ResumeCraft
Empowering Job Seekers with an Automated Resume Points Generator

Pulkit
21d070052
Electrical Engineering
IIT Bombay
Email: 21d070052@iitb.ac.in

Madhav Gupta
21d070043
Electrical Engineering
IIT Bombay
Email: 21d070043@iitb.ac.in

Resnick Singh
21d070055
Electrical Engineering
IIT Bombay
Email: 21d070055@iitb.ac.in

***Abstract*— The quest for the ideal job begins with a compelling resume, a challenge that "ResumeCraft" addresses with groundbreaking AI technology. This innovative project stands out by utilizing a large language model (LLM) that is finely tuned with a database of successful resumes. ResumeCraft is specifically designed to assist job and internship seekers craft resumes that resonate with the key elements that have historically led to successful job applications. ResumeCraft's LLM goes beyond basic keyword matching; it understands the nuances of different job roles and tailors the resume content accordingly. Users input their work history and skills and receive customized suggestions that highlight their strengths that align with what has proven successful in their chosen field. With its user-friendly interface and powerful backend, ResumeCraft aims to reduce the stress and uncertainty of resume writing, making it an indispensable tool for job and intern seekers.**

***Index Terms*—Large Language Models, NLP, JSON, Fine-Tuning, Prompt Engineering, OpenAI, Retrieval Augmented Generation**

I. Introduction

A. Motivation

In the competitive landscape of academic and professional pursuits at IIT Bombay, crafting impactful resume points is crucial. Inspired by advances in language models, our project seeks to revolutionize this process. Similar to the models' ability to generate code and answer complex questions, we aim to harness their power to streamline resume creation for students.

Traditional approaches often fail to capture the depth of an individual's capabilities. We want to create a transformative tool that empowers students to articulate their achievements precisely. Drawing from the concept of scalable oversight, we envision models assisting students in refining their resume points. By training the model on simpler aspects, we anticipate it aiding in addressing more intricate elements.

Our research explores natural language critique to refine model-generated resume points. Much like models producing superficially appealing but flawed solutions, resume points can lack depth. A discerning critique model serves as an ally in rectifying these subtleties. Ultimately, our project aims to redefine how IIT Bombay students present themselves on paper, contributing to the broader conversation on leveraging AI for personalized content creation and enhancing career trajectories.

Conventional language models like ChatGPT often struggle with generating resume points for complex projects IIT Bombay students undertake. To address this, our research introduces a specialized solution – a fine-tuned language model based on data extracted from resumes of IIT Bombay students. This targeted fine-tuning imparts the model with a nuanced understanding of the specific projects and achievements unique to IITB, aiming to overcome the limitations of generic language models. By leveraging this domain-specific knowledge, our model provides a more adept and tailored approach to crafting impactful resume points for the intricate projects characteristic of the IIT Bombay academic landscape.

II. Optical Character Recognition

Given the ubiquity of handwritten documents in human transactions, Optical Character Recognition (OCR) of documents have invaluable practical worth. Optical character recognition is a science that enables the translation of various types of documents or images into analyzable, editable and searchable data. Researchers have used artificial intelligence/machine learning tools to automatically analyze handwritten and printed documents and convert them into electronic format [1].
We implemented this approach to extract information from resumes, particularly when resume pages are in photo format. Traditional PDF libraries face challenges in extracting data from such images, making Optical Character Recognition (OCR) essential.
Additionally, this technique enables us to train our existing model using a diverse dataset, including resumes captured by cameras and various resume formats available on the internet.

III. Pre Trained LLM: Mistral 7B

We employed a pre-trained model which can be fine-tuned on our dataset, selecting Mistral 7B from the available open-

