NAMED ENTITY EXTRACTION IN STRUCTURED FORMAT FROM RESUME USING LLAMA 2

Dissertation Phase 2 Report

Submitted by

Ms. ANN B MERIN

REG NO : TKM22MEAI05

the APJ Abdul Kalam Technological University in partial fulfillment for the award of the degree of

MASTER OF TECHNOLOGY
IN
Artificial Intelligence

Under the guidance of
Dr. Anzar S M

Centre for Artificial Intelligence

TKM College of Engineering, Kollam

JUNE 2024

Thangal Kunju Musaliar College of Engineering
Centre for Artificial Intelligence

C E R T I F I C A T E

This is to certify that, this report titled ***NAMED ENTITY EXTRACTION IN STRUCTURED FORMAT FROM RESUME USING LLAMA 2*** is a bonafide record of the **Dissertation Phase 2** work presented by **ANN B MERIN (TKM22MEAI05)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M. Tech in Artificial Intelligence** in **APJ Abdul Kalam University**.

Internal Supervisor	Project Coordinator	Head of the Department
Dr. Anzar S M Assistant Professor Dept. of ECE TKMCE	Dr. Sumod Sundar Associate Professor Centre for AI TKMCE	Dr Imthias Ahamed T P Professor Centre for AI TKMCE

ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the Almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hameed**, Principal of T.K.M College of Engineering for allowing me to present my dissertation phase 2. I would like to thank **Dr. Imthias Ahamed T P**, Professor and Head of the Department, Centre for Artificial Intelligence, TKM College of Engineering, Kollam, for his constant support and encouragement throughout the work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide **Dr. Anzar S M** , Assistant Professor, Department of Electronics and Communications Engineering and Project Coordinator, **Dr. Sumod Sundar**, Associate Professor, Centre for Artificial Intelligence(AI), TKM College of Engineering, Kollam for their expert guidance, cooperation, and immense encouragement. I also extend my thanks to the entire faculty and staff members of the Centre for AI, TKMCE, who have encouraged me throughout this work.

I express my deepest gratitude to **Mr. Ajish M J**, Manager, Tata Elxsi, Trivandrum, for entrusting me with the project idea and giving me the opportunity to work on it. Sincere thanks to **Mr. Vishnu A K**, Senior Engineer, Tata Elxsi, Trivandrum, for his mentoring and guidance throughout the project. I also extend my thanks to everyone at Tata Elxsi, Trivandrum, for their support and help throughout this project.

I also express my thanks to my loving parents and friends, for their support and encouragement in the successful completion of this work.

Ann B Merin

Abstract

Recruitment agencies and firms encounter a significant challenge in handling the multitude of resumes they receive daily, which come in various formats such as PDFs, DOCX files, and with diverse layouts. Extracting relevant information and identifying suitable candidates from this diverse pool is a time-consuming process due to the resumes' unstructured nature and variations in format, style, and content. To address this challenge, a system has been proposed to automate the conversion of unstructured resumes into standard structured formats. The system aims to generate resumes in a standard template, ensuring uniformity across all resumes. This automation process is crucial for streamlining the resume screening process, saving time and effort for recruiters. The proposed system utilizes the qlora parameter-efficient fine-tuning technique with the Llama 2 model. This technique minimizes the need for extensive GPU resources while achieving effective fine-tuning. The fine-tuned model yielded an impressive F1 score of 0.8928, surpassing the performance of the previously instruction-tuned model. Overall, the proposed system offers a robust solution for automating the resume screening process. By improving the efficiency and effectiveness of candidate selection, it provides significant benefits to recruitment agencies and firms, allowing them to focus on more strategic tasks.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In a competitive job market saturated with highly qualified candidates, securing a job can be a daunting task. The first hurdle in the recruitment process is often the resume, which serves as a crucial tool for making a positive first impression. However, in an effort to stand out, applicants often resort to creating resumes with fancy styles and layouts, potentially overshadowing the essential content needed for the specific job. This overzealous approach can lead to resumes filled with unnecessary information, making it challenging for recruiters to efficiently sift through the hundreds of applications they receive, each in a different format, font, and style. This cumbersome task not only poses a significant challenge in identifying the right candidate but also consumes a considerable amount of time. An automated resume parser, powered by large language models and natural language processing (NLP), presents a solution to this challenge, streamlining the resume screening process for recruiters and ensuring that each candidate is evaluated fairly and efficiently based on relevant criteria.

Resume parsing is a crucial task in the recruitment process, automating the extraction of relevant information from resumes to streamline candidate evaluation. Over the years, various techniques and approaches have been proposed to tackle the challenges posed by the diverse formats and structures of resumes. In this paper, we provide an overview of different resume parsing techniques used in recent research. Traditional resume parsing methods often rely on rule-based algorithms and heuristics to extract information. These methods involve the use of regular expressions and pattern matching to identify and extract key sections such as personal details, work experience, and education. While effective to some extent, these approaches are limited in their ability to handle the diverse and complex layouts of modern resumes. Machine learning (ML) has been widely adopted in resume parsing to improve accuracy and handle various resume formats. Supervised ML algorithms, such as Support Vector Machines (SVM) [15] and Random Forests, have been used to classify resume sections and extract information. These models are trained on labeled datasets to learn the patterns and structures of resumes.

Deep learning techniques, particularly transformer models like BERT (Bidirectional Encoder Representations from Transformers), have shown great promise in resume parsing. These models can capture complex relationships and dependencies in text, making them suitable for parsing resumes with diverse layouts and structures. BERT-based models have been used for tasks such as named-entity recognition (NER) and information extraction from resumes.

Hybrid approaches that combine traditional rule-based methods with machine learning and deep learning techniques have also been proposed. These approaches aim to leverage the strengths of each approach to improve parsing accuracy and handle various resume formats effectively. Recent research has focused on enhancing the performance of resume parsing through the use of large pre-trained language models, domain-specific fine-tuning, and data augmentation techniques. These advances have led to significant improvements in parsing accuracy and the ability to handle a wide range of resume formats[4][2].

In this work, the focus is on addressing the challenges faced by agencies and firms in handling resumes in various formats, such as PDFs and DOCX files, with diverse layouts and styles. The unstructured nature and format variations of these resumes make it difficult to extract relevant information and identify suitable candidates. To automate this process, the authors propose a system that converts unstructured resumes into standard formats and generates resumes in a uniform template. The system uses the Qlora parameter-efficient fine-tuning technique with the Llama 2 model, which requires minimal GPUs. Through fine-tuning, the model achieves an impressive F1 score of 0.8928, surpassing the performance of the previously instruction-tuned model. This high accuracy indicates the effectiveness of the proposed system in extracting and processing resume information. Overall, the proposed system offers a robust solution for automating resume screening, which not only saves recruiters time and effort but also enhances the efficiency and effectiveness of candidate selection.

1.0.1 Objective

The primary objective of this work is to fine-tune a large language model (LLM) using a minimal number of GPUs to create a resume parser capable of extracting 33 categories of named entities from resumes. To achieve this, we employed a technique known as Parameter Efficient Fine-Tuning (PEFT) with a method called QLoRA. QLoRA involves quantizing the weights of the LLM and performing Low-Rank Adaptation, which allows us to fine-tune the model efficiently without requiring extensive computational resources. Typically, fine-tuning large language models like Llama, GPT, and Mistral demands a significant number of GPUs and substantial resources, making it an expensive process. However, with PEFT, we can achieve optimal results using fewer resources.

The developed resume parsing system is designed to automatically extract essential company information from resumes, regardless of their layout, style, or file format, including DOC, PDF, and text files. The system then uses the extracted information to generate a resume in a standard company template, containing only the necessary details. This approach ensures that the output is uniform and meets the company's requirements, making it easier for HR professionals to review and process resumes.

To evaluate the performance of the fine-tuned Llama 2 model, it is compared with the prompt fine-tuned Llama 2 model, incorporating few-shot learning techniques. The fine-tuned model is expected to produce output in a JSON-like format, capturing the required information accurately and efficiently. This comparison helps us understand the effectiveness of our fine-tuning approach and the improvements it brings to the resume parsing system. By leveraging advanced techniques like QLoRA, the aim is to create a robust and efficient

resume parser that can handle diverse resume formats and provide valuable insights for the recruitment process.

This work aims to offer insights into the effectiveness of fine-tuning large language models (LLMs) using Parameter Efficient Fine-Tuning (PEFT), demonstrating that it is possible to achieve good results without the need to fine-tune the entire Llama 2 model. This approach is more reliable compared to relying solely on prompting, as the output from prompting can be inconsistent and variable. Additionally, using prompting often involves API calls to the model, which can lead to issues such as runout errors or other reliability concerns, making it less dependable. Furthermore, there is a significant risk of data security when using external API calls, as sensitive information may be exposed or compromised.

By using PEFT, we can mitigate these issues. Fine-tuning with PEFT allows us to adapt the model to specific tasks efficiently, ensuring consistent and reliable outputs. This method not only reduces the computational resources required but also enhances the stability and security of the model. By retaining control over the fine-tuning process, we can ensure that the model performs optimally for our specific use case, in this case, extracting information from resumes. This approach is particularly valuable for applications where data security and consistency are paramount. By avoiding external API calls and keeping the fine-tuning process in-house, we can better protect sensitive data and ensure that the model behaves predictably. In summary, this work highlights the benefits of using PEFT to fine-tune LLMs, offering a more reliable, secure, and efficient alternative to traditional prompting methods.

1.0.2 Major Contributions

The major contributions of this work are multifaceted and demonstrate significant advancements in the fine-tuning of large language models (LLMs). This work proves that effective fine-tuning of LLMs can be achieved using Parameter Efficient Fine-Tuning (PEFT) and weight quantization, allowing for high-quality results with minimal computational resources. This approach challenges the conventional notion that fine-tuning large models requires extensive GPU resources, making it a cost-effective solution. Synthetic resume data was generated using GPT-3.5 to fine-tune the model. This synthetic data provided a diverse and extensive dataset, enhancing the model's ability to accurately extract named entities from resumes. This also proves that using LLMs, data creation becomes much easier and faster.

Mainly, the work explores and compares two fine-tuning techniques: PEFT-based fine-tuning and prompt-based fine-tuning. In prompt-based fine-tuning, the model was trained using few-shot prompting and instruction-based examples, where specific input-output pairs and instructions were provided to guide the model. By comparing these two methods, valuable insights were gained into their effectiveness and efficiency. For the experiments, the smallest version of the Llama 2 model was used for fine-tuning with PEFT, while the largest version of the Llama 2 model was used for prompt engineering. This comparison highlighted the advantages of each approach and demonstrated that even with the smallest model, PEFT can yield substantial results.

Another major contribution of this work is the development of a system for automating

the extraction and structuring of information from resumes in various layouts and formats, including PDF and DOC files. The initial step involves converting resumes to text, utilizing pdfminer for PDF files and the python-docx library for DOCX files. One significant contribution lies in the extraction of required categories from resumes. Employing advanced natural language processing models, like Llama 2 and GPT-3.5 Turbo, this work demonstrates the extraction of 30 named entities and their relational information. The output is structured in a JSON-like format, providing a clear and organized representation of the extracted information. Furthermore, in phase one of the work, a novel aspect is introduced by incorporating few-shot instruction tuning into the extraction process. While acknowledging the efficacy of GPT-3.5 Turbo, the practicality of achieving similar results with the Llama 2 13b model through few-shot instruction tuning is higlighted. This finding not only expands the choice of models but also emphasizes the versatility of the proposed approach, accommodating different language models based on availability and accessibility. A significant contribution extends to the final step of generating a standardized resume in a company template format. Leveraging the extracted information and their relationships, the python-docx library is employed to craft a well-structured DOCX file. This output serves as a practical and valuable resource for companies seeking an efficient and uniform representation of candidate information in the hiring process.

Some noteworthy contributions to the field of automated resume processing have been made through this work. From the initial conversion of diverse resume formats to text, through the advanced extraction of categorized information using state-of-the-art language models, to the generation of standardized resumes, the proposed methodology presents a holistic and effective solution for streamlining the hiring process. The introduction of few-shot instruction tuning further enhances the adaptability of the approach, making it a valuable contribution to the broader domain of natural language processing and automated document structuring. In summary, this work significantly contributes to the field by showing that PEFT, combined with weight quantization, can fine-tune LLMs effectively with minimal resources. It also provides a comprehensive comparison between PEFT-based fine-tuning and prompt-based fine-tuning, offering valuable insights into their respective strengths and applications. Through the development of an automated resume processing system, this work demonstrates a practical and versatile solution for enhancing the efficiency and effectiveness of the hiring process.

Chapter 2

Literature Survey

Natural language processing (NLP) algorithms for materials, and named entity recognition (NER) models in particular, have made significant advances over the past half-decade towards structuring the existing body of textual materials science knowledge[3][4][5]. A key outstanding challenge in materials NLP is the development of relation extraction (RE) techniques to extract structured information that accurately describes the links between these entities. Data can be classified into three types. structured data, semi structured data and unstructured data. Resumes in pdf/doc formats can be classified as an unstructured data. They are converted to text and resume itself can be called a semi structured data as there is some organizational properties in a resume that make it easier to analyze compared to unstructured data [1]. The machine learning models for direct property prediction have been increasingly employed as screening steps for materials discovery and design workflows [1] [2], this approach is limited by the amount of training data available in tabulated databases

Diverse methodologies have been explored across different linguistic and cultural contexts in resume information extraction. The RINX system[1] primarily focused on the Banking Domain dataset, employing a custom Spacy Model for tokenization, entity extraction, and relation extraction through linguistic patterns, CRF Extractor, LSTM CRF Extractor, rule-based and supervised techniques. On the other hand, an end-to-end framework for Italian resumes[2] utilized the Italian BERT model for Named Entity Recognition (NER), emphasizing skill-based searching. The segmentation process involved the manual labeling and division of resumes into personal information, education, work experience, and skills blocks using the CvExtractor tool. A similar emphasis on segmentation was observed in the Brazilian Portuguese resume extraction system[3], which incorporated EfficientNet-B0 for resume ordering, BERT, and CRF for segmentation, with a focus on section and item extraction. Meanwhile, the semi-supervised SCRNER model[4], applied to Chinese cultural relics recognition, introduced a sample selection strategy and utilized a BiLSTM-CRF architecture with word embeddings pretrained by ELMo. Lastly, the Transformers-based information extraction system for Japanese business documents[5] employed BERT, CNN, and a question-answering model for tagging and information extraction, achieving notable results with a combination of BERT and CNN for bidding and sales documents. These studies collectively highlight the diverse approaches and language-specific considerations in the field of information extraction from resumes.

These works had a spectrum of innovative approaches to structured information extraction from diverse textual sources, shedding light on the methodologies adopted in various domains. The first paper explores a sequence-to-sequence strategy for document-level joint named entity recognition and relation extraction (NERRE) within the realm of scientific text. Employing the fine-tuned GPT-3 language model, the approach hinges on approximately 500 document-completion examples to yield structured records, such as JSON documents, containing the targeted information.

Automatation of resumes have been studies for a long period now and one such paper using machine learning is given in [15]. It proposes a cascaded information extraction framework for resume parsing. The framework utilizes a two-pass approach. In the first pass, a Hidden Markov Model (HMM) segments the resume into labeled blocks indicating information types (e.g., Education, Personal Information). In the second pass, depending on the block label from the first pass, specific models are used for detailed information extraction. For example, another HMM is used for educational information extraction within the "Education" block, while a Support Vector Machine (SVM) is employed for personal information extraction within the "Personal Information" block. This approach leverages the hierarchical structure of resumes and achieves better performance compared to flat models that don't consider this structure. The study demonstrates the effectiveness of using HMMs for tasks with a strong sequence of information like general information extraction and educational background extraction, while SVMs are found to be more suitable for personal information extraction with less structured formats. It demonstrated that the cascaded model improved the result compared to the flat models. The system did have the issue of error propagation as it had a pipeline approach. If the first part of the extraction has an error, it will be carried to the next part thereby reducing the accuracy further down below for extraction of valuable information. The model applied multiple information extraction approached in a single resume itself so a fuzzy block selection strategy was used to enlarge search scope. Even then the precision and recall of some personal details did not achieve good result.

