Fine-tuning de Modèles de Question Answering Extractif

De l’analyse du dataset SQuAD v1.1 au déploiement MLOps sur Hugging Face Spaces

Projet réalisé par :

- Sarah HARROUCHE
- Khaled BOUADBALLAH
- Mouad TAHIRI
- Dhai Eddine Zebbiche

Encadrant :

- Mustapha LEBBAH

Master 2 DataScale | Université Paris-Saclay

1 | Introduction

Du mot-clé à la compréhension sémantique

Recherche d'Information (RI)

- Recherche par mots-clés
- Manque de précision
- Pas de contexte

Question Answering (QA) Extractif

- Identification d'un span (segment) précis
- Compréhension syntaxique et sémantique

La Révolution Transformer

Repose sur le mécanisme d'attention pour gérer les dépendances textuelles et sur le Transfer Learning pour adapter des connaissances générales au dataset spécifique SQuAD.

2 | L’Objectif

Maîtriser le Question Answering Extractif

Appliquer les méthodes de fine-tuning de Transformers pour passer d'un modèle généraliste à un expert en QA extractif.

Données : SQuAD v1.1 (87k exemples d'entraînement)

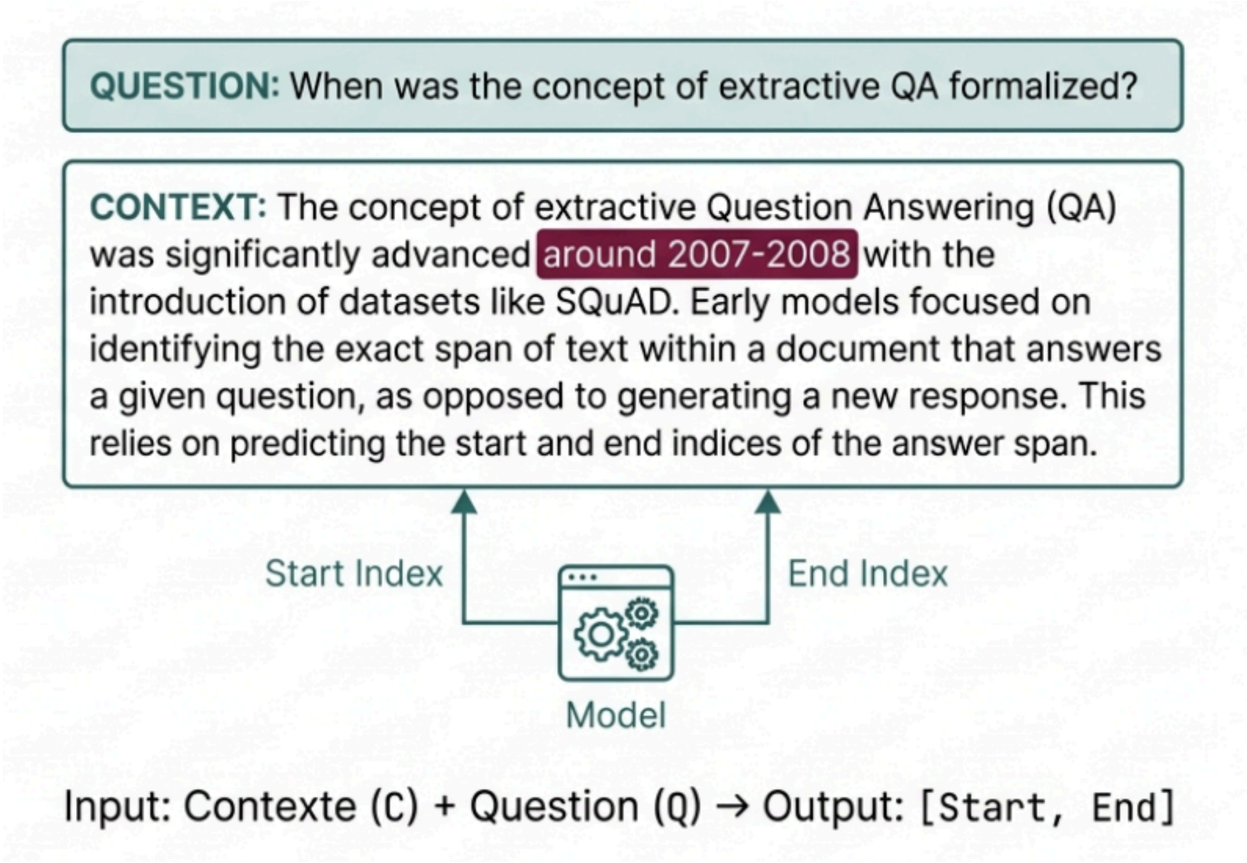
Modèles: Comparaison de 3 architectures (DistilBERT, RoBERTa, DeBERTa)

Résultat: Une application Web interactive déployée via Gradio

3 | Question Answering Extractif

L'art d'extraire plutôt que de générer.

