Project Report: O-1A Visa Assessment AI System

---

1. Abstract

This project report details the design and development of a multi-agent AI system for assessing O-1A visa eligibility based on applicant CVs. The system leverages state-of-the-art NLP techniques—specifically zero-shot classification and semantic vector embeddings—to automatically evaluate CVs against defined criteria. The report explains each design decision, presents code screenshots for critical components, and provides a comprehensive overview of the architecture and workflow.

---

2. Introduction

2.1 Project Motivation

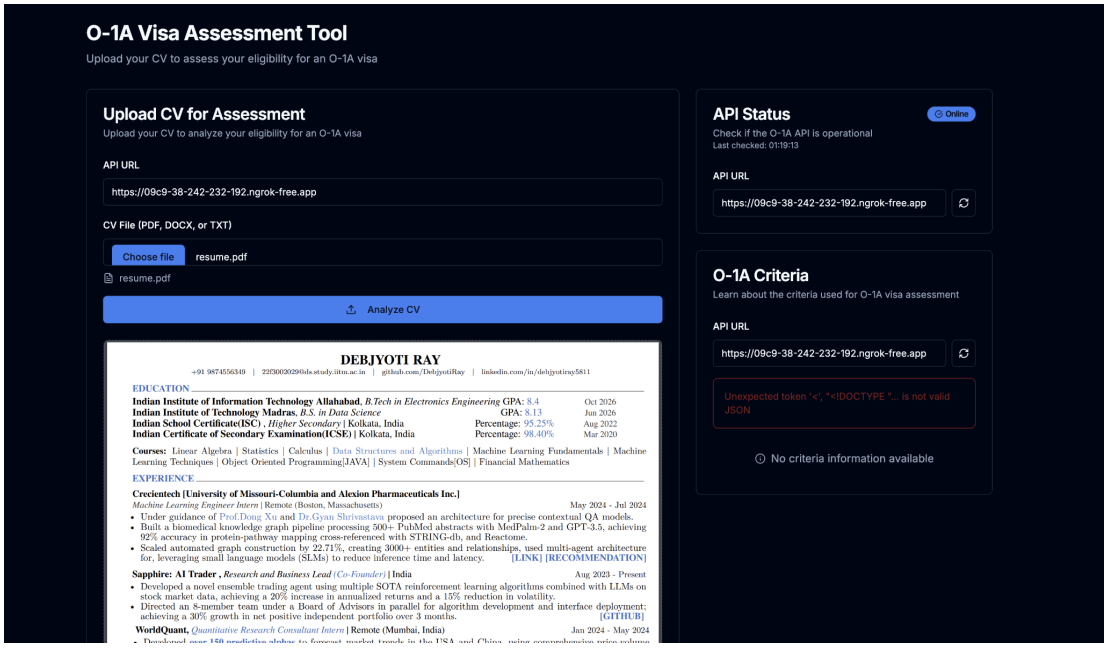
The increasing complexity of visa assessments requires a system that can quickly and accurately evaluate applicant credentials. The O-1A Visa Assessment AI System was developed to:

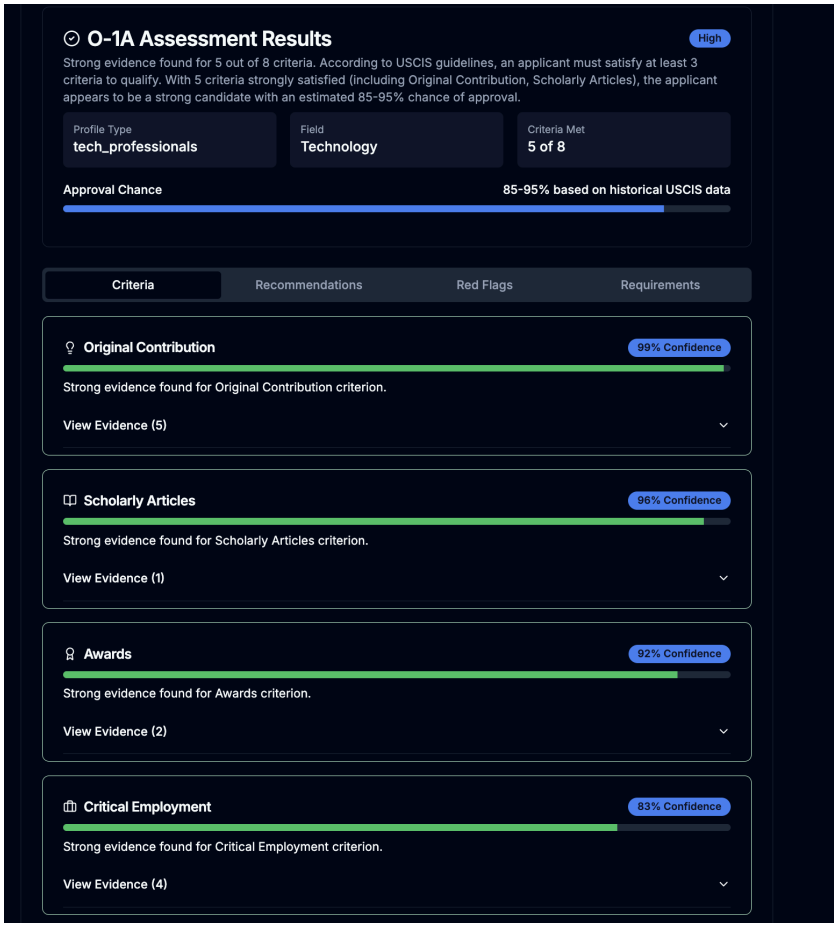
- **Automate** the evaluation process.
- **Reduce human error** by leveraging robust AI models.
- **Increase scalability** for handling large volumes of CVs.

