Resume Generation Project: A Comprehensive Overview

Group Members:-

1. Vaishali Lalit – Student Id - 23104512
2. Yash Garg- Student ID - 23102980
3. Vineela Parepalli Laxmi- Student ID – 23103775

Project Description

Our project, Resume Generation Tailoring to Job Posting, utilizes the Gemini 1.0 Pro model and innovative prompting techniques within the Stream-lit framework to provide users with a seamless and intuitive platform for resume creation. Users can effortlessly input their personal details and job preferences, which are transformed into JSON format and transmitted via the Gemini API. Through in-context learning, the Gemini model adapts dynamically to user input and job requirements and generates personalized resumes guided by meticulously tailored prompts for each section. The generated text is then formatted into a professional PDF resume, empowering users to highlight their qualifications and experiences effectively.

Team Contributions

Associated Task	Description	Contributor
Model Selection and Implementation	• Evaluate various AI models for resume generation based on performance, compatibility, and ease of integration. • Select the most suitable AI model for implementation within the system. • Implement in-context learning techniques to fine-tune the model for personalized resume generation. • Develop a robust pipeline for data collection, model Integration, and inference to streamline the resume generation process.	Ms. Vaishali Lalit
Prompt Refinement and Conversion to PDF Format	• Refine prompts to elicit relevant and contextually appropriate responses from the AI model. • Fine-tune prompts to ensure effective highlighting of user qualifications and experiences in the generated resumes. • Format and structure the generated text to meet professional standards. • Convert the formatted text into a PDF document for easy distribution and printing.	Mr. Yash Garg
Data Collection and Crafting StreamLit Interface	• Design input forms for collecting user data, including personal details and job preferences. • Develop an intuitive and user-friendly interface using the Streamlit framework. • Present input forms in a visually appealing and easily navigable manner. • Ensure seamless interaction and user experience throughout the resume creation process.	Ms. Vineela Parepalli Laxmi

Tools Used:-

Google Generative AI (genai):

- ☐ **Purpose:** Utilized for content generation using AI models.

- **Features:** Provides access to pre-trained AI models such as Gemini for generating text-based content.
- **Performance Evaluation:** The Gemini model was evaluated based on its ability to produce high-quality and contextually relevant text outputs.
- **Customization:** The model's parameters, such as temperature, top-k, and top-p values, were configured to optimize its performance for resume generation.

Configuration: API key authentication for accessing the Google Generative AI platform.

Python Libraries:

- **ReportLab and FPDF:** Employed for converting generated text into professional PDF files.

Frameworks and APIs:

- **Streamlit:** Chosen for building an interactive user interface for the resume generation system.
- **Google-Gemini API:** Integrated and configured to leverage the Gemini model for content generation.
- **Replicate API:** Initially used with Lang-chain and later replaced with Google's API for the Gemini model, facilitating communication between the system and external resources.

Methodology:-

Data Collection and Annotation:

We collected user data using Stream-lit, including personal info, education, and job preferences. This data was systematically analysed to extract qualifications, experiences, and skills. This automated annotation organized the data effectively for accurate resume generation tailored to each user's career goals.

Initial Model Selection and Transition:

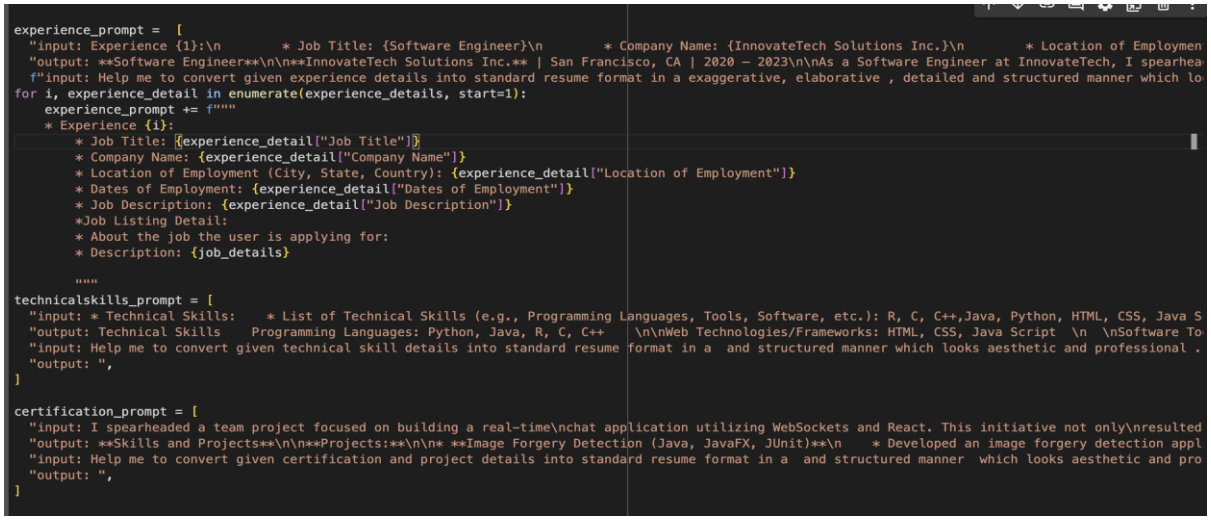
Our initial attempt with the LLama2 model highlighted challenges in achieving contextually relevant outputs aligned with the job descriptions provided. Despite employing various prompting strategies like few-shot prompting to enhance performance, the model struggled to meet our expectations. Recognizing the limitations of static prompting alone, we sought a solution that could adapt and learn from the context of the resume generation task. Transitioning to the Google Gemini API proved beneficial as it not only offered robust text generation capabilities but also demonstrated proficiency in in-context learning. By leveraging the Gemini model's ability to adapt and refine its responses based on the provided input and task context, we achieved more tailored and effective resume generation, better meeting the needs of our project.

