Multi-faceted Resume Parser using LLM and RAG

Team 19
Ali Rehman, Ayan Khokhar, Ibrahim Rana, M.A. Basit, Mubashir, Munam Tariq

May 6, 2024

Abstract

In an era of rapid technological advancement, talent acquisition, and recruitment remain a pivotal challenge for organizations worldwide. The influx of resumes inundates recruiters, making manual screening time-consuming and prone to human error. Conversely, job applicants often navigate uncertainties regarding aligning their resumes with job descriptions and anticipating interview questions. To address these challenges, we present a solution as a Resume Parser, leveraging Large Language Models (LLMs) and Natural Language Processing (NLP) techniques. Our system aims to streamline the hiring process by providing automated resume evaluation and tailored feedback for both recruiters and job seekers. By integrating various LLMs and sophisticated NLP models, our approach facilitates accurate resume parsing and enhances the efficiency of candidate selection. Additionally, our system offers job applicants personalized insights into resume optimization and prepares them for interviews.

Key words: LLMs(Large Language Models), RAG(Retrieval Augmented Generation, Fine-tuning, NLP(Natural Language Processing)

1 Motivation

This academic project is driven by the ambition to explore the complexities of resume parsing and address the multifaceted challenges embedded in modern recruitment processes. In today's competitive job market, organizations are flooded with many resumes for every open position. Sorting through this vast volume of documents manually is time-consuming and prone to errors and biases.

Recognizing the need for an efficient and unbiased solution, our project aims to harness natural language processing (NLP) and machine learning (ML) techniques to automate extracting and analyzing information from resumes. Moreover, it extracts and analyzes resumes for recruiters and gives valuable feedback to an applicant looking forward to a respective job role. However, our approach goes beyond mere extraction.

In addition to addressing the challenges of resume parsing, our project aims to delve into the pipeline of transformers and LLMs such as GPT(generative pre-trained transformer) and LLama. By incorporating techniques such as retrieval augmented generation (RAG) with a large language model (LLM), we aim to automate resume parsing and effectively understand and utilize advanced NLP architectures. Through the implementation of query prompting and other techniques, our goal is to develop a comprehensive solution that showcases the capabilities of these state-of-the-art models in real-world applications, thus contributing to both academic research and practical advancements in the field.

2 Related work

In recent years, numerous studies have explored applying natural language processing (NLP) and machine learning (ML) techniques to automate various aspects of the recruitment process, including resume parsing and candidate matching. These studies have contributed significantly to advancing state-of-the-art recruitment technology and have highlighted the potential benefits of leveraging automated systems in this domain.

One common approach in the literature involves the development of resume parsing algorithms that extract relevant information, such as skills, education, and work experience, from resumes in a structured format [1]. Several studies have proposed innovative methods for information extraction, ranging from rule-based systems to machine-learning models trained on large datasets of annotated resumes. While these approaches have shown promising results in automating the initial screening process for recruiters, they typically focus on parsing resumes in one direction – from the candidate to the recruiter [2].

Additionally, some research has explored using NLP techniques to match candidates to job descriptions. These studies often rely on keyword matching or semantic similarity measures to identify candidates whose skills and qualifications align with the requirements of a given job [3]. While effective in some cases, these approaches may overlook nuanced contextual information and struggle to capture the full range of a candidate's capabilities.

Notably, there is a gap in the existing literature regarding resume parsers that operate bidirectionally – extracting information from resumes and matching them to job descriptions simultaneously. Our project addresses this gap by introducing a novel resume parsing system that works both ways, facilitating a more holistic and efficient approach to candidate evaluation. By leveraging techniques such as retrieval augmented generation (RAG) with embeddings and a large language model (LLM), our parser extracts relevant information from resumes and matches and highlights sections that align with specific job descriptions. This bidirectional functionality sets our approach apart from existing methods, making it uniquely suited to meet the needs of both recruiters and job seekers in today's competitive job market.

3 System description

The following figure describes the basic methodology, which will be further elaborated on in the following subsections.