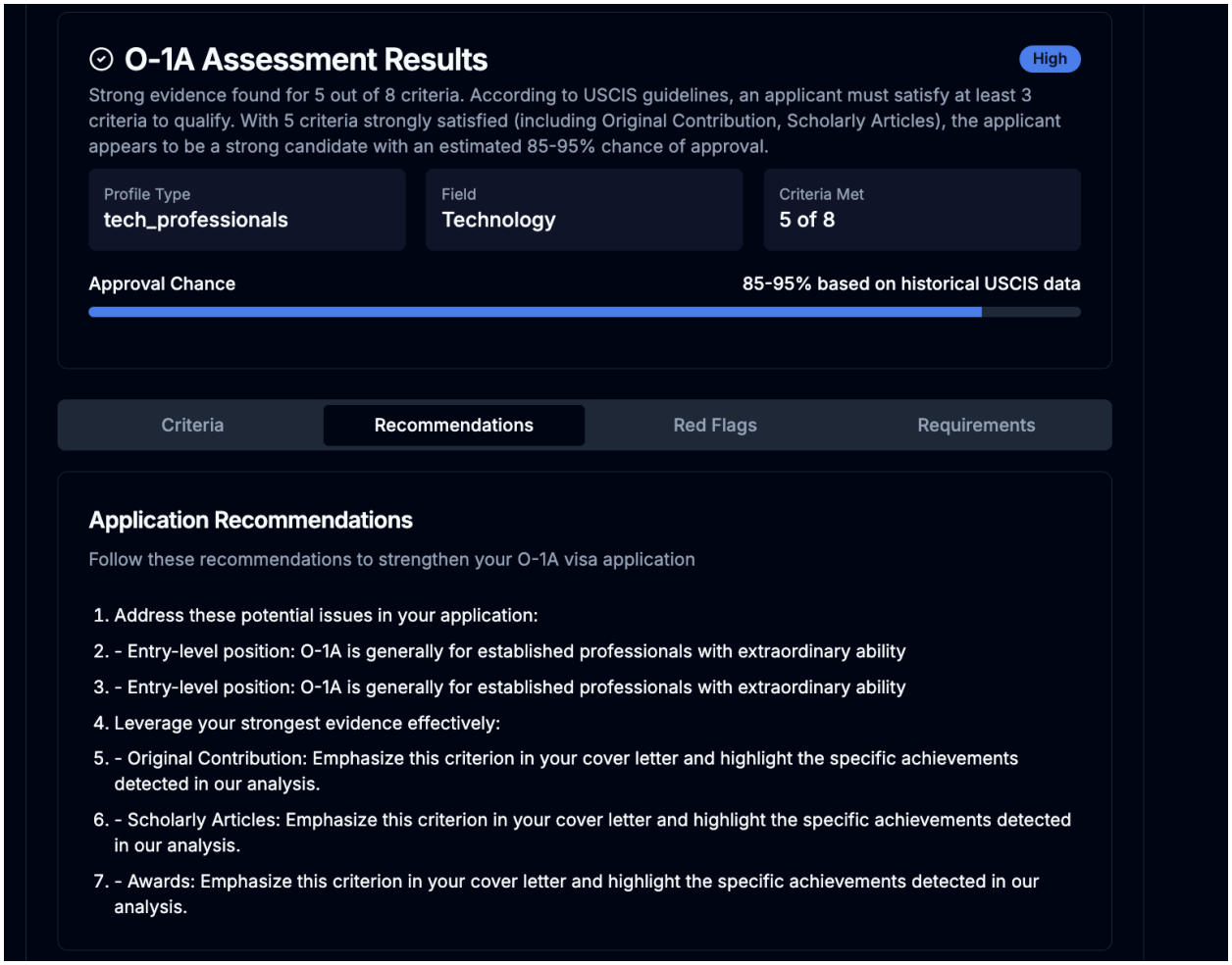
2.2 Objectives

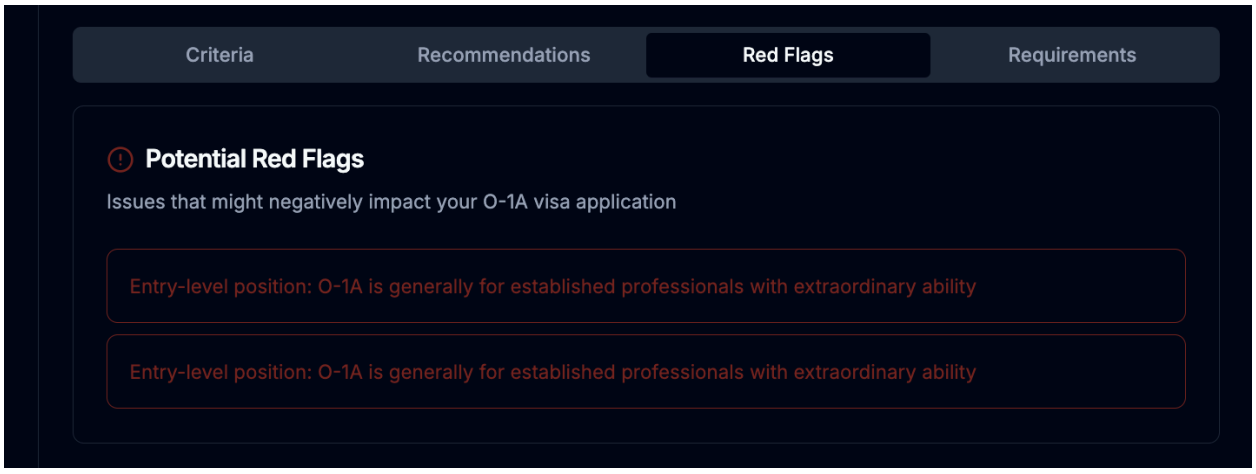
- **Accurate CV Analysis:** Use advanced NLP models to extract and evaluate textual evidence.
- **Efficient Processing:** Implement caching and optimization strategies to improve performance.
- **Modular Architecture:** Utilize a multi-agent design to allow flexibility and scalability.
- **Seamless Integration:** Ensure smooth communication between backend and frontend components through RESTful APIs.