Contrairement au QA génératif, le modèle ne rédige pas de texte. Il doit identifier un segment continu (span) existant dans le contexte.



4 | Les Données

Stanford Question Answering Dataset (SQuAD v1.1)

87 599

Exemples d'entraînement

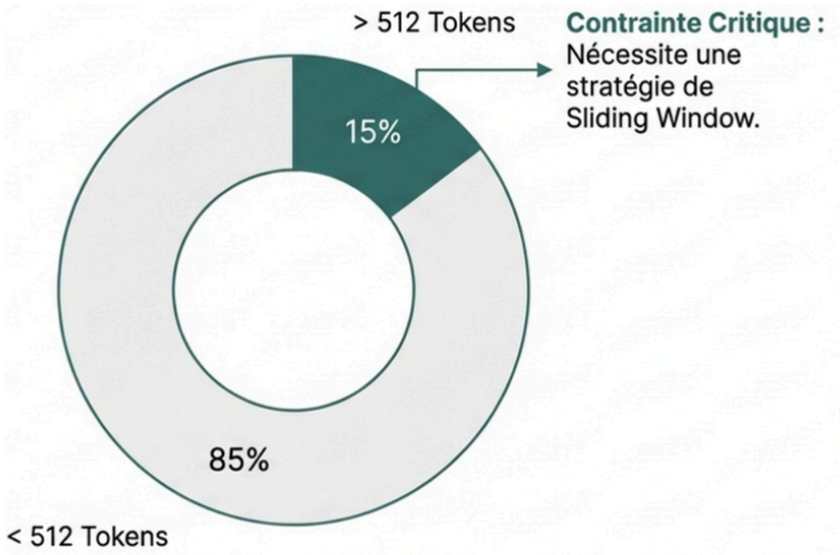
10 570

Exemples de Validation

536

Articles Wikipédia

```
{
  'context": "The Amazon rainforest, also known as
  Amazonia, is a moist broadleaf tropical
  rainforest in the Amazon biome that covers most
  of the Amazon basin of South America. This
  basin encompasses 7 million square kilometers
  (2.7 million square miles), of which 5.5
  million square kilometers are covered by
  the rainforest."
  },
  question: "How large is the Amazon rainforest?"
  answers: {
    text: "5.5 million square kilometers"
    answer_start: 60
  }
}
```



5 | Comparaison des Modèles

Trois Architectures, Trois Philosophies

DistilBERT

Léger & Rapide

66 M Paramètres

Architecture distillée (version allégée de BERT). Conçu pour une inférence rapide et une empreinte mémoire minimale

RoBERTa

Robuste & Optimisé

125 M Paramètres

Robustly Optimized BERT Approach. Entraînement plus long sur plus de données. Suppression "Next Sentence Prediction".

