

Projet de Traduction Automatique de Wikipédia en Fon, Ewe, Yoruba et Dendi

1. Introduction

L'objectif principal de ce projet est d'automatiser la traduction de l'intégralité des articles Wikipédia en anglais vers quatre langues africaines à faibles ressources : **Fon, Ewe, Yoruba, et Dendi**. Ces langues, bien que parlées par plusieurs millions de personnes, manquent encore de ressources linguistiques suffisantes, notamment dans le domaine numérique. Ainsi, ce projet vise à enrichir la documentation numérique de ces langues en exploitant la richesse du contenu existant sur Wikipédia, tout en utilisant des outils et modèles de traduction de pointe. Il s'inscrit également dans une démarche de préservation et de valorisation des langues africaines en ligne.

2. Objectifs du Projet

Les objectifs généraux de ce projet sont les suivants :

- **Collecte des articles Wikipédia** : Centraliser les liens de tous les articles Wikipédia en anglais (environ 7 millions).
- **Extraction et nettoyage du contenu** : Récupérer le contenu HTML de chaque article, effectuer un nettoyage et une structuration appropriée.
- **Traduction vers plusieurs langues africaines** : Traduire les articles extraits vers les langues Fon, Ewe, Yoruba et Dendi.
- **Optimisation du traitement à grande échelle** : Mettre en place des stratégies de parallélisation et de reprise après erreur pour traiter efficacement des millions d'articles.
- **Assurer la résilience du pipeline** : Garantir la qualité, la stabilité et la reprise des traitements sans perte de données.

3. Environnement Technique

Le projet s'exécute dans un environnement robuste et évolutif, avec les technologies suivantes :

Matériel et Logiciels

- **Machine Virtuelle** : VM Azure avec 64 Go de RAM, processeur AMD EPYC, deux disques NVMe (274 Go + 472 Go), et interface réseau haut débit ConnectX-5 adapté au traitement de grandes quantités de données.
- **Python 3.9+** : Langage de programmation utilisé pour le développement du pipeline.

Bibliothèques Python Utilisées

- **requests**, **csv**, **pandas**, **tqdm** : Pour la collecte, le traitement des données et le suivi d'exécution.
- **trafilatura** : Pour l'extraction du contenu des articles Wikipédia.
- **joblib** : Pour la parallélisation des processus.
- **nltk** : Pour la gestion de la tokenisation des textes.
- **transformers**, **openai** et **python-docx** : Pour l'intégration des modèles de traduction.

Modèles de Traduction

- **Azure OpenAI GPT-4o** : Utilisé pour la traduction des articles en Yoruba, ce modèle est particulièrement performant pour la traduction de l'anglais vers le Yoruba.
 - **Masakhane Models (m2m100_418M_fr_fon_rel et m2m100_418M_fr_ewe_rel)** : Modèles de traduction multilingues, chargés localement pour la traduction des articles en Fon et Ewe. Ces modèles, issus du fine-tuning de modèles multimodaux et de corpus anglais, sont performants pour la traduction vers ces langues africaines.
-

4. Architecture du Pipeline

Le pipeline se compose de plusieurs étapes clés, automatisées pour un traitement efficace et scalable des millions d'articles Wikipédia.

4.1 Collecte des Liens d'Articles

- **Script** : [wikipedia_all_articlelinks_getter_en.py](#)
- **Objectif** : Collecter tous les liens d'articles Wikipédia en anglais.
- **Méthode** : Utilisation de l'API MediaWiki pour récupérer les liens des articles via une requête avec pagination.
- **Fichier de Sortie** : [wikipedia_articles_links\(en\).csv](#) (environ 7 millions de lignes).

4.2 Extraction du Contenu HTML

- **Script** : [scraper.py](#)
- **Objectif** : Extraire le contenu texte de chaque article Wikipédia.
- **Méthode** : Téléchargement des pages HTML, nettoyage avec Trafilatura et expressions régulières.
- **Parallélisation** : Utilisation de [joblib](#) pour paralléliser les processus.
- **Fichiers de Sortie** :
 - Fichier temporaire pour reprendre les traitements.
 - [lienetarticles.csv](#): Fichier final contenant les titres, liens et contenus extraits.

4.3 Traduction vers le Yoruba (via Azure OpenAI GPT-4o)

- **Script** : [translator_yoru.py](#)
- **Objectif** : Traduire les articles extraits vers le Yoruba.

