Université de Versailles
Saint-Quentin-en-Yvelines

Université Paris Saclay

Master 2 Data Scale

Fine-tuning de modèles de Question Answering sur le dataset SQuAD avec déploiement web

---

Participants :

Sarah Harrouche

Khaled Bouabdallah

Mouad Tahiri

Dhai Eddine Zebbiche

Encadrant :

Mustapha Lebbah

Année Universitaire 2025-2026

Table des matières

Table des figures

Liste des tableaux

Introduction

Le Question Answering (QA) est une tâche centrale du traitement automatique du langage naturel (Natural Language Processing, NLP). Elle consiste à concevoir des modèles capables de fournir automatiquement une réponse pertinente à une question posée en langage naturel, à partir d'un contexte textuel donné. Contrairement aux systèmes de recherche d'information classiques, le QA vise une réponse précise et contextualisée, souvent sous forme d'un extrait du texte source.

Avec l'essor des modèles de langage pré-entraînés de type Transformer, les performances des systèmes de Question Answering ont connu des progrès majeurs. Des architectures telles que BERT et ses variantes ont permis d'atteindre, voire de dépasser, les performances humaines sur certains benchmarks de référence.

Dans ce projet, nous nous intéressons au **fine-tuning de plusieurs modèles de Question Answering extractif sur le jeu de données Stanford Question Answering Dataset (SQuAD).**

L'objectif est de comparer différentes architectures pré-entraînées en termes *de performance*, *de coût computationnel* et *de temps d'inférence*, tout en proposant une application web permettant d'interagir facilement avec les modèles entraînés.

Les objectifs principaux de ce travail sont les suivants :

— Fine-tuner et comparer plusieurs modèles de Question Answering pré-entraînés ;
— Analyser leurs performances à l'aide de métriques standard telles que l'Exact Match et le F1-score ;
— Etudier l'impact des choix architecturaux sur la qualité des prédictions et le temps d'inférence ;
— Développer une interface web permettant de tester et comparer les modèles sur des contextes et questions personnalisés.

Ce projet s'inscrit dans une démarche complète allant de l'exploration des données et de l'entraînement des modèles jusqu'au déploiement d'une application utilisable par un utilisateur final.

Chapitre 1

État de l'art et modèles de Question Answering

1.1 Question Answering extractif

Le Question Answering extractif consiste à identifier, au sein d'un contexte textuel, un segment continu de texte correspondant à la réponse à une question donnée. Contrairement au QA génératif, le modèle ne produit pas de nouvelle séquence, mais sélectionne une réponse existante dans le document d'entrée. Cette formulation permet une évaluation plus objective et reproductible, et s'appuie principalement sur des métriques telles que l'Exact Match et le F1-score.

Les approches modernes de QA extractif reposent majoritairement sur des architectures de type Transformer, capables de modéliser des dépendances longues et complexes entre la question et le contexte. Le principe général consiste à concaténer la question et le contexte, puis à prédire les positions de début et de fin de la réponse dans la séquence tokenisée.

1.2 Modèles pré-entraînés basés sur les Transformers

Les Transformers ont profondément modifié le paysage du NLP grâce à leur mécanisme d'attention, qui permet de capturer des relations globales au sein d'un texte. Les modèles pré-entraînés sont généralement entraînés sur de très grands corpus de données non annotées, puis adaptés à des tâches spécifiques via une phase de fine-tuning supervisé.

Dans ce projet, plusieurs modèles représentatifs ont été sélectionnés afin d'étudier l'impact des choix architecturaux sur les performances en Question Answering.

1.2.1 Baseline

Une approche baseline est d'abord mise en place afin de disposer d'un point de référence. Cette baseline permet d'*évaluer les performances initiales d'un modèle pré-entraîné sans optimisation poussée des hyperparamètres*, et sert de comparaison pour mesurer les gains apportés par le fine-tuning et le choix de modèles plus avancés.

1.2.2 DistilBERT

DistilBERT est une version allégée de BERT obtenue par distillation de connaissances. Il conserve une grande partie des performances du modèle original tout en réduisant significativement le nombre de paramètres. Ce compromis en fait un candidat pertinent pour des applications nécessitant des temps d'inférence plus courts et une consommation mémoire réduite, tout en maintenant des résultats compétitifs sur les tâches de QA.

