Summarize-earnings-colab-notebook has the colab run of code ran previously for summarizing text - local tests and PythonAnywhere tests for both Dr Lal Pathlabs and One97 are run there (normal cases and error handling cases included).

requirements.txt

We need following packages -

1) requests - Needed to make POST and GET requests to REST API
2) google-generativeai - Needed to access gemini-1.5-flash to summarize transcript
3) pdfplumber- Needed to parse text from the pdf
4) flask - To create Flask API

run.py

'python3 run.py' starts the Flask app locally on port 5000.

app/routes.py

/earnings_transcript_summary API -
We first read the input JSON into *data* variable.
The input JSON looks like -

{
    company_name: *company_name*,
    transcript_text: *transcript_text*
}

Error handling - We then check if *transcript_text* and *company_name* fields are present in the JSON. If any of the fields is not present, we return a 404 error that the particular field is not present. If both fields are present, we again check if the value held by any of those two fields is an empty string, and if so, we return that the respective field is not provided.

If both *transcript_text* and *company_name* fields are present, then, we get summaries of each section from *sections={financial_performance, market_dynamics, expansion_plans, environmental_risks, regulatory_or_policy_changes}* by calling -

summarize_text(transcript_text,section_name,company_name) function, which internally calls Gemini-1.5-flash API to get summary of *section_name* from *transcript_text* document.

The summary of each *section_name* is then inserted as a respective field into the final JSON to be returned. Thus, our final JSON looks like -

{
    company_name: *company_name*,
    financial_performance*: financial_performance,*
    market_dynamics*: market_dynamics,*
    expansion_plans*: expansion_plans,*

environmental_risks*: environmental_risks,*
regulatory_or_policy_changes*: regulatory_or_policy_changes*
}

app/utils.py

It defines the function *summarize_text(transcript_text,section_name,company_name)*.
It makes the API call - "Please summarize the {*section_name*} from {*company_name*} company's earning transcript:\n\n{transcript_*text*}" - to Gemini-1.5.Flash Generative AI, which then summarizes the section called *section_name* from *transcript_text.*

For this, we need a Gemini-1.5-Flash API to be inserted in code to run.

Instructions for running API locally and hosting it

Method 1 -

1) Add your Gemini 1.5 flash key to app/utils.py
2) Start app - "python3 run.py".

Method 2 -
Jupyter notebook *summarize_earning.ipynb* is the notebook downloaded from Google Colab that has the run results cell-by-cell for this code. *summarize_earnings.ipynb* can also be started and run cell by cell. Make sure to put *earning_call_dr_lal_pathlabs* and *earning_call_one97* PDFs in the first folder (MyDrive) of Google Drive for tests to run.

Testing

Method 1 (local)-
Run tests from tests/ folder-
1) python3 test1.py:- Summarizes earning transcript of earning_call_dr_lal_pathlabs.
2) python3 test2.py:- Summarizes earning transcript of earning_call_one97.
3) python3 test3.py:- company_name field is empty in input json. This should return "No company_name provided".
4) python3 test4.py:- transcript_text field is empty in input json. This should return "No transcript_text content provided".
5) python3 test5.py:- company_name field is not present in input json. This should return "No company_name provided".
6) python3 test6.py:- transcript_text field is not present in input json. This should return "No transcript_text content provided".

All these tests (local tests and PythonAnywhere tests for both Dr Lal Pathlabs and One97 are run there - normal cases and error handling cases).

All these tests create the input JSON by parsing input PDF file to extract text, and this input JSON is then used as input for local API call -
http://127.0.0.1:5000/earnings_transcript_summary

Method 2 (local) -
Open *summarize_earning.ipynb*. Tests and results are present in cells there.

Public (PythonAnywhere) -

Open *summarize_earning.ipynb*. It has cells that send input JSON to -

URL of hostel API -
https://kshitijdegg.pythonanywhere.com/earnings_transcript_summary