In [7], the spotlight is on HealthPrompt, a pioneering prompt-based clinical Natural Language Processing (NLP) framework. This framework operates under a zero-shot learning paradigm, emphasizing the capability to handle clinical text with minimal to no training data. The methodology involves prompt-based learning, a zero-shot learning (ZSL) technique in NLP, wherein task-specific templates are defined. HealthPrompt, leveraging this approach, effectively captures the contextual nuances of clinical texts, demonstrating robust performance across clinical NLP tasks. The experiments conducted on six different pretrained language models underscore the efficacy of HealthPrompt in extracting pertinent information from unstructured healthcare data.

Based on the extensive review done on few shot learning [7], Few-Shot Learning (FSL) is used to address the challenges from limited dataset. FSL proves particularly advantageous in scenarios where prior knowledge is leveraged to rapidly generalize to new tasks with only a few samples. The proposed paradigm can alleviate the need for extensive supervised data collection, proving especially useful in scenarios where data acquisition is constrained. By exploring applications in tasks such as image classification, the paper gives insight on how FSL enhances accuracy through knowledge transfer from labeled images for specific classes

and contributes to the overall generality of the model. This comprehensive survey highlights the nuanced strategies employed in information extraction, ranging from leveraging large language models to embracing innovative learning paradigms like zero-shot and few-shot learning. A sequence-to-sequence approach like in [10], termed LLM-NERRE, for extracting complex scientific information from unstructured text using GPT-3, a large language model is used for conversion of unstructured text to structured format. The evaluation encompasses sequence-level metrics, detailed assessments of materials NERRE tasks, and comparisons with BERT-based and seq2rel approaches. The approach's non-technical nature allows scientists without NLP training to leverage existing models for structured relational datasets. This approach is applicable to other large language models too.

The paper [17] introduced an innovative end-to-end system for resume parsing, utilizing neural network-based classifiers and distributed embeddings. This method leverages positional line information and comprehensive word representations within each text block. The authors justified this approach by stating that neural network based feature extraction can capture more semantic information from texts compared to traditional methods. Pretrained word embeddings were used as they are easy to implement and adapt to specific tasks. Their proposed classification method involves training two types of line classifiers: the line type classifier and the line label classifier. The line type classifier categorizes resumes into four main sections: header, content, metadata, and footer. Meanwhile, the line label classifier further divides these sections into six specific information areas: personal, education, work, project, skill, and publishing. To generate domain-specific word embeddings, the authors utilized the Word2Vec model from the Gensim tool, which effectively transforms lines of text in resumes into word vectors. They also experimented with other models such as GloVe and BERT, fine-tuning these embeddings during training. Various neural network architectures were considered for text block classification, including Text-CNN, RCNN, Adversarial LSTM, Attention BLSTM, and Transformer. For text sequence labeling, using sub-categories from Table 1 for named entity recognition (NER). Four sequence labeling classifiers—Bi-LSTM-CRF, Bi-GRU-CRF, IDCNN-CRF, and BLSTM-CNNs-CRF—were trained and evaluated for performance and decoding speed. The authors applied the k-means algorithm to cluster detected named entities and used cosine similarity based on Term Frequency-Inverse Document Frequency (TFIDF) to map these entities to standard attribute names. The result showcased how the attention BLSTM classifier outperformed others in line type classification, achieving F-1 scores of 0.96, 0.93, 0.96, and 0.97 for header, content, metadata, and footer, respectively. Attention BLSTM and Adversarial LSTM excelled in handling long sentences with higher recall and F-1 measures, while Text-CNN was superior for short phrases. RCNN showed better performance than Text-CNN for long sentences. Given its strong performance and robustness, the authors chose Attention BLSTM for practical text block segmentation. The work also made a comparison of the performances of the four sequence labeling classifiers in identifying resume information. The study found that BLSTM-CNNs-CRF outperformed the others. The greedy ID-CNN paired with Viterbi-decoding surpassed both Bi-LSTM and Bi-GRU, and Bi-GRU slightly outperformed Bi-LSTM in NER tasks when combined with CRF. In terms of decoding speed, IDCNN-CRF was the fastest. Additional findings include the effectiveness of the CNN layer in BLSTM-CNNS-CRF as a text feature extractor compared to the LSTM layer, the superior performance of the proposed method over Writing-Style and CHM, and the advanced capabilities of BERT over other

word representation algorithms like Word2Vec and GloVe.

The proposed resume parsing system in [12] leverages contemporary Natural Language Processing (NLP) techniques to automate the analysis and extraction of key information from resumes. The system addresses the challenges posed by the unstructured nature of resumes, which makes manual screening error-prone and inefficient in corporate recruitment environments. The system's architecture consists of four main phases: Text Segmentation, Named Entity Recognition (NER), Text Normalization, and Co-reference Resolution. Text Segmentation is crucial, as it splits the extracted text into segments based on attributes like Name, Phone, Email, and Web information. NER utilizes specialized chunkers to recognize named entities within these segments, using cues like prepositions and famous names. Text Normalization ensures coherence by expanding abbreviations and standardizing entities. Co-reference Resolution links proper names to the same entity, improving data consistency. The system's performance is validated through extensive experiments on familiar datasets, achieving an impressive overall accuracy of 94.19%. This high accuracy demonstrates the system's potential as a robust recruitment tool, capable of handling resume screening and ranking tasks efficiently. Future research will focus on enhancing the segmentation phase's accuracy and expanding the resume corpus collection. Improving NER accuracy is also a priority to further enhance the system's overall performance and relevance in real-time applications.

The work in [13] addresses the challenge faced by employers in manually sorting through a large number of job applications to find suitable candidates. To address this, the authors propose a two-stage solution. Firstly, they develop a resume parser capable of extracting comprehensive information from candidate resumes. This parser is made accessible to the public through a web application. The parser is particularly adept at handling resumes in both LinkedIn and non-LinkedIn formats. For LinkedIn resumes, which follow a more structured format, the parser achieves 100% information extraction accuracy, ensuring no loss of structure. This is crucial as it allows for the extraction of key information such as personal details, work experience, and education. Secondly, the authors utilize BERT, a state-of-the-art language representation model, for sentence pair classification to rank candidates based on their suitability to a given job description. To approximate the job description, the authors use the descriptions of past job experiences listed in the candidate's resume. This approach leverages BERT's ability to understand contextual nuances and relationships between sentences, enabling it to effectively match candidate experiences with job requirements. The ranking model achieves a commendable accuracy of 73% for candidate suitability, providing a reliable baseline for candidate evaluation. Overall, the work not only provides a practical solution for automating the candidate ranking process but also contributes to the research by exploring the feasibility of a generic resume parser and offering a user-friendly tool for employers to parse resumes efficiently.

A ranking system based work with resume parsing is given in [14]. It propose a two-stage approach consisting of resume parsing and ranking the resumes using BERT. For resume parsing they developed a public web application parser that extracts comprehensive information from candidate resumes. The work addressed the challenge in resume processing, where inconsistent formats can hinder data extraction accuracy. For ranking of candidates,

by leveraging the power of BERT sentence pair classification, the authors rank candidates based on their suitability for a specific job description. They achieve this by approximating the job description using the candidate's past work experiences mentioned in the resume. This approach presents an alternative to relying solely on the provided job description, which might not always capture the ideal candidate profile. A custom dataset was used consisting of both LinkedIn and non-LinkedIn resumes. According to the paper 100% accuracy was achieved for parsing LinkedIn resumes and establish a strong baseline accuracy of 73% for candidate suitability ranking. For structuring and converting the general resumes they mention usage of heuristic rules based on spacing to preserve the logical flow of texts. However, the paper failed to mention the accuracy of the extracted text using this method. Detailed methodology of the heuristic rules used were also not mentioned. These resumes were segmented based on headings. They tried to use unmentioned image techniques to extract tables and section boundaries of the resumes but mentions that due to variations in the resume formats they failed to do so accurately.

The work in [16] presents a novel semi-supervised Named Entity Recognition (NER) approach designed to extract educational qualifications, such as degrees and institutes, from resumes. The method addresses the common challenge of limited annotated resume data by employing a multi-step workflow that incrementally enhances the model's accuracy. Initially, a small manually annotated seed set of 550 resumes is prepared, involving preprocessing steps like converting resumes to JSON format, extracting words, and Part-of-Speech (POS) tagging to create features. The entities are annotated using BIO encoding (Beginning, Inside, Outside). An initial NER model is then trained using a combination of Word Embeddings for words and POS tags, a Convolutional Neural Network (CNN) for character-level features, and Bidirectional Long Short-Term Memory (Bi-LSTM) layers. This model is applied to a larger unlabeled dataset of 360 resumes to predict entities. A critical component of the approach is the correction module, which refines these predictions using two strategies: string matching with pre-defined lists of institute and degree names, accommodating potential abbreviations, and employing Fuzzy Wuzzy for similarity matching when exact matches are unavailable. The iterative improvement process involves adding the corrected entities from the unlabeled data to the seed set and retraining the model, resulting in progressive accuracy enhancements. This methodology achieves a high accuracy of 92.06% for NER on educational qualifications in resumes, demonstrating the significant impact of the correction module that leverages both predefined lists and fuzzy matching. This approach offers several advantages over previous methods that relied heavily on extensive manual annotation or exhibited lower precision and recall rates. Additionally, the model holds potential for extension to other resume sections, such as identifying majors and specializations. The authors acknowledge certain limitations, including the challenge of fairly comparing this technique with other NER approaches due to the scarcity of prior work on resume parsing and the absence of a large, standardized dataset. Moreover, the accuracy improvements may eventually plateau, given the privacy concerns that restrict access to extensive amounts of resume data.

All above supervised learning techniques have the drawback of requiring a large amount of annotated resume data. Collecting extensive annotated datasets is very challenging and time consuming. To overcome this, the authors propose a new resume parsing framework

that addresses the limitations of previous methods, including rule-based, supervised, and semantics-based approaches. In the proposed system, initially like previous methods, raw text is extracted from resumes using PDFBox. It converts PDF files into text, handling non-ASCII characters, missing or extra spaces, and punctuation marks accurately. Boolean Naive Bayes (BNB) is then employed to divide the text into blocks for categorization. Here the texts are divided into separate text blocks based on personal information, education, interests, work experience and skills. From these segmented texts, resume facts are identified using Named Entity Recognition (NER) using BERT. Here the named entities like education, experience, skills, location and job title are extracted. Unlike earlier methods, this approach extends named entities with a custom-built ontology to address data sparsity by incorporating skills from online job portals, specifically the CSO and European Skills/-Competence, Qualification, and Occupation ontologies (ESCO). An ontology represents a knowledge base that helps create a semantic model of data associated with specific domain knowledge. It also establishes relationships between different types of semantic knowledge within a domain. By using an ontology, the system can understand and categorize the information in resumes better than in previous methods. It can recognize different terms with same meaning and understand the relationships between different qualifications and job requirements. This makes the resume parsing process more accurate and effective.

The diverse array of approaches utilizing Natural Language Processing (NLP) models for the conversion of unstructured text to structured formats is present. The studies showcase the versatility and effectiveness of various techniques across different domains. The advancements in leveraging NLP models for structured information extraction. The diversity of approaches, ranging from domain-specific frameworks to general-purpose models like GPT-3, demonstrates the adaptability and wide applicability of NLP techniques. As the field evolves, the potential for accelerating data extraction and knowledge generation in various domains becomes increasingly evident. The studies collectively contribute to the growing body of knowledge in the field of NLP-based information extraction, highlighting both the opportunities and challenges in harnessing these models for effective conversion of unstructured text to structured information.

2.1 Motivation

The current state of research in resume processing reveals several critical gaps that hinder the development of effective models for standardizing diverse resume layouts. Firstly, the absence of a model capable of converting various formats, such as two-column, multicolumn, and tables, into a standardized template is a significant challenge. This gap underscores the need for a versatile solution that can accommodate the varied formatting choices made by job applicants, thereby enhancing the efficiency of automated resume parsing systems.

Another substantial research gap pertains to the limited focus on relation extraction within resumes. Existing models often struggle to identify and extract meaningful relationships between different pieces of information within a resume. Addressing this gap is crucial for providing context to the extracted data, enabling recruiters and hiring systems to make more informed decisions about a candidate's qualifications and experience.

Furthermore, the insufficient emphasis on fine-tuning large language models for the conversion of unstructured resume text data into a structured format is a notable research limitation. Particularly, when dealing with limited datasets, the need for tailored models becomes apparent. Future research should explore fine-tuning strategies to optimize the performance of pre-trained models specifically for the intricate task of converting unstructured resume data into a structured and analyzable format.

Additionally, the research landscape lacks comprehensive methodologies for extracting work and result information from diverse resume layouts. This gap signifies a deficiency in the ability of existing models to capture detailed information about a candidate's professional background and achievements. Addressing this gap is essential for enhancing the overall understanding of an applicant's qualifications and performance during the hiring process.

The motivation for future research in this domain is derived from a thorough literature review that identifies these gaps. By synthesizing insights from existing studies, researchers can emphasize the urgency of developing more robust models. The literature review underscores the importance of advancing the state-of-the-art in automated resume processing and analysis. Ultimately, bridging these research gaps promises to bring about significant improvements in the efficiency and accuracy of automated resume parsing systems, thus revolutionizing the hiring process.

Chapter 3

Methodology

3.1 Introduction

There are two main approaches to extracting information from resumes: the traditional method using Named Entity Recognition (NER) and Relational Extraction, and the advanced method using Generative AI with Large Language Models (LLMs). In the traditional approach, NER models like those available in Spacy can identify basic entities such as names, places, and organizations. However, when using a pretrained BERT-based NER model in Spacy, it struggled to accurately recognize Indian names and places. Additionally, this approach was unsuitable for extraction of different entities along with relational data extraction, which involved extracting 32 different entities. Building a custom NER model would require an extremely large dataset, making it time-consuming and impractical. Even with a large dataset, NER fails to effectively identify varied and complex relational entities such as project skills and descriptions and will not be able to generate experience summary based on the resume context.

Given these limitations, we turned to more advanced techniques using Generative AI. A text generation model is required for proper geenration of Large Language Models, pre-trained on a wide range of topics, can identify complex relational entities quickly and with minimal data. We tested state-of-the-art LLMs like GPT, Gemini, and Cohere, and found that with few-shot prompting, they delivered the best results. Among these, OpenAI's GPT-3.5 turbo performed exceptionally well, accurately extracting details such as experience summaries and project information, which was verified manually. Few-shot prompting via API calls can be a convenient way to extract information from resumes, but it comes with several disadvantages.

One of the primary issues is the occurrence of time-out errors. When processing a large number of resumes, multiple API calls are required, and this can lead to interruptions in the extraction process. These time-out errors can cause significant delays, hindering the efficiency of the workflow. Another disadvantage is latency which can slow down the process, particularly if the server handling the requests is experiencing high traffic or network issues. This can be frustrating when dealing with tight deadlines or large batches of resumes that need to be processed quickly. Cost is also a significant factor. Frequent API calls can become expensive, especially when processing high volumes of resumes. The costs can add up over time, leading to substantial expenses that might not be sustainable for all organizations.

Along with this data privacy concerns are another critical issue. Sending sensitive resume data to an external API raises questions about the security and privacy of the information. Companies must be cautious about how they handle and transmit personal data to avoid potential breaches or misuse. Another major issue is that the response generated can get varied especially if free models are used which might still be undergoing updates and fine-tuning like Gemini.