1.2.3 RoBERTa

RoBERTa est une amélioration de BERT basée sur un entraînement plus long, l'utilisation de corpus plus volumineux et la suppression de certaines contraintes du pré-entraînement initial. Ces choix permettent au modèle de mieux généraliser et d'obtenir de meilleures performances sur de nombreuses tâches de compréhension du langage, y compris le Question Answering extractif.

1.2.4 DeBERTa

DeBERTa introduit une attention dite « disentangled », séparant explicitement les représentations du contenu et de la position des tokens. Cette architecture améliore la modélisation des relations syntaxiques et sémantiques complexes, ce qui se traduit généralement par de meilleures performances sur les benchmarks de QA, au prix d'un coût computationnel plus élevé.

L'étude conjointe de ces modèles permet ainsi d'analyser le compromis entre performance, complexité et temps d'inférence, éléments essentiels dans le cadre d'un déploiement applicatif.

Chapitre 2

Données et prétraitement

2.1 Présentation du dataset SQuAD

Le Stanford Question Answering Dataset (SQuAD) est un jeu de données de référence largement utilisé pour l'évaluation des systèmes de Question Answering extractif. Dans ce projet, nous utilisons la version SQuAD v1.1, qui se compose exclusivement de questions dont la réponse est explicitement contenue dans le passage fourni.

Le dataset repose sur des articles encyclopédiques issus de Wikipedia et couvre une grande diversité de thématiques telles que l'histoire, la géographie, les sciences, la culture ou encore la technologie. Cette diversité contribue à la robustesse et à la capacité de généralisation des modèles entraînés.

2.1.1 Caractéristiques du dataset

Les principales statistiques descriptives du dataset SQuAD v1.1 sont présentées dans le tableau ci-dessous :

Caractéristique	**Valeur**
Nombre d'exemples d'entraînement	87 599
Nombre d'exemples de validation	10 570
Nombre total d'exemples	98 169
Nombre d'articles sources	536
Longueur moyenne des contextes	122 mots
Longueur moyenne des questions	11 mots
Longueur moyenne des réponses	3,2 mots
Contextes $>$ 512 tokens	~15 %

TABLE 2.1 – Statistiques du dataset

Ces caractéristiques font de SQuAD un jeu de données particulièrement adapté à l'apprentissage supervisé du Question Answering extractif, en fournissant des annotations précises et de haute qualité.

2.1.2 Structure des données

Chaque exemple du dataset est structuré sous la forme d'un triplet :

— **Contexte :** un paragraphe extrait d'un article Wikipedia ;

— **Question :** une question formulée en langage naturel portant sur le contexte ;
— **Réponse :** un segment de texte extrait du contexte, défini par :le texte exact de la réponse, la position du caractère de début dans le contexte.

Exemple illustratif :

Contexte : "The Amazon rainforest, also known as Amazonia, covers 5.5 million square kilometers..."

Question : "How large is the Amazon rainforest ?"

Réponse : - Texte : "5.5 million square kilometers" - Position de début : 60

Cette structure permet un apprentissage supervisé précis des positions de début et de fin de la réponse au sein de la séquence tokenisée.

2.2 Séparation des ensembles

Le jeu de données est divisé en ensembles d'entraînement et de validation conformément aux recommandations standards associées à SQuAD. L'ensemble d'entraînement est utilisé pour l'optimisation des paramètres des modèles lors du fine-tuning, tandis que l'ensemble de validation sert à l'évaluation des performances et à la comparaison entre les différentes architectures.

Cette séparation garantit une évaluation équitable et permet de mesurer la capacité de généralisation des modèles sur des exemples non vus durant l'apprentissage.

2.3 Prétraitement et tokenisation

Le prétraitement des données constitue une étape cruciale pour les modèles de Question Answering basés sur les Transformers. La question et le contexte sont concaténés puis tokenisés à l'aide du tokenizer associé à chaque modèle pré-entraîné. Cette tokenisation transforme le texte brut en une séquence de tokens numériques compréhensibles par le modèle.

Étant donné la longueur maximale des séquences imposée par les modèles (généralement 512 tokens), une stratégie de sliding window est appliquée pour les contextes trop longs. Cette approche consiste à découper le contexte en plusieurs segments chevauchants afin de garantir que la réponse soit incluse dans au moins une fenêtre d'entrée.

