SMART APPLICATION TRACKING SYSTEM: LEVERAGING GENERATIVE AI AND LANGCHAIN

Group Members-

2029197 - Shivendra Singh

2029202 - Srijan Kumar

2029203 - Sneha Singh

2029204 - Jyoti Kumar

PROF.Amiya Kumar Dash

PROBLEM WITH TRADITIONAL ATS

- Traditional ATS systems rely heavily on rigid keyword matching.
- This approach often misses qualified candidates whose resumes express relevant skills or experience differently.
- These systems offer limited personalization and provide generic or unhelpful feedback to applicants.

INTRODUCING THE AI-POWERED SMART ATS

- Our Smart ATS leverages Google Gemini for semantic matching, ensuring it understands the nuances of both resumes and job descriptions.
- Instead of just keywords, it calculates a matching score based on deeper contextual relevance.
- It empowers job seekers with personalized project suggestions to enhance their resumes.
- The system offers relevant certification recommendations aligned with current job requirements.
- It streamlines the application process by drafting tailored cover letters.

HARNESSING THE POWER OF GENERATIVE AI

- We use large language models (LLMs), like Google Gemini, because they excel at understanding the nuances of human language.
- Gemini enables matching beyond keywords, considering synonyms and the overall context of a text.
- This results in a more accurate and relevant matching score.
- Used Generative AI to parse the resume and generate the match percentage.

```python
# Function to get response from Generative AI model
def get_gemini_response(input):
    model = genai.GenerativeModel('gemini-pro')
    response = model.generate_content(input)
    return response.text
```

POWERING THE SMART ATS WITH LANGCHAIN

- We used LangChain for modular and flexible interactions with Google Gemini.
- LangChain's Sequential Chains allowed us to structure different analysis tasks into a logical pipeline.
- This enabled us to easily experiment with different prompts and chain configurations for generating project suggestions, certifications, and cover letters.
- LangChain facilitated the creation of structured prompts that we sent to Gemini for analysis.

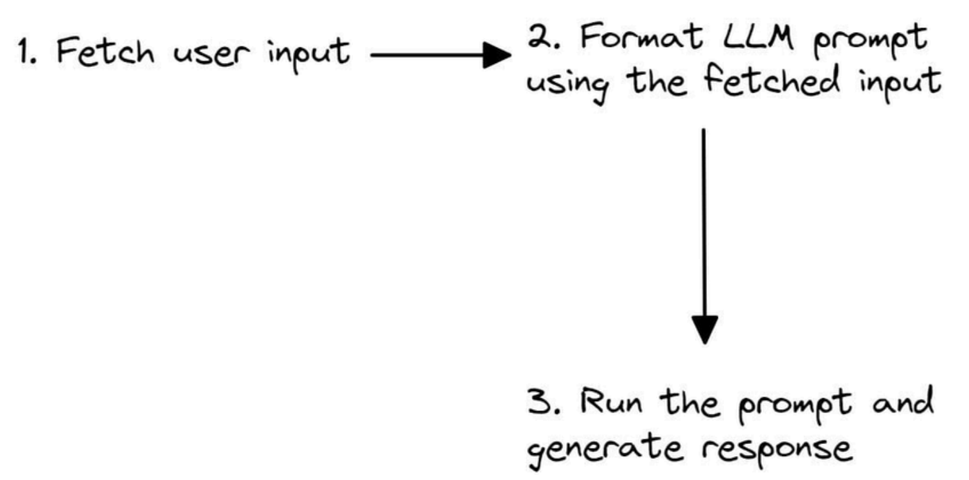
```
cover_template = PromptTemplate(
    input_variables=['topic', 'resume'],
    template='Write a cover letter with help of resume based on the job description {topic}'
)
llm = ChatGoogleGenerativeAI(model="gemini-pro", google_api_key=GOOGLE_API_KEY)
project_chain = LLMChain(llm=llm, prompt=project_template, verbose=True)
```

HOW THE SMART ATS WORKS

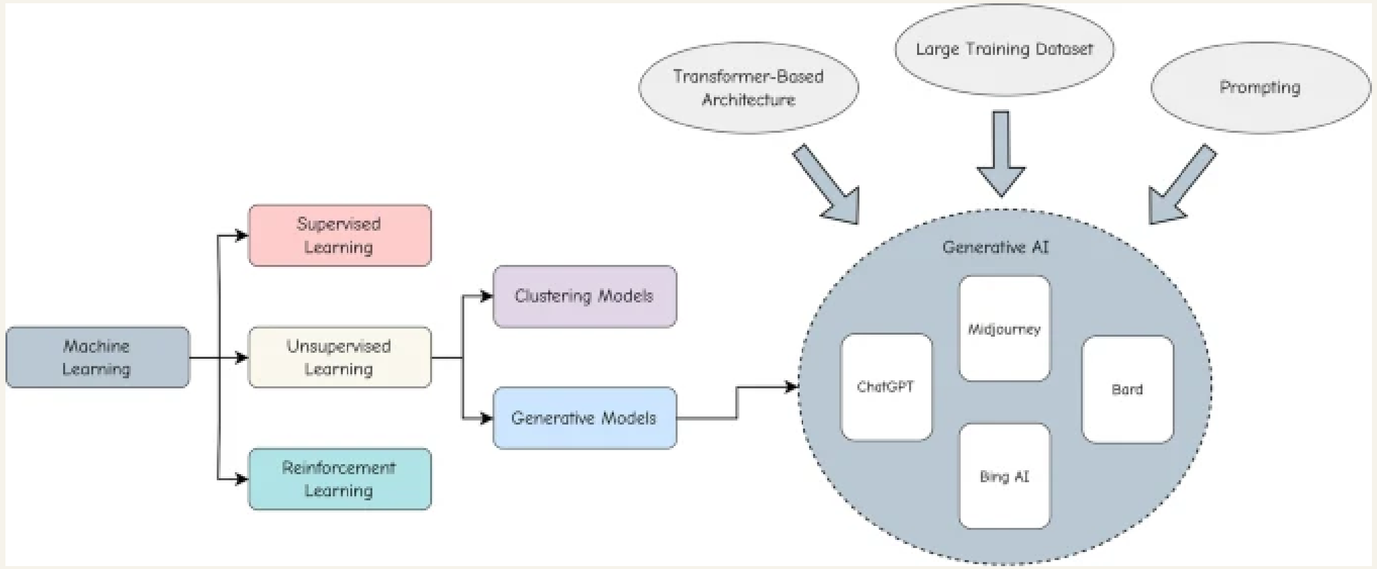
Parsing & Cleaning:

