Talent Recommendation Engine: RanKIT

Prepared by Edgard El Cham and Roy Aad

Introduction

Motivation

- Today, connecting the right candidates with the appropriate job position is more critical than ever.
- Manual screening is time-consuming and prone to errors and biases.
- Businesses are looking for more effective ways to match potential employees to opportunities.

Previous Solutions

Keyword Matching

Job Description	Resume
Javascript	Javascript
Node.js	Node

Early Machine Learning Approaches

Document Extraction and Cleaning
↓
TF-IDF Vectorizer
↓
Similarity Computation
↓
Ranking

Problem Definition

- Previous solutions are **inefficient**, **time-consuming** and prone to **errors** and **biases**.
- They **fail** to capture the nuanced connections of experience and skills leading to **mismatches**.

Objectives

- Automating the matching of candidates to job opportunities based on **work experience**, **language** and **technical skills**.
- Leveraging AI to achieve more accurate recommendations.

Experimental Setup

Dataset

- 13 unique job descriptions.
- 126 unique resumes, of which 2 were corrupted.
- All documents in .pdf or .docx format.

Limitations of Old Parsers

- spaCy was not able to detect basic information such as languages as shown below:

- While, widely known resume parsers, like PyResParser, failed to correctly extract most information.


```
"Facebook",
"Content",
"Analysis",
"Jsp",
"Jira",
"Modeling",
"Servers",
"Php",
"Cms",
"Communication",
"Benchmarking",
"Iphone",
"Hotel",
"Database",
"Technical",
"Apis",
"Linux",
"Html",
"Website",
"Audit",
"Rest",
"Sql",
"English",
"Hotels",
"Programming",
"Design",
"French",
"System",
"Windows",
"C",
"Research",
"Unix",
"Hospital",
"Html5",
"Aws",
"Mobile",
"Requests",
"Xml",
"Oracle",
"Presentation",
"Javascript",
"Administration",
"Coding",
"Github",
"Shell",
"Ubuntu",
"C++",
"Mysql"
],
"college_name": null,
"degree": null,
"designation": null,
"experience": null,
"company_names": null,
"no_of_pages": 4,
"total_experience": 0
```

Exploration of Large Language Models (LLMs)

Model	Computational Requirements	Accuracy	Cost
Free Dolly	Low	Low	Free
Llama 2	High	High	Free
Falcon	High	High	Free
Llama 2 (Small)	Low	Low	Free
Falcon (Small)	Low	Low	Free
ChatGPT	None (API)	High	$0.005/CV

- Given our limited computational resources, we chose OpenAI's ChatGPT for its superior performance at a relatively low cost.

Prompt Engineering

- **Prompt Engineering** played a vital role in **accurately extracting data**.
- LangChain allows us to specify a response schema to format the output. In our case, we went for the **JSON format**.
- For the **job descriptions**, we crafted a **single prompt** to extract all the necessary information.
- For **resumes**, we found that using one prompt led to inaccuracies. We opted for **smaller distinct prompts to target the different sections** of the resumes.