2.3.1 Alignement des réponses

Pour chaque exemple, les positions de début et de fin de la réponse sont alignées avec les indices des tokens générés par le tokenizer. Cette étape permet de convertir les annotations textuelles du dataset en labels exploitables par le modèle lors de l'apprentissage.

Une attention particulière est portée à la gestion des cas où la réponse n'est pas entièrement contenue dans une fenêtre donnée. Ces exemples sont alors ignorés pour la fenêtre concernée afin d'éviter l'introduction de bruit dans le processus d'entraînement.

2.3.2 Impact du prétraitement

Les choix de prétraitement, notamment la tokenisation et la gestion des contextes longs, ont un impact direct sur les performances des modèles. Un mauvais alignement des réponses ou une stratégie de découpage inadaptée peut entraîner une dégradation significative des résultats. Le prétraitement mis en place vise ainsi à assurer un compromis entre couverture du contexte, efficacité computationnelle et qualité des annotations utilisées pour le fine-tuning.

Ce chapitre a présenté en détail le dataset SQuAD v1.1 et le pipeline de prétraitement mis en œuvre. Une attention particulière a été portée à la tokenisation, à la gestion des contextes longs et à l'alignement précis des réponses. Ces choix méthodologiques constituent une base essentielle pour garantir des performances élevées lors de l'entraînement des modèles de Question Answering extractif.

Chapitre 3

Méthodologie de fine-tuning

3.1 Vue d'ensemble du pipeline

Le fine-tuning des modèles de Question Answering repose sur un pipeline standard inspiré des bonnes pratiques de la bibliothèque Hugging Face Transformers. L'objectif est d'adapter des modèles de langage pré-entraînés à la tâche spécifique du Question Answering extractif sur le dataset SQuAD.

Le pipeline global comprend les étapes suivantes :

— chargement des modèles pré-entraînés et de leurs tokenizers associés ;

— préparation des données à l'aide du prétraitement décrit au Chapitre 2 ;

— configuration de l'entraînement et des hyperparamètres ;

— entraînement supervisé sur l'ensemble d'entraînement ;

— évaluation des performances sur l'ensemble de validation.

3.2 Paramètres d'entraînement

Le fine-tuning est réalisé en optimisant la fonction de perte associée à la prédiction des positions de début et de fin de la réponse. Les principaux hyperparamètres utilisés incluent le taux d'apprentissage (learning rate), la taille des batchs, le nombre d'époques et la stratégie d'optimisation.

Hyperparamètre	**Valeur**
Learning rate	3e-5
Batch size	16
Nombre d'époques	3
Optimizer	AdamW
Weight decay	0.01
Scheduler	Décroissance linéaire

TABLE 3.1 – Hyperparamètres d'entraînement

Un taux d'apprentissage relativement faible est privilégié afin de préserver les connaissances acquises lors du pré-entraînement tout en permettant une adaptation efficace à la tâche spécifique de Question Answering. La taille des batchs est choisie en fonction des contraintes mémoire, et le nombre d'époques est limité afin d'éviter le sur-apprentissage.

3.3 Stratégie d'entraînement

L'entraînement est réalisé de manière supervisée en minimisant une fonction de perte basée sur l'erreur de prédiction des positions de début et de fin de la réponse. À chaque itération, le modèle ajuste ses poids afin d'augmenter la probabilité associée aux tokens correspondant à la réponse correcte.

Les performances sont évaluées régulièrement sur l'ensemble de validation afin de suivre la convergence des modèles. Le modèle présentant le meilleur score F1 sur l'ensemble de validation est conservé pour chaque architecture.

Afin de limiter le sur-apprentissage, l'entraînement est arrêté après trois époques, même si les performances continuent d'augmenter marginalement.

3.4 Ressources computationnelles

Les expériences ont été menées sur des environnements disposant d'accélérateurs GPU, permettant de réduire significativement les temps d'entraînement. Les ressources utilisées sont résumées dans le tableau ci-dessous :

Composant	**Spécification**
GPU	NVIDIA Tesla T4 (16 Go)
Framework	PyTorch et Transformers
Environnement	Google Colab / Kaggle

TABLE 3.2 – Environnement d'entraînement

Les temps d'entraînement observés varient selon la taille des modèles. Les architectures les plus complexes nécessitent un temps de calcul plus important et une consommation mémoire plus élevée, illustrant le compromis entre performance et coût computationnel.

