NILFRAUD: A Resume Fraud Detector

1 Introduction

In the evolving landscape of talent acquisition, the integration of artificial intelligence (AI) is becoming increasingly essential. This report outlines an AI-powered solution designed to streamline HR processes by automating the management and evaluation of resumes and recommendation letters (LORs). The system employs advanced natural language processing (NLP) techniques to enhance decision-making and improve the quality of candidate assessments.

Contents

2 PDF Management

We started with the dataset, and the first task was to find the relevant content from the resumes.

- **Implementation:** The system begins with the conversion of PDF resumes into raw text data. Using libraries such as Fitz, it extracts relevant content while ensuring unnecessary information is filtered out. This is crucial for creating a clean dataset for further analysis.

  We used the Gemini 1.5 Flash API to Restructure all the data in the raw text into 4 main sections: **Education**, **Experience**, **Skills**, **Sector**. The API was used only to process and re-frame the data. The prompt was specifically designed, and Temperature set to be very low, so that it almost deterministically processes data in correct format.

- **Requirement:** A reliable extraction process enables HR teams to focus on meaningful data without getting bogged down by irrelevant information, making it easier to analyze candidates' qualifications effectively.

- **Technical Approach:** The implementation of a scalable architecture allows the system to process thousands of resumes simultaneously. Leveraging cloud-based services ensures that the text extraction process can handle high volumes of data without performance degradation. The introduction of LLMs and the Gemini API adds another layer of intelligent data extraction by making use of natural language understanding to automatically detect and categorize important resume sections.

3 Skills Extraction and Cleaning

After reading, we have to divide the meaningful data. The foremost of these was skills. Skills are most important as, when manually checking the data, LORs and work experience had direct dependence on skills.

- **Implementation:** The solution utilizes the LLAMA-3.2 90B text preview API to refine and filter the skills section from resumes. It was observed that most resumes have unnecessary skills mentioned, or the mentioned highlights are not really skills, but jargon. From the standard format generated before, we extracted and saved the words that were actual skills and not just key words. They were further verified and filtered against the experience for relevance using a mixbread ai model.

- **Requirement:** A curated skills list enhances clarity for HR evaluators, allowing them to quickly identify the qualifications that match job requirements. This ensures that candidates are assessed based on their true capabilities.

- **Technical Approach:** The skills extraction process is designed to be modular, allowing for easy updates to the skill filtering logic. Machine learning models can be retrained with new skills data, ensuring the system adapts to changing job markets. Furthermore, the skills processing module can run in parallel with other data extraction tasks, improving throughput and reducing overall processing time.

4 Scoring System for Experience

This section outlines the methodology used to calculate relevance scores between a candidate's experiences and their skills, which is essential for detecting potential inconsistencies or fraud in resumes.

- **Implementation:** The system processes the candidate's experience and skill information by first reading and removing stopwords to focus on key content. The relevant experiences and skills are then compared to assess how well they match, using a pre-trained embedding model.

  The system reads candidate experiences from text files and extracts them into sections labeled under "Experience" and "Skills". Skills are further processed to remove any irrelevant words (stopwords) to better match them against the processed experience data.

  The experience data and skill data are then encoded into embeddings using a pre-trained model, such as 'SentenceTransformer'. For each skill, the cosine similarity between its embedding and the embeddings of experiences is computed. The mixbread model was used here again

- **Formula for Cosine Similarity:** Cosine similarity measures the alignment between a skill and experience, providing a numerical relevance score between 0 and 1. The cosine similarity between two vectors A (skill) and B (experience) is computed as:

$$\cos\_\text{sim}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

  Where:

  - $A \cdot B$ is the dot product of vectors A and B
  - $\|A\|$ and $\|B\|$ are the magnitudes (L2 norms) of vectors A and B

- **RMS (Root Mean Square) Score:** To further evaluate the relevance of skills across all experiences, an RMS (Root Mean Square) score is calculated for each skill's relevance scores across the set of experiences. The RMS is given by:

$$\text{RMS} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \text{score}_i^2}$$

