Religion. Nous avons comparé les performances de nouveaux LLM, ce qui nous a permis d'accélérer le processus de fine-tuning et d'en accroître l'efficacité en utilisant l'architecture LoRA via la librairie Unsloth. Ces techniques nous ont permis d'atteindre des résultats satisfaisants.

Dans un second temps, nous avons développé AraQuiz, un outil d'apprentissage de la culture générale en langue arabe, en utilisant la librairie Streamlit. AraQuiz se veut adaptatif et ludique, intégrant des éléments de gamification pour rendre l'expérience d'apprentissage plus engageante. Le score obtenu par l'utilisateur joue un rôle clé dans l'adaptabilité de l'application, en ajustant le niveau de difficulté des questions et en proposant un contenu plus pertinent en fonction des performances individuelles. Parmi ses principales fonctionnalités, AraQuiz offre également la génération de quizzes variés et stimulants, une interface conviviale, un suivi des progrès des utilisateurs, ainsi qu'une personnalisation des questions en fonction des intérêts et des performances individuelles.

Cependant, comme tout produit performant voué à l'amélioration continue, en particulier dans le domaine de l'éducation, nous visons à continuer à améliorer notre application. Voici les améliorations futures proposées :

- Diversité du contenu : Augmenter la variété et la profondeur des questions sur différents sujets et niveaux de difficulté. Cela pourrait inclure l'ajout de questions sur l'actualité, la culture populaire, la science et d'autres domaines d'intérêt pour les utilisateurs.
- Apprentissage automatique : Utiliser des algorithmes d'apprentissage automatique pour prédire plus précisément le niveau de difficulté approprié en fonction des modèles de réponse d'un utilisateur, améliorant la personnalisation au fil du temps. Cela pourrait impliquer l'utilisation de techniques d'apprentissage par renforcement pour optimiser l'expérience d'apprentissage pour chaque utilisateur.
- Réalisations et récompenses : Introduire des badges, des classements et d'autres récompenses pour motiver les utilisateurs et favoriser un sentiment de réussite. Cela peut inclure des récompenses virtuelles, des certificats ou même des prix physiques pour les utilisateurs qui atteignent certains objectifs.

- Défis : Créer des défis à durée limitée ou des compétitions pour stimuler l'engagement et rendre l'apprentissage plus dynamique. Cela peut impliquer des défis individuels ou d'équipe, avec des classements et des récompenses pour les meilleurs joueurs.
- Génération de questions en temps réel : La mise en œuvre de RAG permettra la génération de questions contextuellement pertinentes en temps réel en récupérant et en synthétisant des informations provenant d'un vaste corpus. Cela garantira que les utilisateurs reçoivent toujours les questions les plus récentes et les plus précises, en tenant compte de leurs intérêts et de leurs progrès.
- Pensée critique : Incorporer des questions ouvertes pour promouvoir la pensée critique et développer les capacités de rédaction des apprenants. Cela permettrait également d'évaluer les compétences cognitives de niveau supérieur et d'encourager une réflexion plus approfondie sur les sujets abordés.

CHAPITRE 4

ANNEXE

a LoRA : Une approche efficace pour le fine-tuning

LoRA, ou Low Rank Adaptation, est une méthode de fine-tuning (Parameter-efficient Fine-tuning, PEFT) ; elle réduit le nombre de paramètres formables. Au lieu de mettre à jour tous les poids, cette approche se concentre sur l'adaptation d'un sous-ensemble de poids qui est considéré comme le plus pertinent pour la nouvelle tâche en cours. Le principe est purement algébrique, la matrice de poids est décomposée en vecteurs, de telle manière à ce que leur multiplication donne la première matrice. Ces vecteurs sont ce qu'on appelle les adaptateurs, une approximation "Low Rank". Après le fine-tuning, ces adaptateurs sont fusionnés avec le LLM de base.

b Exploiter Unsloth pour le fine-tuning

Les méthodes usuelles pour le **fine-tuning** telles que **LoRA** et **QLoRA**, la plus connue et utilisée étant par l'auxiliaire de la bibliothèque **PEFT** et **Hugging Face** pour le chargement du modèle, présentent plusieurs défis. Le principal défi que nous avons rencontré est l'épuisement des ressources offertes par le **notebook Colab** avant de commencer l'entraînement, parfois même en phase de chargement. Dans le meilleur des cas, l'entraînement prend du temps et les résultats sont loin d'être satisfaisants.