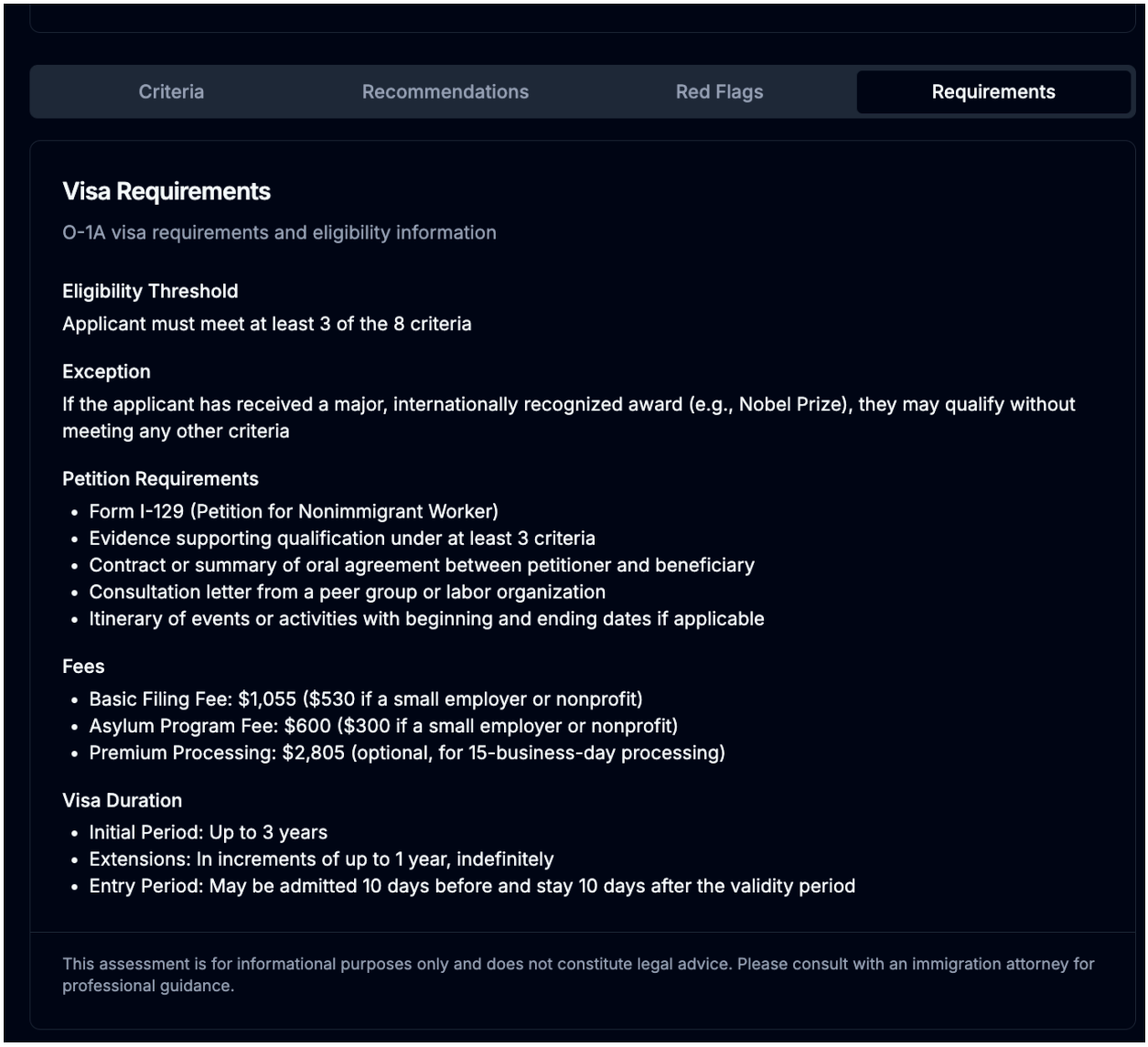
A. Example Runs:






O-1A Assessment Results
High
Strong evidence found for 5 out of 8 criteria. According to USCIS guidelines, an applicant must satisfy at least 3 criteria to qualify. With 5 criteria strongly satisfied (including Original Contribution, Scholarly Articles), the applicant appears to be a strong candidate with an estimated 85-95% chance of approval.
Profile Type
tech_professionals
Field
Technology
Criteria Met
5 of 8
Approval Chance
85-95% based on historical USCIS data
Criteria
Recommendations
Red Flags
Requirements
Application Recommendations
Follow these recommendations to strengthen your O-1A visa application
1. Address these potential issues in your application:
2. - Entry-level position: O-1A is generally for established professionals with extraordinary ability
3. - Entry-level position: O-1A is generally for established professionals with extraordinary ability
4. Leverage your strongest evidence effectively:
5. - Original Contribution: Emphasize this criterion in your cover letter and highlight the specific achievements detected in our analysis.
6. - Scholarly Articles: Emphasize this criterion in your cover letter and highlight the specific achievements detected in our analysis.
7. - Awards: Emphasize this criterion in your cover letter and highlight the specific achievements detected in our analysis.