DeBERTa

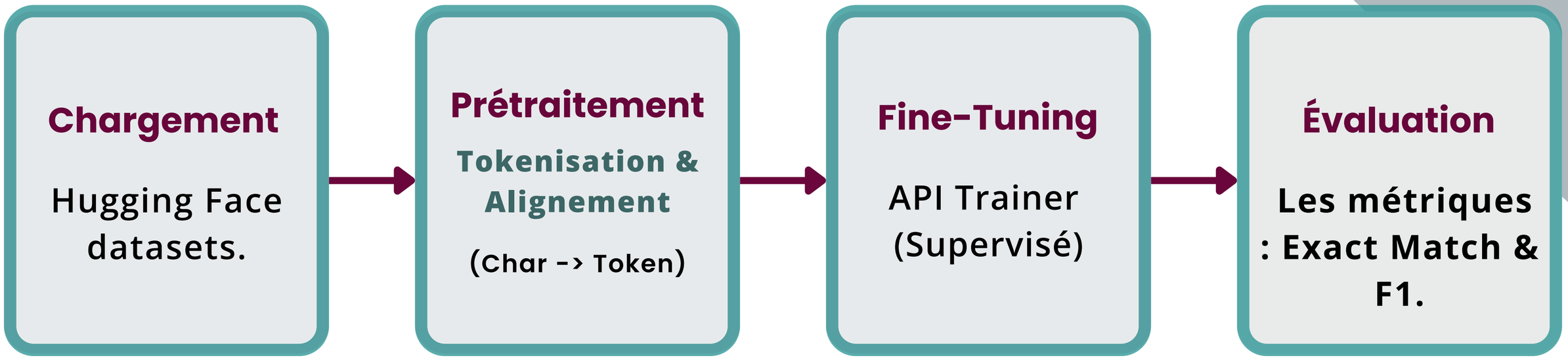
Haute Performance

184 M Paramètres

Decoding-enhanced BERT with disentangled attention. Sépare le contenu et la position pour capturer les relations sémantiques fines.

6 | Pipeline de Fine-Tuning

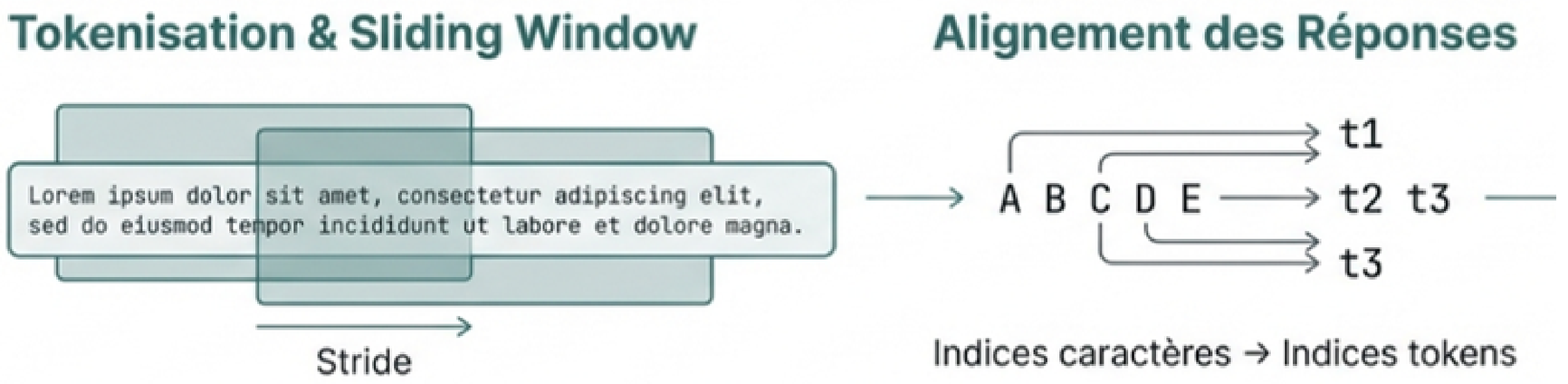
Vue Globale



Ce pipeline standardisé assure la reproductibilité de l'expérience, de la gestion des données brutes jusqu'à l'obtention de métriques de performance fiables (Exact Match et F1).

6 | Pipeline de Fine-Tuning

Prétraitement et Stratégie de Tokenisation



Optimisation du traitement des textes longs par Sliding Window afin de prévenir toute perte d'information lors du découpage.

6 | Pipeline de Fine-Tuning

Stratégie de Fine-Tuning

Configuration (Entraînement)

```
Library: Hugging Face Transformers
(Trainer API)
Optimizer: AdamW (Linear Decay)
Hyperparameters {
    learning_rate: 3e-5,
    batch_size: 16,
    num_epochs: 3
}
```

Infrastructure & Stratégie

Matériel :