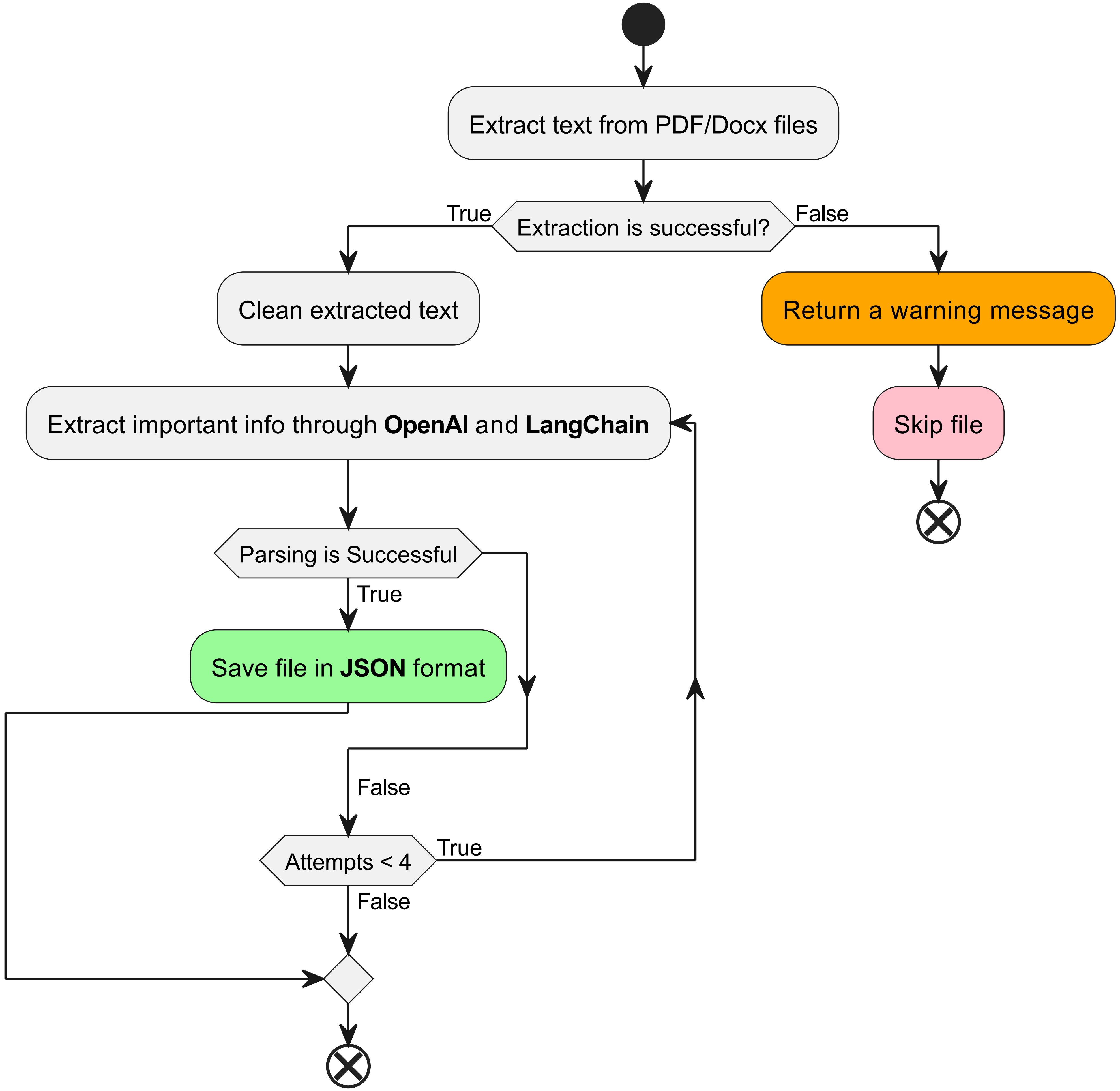
Choice of Embeddings

- **Word2Vec** and **GloVe** vocabulary is **deficient** in encompassing domain-specific terms such as "Kubernetes" and "Kotlin".
- They do not allow for words featuring punctuations like "C#," "C++," "Node.js," and "React.js."
- They do not allow for terms such as "project management," "distributed systems design," and "relational database."
- OpenAI's text embeddings such as **DaVinci** and **Curie** are **less accurate** and **costly**.
- Our best option was OpenAI's **ada-002** embedding which is **accurate** for semantic search and has a **low cost** of 0.0001$/1,000 tokens.