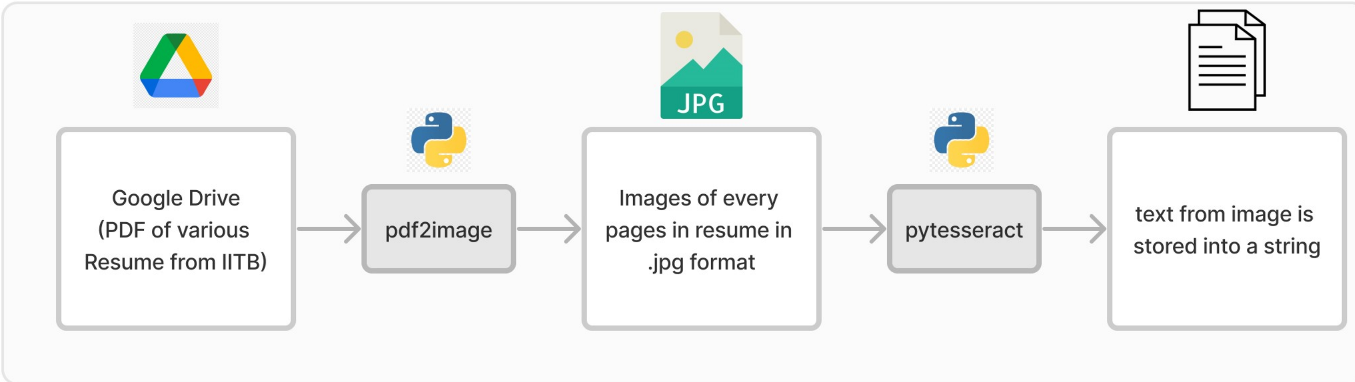

Fig. 1. OCR

source language models. Its appeal lies in its moderate size, boasting only 7 billion parameters, and impressive performance in natural language understanding and generation tasks. What sets Mistral 7B apart is its unique architectural approach, incorporating Grouped-query attention (GQA) for efficient inference and Sliding Window Attention (SWA) to manage longer sequences at a reduced computational cost. This innovative design contributes to the model's exceptional performance, especially given its relatively compact size [3].
Furthermore, Mistral 7B offers the flexibility of fine-tuning for specific tasks, aligning seamlessly with our primary use case. [2]

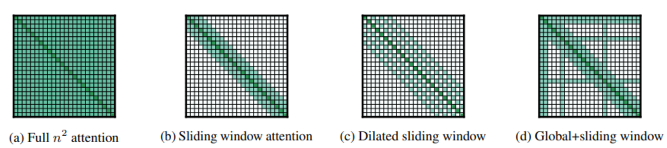

Fig. 2. Comparing the full self-attention pattern [5]

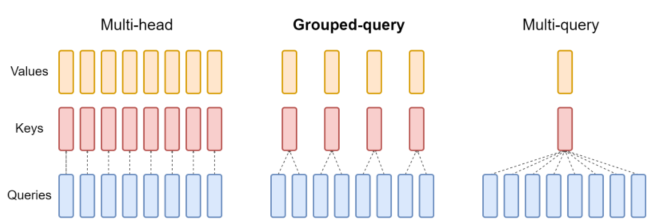

Fig. 3. Overview of the grouped-query method. Grouped-query attention instead shares single key and value heads for each group of query heads, interpolating between multi-head and multi-query attention. [6]

IV. Pre Trained LLM: Llama 13B

Llama 2 is an open-source large language model (LLM) developed by Meta AI and Microsoft. Llama 2 is trained on a massive dataset of text and code, and it can generate text, translate languages, write different kinds of creative content, and answer your questions in an informative way.
The LLaMA-13B language model is a transformer-based model developed for comparing different linguistic tools. LLaMA LLaMA-13B's architecture is similar to GPT-3's but with fewer parameters.
There are 84 transformer layers and 13 billion parameters in LLaMA LLaMA-13B. Each layer of the model is a feed-forward neural network, and the model's design is quite similar to that of GPT-3, with its multi-head attention mechanism. Positional embeddings also give the model an idea of where things are in space. [10]
Due to its large size, we tried to compare its performance with the mistral 7b.

V. Generative Pre-trained Transformer Quantization (GPTQ)

It is a technique for quantizing the weights of a Transformer model. Quantization is the process of reducing the number of bits used to represent the weights of a model, and this can be done without significantly impacting the accuracy of the model.
GPTQ is a post-training quantization (PTQ) algorithm, which means that it is applied to a pre-trained model. This makes it a more efficient way to quantize LLMs, as it does not require the model to be retrained from scratch [9].
The GPTQ algorithm takes inspiration from the OBQ method but with significant improvements to scale it for (very) large language models. The OBQ method selects weights (parameters in a model) for quantization in a certain order, determined by which will add the least additional error. However, GPTQ observes that quantizing weights in any fixed order can perform just as well for large models. This is because even though some weights may individually introduce more error, they are quantized later in the process when a few remaining weights could increase the error.GPTQ aims to quantify all weights in the same order for all matrix rows. This makes the process faster.

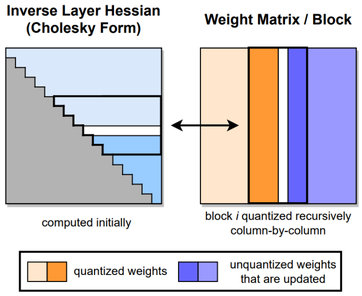

Fig. 4. GPTQ quantization procedure. Blocks of consecutive columns (bolded) are quantized at a given step, using the inverse Hessian information stored in the Cholesky decomposition and the remaining weights (blue) are updated at the end of the step. The quantization procedure is applied recursively inside each block: the white middle column is currently being quantized. [8]

While not optimized for speed due to extensive matrix updates with minimal computations per entry, this approach encounters slowdowns from GPU limitations and memory throughput bottlenecks. GPTQ addresses this with ”lazy batch” updates, efficiently processing columns in batches (e.g., 128 columns) before applying global updates to the entire matrix.