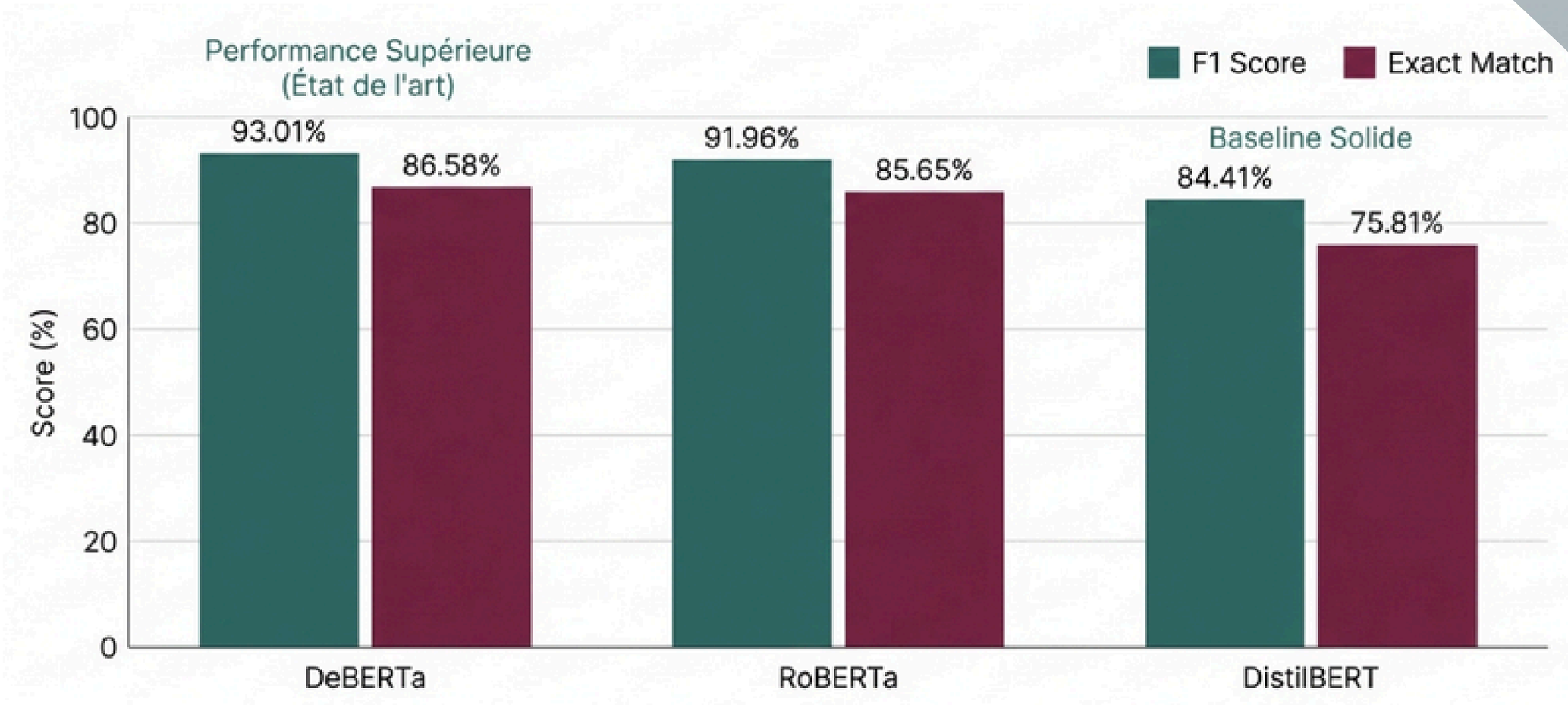
- GPU NVIDIA Tesla T4 (16 Go VRAM)
- Environnements Google Colab / Kaggle

Stratégie :

- Sauvegarde du meilleur modèle (Best Model Checkpoint)
- Critère : F1-Score sur Validation
- Early Stopping pour éviter l'overfitting

