Job Retriever Model: Matching Resumes with Job Openings

Group 15

Anay Chauhan, Dhruv Sood, Divyansh Mishra, Hardik Singh, Pranjal Bharti, Priyash Shah

Motivation

Nearly 60% of job seekers face difficulty finding relevant job openings, varying by study and methodology.

Aligning individuals with roles matching their skills and preferences leads to increased job satisfaction, boosting productivity and innovation.

Facilitating suitable job opportunities initiates a positive cycle of growth, where employment contributes to economic activity, stimulating further job creation and economic expansion.

Connecting individuals with opportunities that align with their skillsets, we empower them to achieve better work-life balance, enhanced pay, and increased job satisfaction.

Overqualified Vs Underqualified!

Table 4.1 Rates of match, overqualification and underqualification (percentages)

	1 Matched	2 Overqualified	3 Underqualified
Armenia	66.2	28.0	5.8
Bolivia, Pluri. State of	40.1	34.6	25.2
Georgia	66.4	29.4	4.0
Ghana	47.7	39.5	12.8
Kenya	34.5	24.9	40.4
Lao PDR	45.1	41.1	13.7
North Macedonia	72.6	22.3	5.1
Sri Lanka	43.5	46.1	10.4
Ukraine	72.1	24.0	3.8
Viet Nam	26.0	70.0	4.0
Yunnan-China	56.6	32.6	10.7
Mean	51.9	35.7	12.4

Note: Means in bottom row are unweighted.

Source: STEP survey.

Country	Overqualified	Underqualified	Matched	Income classification
Albania	18	17	65	Upper middle-income
Argentina	35	22	43	Upper middle-income
Bangladesh	31	11	58	Lower middle-income
Cambodia	22	3	75	Lower middle-income
Ecuador	21	17	62	Upper middle-income
The Gambia	30	7	63	Low-income
Guatemala	33	23	43	Lower middle-income
India	18	29	53	Lower middle-income
Liberia	27	17	57	Low-income
Mongolia	11	20	69	Lower middle-income
Namibia	20	16	63	Upper middle-income
Pakistan	28	21	51	Lower middle-income
Peru	17	29	54	Upper middle-income
Philippines	22	27	51	Lower middle-income
Samoa	11	14	75	Upper middle-income
Serbia	22	21	56	Upper middle-income
South Africa	32	24	45	Upper middle-income
Tanzania, United Rep. of	25	7	68	Low-income
Uganda	26	7	67	Low-income
Viet Nam	25	17	59	Lower middle-income

Source: Own calculations based on LFS 2012 or nearest available year.

When it comes to searching for a job on job boards like Indeed, many people simply type in the job title they desire for their next role.

Although this approach can uncover some relevant job opportunities in your field, it only reveals a small portion of the positions that you may be qualified for.

The reason is that job titles differ across companies and industries.

For instance, should a software developer confine their job search solely to listings with the title "Software Developer"? what about job titles like CS programmer or Frontend engineer? similarly, a senior project manager might wish to consider other positions that highlight similar skills, such as product manager or program manager. It's evident that there's bias in the variation of job titles, potentially causing users to expend additional time generating alternative title variations or seeking roles emphasising transferable skills.

Why is this problem of interest to us?

This problem is of interest to us because it addresses a significant inefficiency in the job market. Traditional job search methods often lead to mismatches between job seekers and opportunities due to limitations in keyword-based searches and manual screening processes. By leveraging IR techniques and ML, we can streamline the job search process, enhance job matching accuracy, optimise resume and improve overall user experience.

Who are the people who benefit from this problem being solved?

1. Job seekers: Individuals searching for employment benefit from this problem being solved as it increases their chances of finding relevant job openings that match their skills and preferences.
2. Employers: companies benefit from improved job matching as it allows them to find candidates who are better suited for their roles, leading to reduced recruitment costs and increased productivity.
3. Recruiters: recruitment agencies and hr professionals benefit from streamlined processes and access to a larger pool of qualified candidates, improving their efficiency and effectiveness.