- **The system begins by parsing both the uploaded resume and the job description to extract relevant text.**
- **Basic cleaning techniques might be applied to remove unnecessary characters or formatting.**

Lang chain



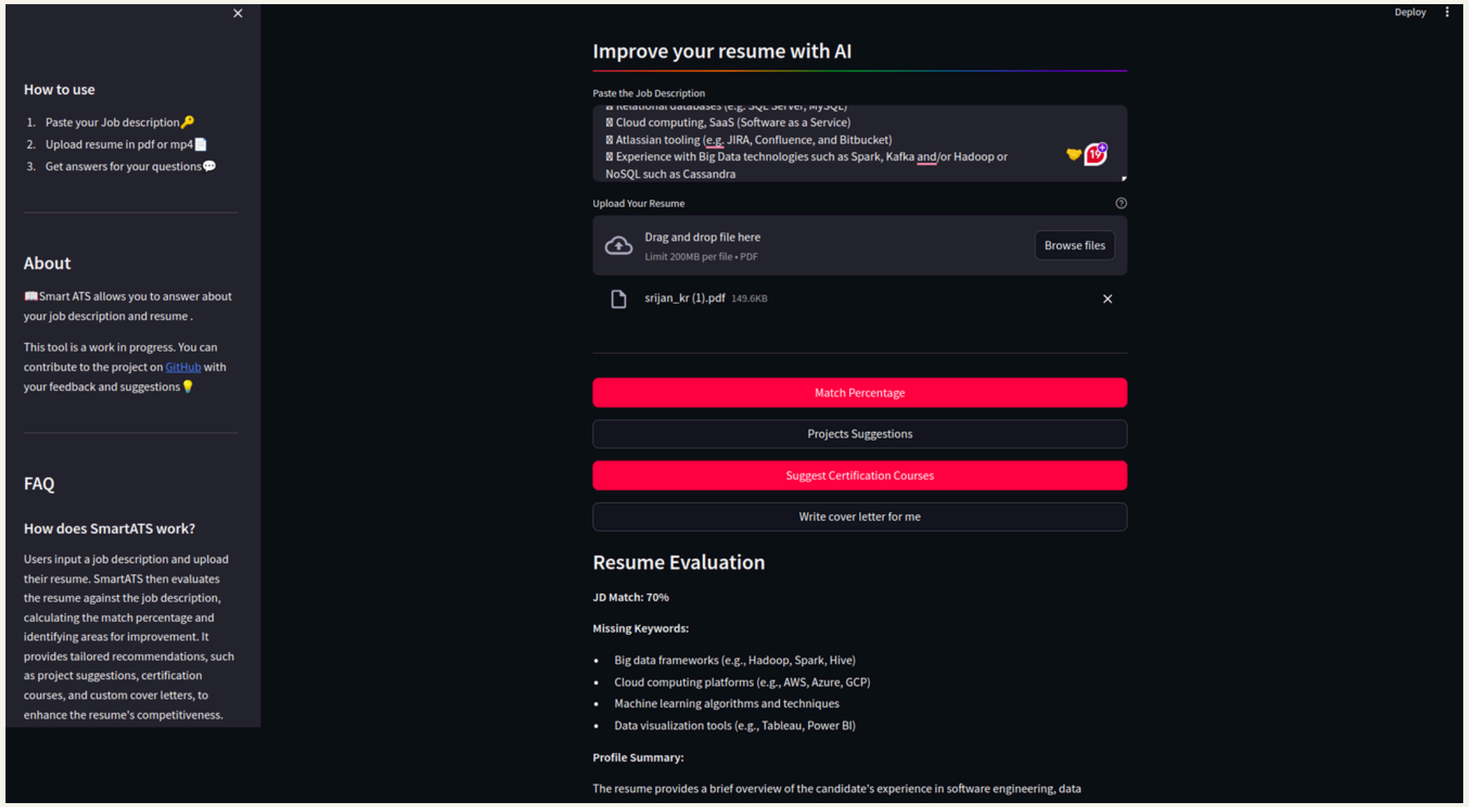
Generative AI



- The processed text is fed into the Langchain engine.
- Langchain utilizes a Sequential Chain to structure the analysis through Gemini, the LLM.
- Depending on the desired output (project suggestions, cover letter, etc.), Langchain sends specific prompts to Gemini, leveraging its semantic understanding.

IMPACT AND RESULTS

Lets see the demo and working of our SMART ATS



CONCLUSION AND FUTURE SCOPE

- Recap: Our Smart ATS offers a more accurate and personalized experience than traditional ATS systems.
- Potential Possibilities for expansion include refined resume parsing, providing explanations along with matching scores, and integrating directly with job search platforms.

KIIT University | 2024

THANK YOU

Presented By :

2029197 - Shivendra Singh

2029202 - Srijan Kumar

2029203 - Sneha Singh

2029204 - Jyoti Kumar