Considering these disadvantages, companies need to carefully evaluate whether few-shot prompting via API calls is the most suitable approach for their needs or if alternative methods, such as fine-tuning custom models, might provide better results. To overcome this, the solution was to fine-tune a custom LLM to handle the data extraction more efficiently and meet the requirements. The primary objective of this project is to fine-tune the Llama 2 7b model using the Low-Rank Adaptation technique (LoRA) to address memory and computing challenges. The goal is to train the model using a free version of Kaggle Notebook with a single GPU (P100) to generate structured output from unstructured resume text. To achieve this, a synthetically generated dataset is employed, created using GPT 3.5 turbo.

This model facilitates the automatic conversion of resumes, regardless of layout or format, into a standardized resume template. The template includes a fixed number of extracted entities or information from the resume, such as name, technical skills, educational qualifications, projects undertaken, and work experience. This standardization streamlines the recruitment process, enabling recruiters to focus on job-specific factors when selecting candidates. Moreover, standardized resumes are beneficial for customer-oriented companies, as they can be easily presented to clients. Additionally, they simplify the task of assigning employees to projects based on the information provided in the resumes. Initially, resumes submitted by candidates or job seekers in PDF or DOC format are converted to text using the pdfminer library in Python. This extracted text serves as input for the fine-tuned Llama 2 model, which transforms it into a structured JSON-like string containing the required entities. If these entities are found in the resume, they are extracted; otherwise, they are marked as null. Subsequently, this JSON-like string is converted to actual JSON format. Using the python-docx library, a new document is then generated with a standard template, into which the extracted information is inserted.

3.2 Pipeline

The proposed pipeline efficiently processes resumes in PDF, DOC, and text formats. It extracts the required named entities from the resumes, if available, and converts them into a standard template with the extracted named entities. Figure 3.1 illustrates the pipeline of the entire proposed system. The system first checks the format of the resume (PDF, DOCX, or text). If it's not in one of these extractable formats, an error message is sent to the user. If the resume is in an extractable format, the raw text is extracted. This unstructured text then undergoes preprocessing to remove multiple blank spaces and symbols if they are present. Using a Large Language Model (LLM), thirty named entities are extracted from the unstructured resume text if they are present, and the output is structured into a JSON-like format string using the LLM. From this JSON-like string, where the named entities and their corresponding extracted values form key-value pairs, a new document is created. In

this new document which has a predetermined template, the extracted required information is mapped.

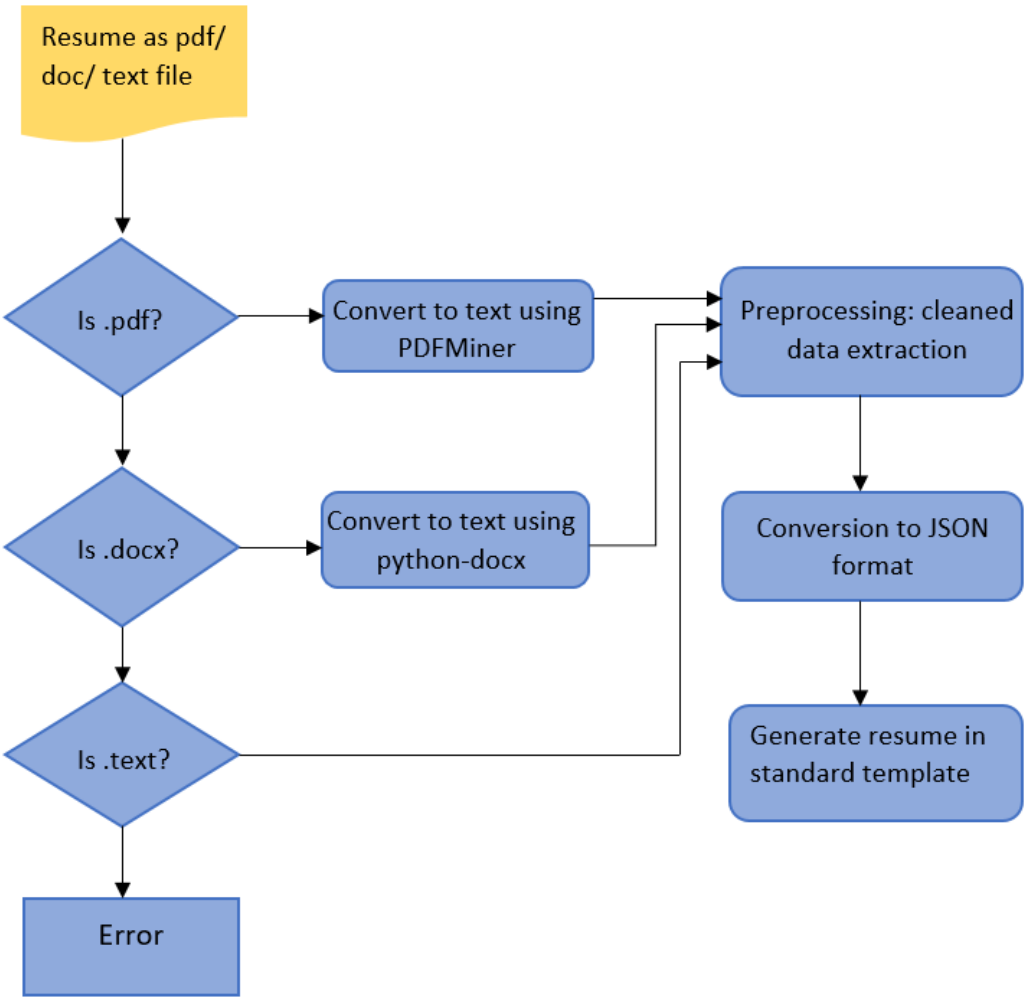


Figure 3.1: Proposed Pipeline

The pipeline mainly consists of three steps:

- **Extraction of Text from Resumes:** In the process of extracting text from resumes in PDF format, the primary focus was on maintaining the natural reading order. The pdfminer.six python library was used for converting pdf to text, as it allowed for the extraction of text in a way that closely resembled the original document's reading flow, even in cases of multicolumn resumes or complex layouts. This was optimised by customising the LAParams object from pdfminer.layout function. The parameters in LAParams include:
  - line_overlap : the minimum overlap between two lines to consider them as one.

- char_margin : the maximum distance between two words to merge them into one line.
- word_margin : the maximum distance between two characters to merge them into one word.
- line_margin : minimum distance between lines to be considered part of same paragraph.
- boxes_flow : determines how much a horizontal and vertical position of a text matters when determining the order of text boxes. for horizontal only -1.0 and +1 for vertical position only.
- detect_vertical : True if vertical text should be considered during layout analysis.
- all_texts : True if layout analysis should be performed on text in figures.

The custom values for this particular system are: line_overlap= 0.5, char_margin=1.0, word_margin=0.5, line_margin = 0.5 ,boxes_flow= -1, detect_vertical=False, all_texts=True When it comes to resumes in document format, converting them to plain text in the natural reading order is much easier than with PDFs. However, extracting tables in the correct reading order can be challenging. This was managed with the help of the python-docx library. It was noted that in some cases, headings separately added to the document could not be extracted correctly.

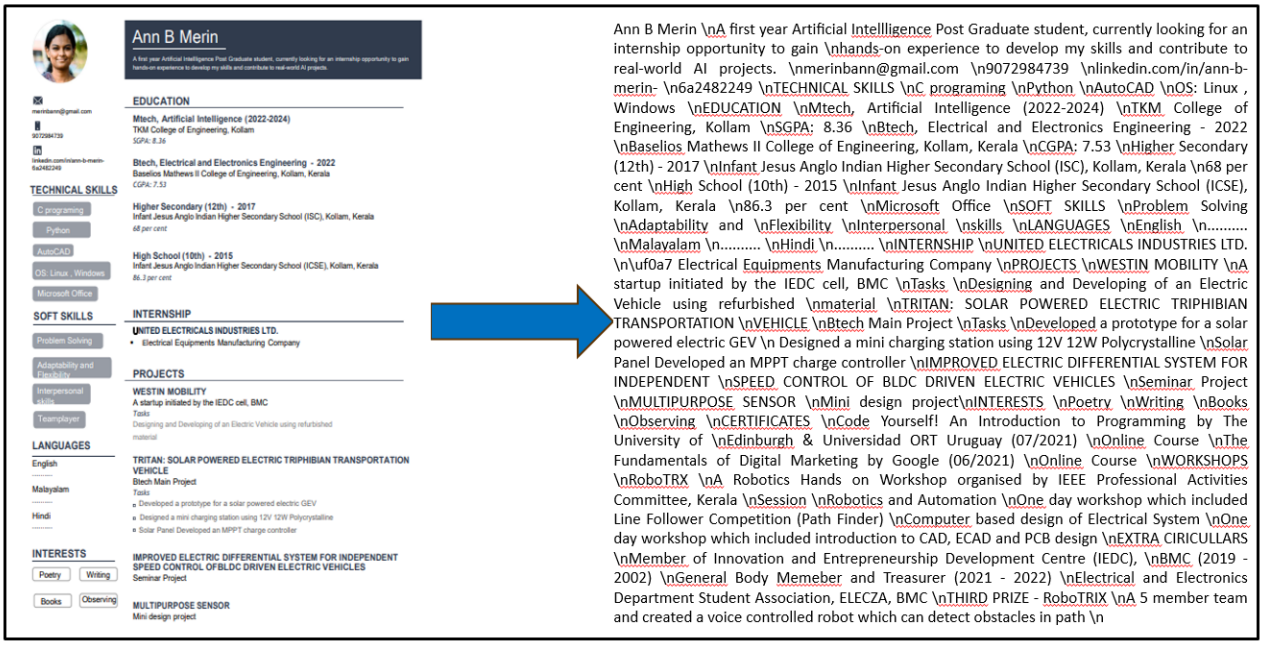


Figure 3.2: Example of Conversion of PDF to text using pdfminer

- **Extraction of Required Information and Conversion to Structured Format:** The extracted text from resumes is used to generate a structured resume in a standard

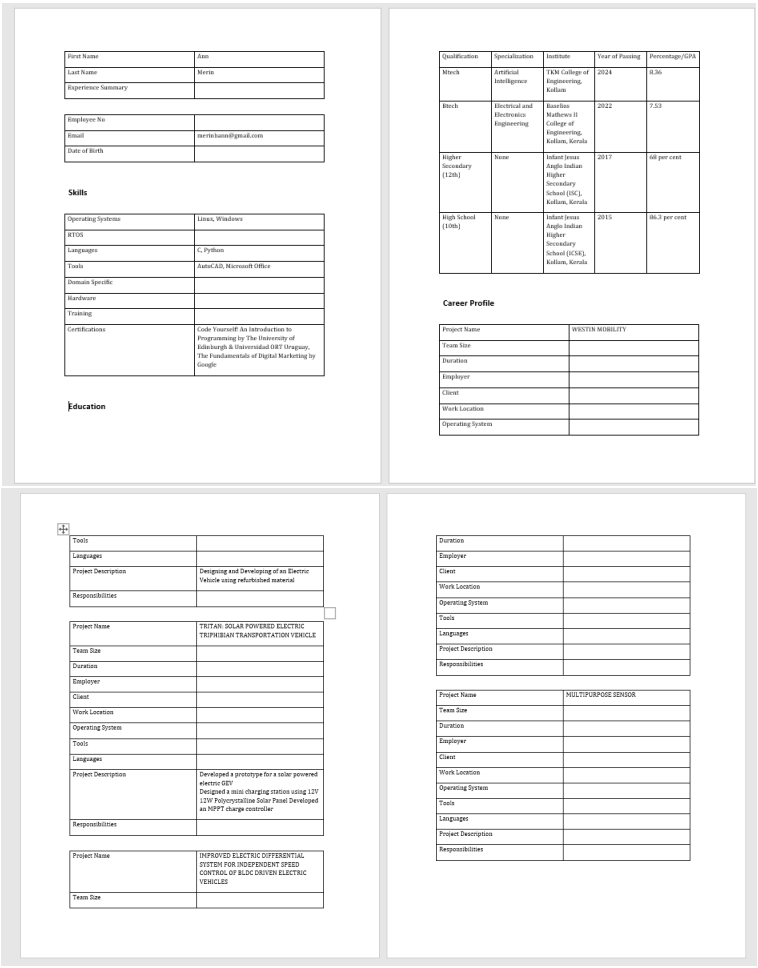
template, including only the necessary information required by the company for efficient recruitment. The core focus of Phase-2 of this dissertation is the extraction of required named entities and their conversion to a structured JSON-like format using fine tuning techniques of LLMs. For this structured information extraction, a Large Language Model (LLM) called Llama 2 is used, which is fine-tuned using a custom dataset generated with GPT-3.5 Turbo. The fine-tuning is implemented along with the Parameter Efficient Fine-Tuning Technique called Low Rank Adaptation (LoRA), which significantly reduces the number of GPUs needed for fine-tuning. The fine-tuned Llama 2 model is trained to extract the required information from unstructured resume text and generate a structured output with the necessary details.
In Phase-1 of the dissertation, the extraction and conversion to a structured format were achieved using instruction fine-tuning of the LLM along with few-shot prompting. Instruction-based few-shot prompting and fine-tuning using PEFT are two of the latest techniques that can be implemented with fewer resources and lower costs when dealing with LLMs.
The entities extracted from the resumes are: First Name, Last Name, Experience Summary, Employ No, Email, Date of Birth, Skills: Operating Systems, RTOS, Languages, Tools, Domain Specific, Hardware, Training, Certifications, Education: Qualification, Specialization, Institute,Year of Passing, Percentage/GPA, Career Profile Project Name, Team Size, Duration, Employer, Client, Work Location, Operating System, Tools, Languages, Project Description, Responsibilities.