How does solving this problem help in solving other problems?

1. Economic growth: by reducing unemployment rates and facilitating better job matches, we contribute to economic growth and stability.
2. Skills development: matching individuals with roles that align with their skills and preferences can lead to better job satisfaction and retention, fostering continuous skill development and career advancement.
3. Diversity and inclusion: by improving job matching accuracy, we can help address biases and promote diversity and inclusion in the workforce by ensuring opportunities are accessible to a wider range of candidates.

Literature Review - I

Bhatia, Vedant, Prateek Rawat, Ajit Kumar, and Rajiv Ratn Shah. "End-to-end resume parsing and finding candidates for a job description using bert." *arXiv preprint arXiv:1910.03089* (2019)

Method

Resume parsing: Differentiate between linkedin and non-linkedin resumes using heuristic methods based on font size frequency and order of occurrence. Structure extracted text into predefined segments using heuristics and bert-based sequence classification.

Standard format conversion: Attempt to convert non-linkedin resumes into linkedin format using bert for sequence classification. Segment text based on heuristics and classify it into linkedin format sections.

Candidate ranking: Utilize bert for sequence pair classification to rank candidates based on their suitability to a job description.simulate job descriptions using a candidate's past job experiences and train bert to predict the similarity between a job description and a candidate's work experience.

Results

- 100% accuracy in differentiating between linkedin and non-linkedin resumes.
- 100% accuracy in structuring extracted text from linkedin resumes into predefined segments.
- 97% accuracy in converting non-linkedin resumes to linkedin format.
- 72.77% accuracy in predicting the similarity between job descriptions and candidates' work experiences using bert-based sequence pair classification.

Limitations

1. Format dependency: the system heavily relies on structured linkedin resumes, potentially leading to information loss and reduced accuracy when parsing resumes in varied formats.
2. Limited generalisation: focusing primarily on linkedin format resumes and simulating job descriptions from candidate work experiences may limit the system's ability to generalise across diverse resume styles and job requirements.

Literature Review - II

Resume Classification using various Machine Learning Algorithms Riya Pal, Shahrukh Shaikh, Swaraj Satpute, Sumedha Bhagwat ITM Web Conf. 44 03011 (2022) DOI: 10.1051/itmconf/20224403011

Method

Data collection and preprocessing: gather datasets from sources like kaggle, glassdoor, and indeed. clean the unstructured data by removing spaces, converting text to lowercase, and eliminating stop words. tokenize documents, and apply stemming and lemmatization to standardise vocabulary and simplify word variations.

POS tagging and tf-idf vectorization: associate grammatical information with words based on context and relationships within sentences. simultaneously calculate term frequency-inverse document frequency (tf-idf) to assess word importance. this assigns weights to words based on their frequency and rarity in the dataset, aiding in understanding the text's syntactic structure.

Applying classification algorithm: utilise classification algorithms such as naïve bayes, support vector machine (svm), and random forest to train the model. train the models with cleaned and classified data, evaluating performance metrics including accuracy, precision, recall, and f1 score.

Confusion matrix analysis: generate confusion matrices for each classification algorithm to evaluate true positive, true negative, false positive, and false negative values. random forest demonstrates excellent true positive values across various job classes.

Results

1. Naïve Bayes: accuracy - 45%, precision - 0.521, recall - 0.452, f1 score - 0.448.
   SVM: accuracy - 60%, precision - 0.598, recall - 0.597, f1 score - 0.594.
   Random forest: accuracy - 70%, precision - 0.687, recall - 0.683, f1 score - 0.678.
2. classification algorithms:
   naïve bayes: accuracy - 45%
   support vector machine (svm): accuracy - 60%
   random forest: accuracy - 70%, random forest exhibited the best performance with high true positive values across various job classes.