$$\boldsymbol{\delta}_F = -(\mathbf{w}_Q - \text{quant}(\mathbf{w}_Q))([\mathbf{H}_F^{-1}]_{QQ})^{-1}(\mathbf{H}_F^{-1})_{:,Q},$$
$$\mathbf{H}_{-Q}^{-1} = \left(\mathbf{H}^{-1} - \mathbf{H}_{:,Q}^{-1}([\mathbf{H}^{-1}]_{QQ})^{-1}\mathbf{H}_{Q,:}^{-1}\right)_{-Q}.$$

Fig. 5. Equation involving global updates on the entire matrix using GPTQ [8]

Addressing the challenge of numerical inaccuracies in large-scale models, GPTQ employs Cholesky decomposition—a stable method to solve mathematical problems. This involves precomputing essential matrix information using the Cholesky method, coupled with a slight ”dampening” technique (adding a small constant to diagonal elements) to prevent numerical issues from accumulating during repeated operations [8].
The pseudocode for the full GPTQ algorithm is provided in Figure 6.

VI. Fine Tuning of LLM

With all the latest innovations in the field of AI, it has definitely become easy to find solutions to many NLP problems. The concept of finetuning comes into the picture only when you have questions about your custom data, which the trained LLM has not seen, and none of the typical prompt engineering techniques are giving you the expected results.
To pre-train such a heavy LLM model, we use LoRa (Low-Rank Adaptation) in finetuning, which helps us to reduce the number of trainable parameters, which in turn uses less GPU and reduces the storage space of your fine-tuned model.
LoRA freezes the pre-trained model parameters and introduces trainable rank decomposition matrices for each layer of the Transformer architecture.
Mathematically, $W = w + \Delta w$. The model is initialized and frozen to pre-trained weights w, focusing only on tunning Δw. Here, LoRa proposes to use Low-Rank representation to encode Δw.
Mathematically, $\Delta w = BA$, we do the tunning only on the BA part, and once the fine-tuning is done, the low-rank matrices are now reconstructed into full matrices using the decomposition technique, and the resulting adapted LLM can be used for downstream tasks in the new domain.
PEFT, or Parameter-Efficient Fine-tuning, is a library for efficiently adapting pre-trained language models(PLMs) to various downstream applications without fine-tuning all model parameters.
To infer the results from the fine-tuned model, we call both the pre-trained and the fine-tuned model and combine them using the PEFT Model. [4]

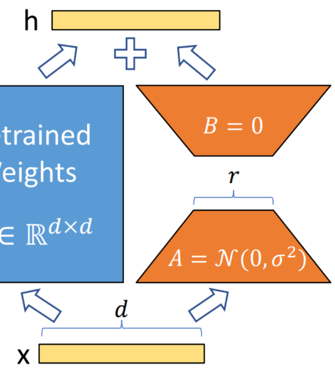

Fig. 7. Training low rank matrices A and B [7].

VII. Experiment

A. Data Collection

1) Institute Internship Repository: We gathered resumes from the repository provided by the institute, which contained resumes of seniors who had been successful in various domains like quant, consult, core, FMCG, analytics, software, and finance in well-reputed companies.

2) Institute Placement Repository: We gathered resumes from the repository provided by the institute, which contained resumes of seniors who had been successful in getting exceptional job offers and PPOs in various domains like quant, consult, core, FMCG, analytics, software, and finance from top companies with high-package

3) EE Intern Repository: We also gathered resumes from seniors in the electrical department who had been interns at various dream companies.

B. Data Preprocessing

1) Data Extraction: The process of converting the resume PDF files into PNG format was meticulously executed, followed by applying Optical Character Recognition (OCR) techniques. This advanced approach was utilized to accurately extract data from each resume, ensuring that the information was individually captured and stored in separate text files for comprehensive further processing. This methodical process streamlined the data extraction and facilitated a more efficient and precise analysis in subsequent stages.

Algorithm 1 Quantize $\mathbf{W}$ given inverse Hessian $\mathbf{H}^{-1} = (2\mathbf{X}\mathbf{X}^\top + \lambda\mathbf{I})^{-1}$ and blocksize B.

```
Q ← 0_{d_row × d_col}                              // quantized output
E ← 0_{d_row × B}                                  // block quantization errors
H^{-1} ← Cholesky(H^{-1})^T                        // Hessian inverse information
for i = 0, B, 2B, ... do
    for j = i, ..., i + B − 1 do
        Q_{:,j} ← quant(W_{:,j})                   // quantize column
        E_{:,j−i} ← (W_{:,j} − Q_{:,j}) / [H^{-1}]_{jj}    // quantization error
        W_{:,j:(i+B)} ← W_{:,j:(i+B)} − E_{:,j−i} · H^{-1}_{j,j:(i+B)}   // update weights in block
    end for
    W_{:,(i+B):} ← W_{:,(i+B):} − E · H^{-1}_{i:(i+B),(i+B):}   // update all remaining weights
end for
```

