

ChatGPT - Gemini Job Description Validation

my persimon project has job description while users type job description gemini need to ask questions about missing jd information . gemini is already installed in fast api app

To enable Gemini to ask questions about missing information in job descriptions, you can leverage natural language processing (NLP) and dialog management features. Here's an approach to integrate it into your FastAPI app:

Steps to Implement

1. Identify Missing Information:

- Define required fields for a job description (e.g., role, responsibilities, qualifications, salary, location).
- Use a pre-defined schema or validation tool to check the completeness of the input.

2. Process the Job Description:

- When a user types the job description, pass the input to Gemini.
- Use an NLP model to extract entities from the text and identify gaps by comparing the extracted entities against the required schema.

3. Generate Questions:

- For each missing field, generate a question. For instance:
 - Missing "Location" → *"Where is this job located?"*
 - Missing "Salary" → *"What is the salary range for this position?"*

4. Integrate Gemini's Questioning:

- Configure Gemini to dynamically generate and display questions to users.
- Capture user responses and append them to the job description.

5. FastAPI Endpoint:

- Create an endpoint that:
 - Receives the job description.
 - Invokes Gemini for validation and questioning.
 - Returns updated job descriptions after user input.

Sample Code Snippet

python

```
from fastapi import FastAPI, HTTPException
from pydantic import BaseModel
from gemini import Gemini # Hypothetical Gemini import

app = FastAPI()
gemini = Gemini()

class JobDescription(BaseModel):
    title: str
    description: str
    location: str = None
    salary: str = None
    qualifications: str = None

@app.post("/validate-job-description/")
async def validate_job_description(jd: JobDescription):
    # Check for missing fields
    missing_fields = []
    if not jd.location:
        missing_fields.append("location")
    if not jd.salary:
        missing_fields.append("salary")
    if not jd.qualifications:
        missing_fields.append("qualifications")

    if missing_fields:
        # Generate questions for Gemini
        questions = [
            f"What is the {field} for this job?" for field in missing_fields
        ]
        return {"missing_fields": missing_fields, "questions": questions}
    return {"message": "Job description is complete."}
```

6. User Interface:

- On the front end, display questions dynamically and send user responses back to the API for real-time updates.

Would you like help setting up Gemini's question generation or refining the API further?