Unsloth est une startup qui crée des produits de IA. En décembre 2023 ils ont fait leur premier lancement "Unsloth" qui a rapidement gagné une grande popularité à cause de son efficacité, ils affirment que son utilisation rend le fine-tuning des LLM 30 fois plus rapide avec 60% moins d'utilisation de mémoire (la version open source gratuite rend le fine-tuning deux fois plus rapide avec 50% de mémoire en moins) grâce à plusieurs optimisations manuelles et des changements radicaux. Unsloth peut être intégré facilement avec les méthodes utilisées comme PEFT et TRL pour optimiser le processus d'entraînement. On peut également l'utiliser avec les adaptateurs LoRA ou bien QLoRA, ce qui convient à notre plan.

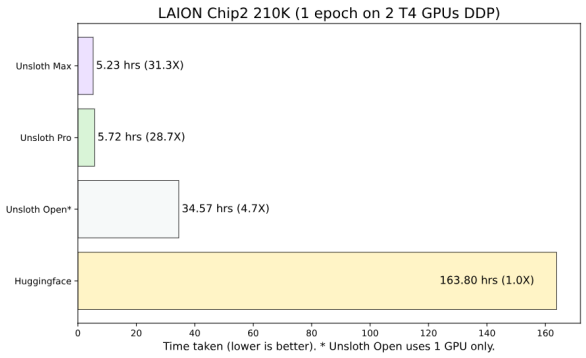

FIGURE 4.1 – Entraînement LAION Chip2 avec DDP sur 2 GPU T4 (1 époque)

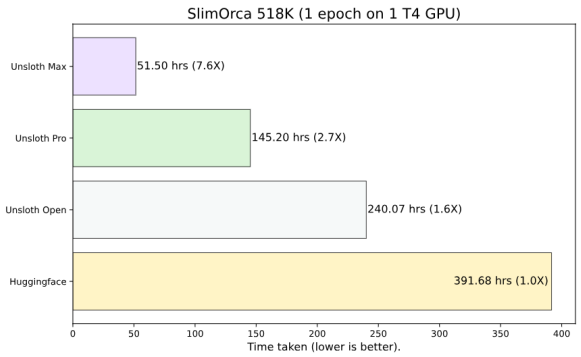

FIGURE 4.2 – Entraînement SlimOrca 518K sur 1 GPU T4 (1 époque)

Comme le montrent les figures 3.13 et 3.14, en utilisant le célèbre dataset Alpaca sur une seule GPU Tesla T4, la mise en œuvre originale de Huggingface prend 23 heures et 15 minutes, tandis que notre solution Max ne prend que 2 heures et 34 minutes, soit 8,8 fois plus rapide [30] .

BIBLIOGRAPHIE

[1] Sowmya Vajjala et al. *Practical natural language processing : a comprehensive guide to building real-world NLP systems*. O'Reilly Media, 2020.

[2] Meltem Huri Baturay. "An overview of the world of MOOCs". In : *Procedia-Social and Behavioral Sciences* 174 (2015), p. 427-433.

[3] Firuz Kamalov, David Santandreu Calonge et Ikhlaas Gurrib. "New era of artificial intelligence in education : Towards a sustainable multifaceted revolution". In : *Sustainability* 15.16 (2023), p. 12451.

[4] Sherry Ruan et al. "Quizbot : A dialogue-based adaptive learning system for factual knowledge". In : *Proceedings of the 2019 CHI conference on human factors in computing systems*. 2019, p. 1-13.

[5] Thimira Amaratunga. "Understanding Large Language Models Learning Their Underlying Concepts and Technologies". In : *(No Title)* (2023).

[6] Anne Darche. "L'intelligence artificielle pour les nuls". In : *Gestion* 40.2 (2015), p. 112-114.