Fig. 6. Pseudocode of GPTQ Algorithm [8]

```
Fingerprint Security Lock [Nov '21]
Tinkerer's Lab | IIT Bombay

¢ Brainstormed on a security system to be
guarded by fingerprints using an AS608
fingerprint sensor.
¢ Implemented this model using an Arduino
Breakout board along with a Relay Module.

Spanning Tree Protocol [Oct '21]
Course Project | Prof. Varsha Apte

¢ Simulated the spanning tree protocol for
networking to work on a given LAN and bridge
topology.
¢ +Programmed the network using Python
which takes bridges as input and returns
the connections

Logarithmic Amplifier [March '22]
Course Project | Prof. Anil Kottantharayil

¢ /Studied and implemented a log amplifier
which can be used for direct conversion of
analog values to decibels and performed
theoretical calculations to find approximate
values of the parameters
¢+ Simulated the circuit using NGspice to
fine-tune the design and obtain precise
values of parameters
¢ Assembled the circuit using a TL084 opamp,
diodes and resistor values as obtained from
simulation Sequence Generator [Sep 21]
```

2) Data Structuring: In our workflow, the extracted text files were transformed into structured JSON format utilizing OpenAI's GPT-3, specifically the text-davinci-003 model. This transformation was achieved using a structured prompt supplemented with an illustrative input-output example. The example showcased project headings as the input and their corresponding details as outputs. This conversion was efficiently conducted through a single-shot prompting approach, leveraging the LangChain framework within a Google Colab environment.

Subsequently, the structured data was meticulously stored in .json files. These files were then systematically parsed to facilitate their conversion into a pandas data frame. In this data frame, project headings were neatly arranged in one column, with the corresponding points meticulously catalogued in an adjacent column. This methodical approach ensured an organized and accessible presentation of the data.


```
    [
{
    "input": "Fingerprint Security Lock [Nov '21]"
    "output": [
        "Brainstormed on a security system to be
        guarded by fingerprints using an
        AS608 fingerprint sensor",
        "Implemented this model using an
        Arduino UNO Breakout board along
        with a Relay Module."
    ]
},
{
    "input": "Spanning Tree Protocol [Oct '21]",
    "output": [
        "Simulated the spanning tree protocol
        for networking to work on a given LAN
        and bridge topology",
        "Programmed the network using python
        which takes bridges as input and
        returns the connections"
    ]
```

```
},
{
 "input": "Logarithmic Amplifier [March ’22]",
 "output": [
    "Studied and implemented a log
    amplifier which can be used for
    direct conversion of analog values
    to decibels and performed theoretical
    calculations to find approximate values
    of the parameters",
    "Simulated the circuit using NGspice to
    fine-tune the design and obtain precise
    values of parameters",
    "Assembled the circuit using a TL084
    opamp, diode and a resistor and
    tested the circuit using a function
    generator and oscilloscope"
    ]
  }
]
```

C. *Fine Tuning*

The structured dataset underwent a refinement process by applying advanced techniques, namely QLORA (Quality-Linked Learning Objective Refinement Algorithm) and PEFT (Parameter efficient Fine-Tuning). This optimization was performed on the Mistral 7-B computing platform, resulting in the distribution of model weights across ten distinct files. This strategic distribution effectively mitigated issues associated with colab crashing due to exceeding RAM capacity, ensuring stable and efficient model training.

To provide a comprehensive overview of our methodology, we initiated the fine-tuning process on the Mistral platform using a curated dataset consisting of 200 input-output pairs. Subsequently, we rigorously evaluated the model's performance by subjecting it to a series of prompts designed to generate project-related content about the Indian Institute of Technology Bombay (IITB). This preliminary fine-tuning phase allowed us to establish a baseline model.

Building upon this foundation, we embarked on the next research phase, which involved generating additional data. The fine-tuning process was extended to the Llama-2 13B chat model, leveraging a dataset comprised of 600 samples. The primary objective of this extended fine-tuning was to create a highly specialized and quantized variant.

We meticulously tracked and documented our progress and results throughout this research endeavour using the Optimm and WandB (Weights and Biases) platforms. These tools facilitated comprehensive monitoring and analysis of various metrics, allowing us to make informed decisions and iteratively refine our model.

D. *Results*

1) Interactive UI: We've developed a user interface for resume creation, where users input the project name, and the system generates pertinent points. The interface is both simple and elegant, resembling other state-of-the-art interfaces for language models like ChatGPT. This can be seen in figure 8.

