

Title: Job Recommendation System Using Hybrid Embedding Models

Proposed Methodology

System Architecture

In this project, I designed a hybrid Job Recommendation System that processes multilingual a dataset from Kaggle and resumes (Arabic and English) and matches them with relevant job postings. The system is organized into four main components:

1. **Data**
2. **Preprocessing and Translation**
3. **Feature Extraction**
4. **Similarity Computation**

The architecture enables the system to handle both Arabic and English resumes while reducing semantic ambiguity in the process of matching.

Data Extraction and Preprocessing

Text Extraction:

I used the *pdfplumber* library to extract text from PDF resumes.

Resume 2.csv dataset, as the data was presented in structured CSV format. The dataset was loaded using the Pandas library: `pd.read_csv`.

Language Normalization:

I evaluated two translation tools: (Google Translate , mtranslate)

The system detects the language of the resume. If it is Arabic, the text is automatically translated to English to keep all documents in the same semantic vector space. For the resume 2 .Applied `.lower()` to standardize text case (lowercasing).

Text Cleaning:

After translation, I have used typical NLP preprocessing, lowercasing, tokenization, and stop-word removal using NLTK. These steps actually help in reducing noise and improving the quality of feature extraction. Also utilized Regex to remove URLs, special characters, and extra whitespaces.

To convert text into numerical vectors, I tried a four different embedding approaches:

1.SimpleWord2Vec:

It creates document vectors by averaging word embeddings, but it does not capture the context fully.

2.TF-IDFWeightedWord2Vec:

Improvement over the previous one by weighting each word with its TF-IDF score. This highlights important skills, such as “Python”, while reducing the impact of generic terms such as “Team”, “Management”.

3.Doc2Vec:

Represents the entire document as a single vector, while word order is captured. However, this typically requires a large dataset to achieve stable performance, something which was a limitation in this project.

4.SBERT(Sentence-BERT):

I used msmarco-distilbert-base-v4, which has been trained for semantic search. SBERT provides deep semantic embeddings, which can be directly compared by cosine similarity. It captures the meaning contextually much better than traditional statistical embeddings.

Similarity Computation

The similarity between a resume vector and a job description vector is calculated using Cosine Similarity:

$$Similarity(V_r, V_j) = \frac{V_r \cdot V_j}{\|V_r\| \|V_j\|}$$

It will match the skills and experience align with a job posting.

Experimental Results and Discussion

Quantitative Analysis

Word2Vec(Simple&Weighted):

Both versions resulted in very high similarity scores, 0.95 to 0.99, which suggested a limited differentiation between job roles. The models over-relied on common corporate vocabulary that resulted in unrelated roles like "Nurse" and "Banker" sounding overly similar.

Doc2Vec:

While it produced more variation in scores, the top recommendations were inconsistent. The small size of the dataset (around 2,000 documents) likely prevented Doc2Vec from learning strong document-level representations..

SBERT

SBERT produced the most realistic score distribution (0.20–0.75) and captured genuine semantic differences between resumes and job descriptions.

For example, even if both roles mention “Machine Learning,” SBERT correctly assigns low similarity when comparing a “Deployment Engineer” resume with a “Model Training” position because the responsibilities are fundamentally different.

Conclusion

The experiments clearly show that contextual embeddings outperform statistical models by a wide margin. Although statistical models are faster, they fail to capture the deeper semantic relationships needed for accurate job matching. In contrast, SBERT provides context-aware, meaningful recommendations and behaves much closer to human judgment, making it the most suitable solution for building a reliable job recommendation system.