{\n "**First name**": "Ann",\n "**Last name**": "Merin",\n "**Experience summary**": null,\n "**Employee Id**": null,\n "**Email**": "merinbann@gmail.com",\n "**Date of birth**": null,\n "**Skills**": {\n "**Operating systems**": "Linux, Windows",\n "**RTOS**": null,\n "**Programming languages**": "C, Python",\n "**Tools**": "AutoCAD, Microsoft Office",\n "**Hardwares**": null,\n "**Domain-specific skills**": null,\n "**Training received**": null,\n "**Certified courses**": "Code Yourself! An Introduction to Programming by The University of Edinburgh & Universidad ORT Uruguay",\n "The Fundamentals of Digital Marketing by Google"\n },\n "**Educational details**": [\n {\n "**Qualification**": "Mtech",\n "**Specialization**": "Artificial Intelligence",\n "**Institution**": "TKM College of Engineering, Kollam",\n "**Year of passing of the course**": "2024",\n "**Mark**": "8.36"\n },\n {\n "**Qualification**": "Btech",\n "**Specialization**": "Electrical and Electronics Engineering",\n "**Institution**": "Baselios Mathews II College of Engineering, Kollam, Kerala",\n "**Year of passing of the course**": "2022",\n "**Mark**": "7.53"\n },\n {\n "**Qualification**": "Higher Secondary (12th)",\n "**Specialization**": null,\n "**Institution**": "Infant Jesus Anglo Indian Higher Secondary School (ISC), Kollam, Kerala",\n "**Year of passing of the course**": "2017",\n "**Mark**": "68 per cent"\n },\n {\n "**Qualification**": "High School (10th)",\n "**Specialization**": null,\n "**Institution**": "Infant Jesus Anglo Indian Higher Secondary School (ICSE), Kollam, Kerala",\n "**Year of passing of the course**": "2015",\n "**Mark**": "86.3 per cent"\n }\n],\n "**Career profile**": [\n {\n "**Project name**": "WESTIN MOBILITY",\n "**Team Size**": null,\n "**Duration of project**": null,\n "**employer**": null,\n "**client**": null,\n "**work location**": null,\n "**operating system**": null,\n "**Tools used for the project**": null,\n "**Programming languages used for the project**": null,\n "**Project description**": "Designing and Developing of an Electric Vehicle using refurbished material",\n "**Responsibility**": null\n },\n {\n "**Project name**": "TRITAN: SOLAR POWERED ELECTRIC TRIPHIBIAN TRANSPORTATION VEHICLE",\n "**Team Size**": null,\n "**Duration of project**": null,\n "**employer**": null,\n "**client**": null,\n "**work location**": null,\n "**operating system**": null,\n "**Tools used for the project**": null,\n "**Programming languages used for the project**": null,\n "**Project description**": "Developed a prototype for a solar powered electric GEV\\nDesigned a mini charging station using 12V 12W Polycrystalline Solar Panel Developed an MPPT charge controller",\n "**Responsibility**": null\n },\n {\n "**Project name**": "IMPROVED ELECTRIC DIFFERENTIAL SYSTEM FOR INDEPENDENT SPEED CONTROL OF BLDC DRIVEN ELECTRIC VEHICLES",\n "**Team Size**": null,\n "**Duration of project**": null,\n "**employer**": null,\n "**client**": null,\n "**work location**": null,\n "**operating system**": null,\n "**Tools used for the project**": null,\n "**Programming languages used for the project**": null,\n "**Project description**": null,\n "**Responsibility**": null\n },\n {\n "**Project name**": "MULTIPURPOSE SENSOR",\n "**Team Size**": null,\n "**Duration of project**": null,\n "**employer**": null,\n "**client**": null,\n "**work location**": null,\n "**operating system**": null,\n "**Tools used for the project**": null,\n "**Programming languages used for the project**": null,\n "**Project description**": null,\n "**Responsibility**": null\n }\n]\n}

Figure 3.3: Example of Output from Llama 2 Model

- **Conversion to standard template:** The final step involves converting the structured resume, in JSON format, into a document using the python-docx library. The document is automatically generated based on a pre-established template, with two-column tables. The left column contains fixed category names representing distinct sections within the resume, while the right column is dynamically populated with information sourced from the JSON-formatted text. This automation process ensures consistency in the layout and structure of the resumes, presenting a uniform and pro-

fessional appearance.



First Name	Ann
Last Name	Merin
Experience Summary	

Employee No	
Email	merinhann@gmail.com
Date of Birth	

Skills

Operating Systems	Linux, Windows
RTOS	
Languages	C, Python
Tools	AutoCAD, Microsoft Office
Domain Specific	
Hardware	
Training	
Certifications	Code Yourself! An Introduction to Programming by The University of Edinburgh & Universidad ORT Uruguay, The Fundamentals of Digital Marketing by Google

Education

Qualification	Specialization	Institute	Year of Passing	Percentage/GPA
Mtech	Artificial Intelligence	TKM College of Engineering, Kollam	2024	8.36
Btech	Electrical and Electronics Engineering	Rajadhani Mathews II College of Engineering, Kollam, Kerala	2022	7.53
Higher Secondary (12th)	None	Infant Jesus Anglo Indian Higher Secondary School (ISC), Kollam, Kerala	2017	68 per cent
High School (10th)	None	Infant Jesus Anglo Indian Higher Secondary School (ICSE), Kollam, Kerala	2015	86.3 per cent

Career Profile

Project Name	WESTIN MOBILITY
Team Size	
Duration	
Employer	
Client	
Work Location	
Operating System	

Tools	
Languages	
Project Description	Designing and Developing of an Electric Vehicle using refurbished material
Responsibilities	

Project Name	TRITAN: SOLAR POWERED ELECTRIC TRIPHIBIAN TRANSPORTATION VEHICLE
Team Size	
Duration	
Employer	
Client	
Work Location	
Operating System	
Tools	
Languages	
Project Description	Developed a prototype for a solar powered electric GEV Designed a mini charging station using 12V 12W Polycrystalline Solar Panel Developed an MPPT charge controller
Responsibilities	

Project Name	IMPROVED ELECTRIC DIFFERENTIAL SYSTEM FOR INDEPENDENT SPEED CONTROL OF BLDC DRIVEN ELECTRIC VEHICLES
Team Size	

Duration	
Employer	
Client	
Work Location	
Operating System	
Tools	
Languages	
Project Description	
Responsibilities	

Project Name	MULTIPURPOSE SENSOR
Team Size	
Duration	
Employer	
Client	
Work Location	
Operating System	
Tools	
Languages	
Project Description	
Responsibilities	

Figure 3.4: Example of Generated Resume in docx format

3.3 Proposed Methodology

Fine-tuning large language models comes with several challenges, mainly related to the resources and infrastructure needed .One of the primary needs for fine-tuning is access to powerful servers equipped with high-end GPUs. These GPUs are necessary to handle the complex computations involved in training the models, even when working with a minimal amount of data. These powerful machines are costly and not easily available to all, especially smaller companies. Fully fine-tuning a large model is a big task, even for large companies. It requires a lot of resources, including advanced servers and extensive computational power, making it impractical for most organizations. Besides the hardware, it also needs a lot of energy and cooling systems to keep the servers running smoothly. Hence, even though fine-tuning large language models can lead to very accurate and customized results, it requires a lot of resources and expertise. The need for powerful servers, high-end GPUs, extensive com-

putational power, and specialized knowledge makes it a challenging and costly process. A way to reduce these expenses and challenges was introduced with Parameter-Efficient Fine-Tuning (PEFT) [19]. PEFT allows for fine-tuning large language models without the need for extensive computational resources and high-end hardware. With PEFT, you can achieve effective fine-tuning using fewer GPUs and less powerful servers, which lowers the overall cost. It also shortens the time required for training, helping to meet project deadlines more easily. Instead of updating all the model parameters, PEFT only adjusts a small subset of them [22]. This approach significantly reduces the computational load, making the process faster and more affordable.

There are numerous PEFT techniques out of which two popular techniques are LoRA (Low Rank Adaptation) and QLoRA (Quantized Low-Rank Adaptation). LoRA focuses on reducing the number of parameters that need to be updated during fine-tuning by decomposing weight updates into low-rank matrices. On the other hand, QLoRA further enhances this approach by considering the sparsity of the weight matrices, leading to even more efficient fine-tuning with fewer computations. QLoRA is considered better than LoRA because it achieves similar performance with even fewer parameters, making it more resource-efficient. In this work, the LLAM2 model is fine-tuned with QLoRA, due to its superior efficiency and performance in parameter-efficient fine-tuning tasks.

The block diagram in figure 3.5 illustrates the steps and techniques used to fine-tune the pretrained Llama 2 model, which has 7 billion parameters. It primarily outlines the steps for fine tuning the Large Language Model (LLM) to create an application, utilizing other Python libraries. The fine tuning of the pretrained Llama 2 model is implemented using QLoRA which is a combination of Low Rank Adaptation and 4bit Quantization. The other part of the block diagram depicts an application which works on this fine tuned Llama 2 model. The application receives a resume (in PDF or DOCX format), which is converted to text using the pdfminer.six library. This unstructured text is then inputted into the fine tuned Llama 2 model, which extracts the required entities and outputs them in a JSON-like format string. This structured string is converted into an actual JSON, and the extracted values are automatically mapped onto a new standard resume template using the JSON keys.

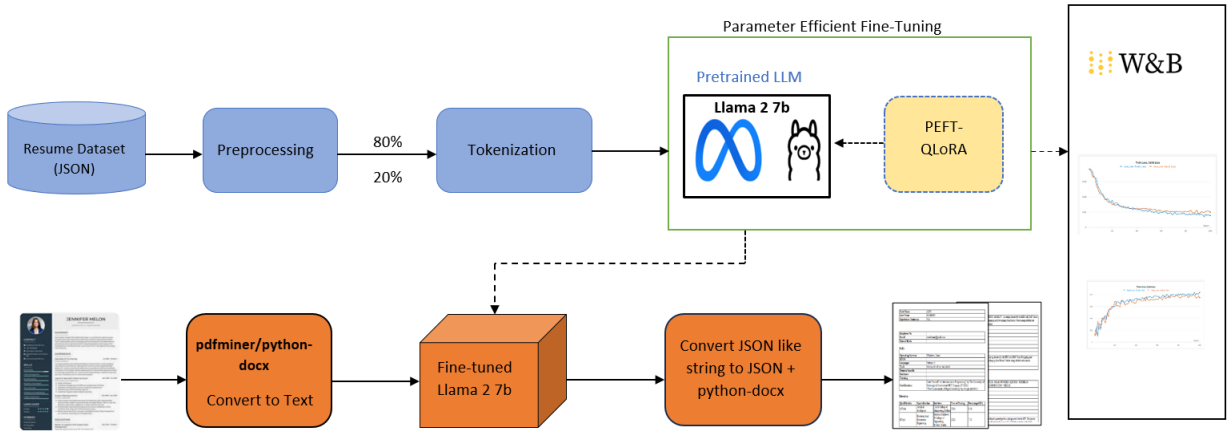


Figure 3.5: Block Diagram of Proposed Methodology

3.3.1 Dataset

The dataset for finetuning the LLM Llama 2 was generated using GPT 3.5 Turbo model, which is a powerful LLM by OpenAI. Using few shot prompt engineering (along with the prompt describing how the output has to be generated some examples of input and output were also given for better accuracy). One fifty unstructured resumes were converted to structured output with only the required information through API calling of the LLM. Thus, with the help of GenAI a custom synthetic dataset was generated. This dataset was manually evaluated for errors. With the same model using prompt engineering, the dataset was converted to a JSONL format that was compatible with the template suitable for finetuning. Each data consisted of the input_text which was the unstructured raw resume text and output_json which consisted of the required information in structured format. The GPT model was specified through prompt not to give any misinformation or generate output based on context outside the given resume input thereby making sure the generated json like output was strictly based on the resume input.

Given below is an example of a data of the created dataset for finetuning the Llama 2 model:
Input data:
{"input_text": "Ann B Merin A first year Artificial Intelligence Post Graduate student, currently looking for an internship opportunity to gain hands-on experience to develop my skills and contribute to real-world AI projects. merinbann@gmail.com TECHNICAL SKILLS C programing Python AutoCAD OS: Linux , Windows EDUCATION Mtech, Artificial Intelligence (2022-2024) TKM College of Engineering, Kollam SGPA: 8.36 Btech, Electrical and Electronics Engineering - 2022 Baselios Mathews II College of Engineering, Kollam, Kerala CGPA: 7.53 Higher Secondary (12th) - 2017 Infant Jesus Anglo Indian Higher Secondary School (ISC), Kollam, Kerala 68 per cent High School (10th) - 2015 Infant Jesus Anglo Indian Higher Secondary School (ICSE), Kollam, Kerala 86.3 per cent Microsoft Office SOFT

SKILLS Problem Solving Adaptability and Flexibility Interpersonal skills LANGUAGES English ..... Malayalam ..... Hindi ..... INTERNSHIP UNITED ELECTRICALS INDUSTRIES LTD. Electrical Equipments Manufacturing Company PROJECTS WESTIN MOBILITY A startup initiated by the IEDC cell, BMC Tasks Designing and Developing of an Electric Vehicle using refurbished material TRITAN: SOLAR POWERED ELECTRIC TRIPHIBIAN TRANSPORTATION VEHICLE Btech Main Project Tasks Developed a prototype for a solar powered electric GEV Designed a mini charging station using 12V 12W Polycrystalline Solar Panel Developed an MPPT charge controller IMPROVED ELECTRIC DIFFERENTIAL SYSTEM FOR INDEPENDENT SPEED CONTROL OF BLDC DRIVEN ELECTRIC VEHICLES Seminar Project MULTIPURPOSE SENSOR Mini design project INTERESTS Poetry Writing Books Observing CERTIFICATES Code Yourself! An Introduction to Programming by The University of Edinburgh Universidad ORT Uruguay (07/2021) Online Course The Fundamentals of Digital Marketing by Google (06/2021) Online Course WORKSHOPS RoboTRX A Robotics Hands on Workshop organised by IEEE Professional Activities Committee, Kerala Session Robotics and Automation One day workshop which included Line Follower Competition (Path Finder) Computer based design of Electrical System One day workshop which included introduction to CAD, ECAD and PCB design EXTRA CIRICULLARS Member of Innovation and Entrepreneurship Development Centre (IEDC), BMC (2019 - 2002) General Body Memeber and Treasurer (2021 - 2022) Electrical and Electronics Department Student Association, ELECZA, BMC THIRD PRIZE - RoboTRIX A 5 member team and created a voice controlled robot which can detect obstacles in path",

Output data:


```
"output_json": {
  "First name": "Ann",
  "Last name": "Merin",
  "Experience summary": "nil",
  "Date of birth": "nil",
  "Skills": {
    "Operating systems": ["Linux", "Windows"],
    "Programming languages": ["C", "Python"],
    "RTOS": "nil",
    "Tools": ["AutoCAD", "Microsoft Office"],
    "Hardwares": "nil",
    "Domain-specific skills": ["Problem Solving", "Adaptability and
       Flexibility", "Interpersonal skills"],
    "Training received": {
      "Certificates": [
          "Code Yourself! An Introduction to Programming by The University
             of Edinburgh & Universidad ORT Uruguay (07/2021)",
          "The Fundamentals of Digital Marketing by Google (06/2021)"
       ],
      "Workshops": [
          "RoboTRX: A Robotics Hands-on Workshop organised by IEEE
             Professional Activities Committee, Kerala",
```

```
        "Robotics and Automation: One-day workshop which included Line
           Follower Competition (Path Finder)",
        "Computer-based design of Electrical System: One-day workshop
           which included introduction to CAD, ECAD and PCB design"
      ]
    },
    "Certified courses": [
      "Code Yourself! An Introduction to Programming by The University of
          Edinburgh & Universidad ORT Uruguay (07/2021)",
      "The Fundamentals of Digital Marketing by Google (06/2021)"
    ]
  },
  "Educational details": [
    {
      "Qualification": "Mtech",
      "Specialization": "Artificial Intelligence",
      "Institution": "TKM College of Engineering, Kollam",
      "Year of passing": "2024",
      "Mark": "8.36"
    },
    {
      "Qualification": "Btech",
      "Specialization": "Electrical and Electronics Engineering",
      "Institution": "Baselios Mathews II College of Engineering, Kollam,
          Kerala",
      "Year of passing": "2022",
      "Mark": "7.53"
    },
    {
      "Qualification": "Higher Secondary (12th)",
      "Institution": "Infant Jesus Anglo Indian Higher Secondary School (
         ISC), Kollam, Kerala",
      "Year of passing": "2017",
      "Mark": "68 per cent"
    },
    {
      "Qualification": "High School (10th)",
      "Institution": "Infant Jesus Anglo Indian Higher Secondary School (
         ICSE), Kollam, Kerala",
      "Year of passing": "2015",
      "Mark": "86.3 per cent"
    }
  ],
  "Career profile": [
    {
      "Project name": "WESTIN MOBILITY",
```

```
        "Tools used for the project": "nil",
        "Work location": "nil",
        "Programming languages used for the project": "nil",
        "Project description": "Designing and Developing of an Electric
           Vehicle using refurbished material",
        "Responsibility": "nil"
      },
      {
        "Project name": "TRITAN: SOLAR POWERED ELECTRIC TRIPHIBIAN
           TRANSPORTATION VEHICLE",
        "Tools used for the project": "nil",
        "Work location": "nil",
        "Programming languages used for the project": "nil",
        "Project description": "Developed a prototype for a solar-powered
           electric GEV. Designed a mini charging station using 12V 12W
           Polycrystalline Solar Panel. Developed an MPPT charge
           controller",
        "Responsibility": "nil"
      },
      {
        "Project name": "IMPROVED ELECTRIC DIFFERENTIAL SYSTEM FOR
           INDEPENDENT SPEED CONTROL OF BLDC DRIVEN ELECTRIC VEHICLES",
        "Tools used for the project": "nil",
        "Work location": "nil",
        "Programming languages used for the project": "nil",
        "Project description": "Seminar Project",
        "Responsibility": "nil"
      },
      {
        "Project name": "MULTIPURPOSE SENSOR",
        "Tools used for the project": "nil",
        "Work location": "nil",
        "Programming languages used for the project": "nil",
        "Project description": "Mini design project",
        "Responsibility": "nil"
      }
    ]
  }
}
```


The dataset was split into training and validation set in the ratio 4:1. The dataset was not further increased for accuracy as this resulted in requiring more GPUs, especially when the largest single data had a maximum token length around 1700. Inorder to sustain using only the Kaggle Notebook provided free GPU, a tradeoff of context length and data size had to be made.