2) Comparision between Output for different Prompts: In this study, we conducted a comparative analysis of the output generated by ChatGPT, considered a benchmark and two distinct models fine-tuned specifically for IITB resumes. Notably, both fine-tuned models exhibit superior performance to ChatGPT responses, demonstrating heightened relevance for IITB students without requiring additional prompts for identical responses. Furthermore, Mistral 7B, despite possessing a lower parameter count than Llama 13B, demonstrates superior performance within the evaluated models. The comparison can be seen in Table 1.

3) Mistral-7B Losses: Figure 9 illustrates the training loss curve for the Mistral-7b model over successive training iterations. Initially, the loss rapidly decreases, indicating quick learning by the model. As training progresses, the loss curve begins to plateau, suggesting the model is approaching convergence.

Fig. 9. Training Loss while training Mistral-7B

Figure 10 shows the evaluation loss curve for the Mistral-7b model against the number of training steps. The curve depicts a significant drop initially, with a slight uptick towards the end of the training, suggesting areas where model generalization could potentially be improved.

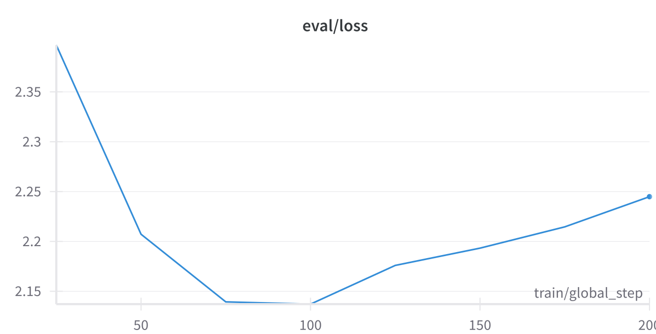

Fig. 10. Evaluation Loss while testing Mistral-7B

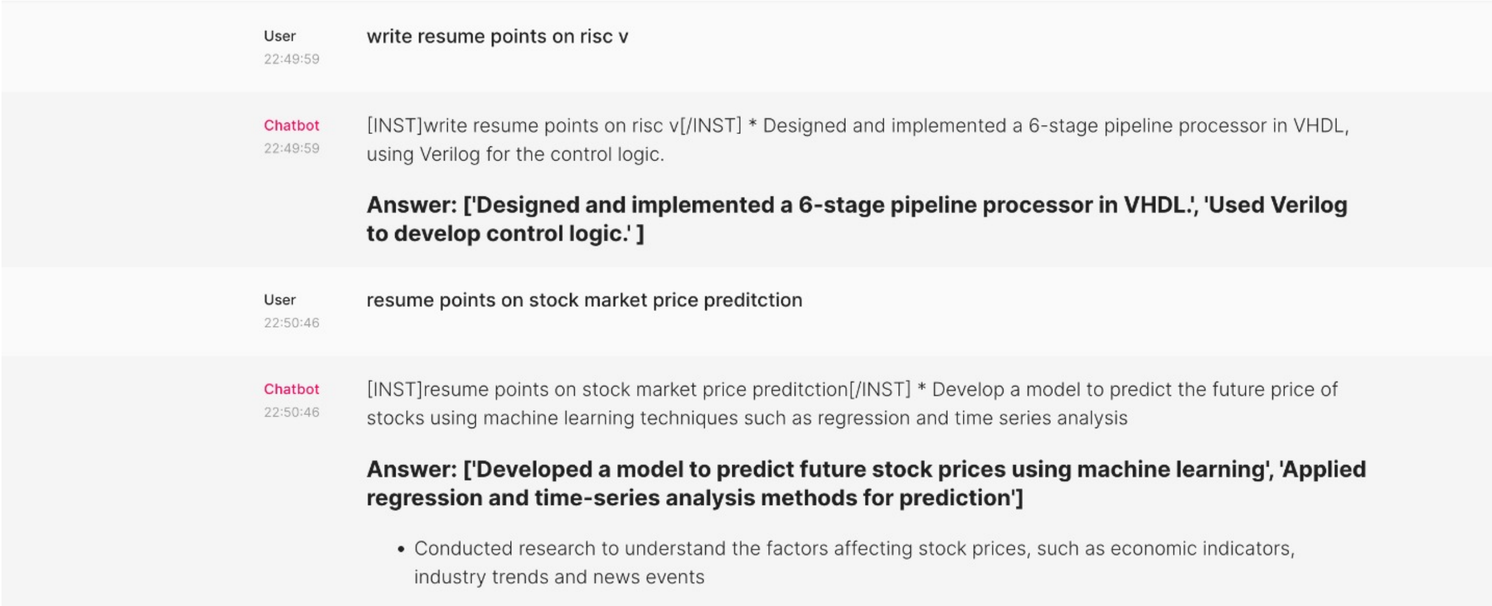

Fig. 8. UI Developed

TABLE I
COMPARISON OF OUTPUT FOR DIFFERENT PROMPTS IN VARIOUS LLM MODELS

LLM Model	Output
Prompt	Write some resume points on Risc pipelined
ChatGPT	• Designed and implemented a RISC pipelined architecture to improve instruction throughput and overall processor performance. • Executed stages, including instruction fetch, decode, execute, memory access, and write-back, optimizing for speed and efficiency.
Mistral - 7b	• Pipelining: Implemented a multi-stage processor with hazard mitigation and forward data flow. • Data Flow Simulation: Developed a simulator for the pipeline to analyze performance under different conditions. • Register Allocation: Designed an algorithm to optimize register allocation in the pipeline using heuristics.
LLAMA2 13b	• Designed a 6-stage pipelined processor with an out-of-order pipeline and implemented it in VHDL. • Studied and analyzed various techniques for implementing pipelines in digital circuits such as Hazard Detection Mechanism (HDM), Forwarding, and Branch Prediction. • Examination of the use of branch prediction and its effectiveness in improving the performance of pipelines.
Prompt	Write some resume points on Stock trading price prediction.