Limitations

1. Data bias: biases in the training data, such as underrepresentation of certain demographics or job profiles, may lead to skewed results and inaccurate classifications.
2. Algorithm selection: although Naïve Bayes, SVM, and Random Forest are frequently employed for resume classification, other algorithms or ensemble methods may yield superior performance, yet were not investigated in this study. While these conventional algorithms demonstrate satisfactory performance, they may not fully capture the semantic nuances of the text.

Understanding our Job Retriever Model

Model Overview

The job retriever model is designed to match resumes with job openings using parsing and job platform APIs. We acknowledged two approaches one using TF-IDF encoding with random forest for probabilistic results and the other utilizing BERT model to capture semantics of words.

Resume Parsing

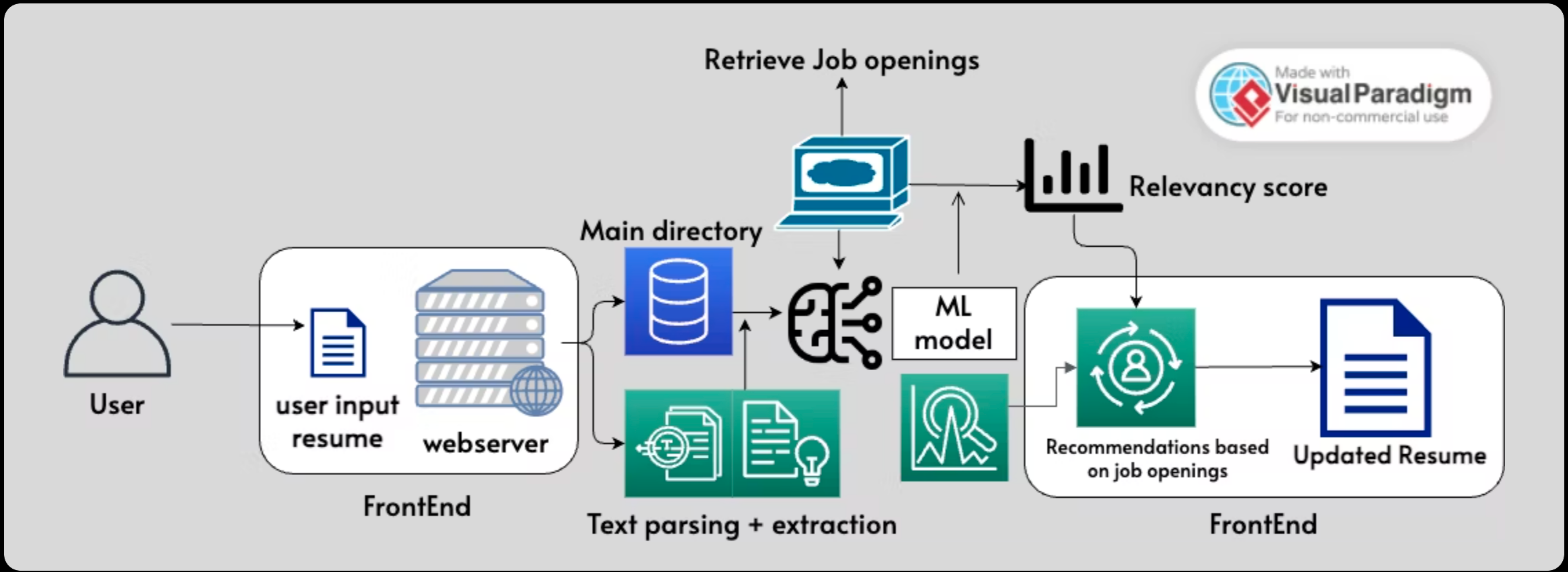
The model parses resumes to extract relevant information such as skills, experience, and education. This information is feeded in the model.

Job Platform APIs

The model interacts with job platform APIs to retrieve job openings and their corresponding requirements. It collects data on job titles, skills, and other criteria.

