Structure de la présentation : Fine-Tuning Multimodal pour l'Extraction de Données Bancaires

Objectif : Présenter les résultats du PoC sur l'extraction d'informations de chèques bancaires via Qwen 32B Fine-tuné. **Audience :** Management / Technique. **Format :** 5 Slides.

Slide 1 : Contexte et Objectifs du PoC

Titre : Automatisation de l'extraction d'informations sur chèques bancaires

Contenu de la slide :

- **L'Objectif Métier :**
  - Extraire automatiquement **7 champs clés** (Montant numérique, Montant lettres, Date, Bénéficiaire, Signature, Lieu, Devise).
  - Passer d'une extraction manuelle/OCR classique à une approche IA Générative Multimodale.
- **Le Protocole Expérimental :**
  - **Modèle choisi :** Qwen-VL 32B (Open Source, État de l'art).
  - **Données :** Création d'un dataset "maison" de 100 images de chèques.
  - **Outillage :**
    - Annotation : Label Studio (Interface visuelle).
    - Entraînement : Librairie Unsloth.
- **Contraintes & Ressources :**
  - **Infra :** Nécessité d'un GPU puissant (H100/A100 ou 4090/A6000 avec optimisation).
  - **Temps :** X heures d'annotation pour 100 documents (Label Studio).

📝 **Notes pour l'orateur (Ce que tu dis) :**

> "Nous avons voulu tester si l'IA générative pouvait surpasser nos OCR classiques sur des documents complexes comme des chèques manuscrits. Nous avons ciblé un modèle Open Source très performant, Qwen en version 32 milliards de paramètres. La première étape critique a été de construire notre propre vérité terrain : nous avons annoté manuellement 100 chèques via Label Studio pour apprendre au modèle exactement ce qu'on attend de lui."

Slide 2 : La Solution Technique - Focus sur Unsloth

Titre : Rendre l'entraînement possible et rapide avec Unsloth

Contenu de la slide :

- **Le Défi :**
  - Entraîner un modèle de 32 Milliards de paramètres demande normalement une infrastructure industrielle massive (coûteuse et lente).
- **La Solution : Unsloth (L'accélérateur intelligent)**

    - **Simplification :** Unsloth optimise "la tuyauterie" mathématique de l'apprentissage. Il supprime les calculs inutiles sans réduire l'intelligence du modèle.
    - **Bénéfices directs :**
        - 🏎️ **Vitesse :** L'apprentissage se fait 2 à 5 fois plus vite.
        - 📉 **Économie :** Permet de faire tourner ce modèle géant sur une seule carte graphique standard, là où il en faudrait normalement plusieurs.
- **L'approche "Chirurgicale" (LoRA) :**
    - Au lieu de rééduquer tout le cerveau du modèle, on insère simplement des petits modules de connaissances (des adaptateurs) dédiés aux chèques.
    - Le modèle conserve sa culture générale mais devient expert sur notre tâche.

📝 **Notes pour l'orateur :**

"Pour entraîner un modèle aussi gros que Qwen 32B, on se heurte normalement à un mur matériel : ça demande trop de mémoire. C'est là qu'intervient Unsloth. Voyez-le comme un outil qui optimise drastiquement l'efficacité de l'apprentissage. Il trouve des raccourcis mathématiques pour arriver au même résultat, mais en consommant beaucoup moins de mémoire.

Couplé à une technique appelée LoRA, cela nous permet de ne pas toucher au cœur du modèle, mais d'ajouter simplement une petite 'couche' de spécialisation pour nos chèques. Résultat : on a pu faire ce projet sur une machine accessible, rapidement, sans louer un supercalculateur."