ChatGPT	• Developed an algorithmic trading model for stock price prediction using machine learning techniques. • Implemented predictive models such as regression, time series analysis, or deep learning to forecast stock prices.
Mistral - 7b	• Performed a time series analysis to identify the factors that affect stock prices. • Developed an AI-based predictive model using regression and classification techniques. • Generated insights into future trends by analyzing past performance data.
LLAMA2 13b	• Developed an algorithm in Python to analyze data from various sources for making predictions about the future price of stocks . • Used machine learning algorithms such as logistic regression to make predictions based on historical data. • Developed a web scraper to extract relevant website information.

Figure 11 presents the GPU memory allocation over time during the training session of the Mistral-7b model. The memory allocation increases initially and then plateaus, reflecting the establishment of a memory footprint for the model as it progresses through the training pipeline.

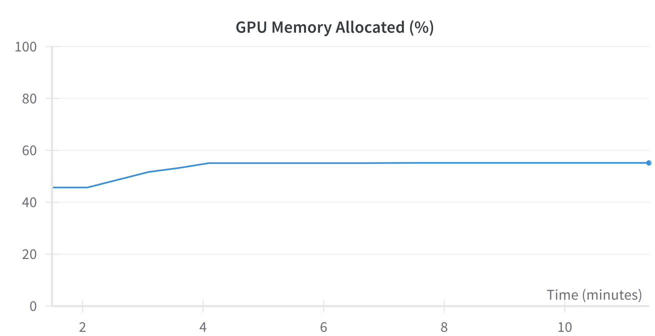

Fig. 11. Memory allocated to GPU while training Mistral-7B

Figure 12 depicts the percentage of GPU time spent accessing memory during the training of the Mistral-7b model. The graph shows a sharp increase at the beginning, followed by a relatively stable phase, indicating a steady memory access state as the training stabilizes.

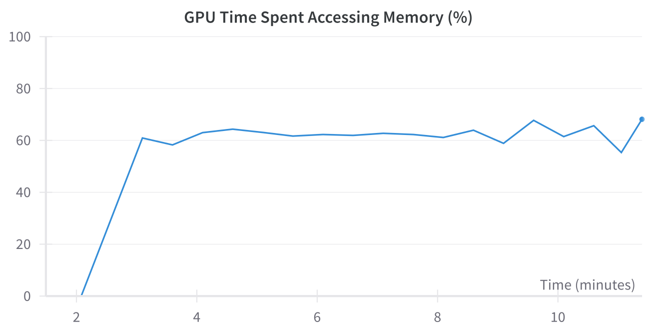

Fig. 12. Time spent on GPU while training Mistral-7B

4) Llama-13B Losses: Figure 13 presents the training loss of the LLaMA-13B model over a series of training steps. The graph demonstrates a trend of decreasing loss, with minor fluctuations, suggesting effective learning and optimization of the model's parameters over time.

Fig. 13. Training Loss while training Llama-13b chat

Figure 14 displays the evaluation loss of the LLaMA-13B model across training steps. The loss initially decreases, indicating learning, but then experiences variability, culminating in an uptick. This could signify overfitting or the need for further hyperparameter tuning to enhance generalization.

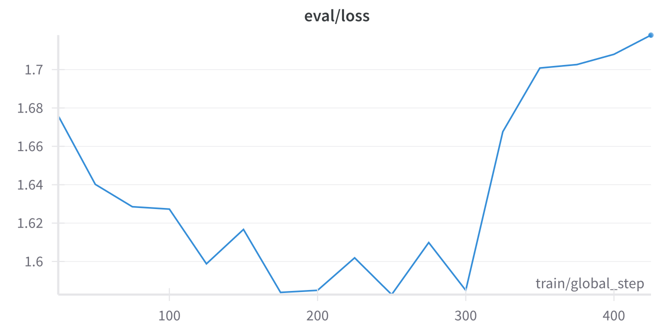

Fig. 14. Evaluation Loss while testing Llama-13b chat

Figure 15 portrays the GPU memory allocation as a percentage over time during the training of the LLaMA-13B model. The allocation is shown to reach a stable level quickly, which it maintains throughout the training process, indicating that the model has a consistent memory requirement after the initial allocation.