- **Méthode** : Utilisation de l'API Azure OpenAI GPT-4o(model performant sur la traduction de l'anglais en français) pour traduire les articles en Yoruba, avec un prompt structuré pour garantir la bonne qualité de la traduction.

-Prompt structuré : messages = [{ "role": "system", "content": ("You are a professional translator. " "Translate the entire input into accurate and complete Yoruba. " "Do not summarize. Do not comment. Return only the full Yoruba translation of the input.") }, {"role": "user", "content": text}]

Le prompt peut optionnellement prendre la colonne Title des articles pour une meilleure contextualisation.

- **Traitement par lots** : 100 articles par lot avec parallélisation et reprise des erreurs.
- **Enregistrement des traductions** : Dans le fichier [translation_yoruba.csv](#).

4.4 Traduction vers le Fon et l'Ewe (via Masakhane Models)

- **Scripts** : [translator_fon.py](#) et [translator_ewe.py](#)
- **Objectif** : Traduire les articles extraits vers Fon et Ewe.
- **Méthode** : Utilisation des modèles Masakhane pour la traduction via [transformers](#).
- **Tokenisation** : Utilisation de [nltk.sent_tokenize\(\)](#) et gestion des limites de tokens (max 1020 tokens).
- **Fichiers de Sortie** : [translation_fon.csv](#) et [translation_ewe.csv](#).

5. Justifications Techniques

- **Ressources de Corpus** : Le corpus anglais est plus riche, permettant d'enrichir les modèles de traduction pour des langues africaines à faibles ressources.

- **Modularité des Scripts** : Chaque étape est indépendante, facilitant la maintenance et l'extension du pipeline.
- **Gestion des Erreurs** : Le pipeline inclut des mécanismes de reprise après erreur et de gestion des timeouts.
- **Parallélisation** : Optimisation du traitement des millions d'articles grâce à la parallélisation.

6. Installation et Exécution

Prérequis

Cloner le dépôt :

```
git clone <url_du_repo>
cd <nom_du_projet>
```

●

Créer un environnement virtuel :

```
python3 -m venv venv
source venv/bin/activate
```

●

Installer les dépendances :

```
pip install -r requirements.txt
```

●

Exécution des Scripts

Exécutez les scripts dans l'ordre suivant pour collecter les liens, extraire le contenu et traduire les articles :

```
python scripts/wikipedia_all_articlelinks_getter_en.py
python scripts/scrapper.py
```

```
python scripts/translator_yoru.py
```

```
python scripts/translator_fon.py # (et variantes pour Ewe)
```

7. Tests et Validation

- Des échantillons manuels ont été comparés pour valider la qualité de la traduction.
- Les traductions sont sauvegardées avec le texte source pour une inspection manuelle.

Bonnes pratiques utilisées

- Sauvegarde incrémentale (mode **a** dans pandas).
- Contrôle d'erreurs avec **try-except**.
- Reprise des traitements grâce à la détection des **NaN**.
- Suivi du progrès avec **tqdm**.


8. Évolutions Futures

- **Intégration de la langue Dendi** (modèle en cours de fine-tuning).
- **Traduction dans les deux sens** (Anglais → Local et Local → Anglais).
- **Interface Web** pour la lecture des traductions.
- **Détection automatique de la langue source**.

9. Conclusion

Ce pipeline robuste et évolutif permet de traduire automatiquement des millions d'articles Wikipédia vers des langues africaines à faibles ressources, contribuant ainsi à enrichir la documentation numérique en **Fon**, **Ewe**, **Yoruba** et **Dendi**. Les perspectives futures incluent l'amélioration de la qualité des traductions par fine-tuning, l'optimisation du traitement de longs paragraphes et l'ajout d'une post-édition humaine pour affiner les traductions.

10. Liens vers les Fichiers de Sortie

Les fichiers de sortie sont stockés sur **Google Drive** pour leur taille importante. Vous pouvez y accéder ici :  `Wikipedia_project`

<https://drive.google.com/drive/folders/1YT56N9i0roOZ-7-ftK7HQR1EcEvd4pI?usp=sharing>

Fichiers de Sortie

- [wikipedia_articles_links\(en\).csv](#) : Liens collectés.
- [liensetarticles.csv](#) : Contenu extrait des articles.
- [translation_yoruba.csv](#) : Traductions en Yoruba.
- [translation_fon.csv](#) : Traductions en Fon.
- [translation_ewe.csv](#) : Traductions en Ewe.

Note : Toutes les lignes pour les traductions ne sont pas encore traitées, mais la logique du pipeline garantit des reprises continues et des exécutions sans perte de données.