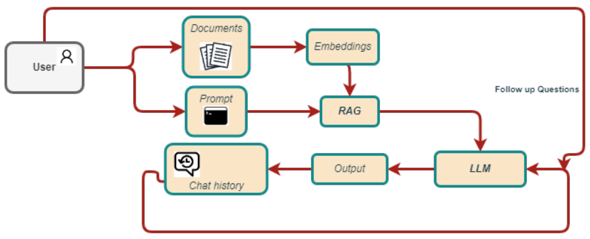


Figure 1: pipeline including all stages of the project

Below is the step-wise sequential explanation of each process involved in the system architecture.

3.1 Pre-Processing

This stage involves preparing the input data for the core processing steps. It consists of the following blocks:

- User: Depending upon the type of users, they interact with the system by uploading resumes and relevant job descriptions. The overarching arrow to the LLM with the heading of follow-up questions would be explained in the core-processing stage.

- Documents: This block represents the uploaded resumes/job descriptions containing the information that needs to be processed and analyzed. These documents can be in various formats, such as PDF, Word, or plain text.

- Embeddings: Embeddings are numerical representations of the textual content within resumes, transforming complex language into a form that machine learning models can process. These embeddings capture the semantic meaning and relationships between words, facilitating more nuanced understanding and operations by language models like LLMs and RAGs. Standard techniques for generating embeddings include Word2Vec, GloVe, and BERT. Each method varies in approach and complexity:

  - Word2Vec generates vector space representations of words by considering the context in which words appear[5].
  - GloVe (Global Vectors for Word Representation) similarly focuses on word co-occurrences over the whole corpus for embedding and balancing the statistical information[6].
  - BERT (Bidirectional Encoder Representations from Transformers), and more advanced transformer models, create embeddings that consider the full context of a word by analyzing words in both directions (left and right of the word)[7].

  These embeddings are computationally intensive as they involve high-dimensional vector calculations and require significant processing power. Their accuracy is crucial; incorrect vector representations can drastically affect the outcomes of model predictions. We evaluated various embedding techniques throughout our project's testing phase to ensure the highest accuracy. Ultimately, we utilized the instructor-large model from Hugging Face, which captures rich semantic information from the surrounding context in text data. These embeddings are critical not only for accurately interpreting the input data but also for maintaining the coherence and relevance of the outputs generated by our models.

Here is a simplified visual representation of what embeddings generically are, illustrating how words are translated into numerical vectors that retain their semantic relationships.

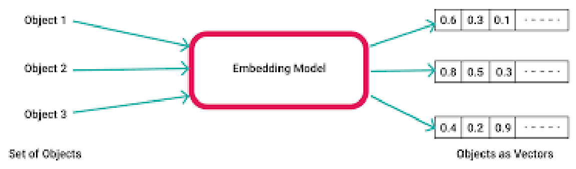
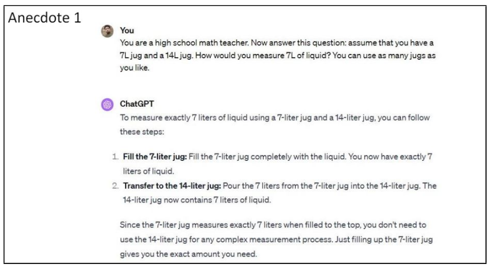


Figure 2: A general overview of what embeddings are

3.2 Core Processing

This stage involves the system's core functionalities, using preprocessed data to generate the desired outputs. It consists of the following blocks:

- The prompt is a crucial element that guides the LLM and RAG model towards the desired output. It can be tailored to specific tasks, such as extracting skills, experience, or qualifications from resumes or generating personalized suggestions for improvement. The prompt should be clear, concise, and relevant to the user's goals. In this context, prompt engineering plays a significant role in fine-tuning the interaction with the model without altering its underlying code.
  Moreover, by thoughtfully crafting the prompt, users can effectively steer the model's responses, enhancing the precision and applicability of the information generated. This technique is vital for leveraging the full potential of language models in specialized applications. Here is a simple example of how an inefficient and efficient prompt leads to two different contrasting responses, hinting towards the importance and significance of the art of prompt engineering.