La méthodologie de fine-tuning adoptée repose sur un pipeline robuste, des hyperparamètres standards et une stratégie d'entraînement maîtrisée. Ces choix assurent une comparaison équitable entre les différentes architectures étudiées tout en maintenant des temps d'entraînement raisonnables.

Le chapitre suivant présente les résultats expérimentaux obtenus à l'aide de cette méthodologie et analyse les performances des différents modèles.

Chapitre 4

Évaluation et résultats expérimentaux

4.1 Métriques d'évaluation

L'évaluation des modèles de Question Answering extractif est réalisée à l'aide de métriques standard largement utilisées dans la littérature. Les principales métriques retenues sont l'Exact Match (EM) et le F1-score. L'Exact Match mesure la proportion de réponses prédites exactement identiques aux réponses de référence, tandis que le F1-score prend en compte le chevauchement entre les tokens de la réponse prédite et ceux de la réponse attendue, offrant ainsi une mesure plus souple et plus informative.

En complément, le temps d'inférence est mesuré afin d'évaluer l'efficacité des modèles dans une perspective de déploiement applicatif. Certaines métriques classiques de classification telles que la précision, le rappel ou les courbes ROC ne sont pas privilégiées ici, car le Question Answering extractif repose sur une prédiction de positions dans un texte plutôt que sur une classification binaire.

4.2 Résultats quantitatifs

Les performances des différents modèles sont comparées sur l'ensemble de validation du dataset SQuAD. Le Tableau 1 présente les résultats obtenus en termes d'Exact Match, de F1-score et de temps moyen d'inférence par requête.

Modèle	**Paramètres**	**F1-score (%)**	**Exact Match (%)**	**Temps d'inférence (ms)**
DistilBERT	66M	84.41	75.81	~50
RoBERTa	125M	91.96	85.65	~120
DeBERTa	184M	93.01	86.58	~320

Table 4.1 – Comparaison des performances des modèles

Ces résultats montrent une amélioration progressive des performances avec l'augmentation de la complexité des modèles. Toutefois, cette amélioration s'accompagne d'une augmentation significative du temps d'inférence.

Les performances humaines sur SQuAD v1.1 sont estimées à environ 91.0 % de F1-score et 82.3 % d'Exact Match.

Dans ce contexte, RoBERTa atteint des performances comparables, tandis que DeBERTa les dépasse légèrement, confirmant l'efficacité des architectures Transformer avan-

cées pour le Question Answering extractif.

4.3 Analyse comparative

4.3.1 Comparaison des performances

La Figure 4.1 compare les scores F1 et Exact Match obtenus par chaque modèle.