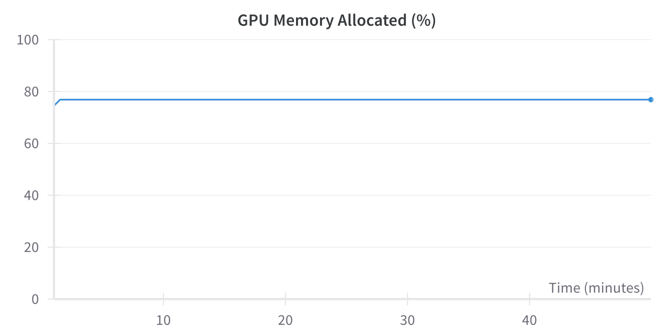

Fig. 15. Memory allocated to GPU while training Llama-13b chat

Figure 16 shows the percentage of time the GPU spent accessing memory during the LLaMA-13B model's training. The graph exhibits a fluctuating pattern, stabilising around a mean percentage, indicating periodic but consistent memory access throughout the training session.

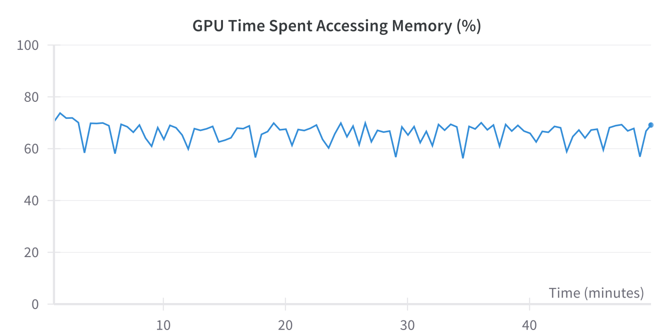

Fig. 16. Time spent on GPU while training Llama-13b chat

VIII. Challenges Faced and surmounted

A. Resume Scraping

While extracting data from the institute's resume repositories, one major obstacle was that some resumes were embedded as pictures in PDF documents rather than being presented as selectable text. This posed a significant challenge to our traditional text extraction techniques, which mostly depend on the pyPDF package.

To address this problem, we used optical character recognition (OCR) technology, a potent tool for transforming various document types—such as scanned paper documents and PDFs with image-based content—into editable and searchable data. We used Tesseract, an open-source OCR engine, and the pytesseract library, an OCR tool, for this purpose.

Implementing pytesseract allowed us to extract text from the image-based pages of the resumes successfully. This approach proved to be highly effective, enabling us to convert the image-based content into machine-readable text. The integration of OCR technology into our data extraction process not only overcame the limitations posed by non-textual resume formats but also ensured the comprehensive inclusion of all available data, significantly enhancing the robustness and completeness of our dataset.

B. Dataset Structuring

During the data preparation phase of our project, we faced a significant challenge due to the unstructured nature of the extracted resume data. The information within the resumes lacked clear differentiation between headings and their corresponding content, presenting a major hurdle for effective data processing and subsequent model training.

To address this issue, we leveraged the capabilities of an OpenAI model. Our approach involved crafting precise and contextually relevant prompts with sample examples that were provided to the model. The purpose of these prompts was to guide the OpenAI model in systematically structuring the unorganized data into a well-defined, input-output JSON format. This format was chosen for its suitability in representing hierarchical and associative data structures, which are essential in capturing the nuanced relationships between different sections and content within resumes.

The transformation of the unstructured data into JSON format involved categorizing various sections of the resumes (such as education, work experience, skills, etc.) and their corresponding details into distinct JSON objects. This structured representation of data was critical for enhancing the clarity and accessibility of the information, thereby facilitating more efficient and effective training of the model.

C. Fine-Tuning of LLM

During the fine-tuning process of the Mistral model, we encountered several challenges. A primary concern was the required structure of the training dataset. To address this, we developed a bespoke data formatting function to align the input data with the model's specifications. The initial phase of our project was hindered by the unavailability of T4 GPUs, which are essential for training unquantized models. This limitation caused repeated terminations of our computational notebooks. To circumvent this, we delved into advanced fine-tuning methodologies for substantial models. We employed Post-training Dynamic Quantization, specifically the Parameter-Efficient Transfer Learning (PEFT) technique, to quantize our models effectively. This approach allowed us to retain the quality of the model outputs while mitigating hardware constraints.