Anecdote 1

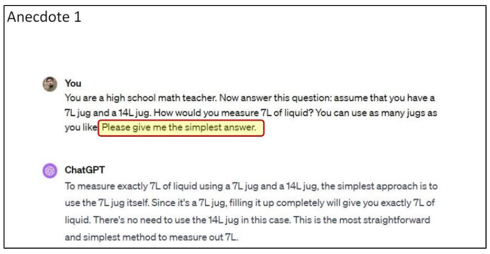
You
You are a high school math teacher. Now answer this question: assume that you have a 7L jug and a 14L jug. How would you measure 7L of liquid? You can use as many jugs as you like.

ChatGPT
To measure exactly 7 liters of liquid using a 7-liter jug and a 14-liter jug, you can follow these steps:

1. **Fill the 7-liter jug:** Fill the 7-liter jug completely with the liquid. You now have exactly 7 liters of liquid.
2. **Transfer to the 14-liter jug:** Pour the 7 liters from the 7-liter jug into the 14-liter jug. The 14-liter jug now contains 7 liters of liquid.

Since the 7-liter jug measures exactly 7 liters when filled to the top, you don't need to use the 14-liter jug for any complex measurement process. Just filling up the 7-liter jug gives you the exact amount you need.

Figure 3: Inefficient response



Anecdote 1

You
You are a high school math teacher. Now answer this question: assume that you have a 7L jug and a 14L jug. How would you measure 7L of liquid? You can use as many jugs as you like Please give me the simplest answer.

ChatGPT
To measure exactly 7L of liquid using a 7L jug and a 14L jug, the simplest approach is to use the 7L jug itself. Since it's a 7L jug, filling it up completely will give you exactly 7L of liquid. There's no need to use the 14L jug in this case. This is the most straightforward and simplest method to measure out 7L.

Figure 4: Efficient response

- RAG: The Retrieval-Augmented Generation model plays a vital role in this system. It leverages the strengths of both retrieval and generation techniques. First, it retrieves relevant information from a pre-populated knowledge base, which, in our case, is the resume bank uploaded by users. Then, it utilizes and ranks the top resumes. After that, the summarised information is passed to the LLM to generate the desired output, incorporating the retrieved information and the prompt's instructions.

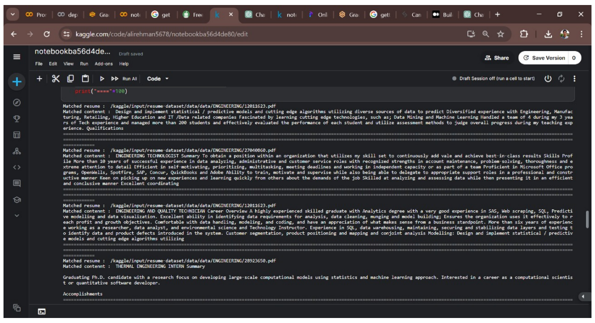
- Chat History: This block records the user's interactions with the system. This history can be used to personalize the experience and provide more relevant suggestions in subsequent interactions. For example, the system can learn the user's preferences for ranking criteria or the types of jobs they are interested in.

- The large language model serves as the core engine or, what you can say, the motherboard for processing and generating text. It leverages its vast knowledge and understanding of language to perform tasks such as further ranking the summarised resumes from the RAG according to the skillset and experiences desired by the user. It also dissects the resumes contextually to find the best match according to the query. For example, if the RAG has listed the top 5 machine learning resumes, LLM would list them further according to the diversity of projects and skills the candidates possess. It is also noteworthy to note that the ranking of RAG and LLM may differ, and the LLM gives the final ranking. Various LLM models can be used, such as GPT-3, Jurassic-1 Jumbo, and Megatron-Turing NLG. However, in our case, we used the LLM **mistral 7b**, which is a highly efficient LLM both computationally and operationally.