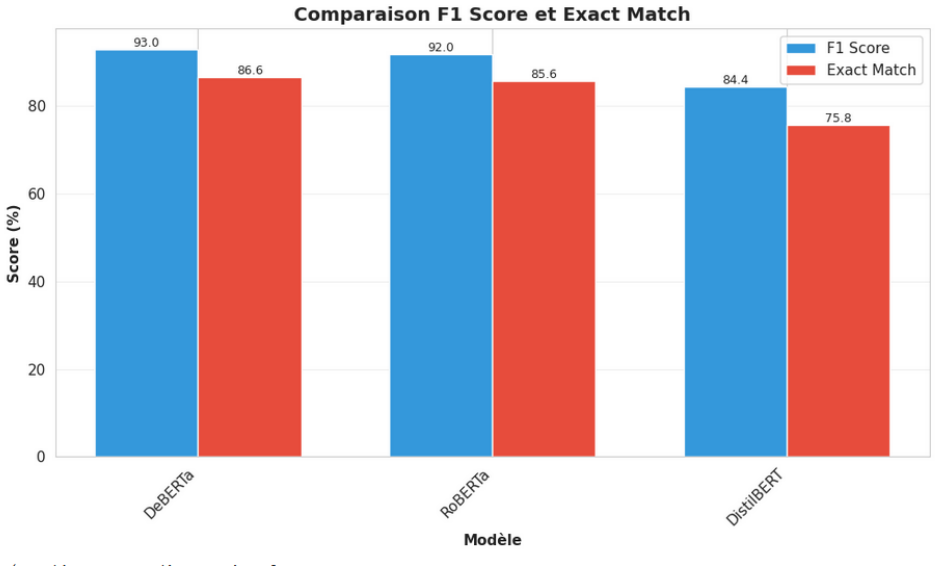

FIGURE 4.1 – Comparaison des scores F1 et Exact Match par modèle

DistilBERT, bien que nettement plus léger, conserve des performances solides et constitue un bon compromis pour des applications à ressources limitées. RoBERTa offre un gain de performance significatif, tandis que DeBERTa obtient les meilleurs résultats globaux grâce à son mécanisme d'attention amélioré.

4.3.2 Trade-off performance et vitesse

La Figure 4.2 illustre le compromis entre performance (F1-score) et temps d'inférence.

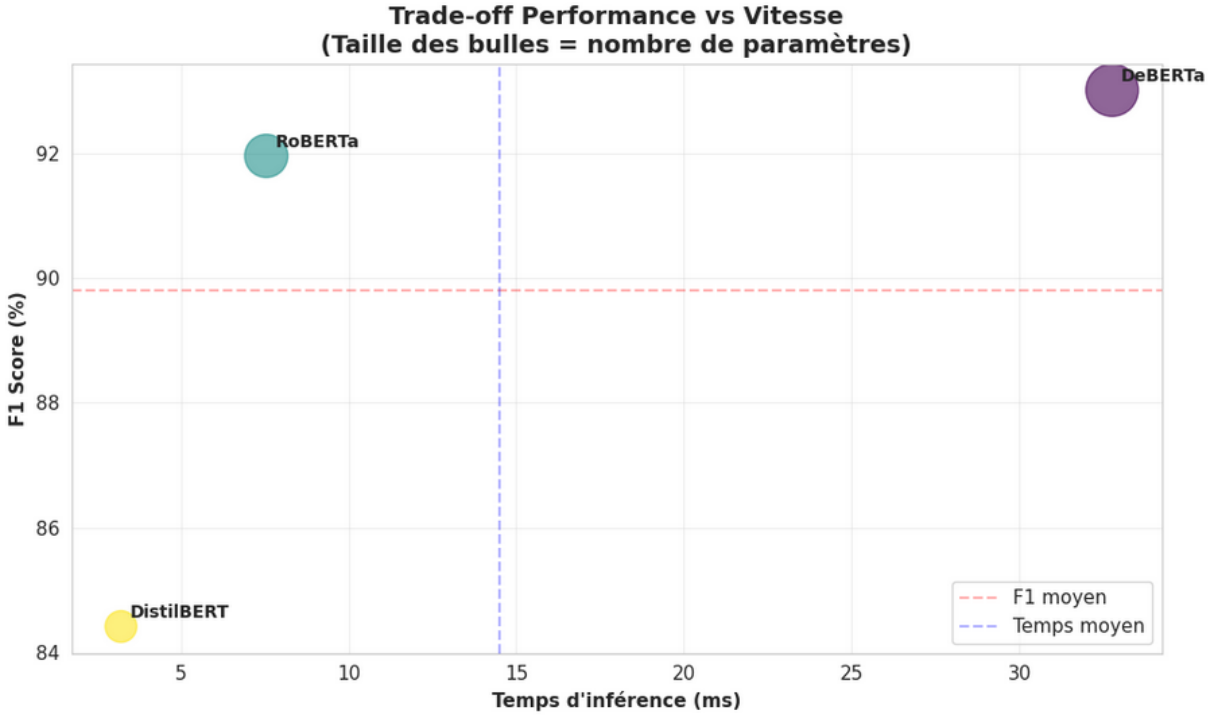

FIGURE 4.2 – Trade-off entre performance et temps d'inférence

DistilBERT est le plus rapide mais le moins performant, tandis que DeBERTa est le plus précis mais également le plus lent. RoBERTa apparaît comme un point d'équilibre pertinent entre précision et efficacité, en particulier pour un déploiement web.