3.3.2 Preprocessing

The dataset further undergo a preprocessing stage where the data is again rephrased to make it compatible for finetuning with the Llama 2 model. The final format of dataset has following template given below:
"###Prompt: From the given input extract following entities as shown in the examples in json format. For every output the format of output and the entities extracted must be same. If any information not in the given input, leave it as null. ### Input: example['input_text']### Output: example['output_json']"

3.3.3 Tokenization

The default tokenizer of Llama 2 model is SentencePiece Tokenizer. SentencePiece is an unsupervised tool designed for tokenizing and detokenizing text, primarily used in Neural Network-based text generation systems where the vocabulary size is set before the model is trained. It incorporates subword units, such as byte-pair encoding (BPE) and the unigram language model, with the ability to train directly from raw text. This approach enables the creation of a completely end-to-end system without relying on language-specific preprocessing or postprocessing steps.
It is language-independent, processing sentences as sequences of Unicode characters without any language-specific logic. SentencePiece supports multiple subword algorithms, including byte-pair encoding (BPE) and the unigram language model, and implements subword sampling for subword regularization and BPE-dropout to enhance the robustness and accuracy of neural machine translation (NMT) models. It is fast and lightweight, with a segmentation speed of approximately 50,000 sentences per second and a memory footprint of around 6MB. The tool is self-contained, ensuring consistent tokenization and detokenization as long as the same model file is used. Additionally, SentencePiece manages vocabulary-to-ID mapping and can directly generate vocabulary ID sequences from raw sentences while performing NFKC-based text normalization.

3.3.4 Parameter Efficient Fine-Tuning of Llama 2

Parameter-efficient fine-tuning of the LLaMA 2 model focuses on adapting pre-trained models to new tasks with minimal changes to their parameters. This work aims to explore methods for fine-tuning very large language models like LLaMA 2, which have billions of parameters, using limited resources. By employing parameter-efficient fine-tuning (PEFT) techniques and model quantization, it is possible to fine-tune these models even with the limited free GPUs available on platforms like Kaggle and Google Colab. Among the various PEFT methods, Low Rank Adaptation (QLoRA) is utilized in this work as it is the most efficient compared to LoRa due to its quantization ability.

LoRA represents a significant advancement in the field of large language models.LoRA has helped in making fine tuning of LLMs with billions of parameters accessible and available to everyone rather than just wealthy companies. Its approach to low-rank adaptation has reduced the trainable parameters by up to 10,000 times, resulting in a three-fold reduction in GPU requirements and comparable or even improved performance, even without fine-tuning the entire model as proven with this work. Fine-tuning traditionally involves modifying a

pre-trained model's weights using new examples, requiring a matrix of the same size. However, by applying the concept of rank factorization, this matrix can be split into two smaller matrices that approximate the original when multiplied together. This is the basic idea of Low Rank Adapatation.

For quantization, the model uses 4-bit precision, the minimum quantization available. Quantization compresses deep learning models by reducing the number of bits used to represent their weights and activations. This technique allows for faster inference and reduced memory consumption, making it feasible to deploy these models on edge devices with limited resources.

The custom generated dataset is fine-tuned on Llama 2 model with the help of QLoRA (a combination of LoRA and 4bit quantization).The details of the hyperparameters used are given in tables 3.1 and 3.2 which are explained in detail.

QLoRA Parameters			
LoRA attention dimension		Bitsandbytes Parameters	
No.	**Parameter**	**No.**	**Parameter**
1	Attention Dimension = 64	1	use 4bit = True
2	Alpha Parameter = 16	2	bnb 4bit compute dtype = float16
3	Dropout Probability = 0.1	3	bnb 4bit quant type = nf4
		4	use_nested_quant = False

Table 3.1: QLoRA Parameters

lora_r: This parameter specifies the attention dimension for LoRA layers. Attention dimension determines the size of the hidden state vectors used in the attention mechanism of the LoRA layers. A higher value generally allows the model to capture more complex patterns but requires more computational resources.

lora_alpha: The alpha parameter is used for scaling in the LoRA layers. It controls the range of values that the model can output. A lower value of alpha can lead to more conservative outputs, while a higher value can result in more extreme predictions.

lora_dropout: Dropout is a regularization technique used in neural networks to prevent overfitting. This parameter specifies the probability that a neuron will be randomly dropped out during training. A dropout value of 0.1 means that each neuron has a 10% chance of being dropped out during training, helping to improve the generalization of the model.

use_4bit = True: This parameter activates the loading of base models with 4-bit precision. 4-bit precision allows for more efficient storage and computation compared to higher precision formats like 32-bit floating-point numbers. This can be particularly useful for deploying models on resource-constrained devices.

bnb_4bit_compute_dtype = "float16": Specifies the compute data type for 4-bit base models. In this case, it is set to "float16," which means that computations involving the 4-bit quan-

tized weights and activations will be performed using 16-bit floating-point numbers. This compromise between precision and efficiency is common in deep learning models.

bnb_4bit_quant_type = "nf4": Specifies the quantization type for 4-bit base models. "nf4" likely stands for "nested float," indicating a nested quantization scheme where 4-bit quantized values are interpreted as floating-point numbers with a specific scaling factor. This type of quantization can further reduce the memory footprint and computational requirements of the model.

use_nested_quant = False: This parameter controls whether nested quantization is activated for 4-bit base models. Nested quantization refers to a technique where quantization is applied recursively to further reduce the precision of weights and activations. However, in this case, it is set to False, indicating that nested quantization is not used.

Table 3.2 includes essential settings such as the number of training epochs, batch sizes for training and evaluation, gradient accumulation steps, and the use of gradient checkpointing. Additionally, it specifies the optimizer (paged_adamw_32bit), learning rate schedule (cosine), and other optimization-related parameters like maximum gradient norm, initial learning rate, and weight decay. The table also provides details on the model's configuration, such as the maximum sequence length, whether to group sequences by length for efficiency, and the device mapping for model loading. Each parameter is paired with its corresponding value, offering a clear and organized reference for configuring the model training process.

No.	Parameter	Value
1	Number of training epochs	10
2	Enable fp16/bf16 training	False (both fp16 and bf16 are False)
3	Batch size per GPU (training)	2
4	Batch size per GPU (evaluation)	1
5	Gradient accumulation steps	1
6	Enable gradient checkpointing	True
7	Maximum gradient norm	0.3
8	Initial learning rate	2e-4
9	Weight decay	0.001
10	Optimizer	paged_adamw_32bit
11	Learning rate schedule	cosine
12	Maximum training steps	-1 (not set, so num_train_epochs used)
13	Warmup ratio	0.03
14	Group sequences by length	True
15	Save checkpoint every X steps	0 (not set, so checkpoints not saved)
16	Log every X steps	1
17	Maximum sequence length	2000
18	Packing	False
19	Device map	{"": 0} (load entire model on GPU 0)

Table 3.2: Model Finetuning Hyperparameters

3.3.5 Evaluation Metrics

amsmath The evaluation of the fine-tuned model includes the F1 score, precision, and recall, providing a comprehensive assessment of its performance.These metrics include precision, recall, and the F1 score, all of which can be calculated using the counts of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). These metrics are crucial in assessing the model's ability to correctly classify instances across different classes.

Precision represents the proportion of true positive predictions among all positive predictions, indicating the model's ability to avoid false positives. The equation for the Precision can be expressed as:

$$P = \frac{TP}{TP + FP} \tag{3.1}$$

Where,
TP stands for True Positive and FP stands for False Positive

Recall, on the other hand, measures the proportion of true positive predictions among all actual positive instances, highlighting the model's ability to capture all positive instances. The equation for the recall can be expressed as:

$$R = \frac{TP}{TP + FN} \tag{3.2}$$

Where,
TP stands for True Positive, FN stands for False Negative and TP stands for True Positive

The **F1 score**, which is the harmonic mean of precision and recall, offers a balanced measure of the model's accuracy, particularly useful when the class distribution is imbalanced.The equation for F1 Score can be expressed as:

$$F1 = 2 \times \frac{P \times R}{P + R} \tag{3.3}$$

Where,
P is Precision and R is Recall

In addition to these evaluation measures, the loss curve provides valuable insights into the model's training process and convergence. By visualizing how the loss decreases over training epochs, analysts can assess the model's learning dynamics and identify any potential issues such as overfitting or underfitting. To facilitate the visualization of this metrics, the model evaluation utilizes W&B (Weights and Biases), a platform that enables efficient tracking and visualization of machine learning and deep learning experiments. W&B offers a comprehensive suite of tools for visualizing model performance, enabling researchers and practitioners to gain deeper insights into the fine-tuned model's behavior and performance.

The **loss curve**, representing the loss with respect to training steps, monitors and evaluates the fine tuning process of the LLM Llama 2. It is visualized as a plot with the training

steps on the x-axis and the corresponding loss values on the y-axis. The equation for the loss curve can be expressed as:

$$\text{Loss}(t) = \mathcal{L}(t) \tag{3.4}$$

Where:
- Loss(t) is the loss at training step t,
- $\mathcal{L}(t)$ is the loss function value at training step t.

The loss function $\mathcal{L}$ typically represents the discrepancy between the predicted outputs of the model and the actual ground truth labels. As the model learns from the training data, the loss decreases, indicating that the model is improving its predictions. The loss curve provides valuable insights into the training dynamics, including information about the model's convergence, overfitting, and underfitting. By analyzing the loss curve, practitioners can make informed decisions about model tuning and optimization strategies to improve the model's performance.

3.4 Architecture of Llama-2

For finetuning to get structured required information from resume, Llama 2 model by Meta is used. Its large context length, crucial for this use case, and performance comparable to GPT models made it a favorable choice. Additionally, its accessibility as a free model added to the appeal. It is a text-generation large language model which currently outperforms every open source alternative. For finetuning the model using QLoRA the basic model of Llama-2 model consisting of 7 billion parameters was used. Although the 70 billion parameter Llama-2 model gave the best result while instruction based fewshot prompting in Phase-1 of the dissertation, the 7 base model was used to have the smallest model. The final evaluation proved that, the 7 billion parametr model after finetuning gave better result than the instruction tuned 70 billion parameter largest model of Llama-2.

In comparison to the Llama 1 model, Llama 2 models are trained on a dataset of 2 trillion tokens, which is 40% larger, and they have a context length of 4096, which is twice the context length of Llama 1.The model, was privately hosted, underwent a quantization process using the llama cpp library to address its weighty nature. This quantization approach aimed at reducing the model's weight to 4 bits without compromising accuracy, thereby reducing the computational burden. Particularly noteworthy is the availability of the 13B and 7B versions, with the former being more powerful albeit requiring compression, achieved through quantization or reduced float precision, to accommodate mid-range GPUs efficiently. The implementation of the llama cpp library played a pivotal role in this optimization, offering a streamlined solution for compression without sacrificing the model's ability in information extraction tasks.

Llama-2 is a decoder only text generation model. To facilitate the handling of 2 trillion tokens and internal weights, Meta employs Root Mean Square Normalization (RMSNorm). Meta has opted for the SwiGLU activation function to determine the activation status of

each neurons. LLaMA-2 incorporates a novel mathematical approach known as RoPE, or "Rotary Positional Embedding" to make the model understand the importance of position of words within sentences. These factors make Llama 2 more efficiant and powerful compared to its counterparts.

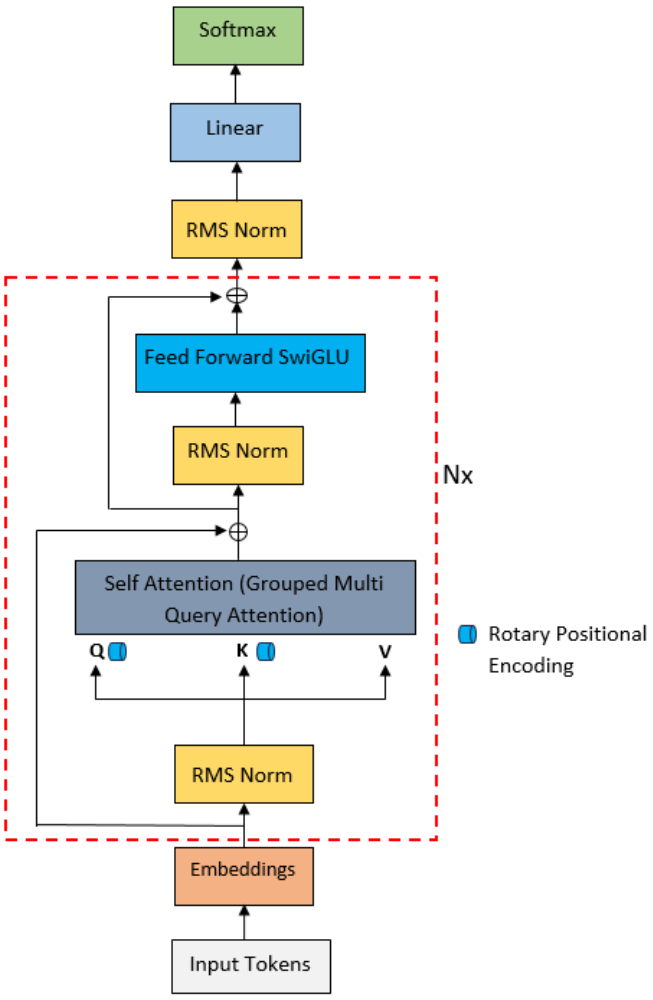


Figure 3.6: Architecture of Llama-2 Model

Figure 3.6 shows the architecture of the LLaMA-2 model. When llama 2 is provide with the training data that includes a prompt, input text and output, it is first tokenized using the sentense piece tokenizer. The model then converts these tokens into numerical representations known as embeddings. These embeddings capture the semantic meaning of the words in the prompt. After the embedding step, the vectors are normalized using Root Mean Square (RMS) normalization. This normalization step scales the embeddings to have a consistent magnitude, improving training stability and convergence. The rotary positional

encodings are added to the token embeddings after RMS normalization to retain information about the position of each token in the sequence.These vectors are transformed to query, key and value vectors. The self-attention mechanism allows the model to weigh the importance of different tokens in the input sequence relative to each other. This mechanism helps the model understand the relationships between words and to understand the context better. In Llama 2, a grouped multi-query attention layer is used. This approach modifies the standard multi-head attention by grouping multiple attention heads to share the same query and key matrices but have separate value matrices. This helps in reducing the number of parameters and computational complexity while still maintaining the ability to capture diverse aspects of the context. The attention scores are calculated using the query and key vectors. For each token, the attention score determines how much focus it should place on other tokens in the sequence. It is passed again to a RMS normalisation and then to a feed forward network with SwiGLU activation function. This is for adding nonlinearity to the vectors. The residual connections can be seen before every RMS normalization. This is to ensure no vanishing gradient issues occur to the parameters while training. The key changes when fine-tuning with Qlora involve adapting the model's output layer for the specific task, optimizing the learning process for parameter-efficient fine-tuning, and applying appropriate regularization and monitoring techniques to ensure optimal performance.As a decoder-only model, Llama 2 generates text one token at a time. It uses the context provided by the prompt and the previously generated tokens to predict the next token. This process continues iteratively until the model generates a complete response or reaches a predefined stopping criterion. The output tokens are then converted back into human-readable text. The generated text is influenced by the prompt and the model's learned patterns from the training data.