3.3 Output

This is the final product of the system. Based on the user's input and the functionalities of the LLM and RAG model, the output can take various forms:

- Ranked list of resumes: The system can rank resumes based on their relevance to a specific job description, considering factors such as skills, experience, and keywords. Below is an output of the model which has ranked top five resumes from the resume bank uploaded by the user.


kaggle.com/code/alirehman5678/notebookba56d4de80/edit
notebookba56d4de...
Draft saved
File Edit View Run Add-ons Help
Share
Save Version 0
Run All
Code
Draft Session off (run a cell to start)

```
    print("===="*100)
```

```
Matched resume :  /kaggle/input/resume-dataset/data/data/ENGINEERING/12011623.pdf
Matched contennt :  Design and implement statistical / predictive models and cutting edge algorithms utilizing diverse sources of data to predict Diversified experience with Engineering, Manufac
turing, Retailing, Higher Education and IT /Data related companies Fascinated by learning cutting edge technologies, such as; Data Mining and Machine Learning Handled a team of 4 during my 3 yea
rs of Tech experience and managed more than 200 students and effectively evaluated the performance of each student and utilize assessment methods to judge overall progress during my teaching exp
erience. Qualifications
================================================================================================
================================================================================================
============
Matched resume :  /kaggle/input/resume-dataset/data/data/ENGINEERING/27040860.pdf
Matched contennt :  ENGINEERING TECHNOLOGIST Summary To obtain a position within an organization that utilizes my skill set to continuously add vale and achieve best-in-class results Skills Prof
ile More than 10 years of successful experience in data analyzing, administrative and customer service roles with recognized strengths in account maintenance, problem-solving, thoroughness and e
xtreme attention to detail Efficient in self-motivating, prioritizing, multitasking, meeting deadlines and working in independent capacity or as part of a team Proficient in Microsoft Office pro
grams, OpenWells, Spotfire, SAP, Concur, QuickBooks and Adobe Ability to train, motivate and supervise while also being able to delegate to appropriate support roles in a professional and constr
uctive manner Keen on picking up on new experiences and learning quickly from others about the demands of the job Skilled at analyzing and assessing data while then presenting it in an efficient
and conclusive manner Excellent coordinating
================================================================================================
================================================================================================
============
Matched resume :  /kaggle/input/resume-dataset/data/data/ENGINEERING/12011623.pdf
Matched contennt :  ENGINEERING AND QUALITY TECHNICIAN Career Overview A highly experienced skilled graduate with Analytics degree with a very good experience in SAS, Web scraping, SQL, Predicti
ve modelling and data visualization. Excellent ability in identifying data requirements for analysis, data cleaning, munging and model building; Ensures the organization uses it effectively to r
each profit and growth objectives. Comfortable with data handling, modeling, and coding, and have an appreciation of what makes sense from a business standpoint. More than six years of experienc
e working as a researcher, data analyst, and environmental science and Technology Instructor. Experience in SQL, data warehousing, maintaining, securing and stabilizing data layers and testing t
o identify data and product defects introduced in the system. Customer segmentation, product positioning and mapping and conjoint analysis Modelling: Design and implement statistical / predictiv
e models and cutting edge algorithms utilizing
================================================================================================
================================================================================================
============
Matched resume :  /kaggle/input/resume-dataset/data/data/ENGINEERING/28923650.pdf
Matched contennt :  THERMAL ENGINEERING INTERN Summary

Graduating Ph.D. candidate with a research focus on developing large-scale computational models using statistics and machine learning approach. Interested in a career as a computational scientis
t or quantitative software developer.

Accomplishments
================================================================================================
```

Figure 5: Top five ranked resumes

- Detailed parsing of skills and experience: The system can extract and present a structured overview of the skills and experience mentioned in a resume, making it easier for recruiters to assess the candidate's qualifications. For example, if the recruiters want a people who have specific knowledge of certain stack for a Software engineering roles, the RAG would give the top software engineering resumes to the LLM who would then list the most well versed resumes as an output.

- Improvements: Based on the identified areas for improvement, the system provides tailored suggestions on revising the content or restructuring the resume.