Prompt Engineering and Resume Generation:

With the transition to the new model, considerable emphasis was placed on prompt engineering to enhance the resume generation process. Leveraging our annotated dataset, we meticulously crafted detailed prompts tailored to guide the model effectively, facilitating in-context learning. The job descriptions provided by users played a pivotal role in extracting and accentuating essential skills and qualifications within the resumes. This amalgamation of refined prompts underwent processing utilizing the Google Gemini model, renowned for its adeptness in handling intricate text structures and adapting to context. Subsequently, the generated text underwent seamless conversion into professional-looking PDF resumes through the utilization of Report-Lab and FPDF libraries. This

ensured that the final output not only delivered comprehensive information but also boasted visually appealing presentation, aligning perfectly with our project's objectives.

Code Snippet :- To Show the Overview of Section Wise Prompts



```
experience_prompt = [
  "input: Experience {1}:\n        * Job Title: {Software Engineer}\n        * Company Name: {InnovateTech Solutions Inc.}\n        * Location of Employmen
  "output: **Software Engineer**\n\n**InnovateTech Solutions Inc.** | San Francisco, CA | 2020 – 2023\n\nAs a Software Engineer at InnovateTech, I spearhea
  f"input: Help me to convert given experience details into standard resume format in a exaggerative, elaborative , detailed and structured manner which lo
for i, experience_detail in enumerate(experience_details, start=1):
    experience_prompt += f"""
    * Experience {i}:
        * Job Title: {experience_detail["Job Title"]}
        * Company Name: {experience_detail["Company Name"]}
        * Location of Employment (City, State, Country): {experience_detail["Location of Employment"]}
        * Dates of Employment: {experience_detail["Dates of Employment"]}
        * Job Description: {experience_detail["Job Description"]}
        *Job Listing Detail:
        * About the job the user is applying for:
        * Description: {job_details}

        """
technicalskills_prompt = [
  "input: * Technical Skills:    * List of Technical Skills (e.g., Programming Languages, Tools, Software, etc.): R, C, C++,Java, Python, HTML, CSS, Java S
  "output: Technical Skills    Programming Languages: Python, Java, R, C, C++    \n\nWeb Technologies/Frameworks: HTML, CSS, Java Script  \n  \nSoftware To
  "input: Help me to convert given technical skill details into standard resume format in a  and structured manner which looks aesthetic and professional .
  "output: ",
]

certification_prompt = [
  "input: I spearheaded a team project focused on building a real-time\nchat application utilizing WebSockets and React. This initiative not only\nresulted
  "output: **Skills and Projects**\n\n**Projects:**\n\n* **Image Forgery Detection (Java, JavaFX, JUnit)**\n    * Developed an image forgery detection appl
  "input: Help me to convert given certification and project details into standard resume format in a  and structured manner  which looks aesthetic and pro
  "output: ",
]
```

Challenges Overcome

- ***Model Performance Issues:*** The initial challenges with the LLaMA2 model's performance were significant, as they threatened the feasibility of the project. Transitioning to the Google Gemini model was a strategic decision that involved reconfiguring our system's integration with new APIs and adapting our data processing to fit the new model's needs accordingly.

- ***Data Privacy Concerns:*** Integrating the Google Gemini API raised concerns regarding data privacy, especially given the strict data protection laws in Europe. We resolved this by implementing VPN solutions to securely manage data transfer without breaching privacy laws. As we had an aim of keeping our model free to use, this could be done by using the resources that were available free for use over the internet. Google Gemini API not charging any cost for its use was one of the important factors that we went ahead with the Gemini model.

Prompt Refinement Challenge:

Crafting clear and effective prompts for guiding resume generation posed a significant challenge. Balancing the need for specificity with flexibility required iterative refinement and close collaboration.

Evaluation of the Approach

The effectiveness of our model's approach can be evaluated through several metrics:

Quality of Generated Resumes: Assess the coherence, relevance, and completeness of the generated resumes compared to user-provided input and job requirements.

User Satisfaction: Gather feedback from users regarding the usability, accuracy, and overall satisfaction with the generated resumes.

Accuracy of Content: Validate the accuracy of information presented in the resumes against the original user input and external sources where applicable.

Adaptability and Flexibility: Evaluate the model's ability to adapt to different user inputs, job descriptions, and formatting preferences.

Alternative or Further Techniques

Ensemble Modeling: Combine multiple AI models for improved resume generation.

Transfer Learning: Adapt pre-trained models to resume generation.

Active Learning: Incorporate user feedback for iterative improvement.

Domain-Specific Fine-Tuning: Tailor the model to industry-specific needs.

Hybrid Approaches: Integrate rule-based systems with AI models.

References:-

1.Akter, S.N., Yu, Z., Muhamed, A., Ou, T., Bäuerle, A., Cabrera, Á.A., Dholakia, K., Xiong, C., & Neubig, G. (2023). An In-depth Look at Gemini's Language Abilities. Carnegie Mellon University, BerriAI. Retrieved from https://github.com/neulab/gemini-benchmark

2. Zhang, T., Madaan, A., Gao, L., Zheng, S., Mishra, S., Yang, Y., Tandon, N., & Alon, U. (2024). In-Context Principle Learning from Mistakes. arXiv preprint arXiv:2402.05403. Retrieved from https://arxiv.org/2402.05403

3. Rane, N., Choudhary, S., & Rane, J. (2024). Gemini Versus ChatGPT: Applications, Performance, Architecture, Capabilities, and Implementation. Retrieved from SSRN: https://ssrn.com/abstract=4723687 or http://dx.doi.org/10.2139/ssrn.4723687