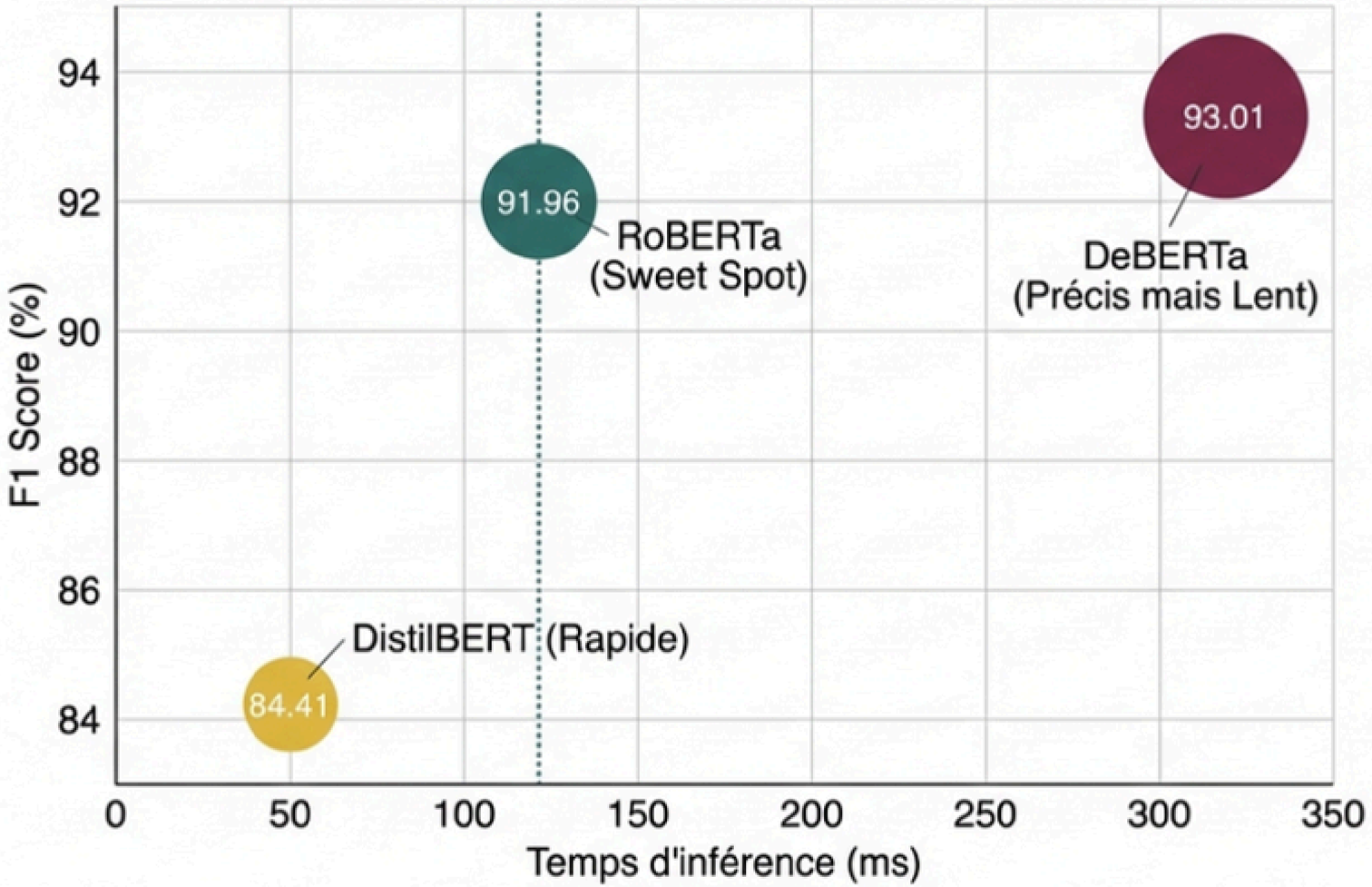
7 | Résultats

DeBERTa domine la précision



7 | Résultats

Le Trade-off Performance Vs Vitesse



RoBERTa → DeBERTa

Gain de 1.5% de précision pour 3x plus de temps

8 | Analyse Qualitative des Erreurs

DistilBERT	Problème de longueur de réponse.	Tendance à prédire des spans trop courts ou trop longs.
DeBERTa	Compréhension Syntaxique.	Excellente gestion des dépendances complexes et des entités nommées.
Défi Commun	Ambiguïté.	Tous les modèles échouent sur les questions mal formulées ou imprécises.

9 | Mise en Production

Architecture de Déploiement

Versionning

Modèles sauvegardés localement.

Upload to Hub

Script `push_to_hub` → **Hugging Face Model Registry**.