[7] Masato Hagiwara. *Real-world natural language processing : practical applications with deep learning*. Simon et Schuster, 2021.

[8] IBM. *What is a neural network ?* Accessed : 2024-04-14. 2024. url : `https://www.ibm.com/topics/neural-networks`.

[9] Geoffrey E Hinton, Simon Osindero et Yee-Whye Teh. "A fast learning algorithm for deep belief nets". In : *Neural computation* 18.7 (2006), p. 1527-1554.

[10] Ruslan Salakhutdinov et Geoffrey Hinton. "Deep boltzmann machines". In : *Artificial intelligence and statistics*. PMLR. 2009, p. 448-455.

[11] Sam Wiseman et Alexander M Rush. "Sequence-to-sequence learning as beam-search optimization". In : *arXiv preprint arXiv :1606.02960* (2016).

[12] Ashish Vaswani et al. "Attention is all you need". In : *Advances in neural information processing systems* 30 (2017).

[13] Simona Petrakieva, Oleg Garasym et Ina Taralova. "http ://ieeexplore. ieee. org/stamp/stamp. jsp ? tp= &arnumber= 7038771". In : (2014).

[14] Humza Naveed et al. "A comprehensive overview of large language models". In : *arXiv preprint arXiv :2307.06435* (2023).

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

[16] OpenAI. *GPT-4*. Accessed : 2024-03-14. 2024. URL : https://www.openai.com/research/gpt-4.

[17] Google. *Google PaLM 2*. Accessed : 2024-03-14. 2024. URL : https://ai.google/discover/palm2/.

[18] Meta. *Llama 3*. Accessed : 2024-03-14. 2024. URL : https://llama.meta.com/llama3/.

[19] Anthropic. *Claude*. Accessed : 2024-03-15. 2024. URL : https://www.anthropic.com/claude.

[20] Falcon. *Falcon LLM*. Accessed : 2024-03-15. 2024. URL : https://falconllm.tii.ae/.

[21] Jais Team. *Jais and Jais-chat : Arabic-Centric Foundation and Instruction-Tuned Open Generative Large Language Models*. Accessed : 2024-03-15. 2023. URL : https://arxiv.org/abs/2308.16149v2.

[22] Technology Innovation Institute. *Technology Innovation Institute Announces Launch of NOOR, the World's Largest Arabic NLP Model*. Accessed : 2024-03-15. 2024. URL : https://www.tii.ae/news/technology-innovation-institute-announces-launch-noor-worlds-largest-arabic-nlp-model.

[23] AraBERT Team. *AraBERT : Transformer-based Model for Arabic Language Understanding*. Accessed : 2024-03-15. 2023. URL : https://arxiv.org/pdf/2312.03727.

[24] All About AI. *Fine-Tuning*. Accessed : 2024-05-20. 2024. URL : https://www.allaboutai.com/ai-glossary/fine-tuning/.

[25] Zihao Fu et al. "On the effectiveness of parameter-efficient fine-tuning". In : *Proceedings of the AAAI Conference on Artificial Intelligence*. T. 37. 11. 2023, p. 12799-12807.

[26] MIT Sloan Educational Technology Office. *Effective Prompts*. Accessed : 2024-03-17. 2024. URL : https://mitsloanedtech.mit.edu/ai/basics/effective-prompts/.

[27] GeeksforGeeks. *What is Prompt Engineering ? The AI Revolution*. Accessed : 2024-05-20. 2024. URL : https://www.geeksforgeeks.org/what-is-prompt-engineering-the-ai-revolution/.

[28] Jim Dowling. *Building Machine Learning Systems with a Feature Store*. O'Reilly Media, Inc., 2025. ISBN : 9781098165239.

[29] Hugging Face. *Wikipedia Dataset*. https://huggingface.co/datasets/wikimedia/wikipedia. Accessed : 2024-05-23. 2024.

[30] UnSloth AI. *Introducing UnSloth AI : Custom Fine-tuning 30x Faster on T4 GPUs*. Accessed : 2024-04-16. 2024. URL : https://unsloth.ai/introducing.