Final Proposed Approach

Parsing

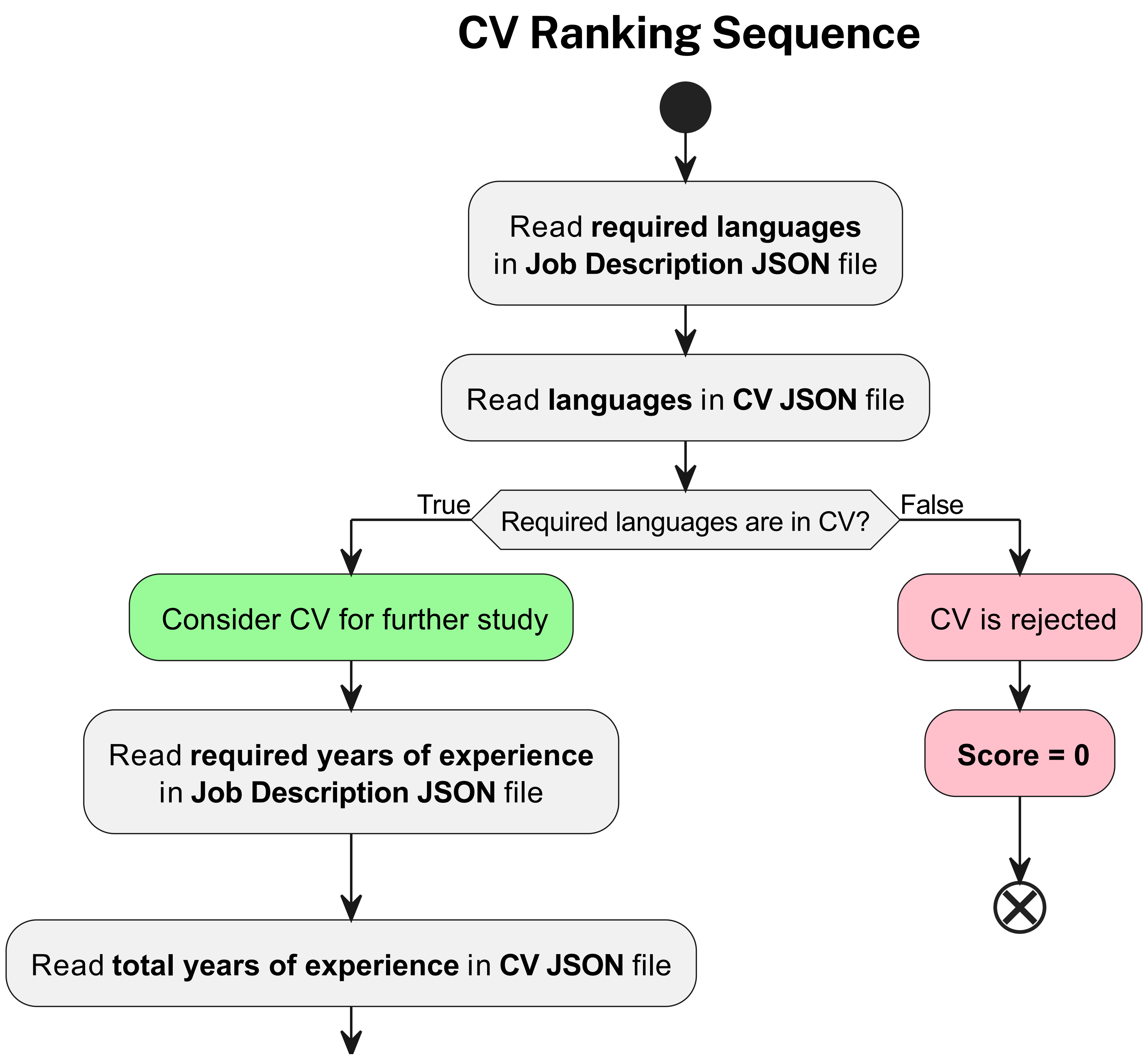
Job Description & CV Parsing Sequence



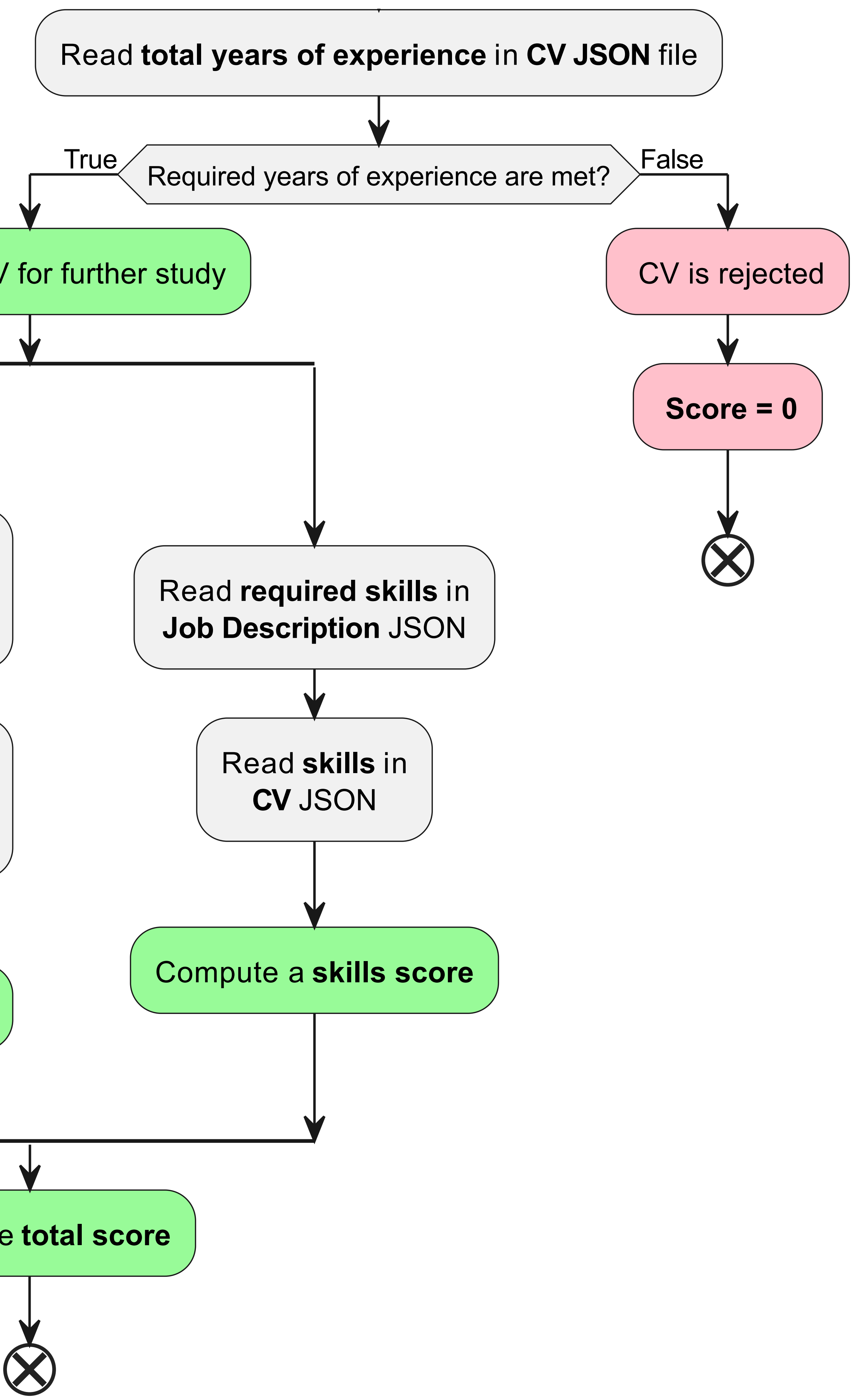
Parsing

- For **job descriptions**: information is extracted with one specific prompt**.**
- For **resumes**: we had to extract information in a step-wise manner using multiple prompts that target the different sections.
  - ***Experience Breakdown*****:** simplifies the resumes, and then extracts job titles and durations.
  - ***Skills*****:** extracts relevant skills from the entire resume.
  - ***Language Proficiency*****:** Extract language skills and proficiencies.

Ranking Flowchart



Ranking Flowchart



Ranking Modes

- We developed 2 language assessment modes:
  - ***Strict***: Candidate is proficient in all the required languages stated in the job description.
  - ***Relaxed***: Candidate is proficient in at least two-thirds of the required languages stated in the job description.

- We developed 3 scoring modes:
  - ***Match***: Uses keywords fuzzy matching.
  - ***ADA***: Uses ada-002 context matching.
  - ***ADAMa:*** Uses a combination of ada-002 and fuzzy matching to prioritize certain desired job titles.

Results

Job Description Parsing

- Job description is parsed within a few seconds and a JSON is returned containing accurate important information.

DevOps Engineer - Infrastructure Focus

Focused Qualifications

1. Bachelor's Degree in Computer Science or related field
2. 1-3 years of professional cloud-based infrastructure (AWS Preferred)
3. 1+ years of experience with Java development
4. 1+ years of experience with SQL database design and development
5. 1+ years of experience working in an agile environment
6. Familiarity with web service design and development


```
