



# Job App Assistant

Automatisation & Optimisation de Candidature par IA

Maxim Quénél | Pierre Louis Brun | Mame Alé Seye | Denis Bereziuc

# Le Contexte

## La barrière des ATS

Les candidats font face à un "trou noir". Les systèmes de suivi des candidatures (ATS) filtrent automatiquement les CVs ne correspondant pas exactement aux mots-clés, rejetant souvent des profils qualifiés.

## Le Défi

Adapter manuellement son CV pour chaque offre est :

⌚ Chronophage

🔁 Répétitif

⚠️ Sujet à l'erreur humaine



# Architecture Technique

## Structure Monolithique Modulaire

Une application web locale conçue pour la confidentialité ("Local-First") et la performance.

### Backend

Flask (Python). Gestion des threads pour les tâches IA longues.

### Frontend

HTML/JS Templates. Polling temps réel des logs.

### Stockage

Système de fichiers (CSV/TXT).  
Pas de SGBD lourd.

### Inférence

Hugging Face Transformers en local.

```
job_opp_ossistant/
├── opp.py
├── requirements.txt
└── msedgedriver.exe
    # Point d'entrée principal (Serveur Web / Orchestrateur)
    # Dépendances Python
    # Driver pour Selenium (Scraping Edge)

└── data/
    ├── jobs_raw.csv
    ├── jobs_rewritten.csv
    ├── cv_converted.txt
    ├── final_matches.csv
    └── final_matches_cross.csv
        # Stockage des données (Entrées/Sorties)
        # Données brutes scrapées
        # Offres redécrivées par IA
        # CV converti en texte brut
        # Résultats du matching simple
        # Résultats du matching avancé (Cross-Encoder)

└── services/
    ├── scraper.py
    ├── cv_converter.py
    ├── cv_rewriter.py
    ├── job_rewriter.py
    ├── matcher.py
    └── cross_encoder_matcher.py
        # Logique métier (Backend)
        # Script de scraping (Selenium)
        # Extraction de texte depuis PDF
        # Optimisation du CV par LLM
        # Nettoyage des offres par LLN
        # Algorithme de matching vectoriel
        # Algorithme de matching précis (Re-ranking)
        # Parsing initial des offres

└── static/
    ├── css/
    └── js/
        # Assets Frontend
        # Styles (Design des résultats et préviews)
        # Scripts côté client (main.js)

└── templates/
    └── index.html
        # Vues HTML
        # Page principale de l'interface

└── utils/
    └── logger.py
        # Utilitaires transverses
        # Gestion des logs
```

# Modèles & Intelligence Artificielle

## Génération & Réécriture

Qwen2.5-1.5B-Instruct



**Type :** SLM (Small Language Model).

**Usage :** Nettoyage des offres, anonymisation et structuration du CV.

**Pourquoi ?** Léger (1.5B paramètres), tourne sur GPU grand public, excellent suivi d'instructions.

## Matching (Niveau 1)

BAAI/bge-m3



**Type :** Bi-Encoder (Sentence Transformer).

**Usage :** Filtrage rapide.

**Méthode :** Calcul de similarité cosinus entre les vecteurs (embeddings) du CV et des offres.

## Reranking (Niveau 2)

BAAI/bge-reranker-v2-m3



**Type :** Cross-Encoder.

**Usage :** Scoring de précision (0-100%).

**Méthode :** Analyse conjointe (CV + Offre) pour capturer les nuances sémantiques fines.

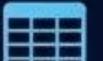
# Pipeline de Données Détailé (Flux)



# Stack Technique Détailée

## Core & Data

 **Python 3.12.8** : Langage principal

 **Pandas** : Manipulation des DataFrames (CSV)

 **Selenium** : Automatisation Web (Edge Driver)

 **pdfplumber** : Extraction texte depuis PDF

## IA & NLP

 **Transformers** : Hugging Face Hub

 **Torch** : Backend Deep Learning

 **Scikit-learn** : Utilitaires mathématiques

 **Sentence-Transformers** : Embeddings

# Interface Utilisateur

## Simplicité & Contrôle

L'interface propose un parcours en 6 étapes distinctes.



**Feedback Temps Réel** : Polling API pour afficher les logs d'exécution.



**Prévisualisation** : Vérification des fichiers intermédiaires (JSON/TXT) avant de passer à l'étape suivante.



**Graphiques** : Visualisation immédiate des scores de matching.

### Job AI Assistant

Automatisation intelligente de votre recherche d'emploi

**01 Recherche d'Offres**  
Scrapes les dernières offres depuis HelloWork.

Per Mois  
Per Loin  
Per Toute Stuff

Mot(s)

Data Analyse

Nombre d'offres

8

Lancer Analyse

En attente

**02 Réécriture Offres (IA)**  
Analysez et résumez les offres avec Owen 1.58.

Lancer Réécriture

En attente

**03 Conversion CV**  
Convertissez votre CV PDF en texte brut nettoyé.

Choisir un PDF

Aucun fichier

Convertir

En attente

**04 Synthèse CV (IA)**  
Génère une synthèse structurée de votre profil.

Générer Synthèse

En attente

**05 Calcul Matching**  
Comparez votre profil aux offres (Semantic Search).

Calculer Scores

En attente

**06 Cross Matching (Ultra-Précis)**  
Celui précis avec Cross Encoder (plus lent mais meilleur).

Lancer Crée Natch

En attente

# Test et Comparaison

## Matching avec CV Correspondant

### 05 Calcul Matching

Comparez votre profil aux offres (Semantic Search).

Calculer Scores

Terminé

Stage Intelligence Artificielle H/F **75.2%**

Proxiad

IA Résumé: - Type de profil recherché: Etudiant(s) en Sciences des Données (ou équivalent) | -...

Voir l'offre →

### 06 Cross Matching (Ultra-Précis)

Calcul précis avec Cross Encoder (plus lent mais meilleur).

Lancer Cross Match

Terminé

Stage Intelligence Artificielle H/F **67.5%**

Proxiad

IA Résumé: - Type de profil recherché: Etudiant(s) en Sciences des Données (ou équivalent) | -...

Voir l'offre →

### 05 Calcul Matching

Comparez votre profil aux offres (Semantic Search).

Calculer Scores

Terminé

Stage Intelligence Artificielle H/F **52.4%**

Proxiad

IA Résumé: - Type de profil recherché: Etudiant(s) en Sciences des Données (ou équivalent) | -...

Voir l'offre →

### 06 Cross Matching (Ultra-Précis)

Calcul précis avec Cross Encoder (plus lent mais meilleur).

Lancer Cross Match

Terminé

Responsable Intelligence Artificielle H/F **50.0%**

UNIKALO

IA Résumé: - Type de profil recherché: Responsable Intelligence Artificielle H/F | - compétences clés...

## Matching avec CV Non Correspondant

# Améliorations & Roadmap



## Robustesse

Remplacer threading par une file de tâches (Celery/Redis) pour gérer les échecs et la montée en charge.



## Données

Migration des fichiers CSV vers **SQLite** pour permettre des requêtes complexes et un meilleur filtrage.



## Performance

Quantification des modèles (4-bit via bitsandbytes) pour réduire la consommation VRAM.

# Questions ?

Merci de votre attention.



[github.com/Maxim-Quenel/job\\_app\\_assistant\\_matching\\_cv](https://github.com/Maxim-Quenel/job_app_assistant_matching_cv)



Démo sur youtube :

[https://youtu.be/R0ofqB8\\_0bw?si=ktxXZxt8ko71zywI](https://youtu.be/R0ofqB8_0bw?si=ktxXZxt8ko71zywI)