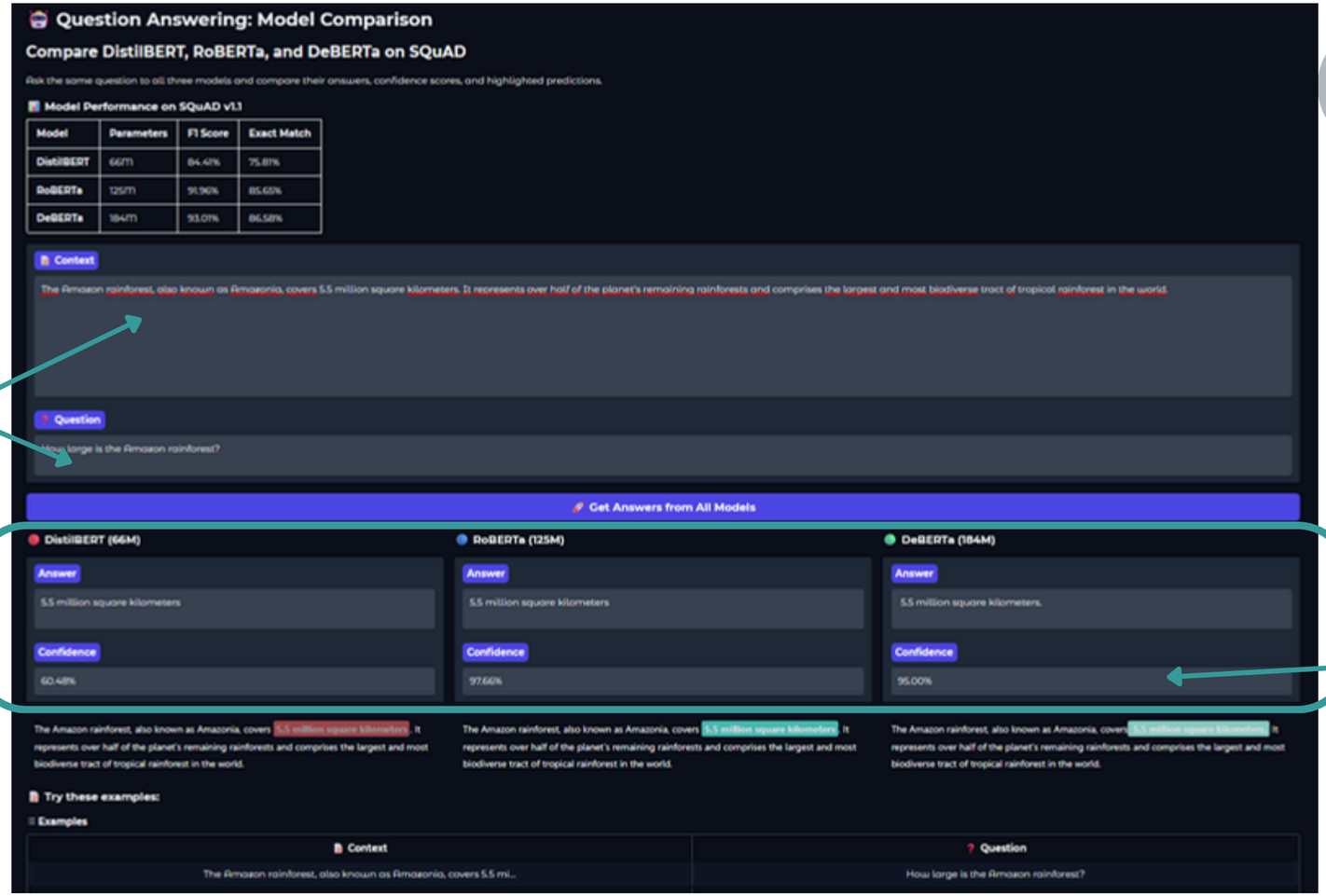
App Hosting

Création d'un **Space** lié au **repository Git**.

Accessibilité : URL publique, aucun setup local n'est requis pour l'utilisateur final

Scannez le QR Code
ou Cliquer ICI Afin d'accéder à l'application

9 | Mise en Production



Context & Question Input

Comparaison Simultanée

Score de Confiance & Métriques

Stack Technique : Python Backend + Gradio UI + Hugging Face Hosting

10 | Discussion et Limites

Contrainte Extractive

- Le modèle ne peut pas répondre si la réponse n'est pas écrite explicitement.
- Il ne peut pas synthétiser des faits dispersés.

Coût Computationnel

- **DeBERTa** est précis mais lourd.
- Son déploiement à grande échelle nécessiterait des ressources GPU importantes.

Gestion de l'Ambiguïté

- SQuAD v1.1 ne contient pas de questions sans réponse.
- Le modèle tente toujours de trouver une réponse, même si la question est absurde par rapport au contexte.

11 | Conclusion et Perspectives

Perspectives

SQuAD v2.0 : Apprendre au modèle à dire 'Je ne sais pas'.

QA Génératif : Explorer T5 ou GPT pour des réponses plus naturelles.

Quantization : Réduire DeBERTa pour accélérer l'inférence.

Pour Conclure, Quel Modèle Choisir ?

Merci pour votre attention !