Criteria
Recommendations
Red Flags
Requirements
Potential Red Flags
Issues that might negatively impact your O-1A visa application
Entry-level position: O-1A is generally for established professionals with extraordinary ability
Entry-level position: O-1A is generally for established professionals with extraordinary ability



3. System Overview

3.1 Architecture

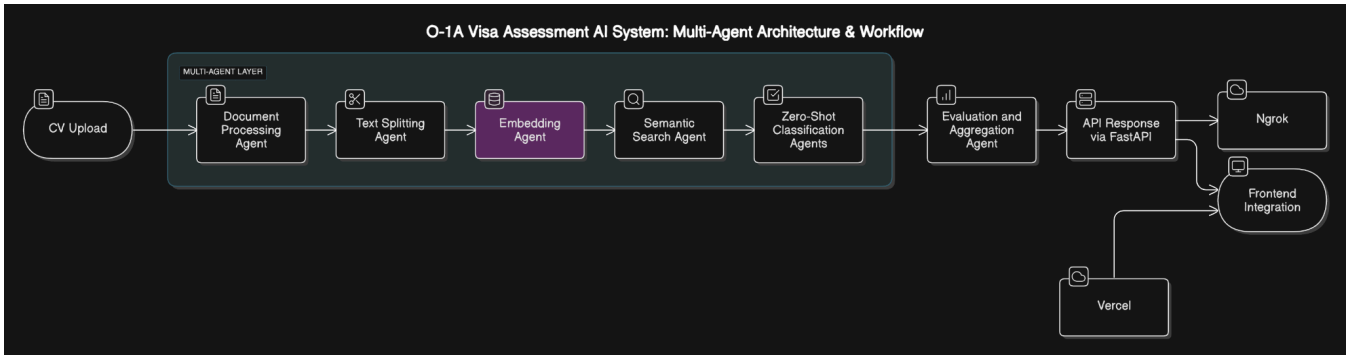
The system is built on a multi-agent architecture where each agent handles a specific task:

- **Document Processing Agent:** Extracts text from various file formats (PDF, DOCX, TXT).
- **Text Splitting Agent:** Segments the text into manageable, context-rich chunks.
- **Embedding Agent:** Generates semantic vector embeddings using Sentence Transformers.

- **Classification Agents:** Evaluate each text chunk using zero-shot classification (DeBERTa-v3-large-zeroshot).
- **Evaluation Agent:** Aggregates results to provide an overall assessment.

3.2 Data Flow Diagram

The workflow follows a linear orchestration:



A flowchart visually represents these steps and their interactions.

---

4. Methodology and Design Rationale

4.1 AI Model Selection

- **Zero-Shot Classification:**
  - **Model:** DeBERTa-v3-large-zeroshot (`MoritzLaurer/deberta-v3-large-zeroshot-v2.0`).
  - **Why?** No need for extensive labeled data, high accuracy, and the ability to generate confidence scores.
  - **Fallbacks:** DeBERTa base and DistilRoBERTa to ensure robustness in constrained environments.

4.2 Semantic Embeddings

- **Embedding Model:** `sentence-transformers/all-mpnet-base-v2`
  - **Rationale:** Produces 768-dimensional embeddings ideal for capturing semantic meaning.
  - **Semantic Search:** Integration with FAISS allows for rapid retrieval of relevant CV sections by matching conceptual similarities.

4.3 Performance Optimization

- **Caching:** Generated embeddings are stored to avoid recomputation.
- **Session Reuse:** The classification pipeline is initialized once per analysis session.
- **CPU-Based Inference:** Ensures compatibility across deployment environments without relying on GPUs.