- **RMS Normalization:** RMSNorm, or Root Mean Square Layer Normalization, is a technique used in LLaMA to normalize the inputs of each transformer sub-layer. Unlike traditional methods that normalize the output, LLaMA takes inspiration from GPT-3 and applies normalization to the input stage. This approach is known as pre-normalization and is aimed at improving the overall efficiency and performance of the model.

  RMSNorm is an advanced variation of Layer Normalization (LayerNorm). The primary reason for adopting RMSNorm over LayerNorm is to address the computational burden associated with LayerNorm. In many cases, the computational demands of LayerNorm can slow down the training process and make it more costly. RMSNorm alleviates these issues by offering a more computationally efficient alternative, thus facilitating faster and more cost-effective improvements.

  It can be expressed as:

$$\bar{a}_i = \frac{a_i}{\text{RMS}(\mathbf{a})} g_i, \quad \text{where} \quad \text{RMS}(\mathbf{a}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n} a_i^2} \tag{3.5}$$

  Where:

  - a_i: activation of the ith neuron
  - $g \in R^n$: the gain parameter used to re-scale the standardized summed inputs

- **SwiGLU:** SwiGLU, or Swish-Gated Linear Unit, is a novel activation function employed in LLaMA 2, inspired by its use in the PaLM model. This function integrates the benefits of Swish and Gated Linear Unit (GLU) activation functions. By utilizing a gating mechanism, SwiGLU selectively activates neurons based on the input it receives, which helps reduce overfitting and enhances generalization capabilities. SwiGLU, is derived from the Swish activation function.

  The Swish activation function can be defined as follows:

  $$\text{swish}(x) = x \cdot \sigma(x) \tag{3.6}$$

  where $\sigma(x) = \frac{1}{1+e^{-x}}$
  $Here, \sigma(x)$ represents the sigmoid function.

  The SwiGLU activation function builds on this concept by applying a gating mechanism that controls the flow of information through neurons. This mechanism is inspired by the GLU, which is designed to allow only certain parts of the input to pass through, based on the gating signal. The Python implementation of SwiGLU demonstrates this concept, where the input x is split into two parts: the output and the gate. The gate undergoes the Swish activation, and then it modulates the output via element-wise multiplication. This process can be succinctly represented by the following equations:

  $$\text{SwiGLU}(x) = \text{out} \cdot \text{swish}(\text{gate}) \tag{3.7}$$

  where x = [out, gate]
  In this function, x is split into out and gate along the specified dimension, the gate is activated using Swish, and the resulting value modulates the output. This selective activation allows SwiGLU to improve the model's performance by enhancing its ability to generalize from training data to unseen data, while mitigating overfitting.

- **Rotary Position Embedding:** RoPE, or Rotary Position Embedding, is a sophisticated method of positional encoding that utilizes rotation matrices to embed absolute positional information while naturally integrating explicit relative positional dependencies into the self-attention mechanism.

  Advantages of RoPE

  - **Scalability**: RoPE can be extended to accommodate any sequence length, making it highly versatile for various applications.
  - **Decaying Dependency**: As the relative distances between tokens increase, RoPE ensures that their inter-dependency diminishes, which aligns well with the characteristics of natural language processing.
  - **Relative Position Encoding**: RoPE effectively equips linear self-attention mechanisms with the ability to encode relative positions, enhancing the model's understanding of context.

  The fundamental concept behind RoPE is to encode relative positional information by multiplying the context representations with a rotation matrix. This approach ensures that the influence of token positions decays with increasing relative distance, a desirable

property for natural language encoding. The inspiration for incorporating RoPE into LLaMA comes from its successful application in GPTNeo.

RoPE introduces a novel perspective by rotating word vectors instead of adding positional vectors. For example, consider a two-dimensional word vector for "dog." To encode its position in a sentence, RoPE rotates this vector by an angle proportional to the word's position. For the first position, the vector is rotated by θ, for the second position by 2θ, and so forth. This rotational method offers several benefits:

1. **Stability of Vectors**: Adding tokens at the end of a sentence does not impact the vectors of words at the beginning, facilitating efficient caching and processing.
2. **Preservation of Relative Positions**: If two words, such as "pig" and "dog," maintain the same relative distance in different contexts, their vectors are rotated by the same amount. This consistency ensures that the angle, and therefore the dot product between these vectors, remains constant across contexts.

RoPE's innovative use of rotation for position encoding significantly enhances the model's ability to understand and maintain contextual relationships, providing a robust framework for natural language processing tasks.

Chapter 4

Results and Discussion

Parameter efficient fine tuning techniques were used for finetuning, to get the required output structure from the Llama 2 model. The result was compared with the dissertation phase-1 results. Apart from that the training and validation testing loss curve was also found. This section introduces the experimental setup along with the findings and discussions.

4.1 Experimental Setup

In order to conduct the programming tasks efficiently, Kaggle Notebook was used, a free Jupyter Notebook which is in a cloud server that can integrate both GPU and TPU. leveraged its GPU resources for accelerated Python programming. GPU P100 was used for fine tuning the Llama model.

4.2 Results and Analysis

4.2.1 Deployment

A user interface was created using Streamlit to deploy the resume parsing application. Users can upload their resumes in PDF or document format, and the application automatically generates a standard template resume with the required entities using the fine-tuned Llama 2 model.

Streamlit is an open-source Python library designed for creating interactive web applications. It enables developers to build custom web apps quickly and efficiently using only Python scripts, eliminating the need for extensive front-end development skills. With Streamlit, data scripts can be transformed into shareable web applications, facilitating easy interaction and visualization.
One of the standout features of Streamlit is its simplicity. The API is straightforward, allowing developers to create interactive widgets such as sliders, buttons, and text inputs with minimal code. This ease of use makes it accessible for developers at various skill levels to build functional and interactive applications.
Another significant advantage of Streamlit is its ability to handle real-time updates. Applications can automatically refresh as the underlying data changes, making it ideal for dynamic data visualization and interactive data exploration. This feature ensures that users always

have access to the most current data without needing to manually refresh the application. Streamlit also integrates seamlessly with popular data science libraries like Pandas, NumPy and Matplotlib. This compatibility allows for comprehensive data manipulation and visualization within the application, making it a powerful tool for data scientists who need to analyze and present data efficiently.
Deployment of Streamlit applications is straightforward, allowing for easy sharing and collaboration. This makes Streamlit an excellent tool for presenting data science projects to a wider audience, including stakeholders who may not have a technical background.
Customization is another key feature of Streamlit, as it allows developers to tailor the appearance and functionality of their applications to meet specific project needs. This versatility makes Streamlit suitable for a wide range of applications, from simple dashboards to complex interfaces for machine learning models. It provides a powerful and accessible platform for creating and sharing interactive applications, enhancing the ability of data scientists and machine learning practitioners to communicate insights and results effectively.

The application screenshots are given below, illustrating the various stages of the resume parsing process using the Streamlit interface.

- The figure 4.1 shows the Home page of the application. It includes a tab to upload the resume. It consist of a button named "Browse files" that allows users to upload their resumes from their system.

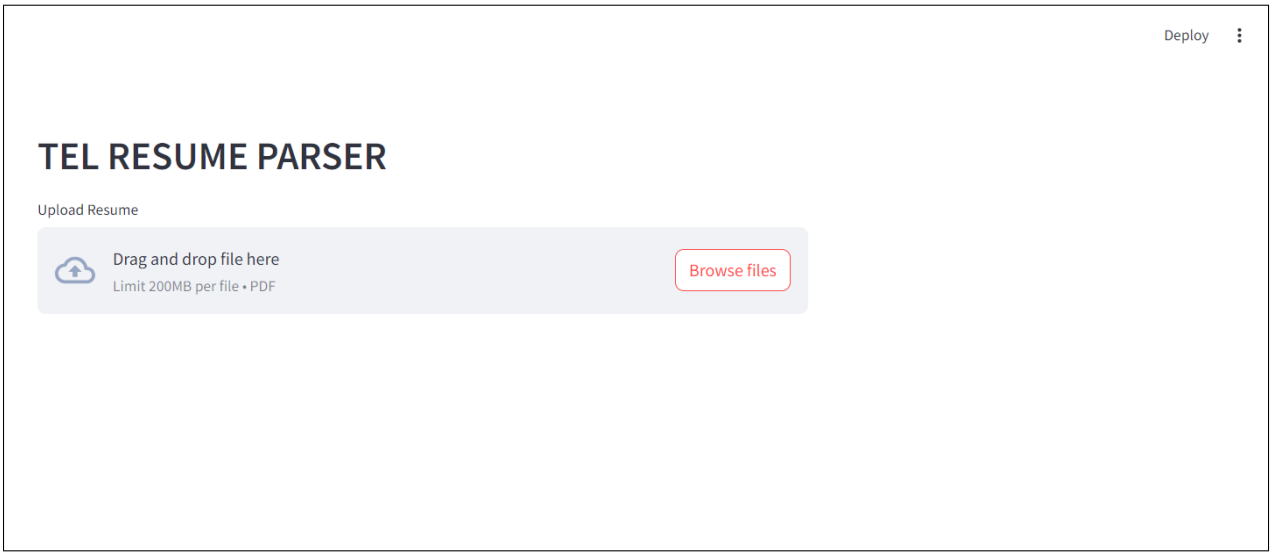


Figure 4.1: Home Page

- The figure 4.2 demonstrates the process of uploading a resume from the system. Using "Browse files" the user can upload the resume which can be a pdf or docx file. It can also be a plain text file. The maximum size of the file that can be uploaded is 200 MB. If the uploaded file cannot be supported an error message asking the user to upload the right file type is displayed.

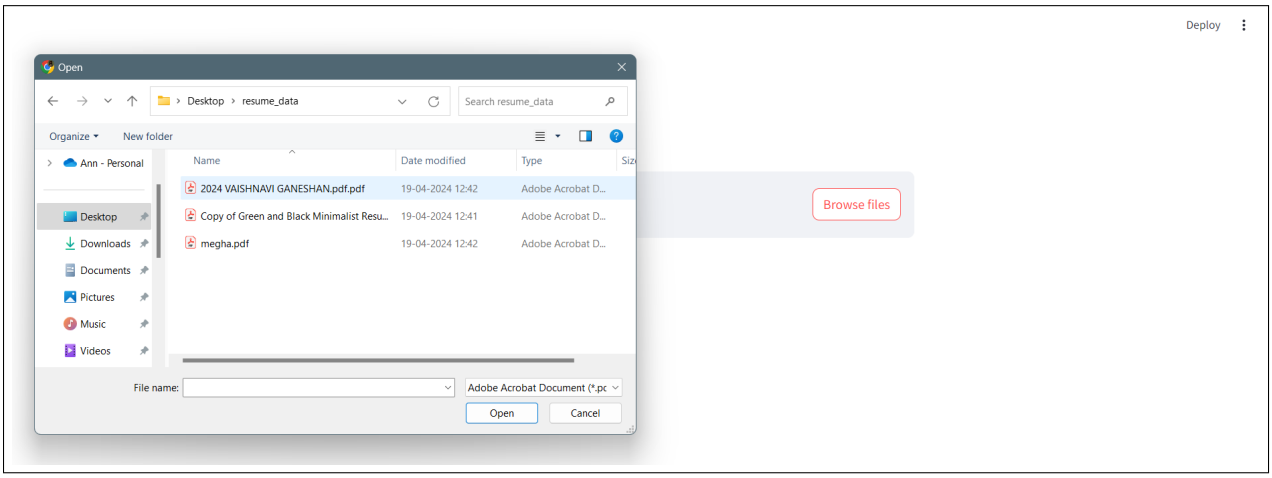


Figure 4.2: Uploading the Resume

- The figure 4.3 displays the application in operation, generating the output based on the uploaded resume. The uploaded resume file name and size is displayed on the page of the application as shown in figure 4.3. On the top right corner of the page, the "Running" indicates the application is running and parsing the uploaded resume to generate the new resume with only the required information from the uploaded resume of the user.

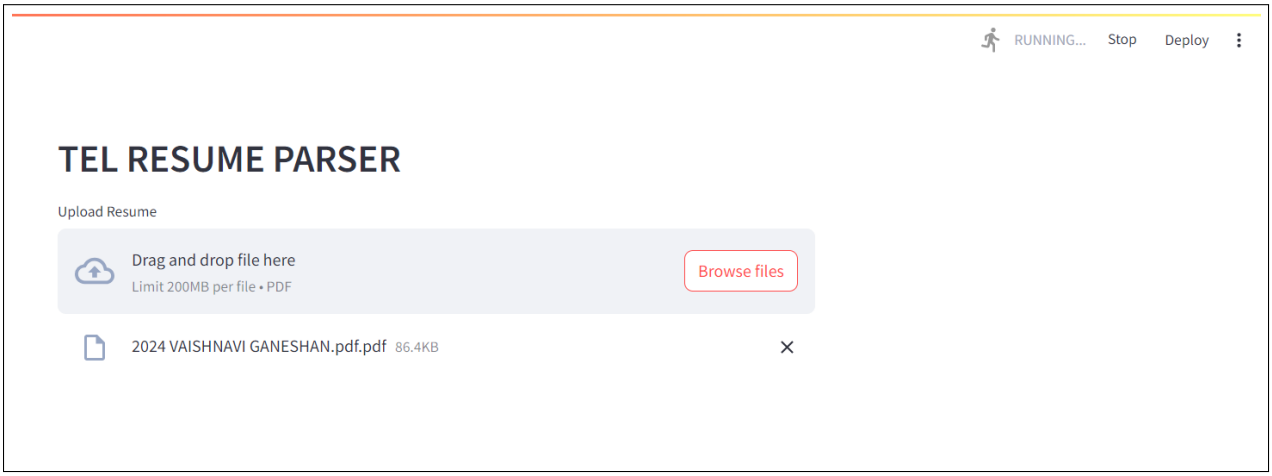


Figure 4.3: Automated Parsing With Uploaded Resume

- The figure 4.4 presents the output resume that has been generated. Users can download this generated resume to their system by pressing the "Download Resume" button. Using the fine tuned llama 2 model the required entities have been extracted and converted to a structured template in docx format which is output to the user.

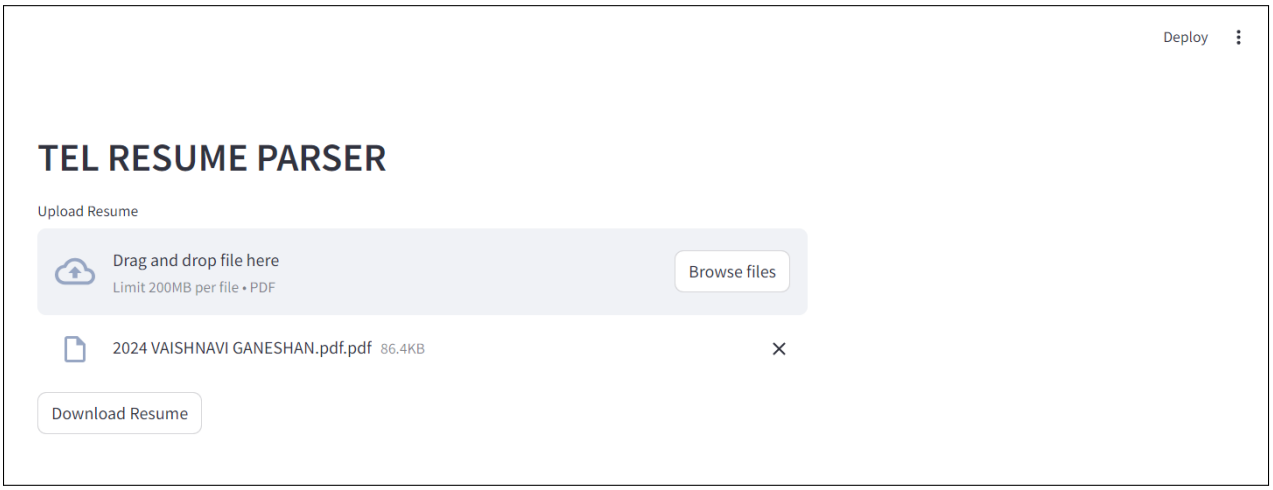


Figure 4.4: Generated Resume Ready For Download

- When the user clicks on the "Download Resume" button, the application is again shown as running. Here the resume generated by the created system is downloaded to the user system. Figure 4.5 shows a screenshot of the application at this stage of downloading.