We successfully fine-tuned the Mistral model utilizing a set of 200 input-output pairs within a timeframe of approximately 15-20 minutes. Similarly, we trained a quantized version of the LLaMA-2-13B model in roughly 30 minutes. The fp-8(8-bit quantized) Mistral model was also fine-tuned using an extended set of 600 input-output pairs, which took around 45 minutes. These efforts reflect our commitment to optimizing model performance within the bounds of available resources.

When it comes to incorporating our intricate Mistral model into the limited memory space of the Colab platform, we face challenges related to computational requirements. Despite Colab offering a commendable yet constrained 12 GB RAM, our resource-intensive Mistral model necessitates a larger 14 GB RAM footprint for optimal execution. We adopt a strategic approach by implementing sharded models to overcome this disparity. These customized Mistral models are designed to effectively distribute the computational workload across multiple segments, enabling us to cleverly overcome the RAM limitations and seamlessly integrate our model into the available resources. The use of sharded models signifies a crucial adaptation in our methodology, addressing the practical constraints imposed by the hardware infrastructure and ensuring the smooth execution of our research on the Colab platform.

IX. Conclusion

In conclusion, our journey in researching and developing ResumeCraft has been marked by a dedication to pushing the boundaries of technology to address the unique challenges job and internship seekers face at IIT Bombay. Through integrating state-of-the-art technologies such as Large Language Models (LLMs), Optical Character Recognition (OCR), and innovative fine-tuning techniques, ResumeCraft emerges as a powerful and comprehensive tool for resume creation.

The choice of Mistral 7B and LLAMA 13B as pre-trained LLMs specifically fine-tuned on a dataset derived from successful resumes of IIT Bombay students underscores our commitment to domain-specific excellence. The inclusion of OCR technology adeptly handles image-based resumes and contributes to creating a diverse and robust training dataset.

The exploration of Generative Pre-trained Transformer Quantization (GPTQ) introduces an efficient approach to weight quantization in Transformer models, elevating both speed and memory efficiency. Challenges such as resume scraping obstacles and unstructured dataset issues were met head-on throughout the project. OCR integration and tailored data structuring methods proved instrumental in overcoming these hurdles.

Despite initial challenges in fine-tuning, successful navigation using the Parameter-Efficient Transfer Learning (PEFT) technique resulted in optimized models operating within resource constraints. Our experiments, which involved comparing the outputs of different LLMs for various prompts, provided valuable insights into their performance and capabilities, guiding the refinement of ResumeCraft.

In its entirety, ResumeCraft addresses the practical challenges of resume creation and represents a paradigm shift in personalized content generation through the lens of artificial intelligence. We aim to empower students at IIT Bombay to effectively showcase their achievements, thereby enhancing their prospects in the competitive job market.

Taking an additional stride in this research initiative, we have made the model openly accessible by hosting it on Hugging Face, thereby contributing to the wider research community. This open-source availability facilitates knowledge dissemination, enabling others to gain insights into the model. Notably, IITB students can leverage the same parameters to generate novel resume points directly from the model. The model is accessible through the following link: Pulkit506/mistral-7b-resume-iitb.

X. Future Scope

The project's future scope involves augmenting ResumeCraft by incorporating pertinent action verbs and industry-specific keywords companies seek. This enhancement aims to optimize resumes further, ensuring alignment with employer expectations and bolstering the tool's efficacy in facilitating successful job placements within distinct professional domains. Additionally, our objective is to quantitatively demonstrate the likelihood, expressed in percentage terms, of a resume being shortlisted by a domain-specific company.

References

[1] Jamshed Memon, Maira Sami et al. "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)"

[2] Albert Q. Jiang, Alexandre Sablayrolles et al. "Mistral 7B"

[3] Rubentak "Mistral 7B: The best 7 billion parameters LLM yet" medium article

[4] Dheeksha Yogananda "In-depth Understanding of LoRa and Finetuning a LLM using LoRa" medium article

[5] Iz Beltagy, Matthew E. Peters, Arman Cohan "Longformer: The Long-Document Transformer"

[6] Joshua Ainslie, James Lee-Thorp et al. "GQA: Training Generalized Multi-Query Transformer Models from Multi-Head Checkpoints"

[7] Edward Hu, Yelong Shen et al. "LORA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS"

[8] Elias Frantar, Saleh Ashkboos et al. "GPTQ: ACCURATE POST-TRAINING QUANTIZATION FOR GENERATIVE PRE-TRAINED TRANSFORMERS"

[9] Aitor Porcel Laburu "Run Llama 2 7b and 13b for free in Kaggle using GPTQ" medium article

[10] Vikash Ruhil "Data Science and Large Language Models (LLaMA vs LaMA-13B and GPT-3 175B): A Revolution That Will Change The World" medium article