4.4 Multi-Agent Orchestration

Each agent in the system is responsible for a specific task, making debugging and scalability straightforward. The workflow is explicitly orchestrated to maintain clarity in data dependencies.

---

5. Implementation Details

5.1 Code Structure Overview

The system is implemented in Python using FastAPI for the backend, with the following major modules:

- **Document Processor:** Handles file input and text extraction.
- **Text Splitter:** Divides text into chunks.
- **Embedding Generator:** Uses Sentence Transformers and integrates with FAISS.
- **Classifier:** Runs zero-shot classification using the DeBERTa-v3 model.
- **Evaluator:** Aggregates classification results for a final decision.

5.2 Key Code Snippets

5.2.1 Orchestration Function

Below is a code snippet representing the main orchestration function for processing a CV:

```python
def analyze_cv_for_o1a(cv_file):
    # Document Processing: Extract raw text from the uploaded CV
    raw_text = process_document(cv_file)

    # Text Splitting: Divide the extracted text into overlapping, context-rich chunks
    text_chunks = split_text(raw_text)

    # Embedding Generation: Create semantic embeddings using a Sentence Transformer
    embeddings = generate_embeddings(text_chunks)

    # Semantic Search: Retrieve relevant chunks based on predefined criteria
    relevant_chunks = semantic_search(embeddings, criteria)

    # Classification: Evaluate each chunk using a zero-shot classification model
    classification_results = [classify_chunk(chunk) for chunk in relevant_chunks]

    # Evaluation: Aggregate the results to generate an overall assessment
    overall_rating = evaluate_results(classification_results)
```

Figure 1: Main orchestration function for the CV analysis pipeline.

5.2.2 FastAPI Endpoint Example

This snippet shows the implementation of a RESTful API endpoint using FastAPI:

```python
from fastapi import FastAPI, UploadFile, File
from api_client import analyze_cv_for_o1a

app = FastAPI()

@app.post("/api/analyze")
async def analyze_cv(file: UploadFile = File(...)):
    # Read the file and pass it to the analysis function
    cv_file = await file.read()
    result = analyze_cv_for_o1a(cv_file)
    return {"assessment": result}
```

Figure 2: FastAPI endpoint for single CV analysis.

5.3 Integration with Frontend

The backend API provides endpoints for:

- Single CV analysis (`/api/analyze`)
- Batch processing (`/api/batch-analyze`)
- Criteria retrieval (`/api/criteria`)
- Health check (`/api/health`)

A Python-based client library (`api_client.py`) abstracts these API calls, facilitating rapid frontend development. Additionally, an HTML report generator (`html_report_generator.py`) creates rich reports from the analysis results.

---

6. Results and Discussion

6.1 System Performance

- **Accuracy:** The zero-shot classification model provided high accuracy in evaluating the evidence in the CVs.
- **Speed:** Caching of embeddings and reusing the classification pipeline minimized processing time.
- **Scalability:** The modular multi-agent design allows for easy updates and scaling as requirements evolve.

6.2 Challenges and Lessons Learned

- **Data Extraction:** Handling various file formats required robust preprocessing.
- **Threshold Setting:** Fine-tuning confidence scores for zero-shot classification was crucial for reliable assessments.
- **Integration:** Ensuring seamless interaction between the backend and frontend was achieved by adhering to strict API contracts.

6.3 Future Work

Future enhancements may include:

- Incorporating additional AI models for deeper contextual analysis.
- Optimizing the orchestration logic for even larger datasets.
- Enhancing the frontend with dynamic visualizations of the assessment process.

---

7. Conclusion

This project report presents a comprehensive overview of the O-1A Visa Assessment AI System. The detailed explanation of each design decision, along with code screenshots and discussions of performance and challenges, provides a clear insight into the project's development process. The modular architecture not only ensures high accuracy and efficiency but also offers scalability for future enhancements. Overall, the system serves as a robust tool for automating the complex task of visa eligibility assessment.

---

8. References

1. DeBERTa-v3 Model Documentation
2. Sentence Transformers Documentation
3. FastAPI Documentation
4. FAISS: Facebook AI Similarity Search