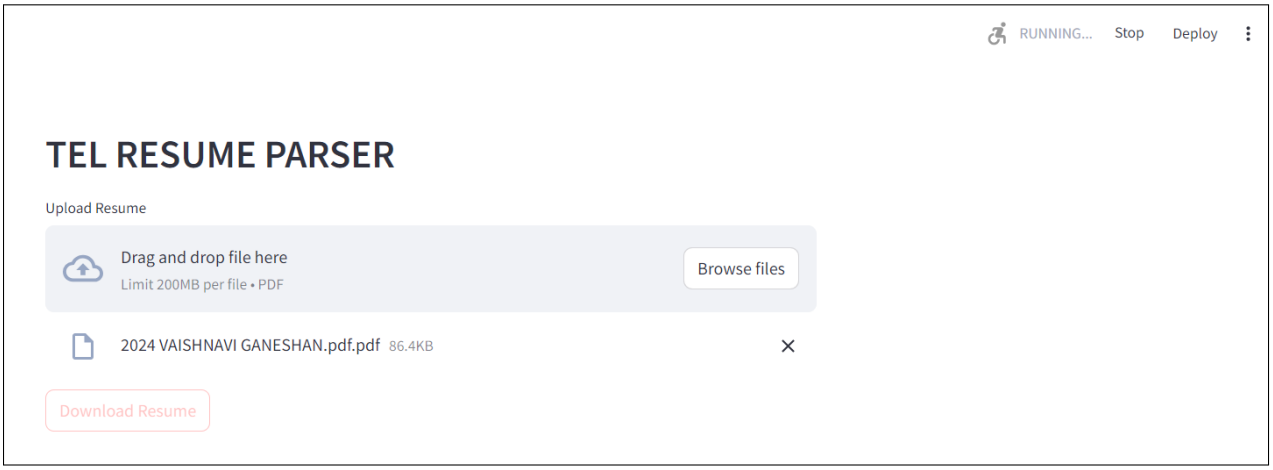
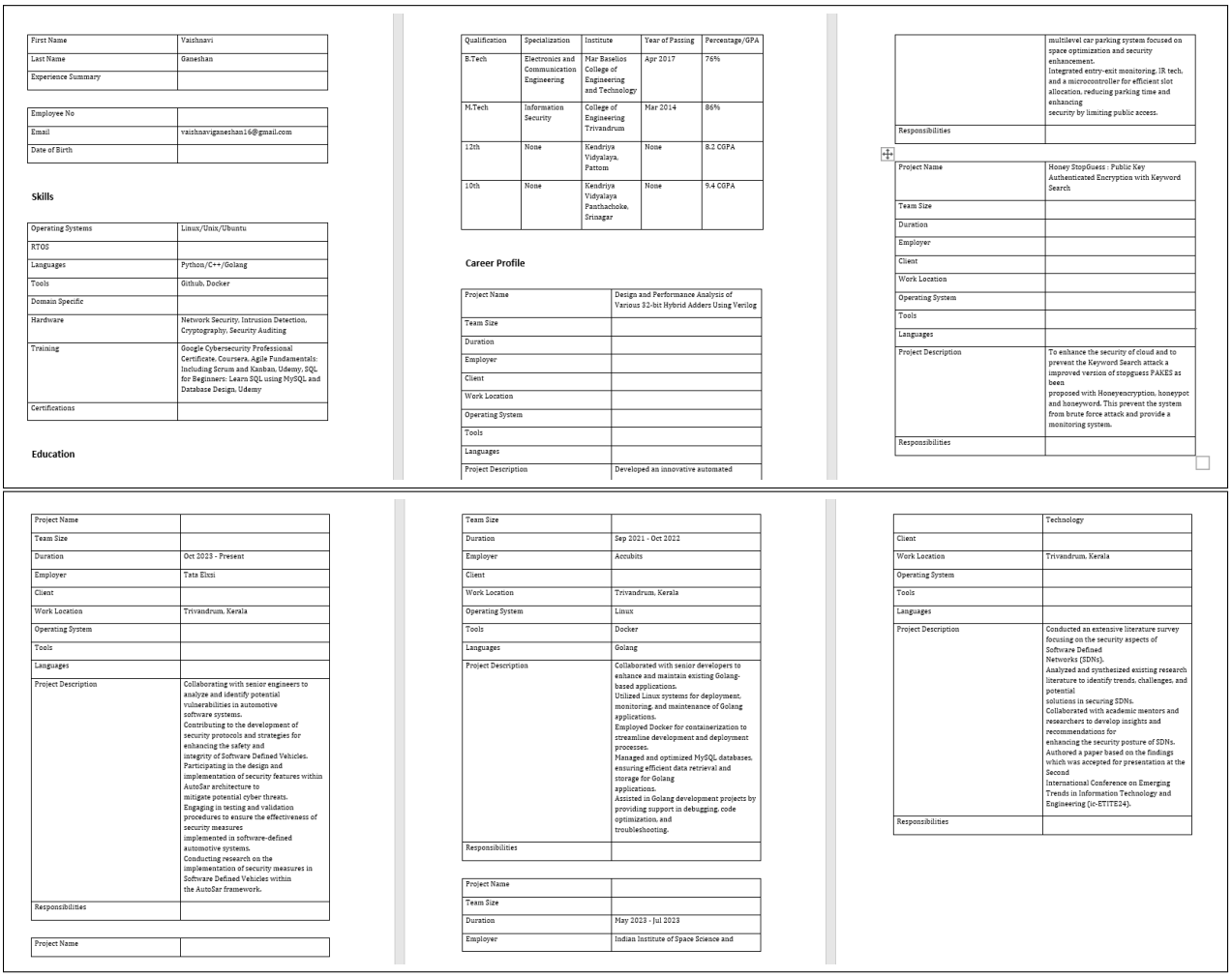


Figure 4.5: Downloading Resume Generated Using The Parser

- Figure 4.6 is the generated resume by the parser. It consists of all the required information needed (by the company) that could be extracted from the resume. If certain information are not provided in the original resume those rows are left blank as shown in the figure. For reference the original resume is shown in figure 4.7



First Name	Vaishnavi
Last Name	Ganeshan
Experience Summary	

Employee No	
Email	vaishnaviganeshan16@gmail.com
Date of Birth	

Skills

Operating Systems	Linux/Unix/Ubuntu
RTOS	
Languages	Python/C++/Golang
Tools	Github, Docker
Domain Specific	
Hardware	Network Security, Intrusion Detection, Cryptography, Security Auditing
Training	Google Cybersecurity Professional Certificate, Coursera, Agile Fundamentals: Including Scrum and Kanban, Udemy, SQL for Beginners: Learn SQL using MySQL and Database Design, Udemy
Certifications	

Education

Qualification	Specialization	Institute	Year of Passing	Percentage/GPA
B.Tech	Electronics and Communication Engineering	Mar Baselios College of Engineering and Technology	Apr 2017	76%
M.Tech	Information Security	College of Engineering Trivandrum	Mar 2014	86%
12th	None	Kendriya Vidyalaya, Pattom	None	8.2 CGPA
10th	None	Kendriya Vidyalaya Panthachoke, Srinagar	None	9.4 CGPA

Career Profile

Project Name	Design and Performance Analysis of Various 32-bit Hybrid Adders Using Verilog
Team Size	
Duration	
Employer	
Client	
Work Location	
Operating System	
Tools	
Languages	
Project Description	Developed an innovative automated multilevel car parking system focused on space optimization and security enhancement. Integrated entry-exit monitoring, IR tech, and a microcontroller for efficient slot allocation, reducing parking time and enhancing security by limiting public access.
Responsibilities	

Project Name	Honey StopGuess : Public Key Authenticated Encryption with Keyword Search
Team Size	
Duration	
Employer	
Client	
Work Location	
Operating System	
Tools	
Languages	
Project Description	To enhance the security of cloud and to prevent the Keyword Search attack a improved version of stopguess PAKES as been proposed with Honeyencryption, honeypot and honeyword. This prevent the system from brute force attack and provide a monitoring system.
Responsibilities	

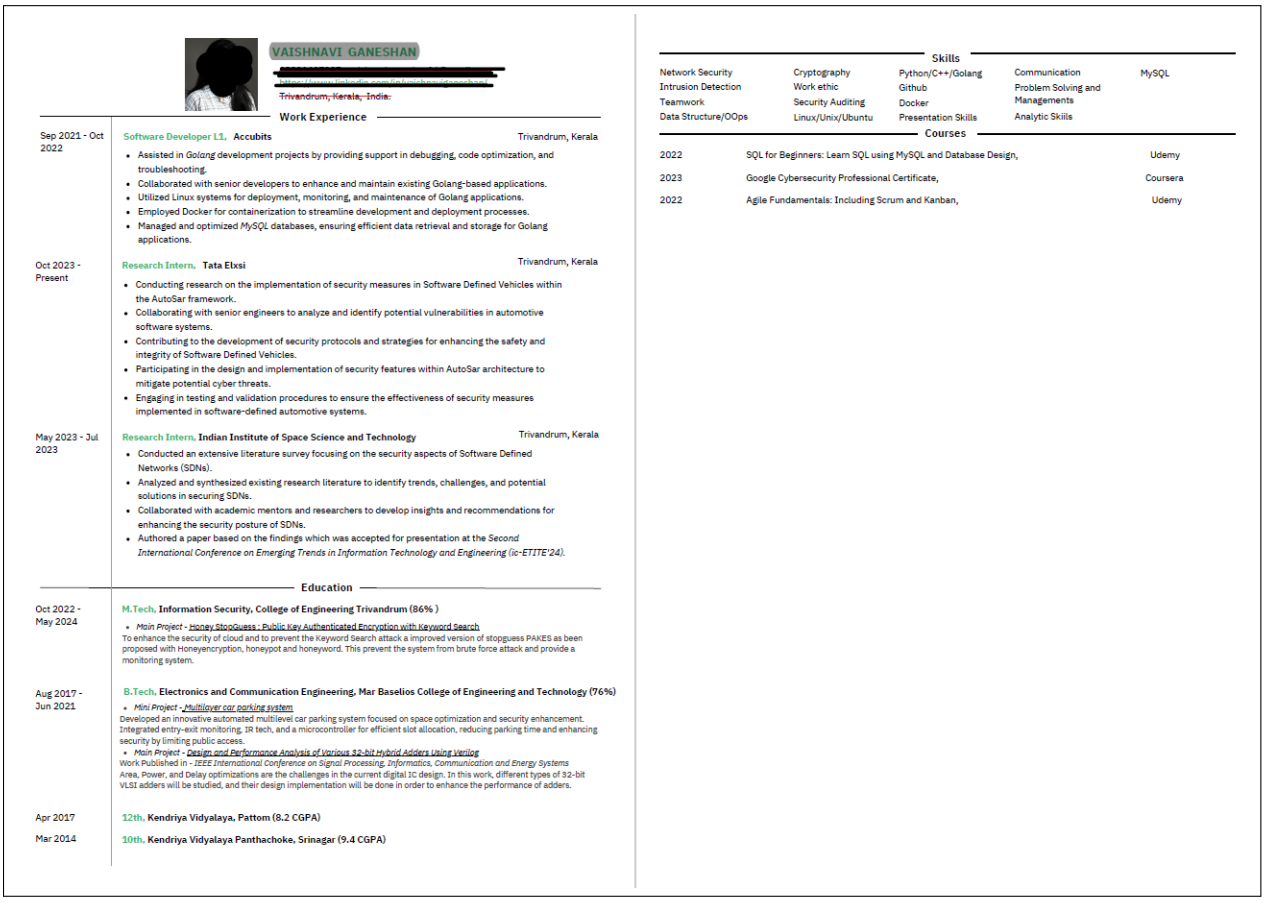
Project Name	
Team Size	
Duration	Oct 2023 - Present
Employer	Tata Elxsi
Client	
Work Location	Trivandrum, Kerala
Operating System	
Tools	
Languages	
Project Description	Collaborating with senior engineers to analyze and identify potential vulnerabilities in automotive software systems. Contributing to the development of security protocols and strategies for enhancing the safety and integrity of Software Defined Vehicles. Participating in the design and implementation of security features within AutoSar architecture to mitigate potential cyber threats. Engaging in testing and validation procedures to ensure the effectiveness of security measures implemented in software-defined automotive systems. Conducting research on the implementation of security measures in Software Defined Vehicles within the AutoSar framework.
Responsibilities	

Project Name	

Team Size	
Duration	Sep 2021 - Oct 2022
Employer	Accubits
Client	
Work Location	Trivandrum, Kerala
Operating System	Linux
Tools	Docker
Languages	Golang
Project Description	Collaborated with senior developers to enhance and maintain existing Golang-based applications. Utilized Linux systems for deployment, monitoring, and maintenance of Golang applications. Employed Docker for containerization to streamline development and deployment processes. Managed and optimized MySQL databases, ensuring efficient data retrieval and storage for Golang applications. Assisted in Golang development projects by providing support in debugging, code optimization, and troubleshooting.
Responsibilities	

Project Name	
Team Size	
Duration	May 2023 - Jul 2023
Employer	Indian Institute of Space Science and Technology
Client	
Work Location	Trivandrum, Kerala
Operating System	
Tools	
Languages	
Project Description	Conducted an extensive literature survey focusing on the security aspects of Software Defined Networks (SDNs). Analyzed and synthesized existing research literature to identify trends, challenges, and potential solutions in securing SDNs. Collaborated with academic mentors and researchers to develop insights and recommendations for enhancing the security posture of SDNs. Authored a paper based on the findings which was accepted for presentation at the Second International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE24).
Responsibilities	

Figure 4.6: Generated Resume by the Parser



VAISHNAVI GANESHAN

[redacted]

Trivandrum, Kerala, India.

Work Experience

Sep 2021 - Oct 2022 — **Software Developer L1,** **Accubits** — Trivandrum, Kerala

- Assisted in *Golang* development projects by providing support in debugging, code optimization, and troubleshooting.
- Collaborated with senior developers to enhance and maintain existing Golang-based applications.
- Utilized Linux systems for deployment, monitoring, and maintenance of Golang applications.
- Employed Docker for containerization to streamline development and deployment processes.
- Managed and optimized *MySQL* databases, ensuring efficient data retrieval and storage for Golang applications.

Oct 2023 - Present — **Research Intern,** **Tata Elxsi** — Trivandrum, Kerala

- Conducting research on the implementation of security measures in Software Defined Vehicles within the AutoSar framework.
- Collaborating with senior engineers to analyze and identify potential vulnerabilities in automotive software systems.
- Contributing to the development of security protocols and strategies for enhancing the safety and integrity of Software Defined Vehicles.
- Participating in the design and implementation of security features within AutoSar architecture to mitigate potential cyber threats.
- Engaging in testing and validation procedures to ensure the effectiveness of security measures implemented in software-defined automotive systems.