Matching Algorithm

The model uses a matching algorithm to compare the profiles of resumes and job openings. It calculates a similarity score based on factors such as job description, user skills, experience etc.
The model recommends changes in the resume based on different matched jobs and their relevant scores.



Expected Input → Output?

Input: User's resume in pdf format with the content inside it being copyable.

Expected output: Top 5 matched job recommendations and optimised resume tailored to the top matched jobs from multiple job platforms such as indeed.com.

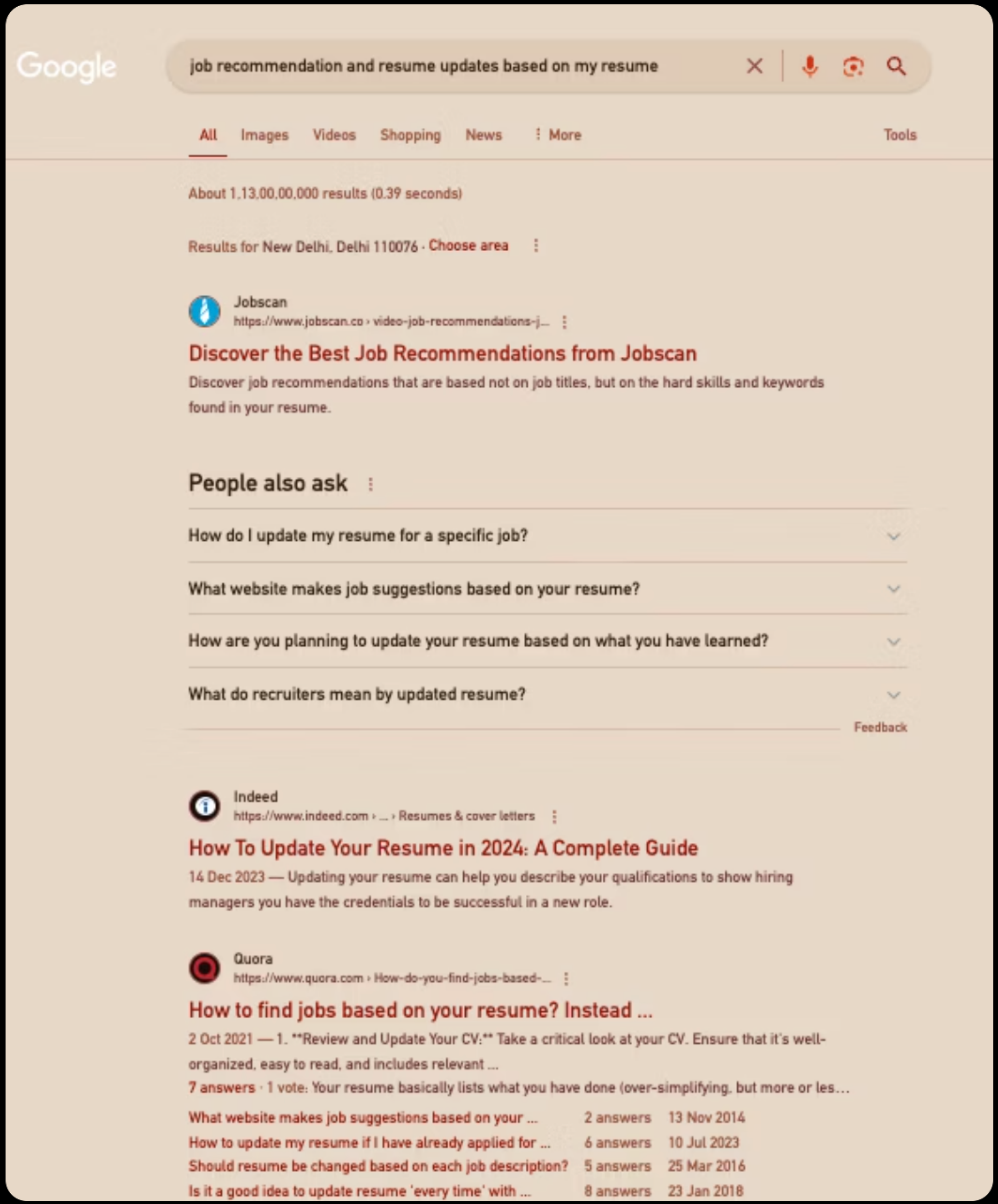
Uniqueness....?