  Where:

  - n is the number of experiences
  - score_i is the cosine similarity score for skill i with each experience

- **Threshold-Based Filtering:** Skills with an RMS score above a specified threshold (e.g., 0.51) are considered highly relevant. The number of such skills is counted to help evaluate the candidate's suitability for the role.
- **L2 Norm of Relevance Scores:** For a more comprehensive assessment, the combined L2 norm of all high-RMS skill relevance scores is computed, providing a measure of the overall alignment of the candidate's skills with their experiences:

$$\text{L2 Norm} = \sqrt{\sum_{i=1}^{m} \text{score}_i^2}$$

  Where m is the number of skills whose RMS exceeds the threshold.
- **Requirement:** This scoring system provides an objective way to quantify how well the candidate's skills align with their job experiences, making it easier to identify inconsistencies or potential embellishments in their resume. By calculating the cosine similarity between each skill and experience, and applying thresholds to the RMS scores, HR teams can focus on the most relevant information during candidate evaluation.
- **Technical Approach:** The system uses pre-trained transformer models (such as 'SentenceTransformer') to convert both skills and experiences into high-dimensional embeddings. These embeddings are compared using cosine similarity to quantify the relevance of each skill to the candidate's experience. The system is built to handle large datasets and can process multiple candidates efficiently.

5 Flagging in LORs

We then moved to the second section of data, i.e. LORs. We have to filter the exaggeration in LORs while keeping a record of that, indicating the vague nature of the corresponding resume points.

- **Implementation:** The solution incorporates a mechanism to flag vague or exaggerated phrases within LORs. By analyzing short phrases in context, it identifies statements that may lack specificity or supporting evidence. The system generates alerts for phrases that could mislead hiring teams. LLAMA-3.2 90B was used here to do a higher order scemantic analysis.
- **Requirement:** Ensuring the credibility of recommendations is crucial for accurate candidate evaluation. By flagging vague statements, HR can take a closer look at potentially misleading endorsements, increasing the reliability of the information used in hiring decisions.

- **Technical Approach:** The vague flagging mechanism leverages advanced NLP models that can be scaled according to the volume of LORs being processed. By deploying models in a microservices architecture, each service can independently analyze LORs, enabling rapid scaling to accommodate fluctuations in demand. This modularity also allows for continuous improvement of the algorithms based on feedback from HR users.

6 Timeline Creation from Resumes

We made a timeline of each person and started looking for any loopholes in the timeline.

- **Implementation:** The system constructs a detailed timeline of each candidate's educational and professional experiences. It calculates metrics related to the density of jobs and educational events over time, providing a clear view of the candidate's career progression.
- **Requirement:** A well-defined timeline allows HR professionals to assess candidates' engagement and identify any gaps in their career. This aids in understanding the individual's professional trajectory, making it easier to spot red flags or confirm consistency.
- **Technical Approach:** The timeline generation utilizes efficient data structures that can quickly aggregate and analyze multiple resumes' timelines simultaneously. With the implementation of caching strategies, frequently accessed timeline data can be retrieved rapidly, enhancing user experience. Moreover, the timeline system is designed for easy integration with visualization tools, enabling HR to present data in an engaging manner.

7 Prompt Engineering

Prompt engineering played a crucial role in improving the effectiveness of AI models, especially when it comes to extracting relevant data from unstructured inputs like resumes. The key to optimizing AI-driven resume analysis lies in designing effective prompts that guide and restrict the model to produce the most accurate and contextually appropriate outputs.

- **Implementation:** To improve the accuracy of information extraction, we craft prompts that clearly define the type of data required from the AI model, whether it's extracting skills, experience, or recommendation insights from LORs. By leveraging natural language understanding capabilities, Large Language Models (LLMs) can interpret these prompts to return concise and relevant information.

- **Importance:** Prompt engineering ensures that the AI model is contextually guided, improving precision in tasks like text extraction, classification, and summarization, which is key when dealing with complex resume documents.