Slide 3 : Résultats Quantitatifs et Analyse Nuancée

Titre : Performance : +15% de gain médian, mais des défis persistent

Contenu de la slide :

- **La Métrique : Exactitude Stricte (Exact Match)**
    - Score binaire (1 ou 0) par champ. Moyenne sur les 7 champs.
- **Les Gains (Base vs Fine-Tuné) :**
    - 📈 **Gain Médian : +15%.** Sur la majorité des chèques "standards", le modèle fine-tuné est nettement plus performant.
    - ➕ **Gain Moyen : +5%.**
- **Analyse Critique (Ce qui n'a pas marché) :**
    - ⚠️ **Stagnation sur les cas critiques :** Les chèques où le modèle de base échouait totalement (0%) restent à 0% après fine-tuning.
    - 📉 **Régression sur le format :** Le modèle de base (*Instruct*) respectait mieux le schéma JSON. Le fine-tuning a introduit une légère instabilité syntaxique (oubli de crochets, clés mal nommées).

📝 **Notes pour l'orateur :**

"Les résultats sont contrastés et très intéressants. D'un côté, nous avons un gain spectaculaire de **15% sur la médiane**. Cela veut dire que pour un chèque 'moyen', le fine-

tuning aide énormément à lire l'écriture manuscrite.

Par contre, nous avons observé deux limites :

1. Les cas difficiles (chèques très raturés ou illisibles) ne sont pas résolus (0% reste 0%).
2. Le modèle de base, qui est un modèle 'Instruct', était paradoxalement plus rigoureux sur le respect du format JSON. En apprenant à lire les chèques, le modèle fine-tuné a un peu 'oublié' la rigueur informatique du code."

Slide 4 : Limites et Complexité

Titre : Analyse Critique : Coûts et Limites

Contenu de la slide :

- **Les Limites Techniques :**
  - **Volume de données :** 100 images est un échantillon faible. Risque d'*overfitting* (le modèle apprend par cœur ces 100 chèques et échoue sur un format de chèque inédit).
  - **Hallucinations résiduelles :** Sur des signatures très abstraites ou ratures.
- **Complexité MLOps :**
  - Contrairement à un appel API (type GPT-4), il faut gérer le cycle de vie du modèle (versioning des poids LoRA).
  - Nécessite de servir (héberger) le modèle en interne (Besoin de GPU pour l'inférence = coût récurrent).
- **Temps Homme :**
  - L'annotation est le goulot d'étranglement principal (plus long que l'entraînement lui-même qui est rapide grâce à Unsloth).

📝 **Notes pour l'orateur :**

"Tout n'est pas magique. 100 images, c'est peu pour généraliser à toutes les banques du monde. De plus, maintenir un modèle fine-tuné demande une infrastructure dédiée (vLLM ou autre) pour le faire tourner, contrairement à l'utilisation simple d'une API externe. Le coût se déplace du 'coût par token' (API) vers un 'coût d'infrastructure fixe' (GPU)."

Slide 5 : Conclusion et Recommandations

Titre : Bilan et Prochaines Étapes

Contenu de la slide :

- **Conclusion :**
  - Le Fine-Tuning est **efficace** pour contraindre le format et spécialiser le modèle sur une tâche visuelle précise.
  - Unsloth rend l'opération **viable** économiquement et techniquement (pas besoin de cluster de calcul).
- **Recommandations / Next Steps :**
  1. **Augmenter le Dataset :** Passer à 500-1000 images pour robustifier le modèle.

2. **Tester un modèle plus petit :** Essayer Qwen 7B (au lieu de 32B). Si les performances sont similaires après fine-tuning, cela divisera les coûts d'inférence par 4.
3. **Industrialisation :** Mettre en place un pipeline d'évaluation automatique.

📝 **Notes pour l'orateur :**

"En conclusion, nous avons prouvé que nous pouvons spécialiser un modèle open source pour nos besoins. Pour la suite, ma recommandation est paradoxale : maintenant que nous savons que le fine-tuning marche, essayons de le faire sur un modèle beaucoup plus petit (7B). Souvent, un petit modèle bien entraîné bat un gros modèle généraliste, et cela nous coûterait beaucoup moins cher à faire tourner en production."