There's a website called jobscan.co which optimises resumes based on job descriptions provided by the user.

- It provides limited flexibility and the website in itself is quite buggy.
- It doesn't provide optimisation based on Indian context and lacks the functionality to suggest job openings.
- It fails to update your resume based on the job opening.

Similarly there are a lot of softwares/websites providing optimisation services but few provide suggestions for job openings. These websites are usually overpriced and there's zero to none for the Indian job market.

Our solution is novel in a sense that it can provide job opening information by considering multiple platforms rather than a single one. It will be able to suggest people jobs on the basis of their resume's content and strength.



Tech Stack

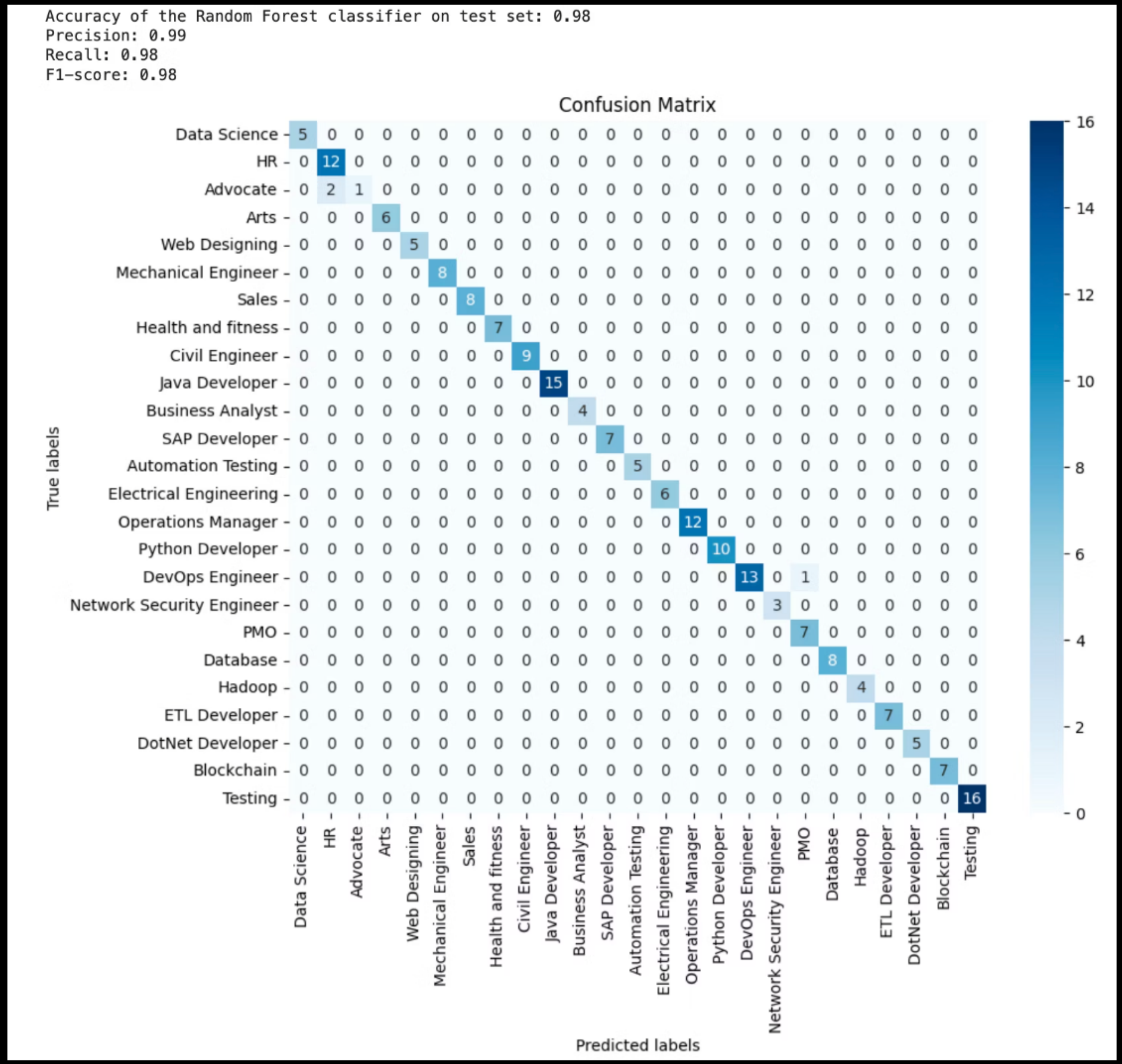
Based on our extensive research we are planning to move with interpretable models like Decision Trees and random forest to understand inherent classification mechanism.