4.4 Analyse qualitative des erreurs

Une analyse manuelle d'un échantillon d'erreurs met en évidence plusieurs types d'erreurs récurrentes :

Type d'erreur	DistilBERT	RoBERTa	DeBERTa
Réponse trop longue	élevée	modérée	faible
Réponse trop courte	modérée	faible	faible
Mauvaise entité	fréquente	moins fréquente	rare
Question ambiguë	présente	présente	présente

TABLE 4.2 – Analyse des types d'erreurs par modèle

Les modèles plus complexes gèrent mieux les relations temporelles et les ambiguïtés, tandis que les questions intrinsèquement ambiguës restent difficiles pour l'ensemble des architectures.

L'évaluation expérimentale met en évidence plusieurs conclusions importantes. Les trois modèles fine-tunés atteignent des performances élevées sur SQuAD v1.1, avec une progression claire entre DistilBERT, RoBERTa et DeBERTa. Un compromis marqué existe entre performance et efficacité computationnelle : RoBERTa offre le meilleur équilibre pour un déploiement applicatif, tandis que DistilBERT convient aux scénarios à forte contrainte de latence et DeBERTa aux contextes où la performance prime.

Ces résultats confirment la pertinence de la méthodologie adoptée et justifient les choix effectués pour le déploiement de l'application présentée dans le chapitre suivant.

Chapitre 5

Application web et déploiement

Afin de rendre les modèles de Question Answering accessibles et exploitables par un utilisateur final, une application web interactive a été développée. L'objectif principal de cette interface est de permettre la comparaison directe des différents modèles fine-tunés sur un même contexte et une même question, tout en offrant une visualisation claire de leurs performances respectives.

Cette approche vise à dépasser le simple cadre expérimental des notebooks en proposant un outil pratique illustrant les compromis entre précision, complexité et temps de réponse des modèles.

5.1 Architecture de l'application

L'application repose sur une architecture légère et modulaire. Les modèles fine-tunés sont chargés côté serveur et exposés via une interface utilisateur permettant de soumettre des requêtes de Question Answering. Chaque requête est traitée indépendamment par les différents modèles, et les réponses générées sont présentées simultanément afin de faciliter la comparaison.

Le choix d'une architecture simple permet de garantir une bonne maintenabilité du code et une intégration fluide des modèles issus des phases de fine-tuning.

5.2 Comparaison multi-modèles

L'un des points forts de l'application réside dans la comparaison multi-modèles. Pour une question et un contexte donnés, l'utilisateur peut observer : (i) la réponse extraite par chaque modèle, (ii) les scores de performance globaux associés au modèle (Exact Match et F1-score), ainsi que (iii) les différences qualitatives entre les réponses proposées.

Cette fonctionnalité met en évidence les variations de comportement entre les architectures, notamment en termes de précision des réponses, de robustesse face à des formulations ambiguës et de temps de réponse.

5.3 Déploiement sur Hugging Face Spaces

L'application a été déployée sur la plateforme Hugging Face Spaces, offrant un accès public et reproductible aux modèles entraînés. Ce déploiement permet aux utilisateurs de tester directement les performances des modèles sans nécessiter d'installation locale ou de ressources matérielles spécifiques.

Le déploiement sur Hugging Face Spaces constitue une étape importante du projet, illustrant la transition entre une expérimentation académique et une application utilisable dans un contexte réel. Il facilite également l'évaluation externe du travail réalisé et renforce la transparence des résultats présentés.

Ce chapitre a présenté l'application web développée pour illustrer et comparer les modèles de Question Answering fine-tunés. L'interface interactive et le déploiement sur Hugging Face Spaces permettent de valoriser les résultats expérimentaux et de démontrer concrètement les compromis entre performance et efficacité des différentes architectures étudiées.

Chapitre 6

Discussion et limites

Les résultats obtenus mettent en évidence l'impact significatif du choix de l'architecture sur les performances en Question Answering extractif. Les modèles plus complexes, tels que RoBERTa et DeBERTa, surpassent nettement les modèles plus légers en termes d'Exact Match et de F1-score, confirmant l'intérêt d'un pré-entraînement plus riche et de mécanismes d'attention avancés.

Cependant, ces gains de performance s'accompagnent de compromis importants. L'augmentation du nombre de paramètres entraîne un coût computationnel plus élevé, tant lors de l'entraînement que lors de l'inférence. Dans un contexte applicatif, ce facteur peut devenir critique, notamment pour des systèmes temps réel ou déployés sur des infrastructures à ressources limitées. À l'inverse, DistilBERT, bien que moins performant, présente des temps de réponse plus rapides et une empreinte mémoire réduite, ce qui en fait une alternative pertinente pour certains cas d'usage.