These suggestions are specific to the candidate's profile and the target job, ensuring the feedback is relevant and actionable. For example, if the system identifies that a candidate lacks a specific skill required for the job, it might suggest adding relevant keywords to the resume or providing examples of projects or experience that demonstrate the skill. If the system detects communication gaps, it might suggest revising the language to be clearer, more concise, or more impactful. If the structure and formatting of the resume are not optimal, the system might suggest reorganizing sections, adding headings, or using bullet points to improve readability.

4 Challenges and Model Intricacies

"In the initial stages of our project, we sought to harness the power of open-source large language models (LLMs) to fulfill our inference requirements[4]. Our quest stemmed from the necessity for a **cost-effective solution** that would **not** rely on **external APIs** since external APIs would mean no novelty, and they are given only for a limited time without a subscription. This led us to explore various options, including Falcon and other publicly available models. However, while these LLMs showcased impressive language understanding capabilities, they presented significant challenges regarding inference time and domain specificity.

Challenges with Falcon and Other LLMs: The primary hurdle we encountered with Falcon and other LLMs was the prolonged inference times, which hindered the real-time responsiveness of our system. Not only this, but it also led to the crashing of our Google Collab notebooks and the exhaustion of our GPU compute units. Additionally, these models struggled to grasp the intricacies of our domain-specific tasks, leading to suboptimal performance and less accurate outputs.

Moreover, A significant task in our journey was to benchmark the performance of our selected LLM against established benchmarks, such as GPT-3.5.

Solution: Implementation of LAMA CPP Python

In our pursuit of efficient and effective inference solutions, we turned to the innovative approach of Language Model for Language Model Augmentation (LAMA) CPP Python.

LAMA CPP Python leverages the efficiency of CPU-based computation to achieve high-speed performance. By utilizing C++ for the core implementation, LAMA CPP Python capitalizes on the language's inherent speed and optimization capabilities. Unlike GPU-based approaches that may require specialized hardware and incur additional overhead, LAMA CPP Python runs entirely on the CPU, making it accessible and efficient for many systems and environments.

C++ allows for direct control over memory management and optimization, resulting in significantly faster execution than Python-based implementations. Additionally, C++ offers low-level access to system resources and hardware, enabling developers to fine-tune performance-critical components for maximum efficiency. As a result, LAMA CPP Python achieves remarkable speed and responsiveness, making it ideal for real-time inference tasks and scenarios where performance is paramount.

5 Conclusion

Our resume parsing system, powered by LLMs and RAG models, revolutionizes the recruitment process for both recruiters and job seekers. Recruiters benefit from streamlined candidate selection and detailed skill extraction, while job seekers receive invaluable feedback for optimizing their resumes. Moreover, our solution is not just innovative—it's also incredibly cost-effective. We've eliminated the need for expensive API subscriptions by leveraging open-source LLMs and RAG models. Plus, we've made it easily accessible and completely free by deploying it on Streamlit—a user-friendly platform that enhances the experience for recruiters and job seekers alike.

6 References

1. Bhatia, Vedant, et al. ”End-to-end resume parsing and finding candidates for a job description using bert.” arXiv preprint arXiv:1910.03089 (2019)

2. Zinjad, Saurabh Bhausaheb, et al. ”ResumeFlow: An LLM-facilitated Pipeline for Personalized Resume Generation and Refinement.” arXiv preprint arXiv:2402.06221 (2024)

3. Azaria, Amos, and Tom Mitchell. ”The internal state of an llm knows when its lying.” arXiv preprint arXiv:2304.13734 (2023)

4. Yang, Jingfeng, et al. ”Harnessing the power of llms in practice: A survey on chatgpt and beyond.” ACM Transactions on Knowledge Discovery from Data 18.6 (2024): 1-32

5. Church, Kenneth Ward. ”Word2Vec.” Natural Language Engineering 23.1 (2017): 155-162.

6. Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. ”Glove: Global vectors for word representation.” Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

7. Alsentzer, Emily, et al. ”Publicly available clinical BERT embeddings.” arXiv preprint arXiv:1904.03323 (2019).