May 2023 - Jul 2023 — **Research Intern, Indian Institute of Space Science and Technology** — Trivandrum, Kerala

- Conducted an extensive literature survey focusing on the security aspects of Software Defined Networks (SDNs).
- Analyzed and synthesized existing research literature to identify trends, challenges, and potential solutions in securing SDNs.
- Collaborated with academic mentors and researchers to develop insights and recommendations for enhancing the security posture of SDNs.
- Authored a paper based on the findings which was accepted for presentation at the *Second International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE'24).*

Education

Oct 2022 - May 2024 — **M.Tech, Information Security, College of Engineering Trivandrum (86%)**

- *Main Project -* Honey StopGuess : Public Key Authenticated Encryption with Keyword Search

To enhance the security of cloud and to prevent the Keyword Search attack a improved version of stopguess PAKES as been proposed with Honeyencryption, honeypot and honeyword. This prevent the system from brute force attack and provide a monitoring system.

Aug 2017 - Jun 2021 — **B.Tech, Electronics and Communication Engineering, Mar Baselios College of Engineering and Technology (76%)**

- *Mini Project -* Multilayer car parking system

Developed an innovative automated multilevel car parking system focused on space optimization and security enhancement. Integrated entry-exit monitoring, IR tech, and a microcontroller for efficient slot allocation, reducing parking time and enhancing security by limiting public access.

- *Main Project -* Design and Performance Analysis of Various 32-bit Hybrid Adders Using Verilog

Work Published in - *IEEE International Conference on Signal Processing, Informatics, Communication and Energy Systems*

Area, Power, and Delay optimizations are the challenges in the current digital IC design. In this work, different types of 32-bit VLSI adders will be studied, and their design implementation will be done in order to enhance the performance of adders.

Apr 2017 — **12th, Kendriya Vidyalaya, Pattom (8.2 CGPA)**

Mar 2014 — **10th, Kendriya Vidyalaya Panthachoke, Srinagar (9.4 CGPA)**

Skills

Network Security	Cryptography	Python/C++/Golang	Communication	MySQL
Intrusion Detection	Work ethic	Github	Problem Solving and Managements	
Teamwork	Security Auditing	Docker		
Data Structure/OOps	Linux/Unix/Ubuntu	Presentation Skills	Analytic Skills	

Courses

2022	SQL for Beginners: Learn SQL using MySQL and Database Design,	Udemy
2023	Google Cybersecurity Professional Certificate,	Coursera
2022	Agile Fundamentals: Including Scrum and Kanban,	Udemy

Figure 4.7: Original Resume

4.2.2 Results

The results and analysis of the study reveal a comparison between the fine-tuned Llama 2 model using QLora and the instruction-based few-shot prompt Llama 2 model output from Phase 1 of dissertation. It was observed that the performance of the fine-tuned model was on par with the instruction-tuned model. The fine-tuned Llama 2 model, with 7 billion parameters, demonstrated performance highly comparable to that of the larger Llama 2 models post fine-tuning.

The evaluation was conducted using F1 score, precision, and recall metrics. Remarkably, the fine-tuned 7 billion parameter Llama 2 model achieved a higher F1 score compared to the instruction-tuned 13 billion parameter Llama 2 model. This comparison was based on an analysis of 20 collected resumes. The model had an F1 Score of 0.8928 compared to the few shot prompt Llama 2 70b model with a F1 score of 0.8888. The comparison of the F1 score, precision and recall of both models are given in table 4.1.

Apart from the F1 score metrics, the loss curve is also determined using the W&B website. W&B Models is the system of record for ML Practitioners who want to organize their models, boost productivity and collaboration, and deliver production ML at scale. Evaluation while training(fine-tuning) and the system can we visualised during W&B. With optimization the loss was reduced from the initial 0.6 to 0.116. The loss graph, Fig:4.8 and the other visual evaluations from W&B are given in Figure:4.9

Model	Precision	Recall	F1 Score
Fine-tuned Llama 2 7b	0.8719	0.9147	0.8928
Instruction-tuned Llama 2 70b	0.8942	0.8834	0.8888

Table 4.1: Comparison of Fine-tuned Llama 2 and Instruction-tuned Llama 2 Models

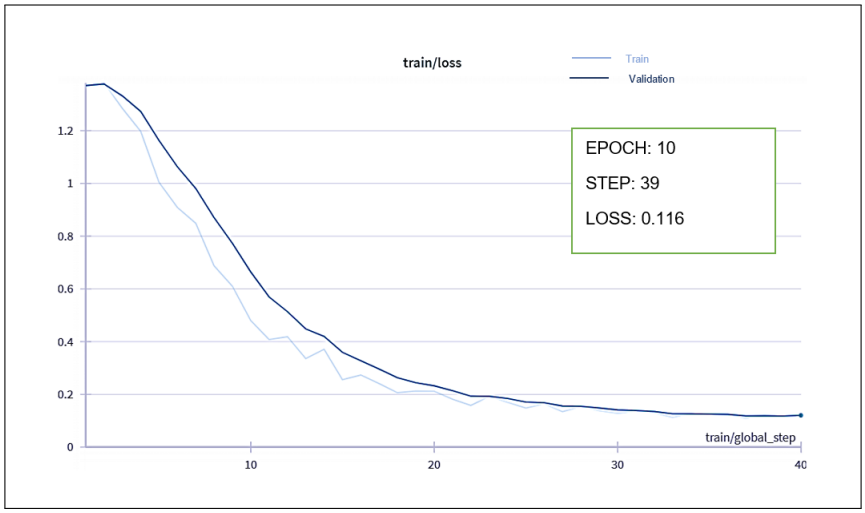


Figure 4.8: Training Steps vs Loss Graph

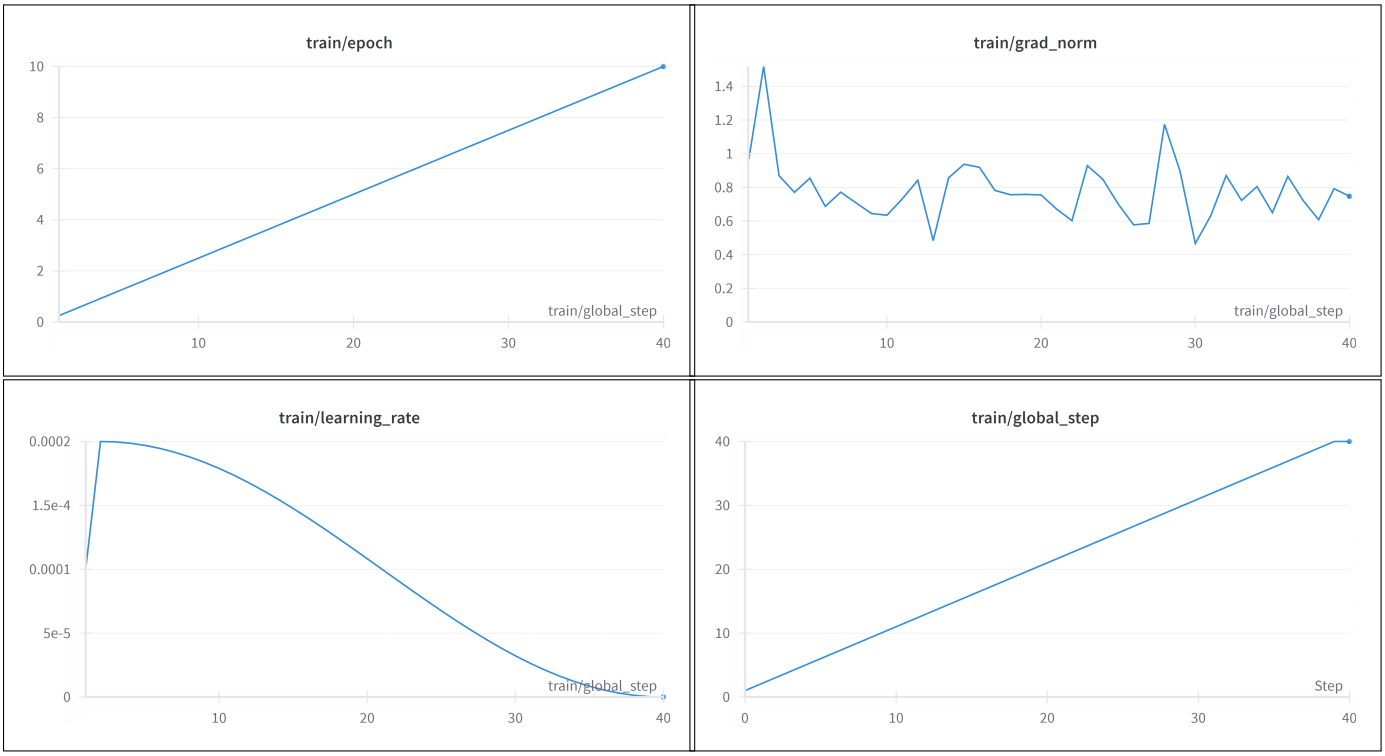


Figure 4.9: Visualised Training PArameters USing W&B

Chapter 5

Conclusion and Future Scope

In this work, a resume parsing system utilizing GenAI and Python libraries was developed. The Llama 2 model was fine-tuned using a parameter-efficient technique called QLoRA, which combines LoRA (Low-Rank Adaptation) and quantization. This method significantly reduces the number of GPUs required for fine-tuning, making the process more accessible and cost-effective. The work showcased the efficency of using PEFT for finetuning large models thereby reducing the cost of finetuning significantly.

The usage of QLoRA had its own advantages. It enables efficient use of hardware resources, reduces training time, and lowers the computational cost. This is especially beneficial when compared to few-shot prompting, where API calls to large models can be unreliable due to time limits and latency issues. With a fine-tuned model, we achieved similar results with much faster output generation. Moreover, fine-tuning large language models (LLMs) has become more manageable with techniques like QLoRA and LoRA. These methods simplify the process, making it feasible to obtain high-quality results without extensive computational resources.

Compared to the work in Phase 1 of this dissertation, the work in Phase 2 demonstrated improved results. The instruction tuned model from phase 1 gave a precision of 0.8942, recall of 0.8834 and F1 score of 0.8888. The fine tuned model gave a precision of 0.8719, recall with 0.9147 and F1 score of 0.8928. The instruction tuned llama was the largest llama 2 model with 70 billion parameters whereas this finetuned model was the base model with only 7 billion parameters. Still the results suggest that by using finetuning results as good as the largest model was obtained wihtout the need for API calling. It is clear from the results that the advancements in fine-tuning techniques contributed to more efficient and effective model performance.

The system significantly reduced the time required to manually convert a resume to the standard format of the company. While it typically took an average of 20 minutes to convert resumes manually, the model was able to convert them within just 12 seconds. Additionally, the system was able to extract implicit information from resumes, such as skills needed for a project, even if these details were not directly mentioned alongside the project information but were listed under the skills section. The model correctly mapped these skills to the respective projects. However, it's important to note that this capability was not achieved using the fine-tuned model but rather with the instruction-tuned model. This aspect presents an opportunity for improvement in future work.

The future scope of this work includes several promising directions. Further exploration of

advanced fine-tuning techniques can help optimize model performance even further. Additionally, integrating more diverse datasets could enhance the resume parsing system's accuracy and generalizability. Exploring and implementing more advanced fine-tuning techniques can lead to even greater performance gains. Techniques like QLoRA not only make the process more efficient but also improve the overall effectiveness of the model. Lastly, incorporating real-time analysis features of the resume could make the system more dynamic and user-friendly. The techniques used in this project have the potential to be applied to more complex tasks beyond resume parsing. For example, generating structured outputs like JSON for requirement generation and other applications can be achieved. By continuing to refine and expand upon these methods, we can create more robust and versatile natural language processing tools. The model can be improved by using various other PEFT techniques or by increasing the number of GPUs more datas can be used to train to achieve better results.

Using the same techniques, other use cases such as creating designs that match customer requirements, generating test cases from customer requirements, and generating managerial processes like creating meeting minutes (MoMs) and report templates can be addressed from an IT firm's perspective. For example, the system can be trained to understand the specific design preferences of customers based on their input or past projects, and then generate design proposals that align with those preferences. Similarly, it can analyze customer requirements to automatically generate test cases that ensure the software meets those requirements. In terms of managerial processes, the system can be trained to understand the structure and content of meeting minutes and report templates typically used in the company. It can then generate these documents based on inputs such as meeting agendas or project status updates, saving time and ensuring consistency in documentation. Overall, by applying the same techniques used in resume parsing to these use cases, IT firms can streamline their processes, improve efficiency, and ensure that deliverables meet customer expectations.

References

[1] Palshikar, Girish K., et al. "RINX: A system for information and knowledge extraction from resumes." Data Knowledge Engineering 147 (2023): 102202.

[2] Barducci, Alessandro, et al. "An end-to-end framework for information extraction from italian resumes." Expert Systems with Applications 210 (2022): 118487.

[3] Werner, Matheus, and Eduardo Laber. "Extracting section structure from resumes in Brazilian Portuguese." Expert Systems with Applications 242 (2024): 122495.

[4] Zhang, Min, Guohua Geng, and Jing Chen. "Semi-supervised bidirectional long short-term memory and conditional random fields model for named-entity recognition using embeddings from language models representations." Entropy 22.2 (2020): 252.

[5] Nguyen, Minh-Tien, Dung Tien Le, and Linh Le. "Transformers-based information extraction with limited data for domain-specific business documents." Engineering Applications of Artificial Intelligence 97 (2021): 104100.

[6] Dunn, Alexander, et al. "Structured information extraction from complex scientific text with fine-tuned large language models." arXiv preprint arXiv:2212.05238 (2022).

[7] Wang, Yaqing, et al. "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53.3 (2020): 1-34.

[8] Sivarajkumar, Sonish, and Yanshan Wang. "Healthprompt: A zero-shot learning paradigm for clinical natural language processing." AMIA Annual Symposium Proceedings. Vol. 2022. American Medical Informatics Association, 2022.

[9] Chung, Hyung Won, et al. "Scaling instruction-finetuned language models." arXiv preprint arXiv:2210.11416 (2022).

[10] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[11] Touvron, Hugo, et al. "Llama 2: Open foundation and fine-tuned chat models." arXiv preprint arXiv:2307.09288 (2023).

[12] Deepak, Gerard, Varun Teja, and A. Santhanavijayan. "A novel firefly driven scheme for resume parsing and matching based on entity linking paradigm." Journal of Discrete Mathematical Sciences and Cryptography 23.1 (2020): 157-165.

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

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

[15] Yu, Kun, Gang Guan, and Ming Zhou. "Resume information extraction with cascaded hybrid model." Proceedings of the 43rd annual meeting of the Association for Computational Linguistics (ACL'05). 2005.

[16] Gaur, Bodhvi, et al. "Semi-supervised deep learning based named entity recognition model to parse education section of resumes." Neural Computing and Applications 33 (2021): 5705-5718.

[17] Zu, Shicheng, and Xiulai Wang. "Resume information extraction with a novel text block segmentation algorithm." Int J Nat Lang Comput 8 (2019): 29-48.

[18] Gill, Jaskaran Kaur, et al. "Large language model based framework for automated extraction of genetic interactions from unstructured data." Plos one 19.5 (2024): e0303231.

[19] Xu, Lingling, et al. "Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment." arXiv preprint arXiv:2312.12148 (2023).

[20] Ding, Ning, et al. "Parameter-efficient fine-tuning of large-scale pre-trained language models." Nature Machine Intelligence 5.3 (2023): 220-235.

[21] Dettmers, Tim, et al. "Qlora: Efficient finetuning of quantized llms." Advances in Neural Information Processing Systems 36 (2024).

[22] Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." arXiv preprint arXiv:2106.09685 (2021).