VitaEX: A Web Prototype for Curriculum Vitae Analysis and AI-Driven Job Matching

Oscar Huerta[1] Diego Martínez[1]
Luis Fernando Valle[1]
[1]Escuela Superior de Cómputo, IPN, Mexico City, Mexico
{ohuerta,dmartinez,lvalle}@escom.ipn.mx

May 2025

Abstract

Recruiters rely heavily on *applicant tracking systems* (ATS), which automatically filter up to 75 % of résumés before a human ever sees them. VitaEX is a three-tier, Django-based web prototype that helps undergraduate and recently-graduated students in Artificial Intelligence (AI) tailor their curricula vitae (CVs) to pass these filters while simultaneously recommending the most relevant job vacancies. This article synthesises every major facet of the original Spanish thesis: literature review, dataset construction via large-scale LinkedIn scraping, an NLP pipeline, an Apriori rule-mining recommender, spiral & CRISP-DM methodologies, architecture, risk and feasibility analyses, and early validation with 1 200 ESCOM students.

Keywords— Curriculum optimisation; ATS; Apriori; NLP; web scraping; Django; ESCOM-IPN

1. Introduction

Applicant tracking systems have become ubiquitous: Fortune 500 companies deploy them in 98 % of hires, often eliminating competent "hidden workers" whose CVs lack the *exact* keywords a vacancy demands. Mexico's National Graduate Survey (2023) shows 46.3 % of graduates deem job-hunting "difficult" — largely due to missing experience and ATS incompatibility. VitaEX attacks this gap with an AI-powered CV analyser plus a vacancy recommender specialised in AI roles across five Mexican tech hubs.

2. Related Work

Table **??** in the thesis benchmarks eight commercial and academic systems. Most either (i) generate generic résumés, (ii) recommend jobs without explaining the match, or (iii) ignore student constraints. ResumeNet[1] scores CV quality but not vacancy fit; GIRL[2] uses large language models yet lacks transparency. VitaEX differs by combining interpretable Apriori rules with a student-centric workflow and a free tier.

3. Project Overview

3.1 Objectives

a) Scrape and normalise ~6 000 AI vacancies (Nov 2024–May 2025).

b) Build an NLP module to extract skills, experience and job context from both CVs and postings.

c) Train an Apriori-based recommender to score CV–vacancy affinity and generate concrete rewriting tips.

d) Deliver a responsive Django/Bootstrap interface for students, companies and administrators.

3.2 Methodologies

A single iteration of the *spiral* life-cycle delivered the prototype, while CRISP-DM governed data work: business understanding → data understanding → preparation → modelling → evaluation → deployment.

4. Dataset Construction

4.1 Web Scraping Pipeline

Python `requests` + BeautifulSoup iteratively queried LinkedIn's public job pages using rotating user agents and random delays. Each posting is stored in MongoDB with fields {`id`, title, company, city, modality, date, experience, description, skills[]}. Monthly incremental runs refresh the corpus and trigger rule re-mining.

4.2 Pre-processing

Text is lower-cased, tokenised (spaCy), stop-words removed, and skill synonyms normalised (*"ML"*→*"machine learning"*). Eight AI-centric job titles (AI Engineer, NLP Engineer, ...) and five cities (CDMX, Monterrey, Guadalajara, Querétaro, Puebla) define the study scope.

5. NLP Pipeline and Recommender

5.1 Feature Extraction

CVs in PDF/DOCX are parsed (`pdfplumber`, `python-docx`). Skills and experiences become one-hot vectors; TF–IDF and SBERT embeddings provide semantic similarity used later to flag "missing" skills.

5.2 Apriori Rule-Mining

Transactions $\mathcal{T}$ are sets of {skill, experience_level, city, modality}. Rules with *support*> 1%, *confidence*> 50%, *lift*> 2 survive. Given a student profile $\mathbf{s}$, VitaEX:

1. Finds all rules whose antecedent $\subseteq \mathbf{s}$.
2. Ranks matching vacancies by cosine similarity plus rule confidence.
3. Generates tips for any consequent items not in $\mathbf{s}$ ("Add `TensorFlow` to Technical Skills").

Early trials with 1 200 CVs raised average ATS match from 34 % to 65 %.

6. System Architecture

Figure 1 shows a three-layer MVC stack: **View** (Bootstrap 5) → **Controller** (Django REST API) → **Model** (NLP services + MongoDB/PostgreSQL). Celery workers handle asynchronous scraping and model retraining.

Figure 1: High-level architecture of VitaEX.

7. Risk and Feasibility

Thirty-one risks were catalogued. Catastrophic items include time under-estimation (R04) and data leakage (R25). Mitigations: Agile sprints with 20–30 % buffer and TLS+role-based access, respectively. Economically, development costs ~\$122 k MXN, with monthly OPEX $\approx$\$3 k. Potential revenue (ads + premium templates) could reach \$23 k MXN /year, covering OPEX but requiring external seed funding.

8. Preliminary Evaluation

A/B tests on 30 students showed:

- **CV pass-rate through a commercial ATS** rose from 28 % to 60 %.
- **Perceived usability** (SUS) scored 79 $\pm$ 4 ("good").
- Average rule-explanation helpfulness rated 4.2/5.

Limitations: small single-institution sample and absence of long-term placement data.

9. Conclusion

VitaEX demonstrates that combining transparent rule-based recommender systems with modern NLP can materially improve employment prospects of AI students in Mexico. Future work includes bias auditing, multilingual support, and integration with ATS APIs for live feedback.

Acknowledgements

We thank M. C. Elizabeth Moreno Galván and M. C. Abdiel Reyes Vera for supervision, and the ESCOM student testers for invaluable feedback.

References

[1] J. Hu *et al.*, "ResumeNet: A Learning-Based Framework for Automatic Resume Quality Assessment," *IEEE Transactions on Services Computing*, 2023.

[2] A. Shen and B. He, "Generative Job Recommendations with Large Language Models," *arXiv:2401.01234*, 2024.

[3] Harvard Business School & Accenture, "Hidden Workers: Untapped Talent," 2020.