Par ailleurs, le cadre du Question Answering extractif présente des limites intrinsèques. Les modèles sont contraints de sélectionner une réponse strictement contenue dans le contexte fourni, ce qui peut conduire à des erreurs lorsque la question est ambiguë, mal formulée ou lorsque la réponse nécessite une reformulation ou une inférence plus poussée. De plus, le dataset SQuAD v1.1 ne contient pas de questions sans réponse, ce qui limite l'évaluation de la robustesse des modèles face à des cas négatifs.

Enfin, bien que l'interface web permette une comparaison qualitative intéressante, elle ne remplace pas une évaluation exhaustive sur des données hors distribution. Les performances observées peuvent diminuer lorsque les modèles sont confrontés à des textes ou des domaines très différents de ceux présents dans SQuAD.

Chapitre 7

Conclusion et perspectives

Dans ce projet, nous avons étudié le fine-tuning de plusieurs modèles de Question Answering extractif sur le dataset SQuAD, en comparant leurs performances et en analysant les compromis entre précision, complexité et temps d'inférence. Les résultats obtenus confirment l'efficacité des architectures de type Transformer pour cette tâche, ainsi que l'apport significatif du fine-tuning supervisé.

La comparaison entre DistilBERT, RoBERTa et DeBERTa a permis de mettre en évidence des profils complémentaires : un modèle léger et rapide, un modèle intermédiaire offrant un excellent équilibre, et un modèle plus complexe atteignant les meilleures performances globales. Le déploiement d'une application web interactive sur Hugging Face Spaces a par ailleurs permis de valoriser ces résultats dans un cadre concret et accessible.

Plusieurs perspectives d'amélioration peuvent être envisagées. Parmi celles-ci figurent l'exploration de techniques d'augmentation de données, l'utilisation de modèles capables de traiter des contextes plus longs, ou encore l'intégration de méthodes de Question Answering génératif ou hybrides. L'ajout de mécanismes d'interprétabilité pourrait également permettre de mieux comprendre les décisions prises par les modèles.

Ce travail fournit ainsi une base solide pour le développement et le déploiement de systèmes de Question Answering performants, tout en mettant en lumière les enjeux et les défis associés à leur utilisation dans des contextes réels.

Bibliographie

[1] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, *SQuAD : 100,000+ Questions for Machine Comprehension of Text*, Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2016.

[2] A. Vaswani et al., *Attention Is All You Need*, Advances in Neural Information Processing Systems (NeurIPS), 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding*, Proceedings of NAACL-HLT, 2019.

[4] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, *DistilBERT, a distilled version of BERT : smaller, faster, cheaper and lighter*, NeurIPS Workshop, 2019.

[5] Y. Liu et al., *RoBERTa : A Robustly Optimized BERT Pretraining Approach*, arXiv preprint arXiv :1907.11692, 2019.

[6] P. He et al., *DeBERTa : Decoding-enhanced BERT with Disentangled Attention*, International Conference on Learning Representations (ICLR), 2021.

[7] T. Wolf et al., *Transformers : State-of-the-Art Natural Language Processing*, Proceedings of EMNLP : System Demonstrations, 2020.

[8] Hugging Face, *Transformers Library Documentation*, `https://huggingface.co/docs/transformers`.

[9] Hugging Face, *Question Answering Fine-Tuning Example Notebook*, `https://github.com/huggingface/notebooks/blob/main/examples/question_answering-tf.ipynb`.

[10] Streamlit, *Streamlit Documentation*, `https://docs.streamlit.io`.

[11] Hugging Face, *Hugging Face Spaces Documentation*, `https://huggingface.co/docs/hub/spaces`.

[12] Hugging Face, *DistilBERT Model Documentation*, `https://huggingface.co/docs/transformers/model_doc/distilbert`.

[13] Hugging Face, *RoBERTa Model Documentation*, `https://huggingface.co/docs/transformers/model_doc/roberta`.

[14] Hugging Face, *DeBERTa Model Documentation*, `https://huggingface.co/docs/transformers/model_doc/deberta`.