As we move further we'll use more advance context aware models like Transformers to parse and optimise resume.

We are planning to deploy our model to a very user friendly web interface using state of the art frameworks.

Information regarding job openings will be retrieved from various online job portals.

The image attached shows the performance of our model which we are taking as a baseline.



Short term goals

1. **Optimal ATS scores:** implement algorithms to identify resumes with the highest ATS scores, enhancing model training and job retrieval accuracy.
2. **Resume and job posting matching:** implement tf-idf with random forest approach for resume-job posting matching, utilising random forest classifier on tf-idf vectors to predict resume suitability, thereby enhancing job retrieval accuracy.
3. **Experimentation:** In initial stages we want to experiment as much as we can in terms of encoding and machine learning models we are using.
4. **User-Friendly web interface:** Create simple web interface for Beta testing (Feedback).

Medium term goals

1. **Resume matching and optimisation using LLMs:** We plan to use LLMs that are specific to our needs. We plan to experiment a lot in order to get best possible model to serve our users.
2. **Expanded reach:** develop solutions tailored to diverse job markets and industries, expanding the platform's reach to cater to a broader audience effectively.

Long term goals

1. **Algorithmic advancement:** advance algorithms to unprecedented levels of accuracy, leveraging cutting-edge technologies such as AI and machine learning for optimal matching between job seekers and positions.
2. **Global impact:** establish the platform as a global leader in job matching, facilitating career opportunities for individuals worldwide and bridging the gap between job seekers and employers on a massive scale.
3. **Continuous improvement:** maintain a commitment to ongoing improvement efforts, ensuring the platform remains efficient, effective, and responsive to the evolving needs of users in an ever-changing job market landscape.

Challenges...

Resume Parsing: Extracting relevant information from resumes can be challenging due to the unstructured nature of the data. Different resumes might have different formats, and parsing them accurately requires robust algorithms capable of handling various formats, languages, and structures.

Performance Optimisation: Optimising the performance of the web application to provide a seamless user experience, especially during peak times or when processing a large number of resumes simultaneously, requires efficient resource utilisation, caching strategies, and load balancing techniques.

Feedback Mechanism: Implementing a feedback mechanism to improve the accuracy of the matching algorithm over time requires collecting and analysing user feedback. Designing an effective feedback loop that encourages user participation and incorporates feedback into the system's learning process is crucial.

Data Quality and Quantity: The effectiveness of the matching algorithm depends on the quality and quantity of the data available. Building a comprehensive database of job titles, descriptions, and requirements can be challenging. Moreover, ensuring that the data is up-to-date and relevant to various industries and domains adds another layer of complexity.

Thank You...

Q/A?