{
    "job_title": "DevOps Engineer",
    "total_years_required": 1,
    "languages": [
        "English"
```

CV Parsing

- Resume is parsed within 20 - 40 seconds and a JSON is returned containing accurate important information.

WORK EXPERIENCE

Software Engineer
PLUGIT Limited
07/2012 - Present
PLUGIT is a certified provider of trading support solutions and services to the global financial trading industry.
Achievements/Tasks
- Develop, test and implement new software programs.
- Clearly and regularly communicate with management and technical support colleagues.

LANGUAGES

Arabic
Native or Bilingual Proficiency

English
Full Professional Proficiency


```
{
    "breakdown_experience": {
        "Software Engineer": 11.08
    },

    "languages": {
        "Arabic": 5,
        "English": 4
    },
```

Ranking

- TF-IDF shows a very poor ranking accuracy.
- ADA shows a huge improvement from TF-IDF & ADAMa a slight improvement from ADA

		Rankings			
		Supportful	TF-IDF	ADA	ADAMa
DevOps Engineer	Elie Youssef	#1	#35	#7	#8
	Hussein Hijazi	#2	#25	#1	#2
	Charbel Nakad	#3	#41	#4	#7
DotNet	Bernard Estephan	#1	#61	#16	#7
	Mohammad Darweech	#2	#78	#23	#3
Quantitative Developer	Tony Mattar	#1	#75	#4	#10
	Patricia Boulos	#2	#79	#29	#9

Conclusion

Conclusion

- We developed a ranking engine with various language, work experience, and skills assessment modes.
- We used state of the art LLMs to accurately parse the job descriptions and resumes.
- We use state of the art ada-002 text embeddings and drastically improved the rankings as compared to old techniques.

Future suggestions

- Data gathering and data cleaning to build and train better models.
- LLM fine-tuning for faster parsing.
- Tokenizer and embeddings training for a higher ranking accuracy.