Annexes

Annexe A – Interface de l'application de Question Answering

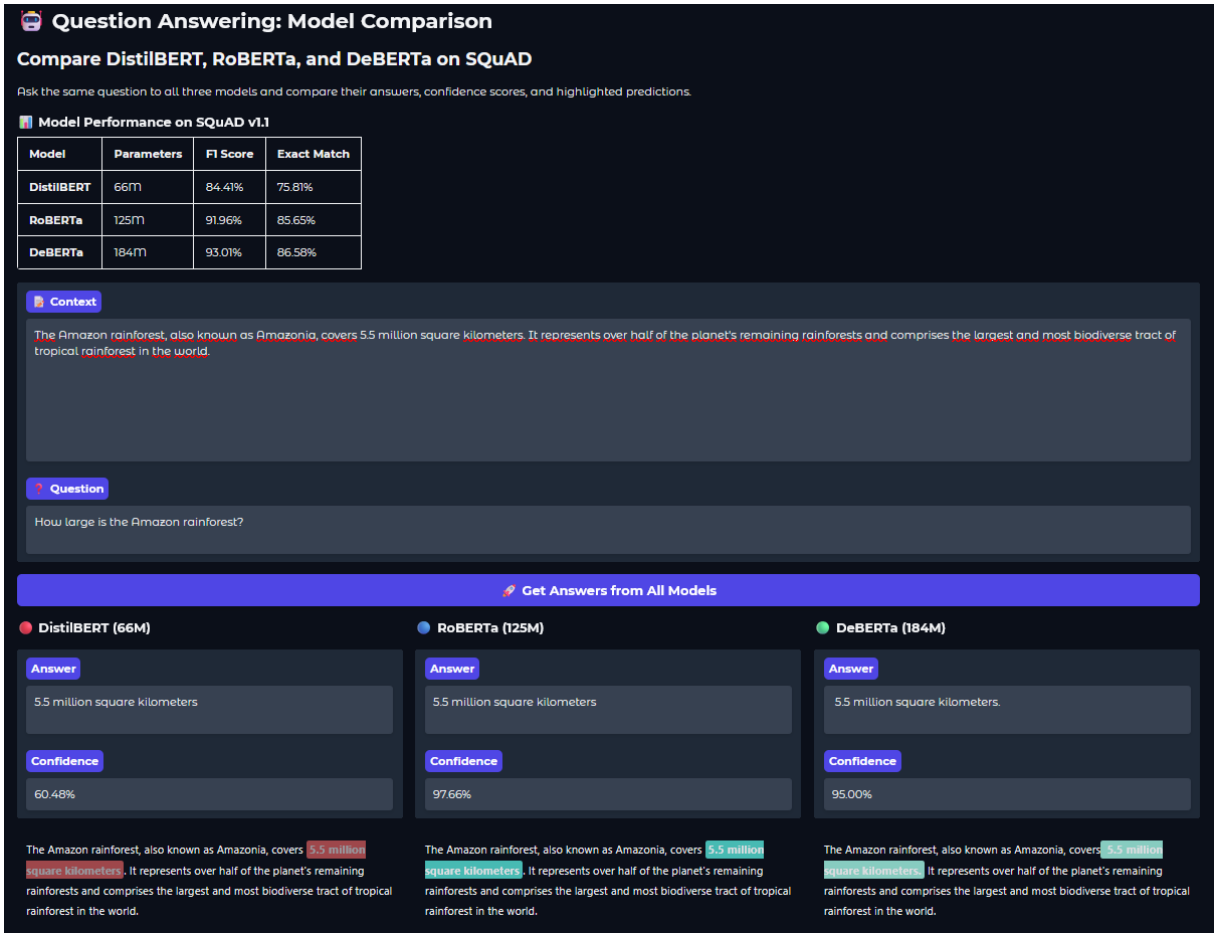

FIGURE 7.1 – Interface web de comparaison des modèles de Question Answering

Cette figure illustre l'interface web développée et déployée sur Hugging Face Spaces. L'application permet à l'utilisateur de fournir un contexte et une question, puis de comparer côte à côte les réponses générées par les trois modèles fine-tunés (DistilBERT, RoBERTa et DeBERTa). Elle met en évidence les différences de comportement, de précision et de temps de réponse entre les architectures, tout en offrant une visualisation claire et interactive des résultats.