8 Cross-Verification with LORs

Not only did we focus on the vagueness present in LORs, but we also examined if the LORs are serving their purpose by directly comparing them with the resume.

- **Implementation:** The solution cross-verifies candidates' experiences with their LORs by calculating cosine similarity between CV entries and LOR content. Trust scores are generated based on how closely LOR endorsements align with claimed experiences. Embeddings of mixbread ai were used again for this purpose.
- **Requirement:** Cross-verifying experiences enhances the reliability of candidates' claims, providing an additional layer of validation. Trust scores help HR teams make informed decisions by identifying candidates whose qualifications are substantiated by external endorsements.
- **Technical Approach:** The cross-verification process is built on a robust data pipeline that can handle multiple iterations of similarity calculations across various LORs efficiently. By leveraging GPU-accelerated computing for the embedding and similarity calculations, the system can scale to process a high volume of LORs and resumes simultaneously without sacrificing performance.

9 Trust Value

We now formulate the untrustworthy elements present in the whole dataset and assign a number for better comparison.

- **Implementation:** The system computes an untrustworthiness score based on various factors extracted from the resume and recommendation letters (LORs), such as the risk factor, vacancy factor, and vagueness of the language in the LORs.

  The vagueness of an LOR is calculated by iterating over its content, checking for low-confidence phrases and assigning a vagueness score. The score is normalized over the length of the text.

$$\text{vague\_value} = \frac{\sum \text{vagueness scores}}{\text{total\_length} + 1}$$

The untrustworthy factor is computed as a weighted sum of the following elements:

- vague_value: The normalized vagueness score.
- RiskFactor: A value taken directly from the resume CSV file, representing the candidate's risk.
- VacancyFactor: Another value extracted from the dataset.
- Flag: A binary indicator for any explicit warning flags associated with the candidate.

The formula for the untrustworthy factor is:

$$\begin{aligned}\text{Untrustworthy\_factor} = 0.1 \times \text{vague\_value} + 0.2 \times \tanh \times \left(\frac{\text{RiskFactor}}{5}\right) \\ + 0.1 \times \tanh \times (\text{VacancyFactor} - 1) + 0.3 \times \text{Flag}\end{aligned}$$

- **Requirement:** This untrustworthiness score helps HR departments assess the overall risk associated with a candidate. The formula considers multiple elements to reflect how vague, risky, or inconsistent the candidate's claims are.

- **Technical Approach:** The cosine similarity between the content of the resume and the LORs is used to assess the alignment between the candidate's self-reported experiences and the recommendations. Cosine similarity is computed as follows:

$$\cos\_\text{sim}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where A and B are the embeddings generated by a pre-trained transformer model from the candidate's work experience and the LORs, respectively. The maximum similarity score for each comparison is recorded, and a trust value is derived.

- **Final Adjustment:** The trust value is incorporated into the untrustworthy factor using the following adjustment:

$$\text{Untrustworthy\_factor} += 0.4 \times (1 - \text{trust\_value})$$

This adjustment reduces the overall untrustworthy factor if there is a strong alignment between the candidate's resume and their LORs, reflecting a higher trustworthiness.

10 Additional Approach of Graphs

The graph-based method is highly scalable as it can efficiently manage large datasets representing candidate-recommender relationships. By employing graph algorithms to identify cycles and analyze connection strengths, HR professionals can visualize and quantify the trustworthiness of recommendations. The use of Python's NetworkX library allows for advanced network analysis techniques, making it easier to derive actionable insights from the graph structure.

11 Conclusion

In today's fast-paced recruitment landscape, leveraging AI technology can significantly enhance the hiring process. Our AI-powered solution not only streamlines resume management and LOR evaluations but also provides meaningful insights into candidates' qualifications through data-driven analysis. By integrating various innovative techniques—from PDF management to graph-based network analysis—we aim to foster a more transparent and reliable hiring process. This holistic approach empowers HR teams to make informed decisions, ultimately leading to better hires and a more engaged workforce. As we continue to refine our methods, we look forward to shaping the future of talent acquisition together.