10-K / 10-Q Digestion + Retrieval

The goal of this project is to create a generalized version of a 10-K / 10-Q question answer chatbot. We feel like a project like this can allow you to show your stuff given your experience at BlueFlame. We intentionally have not given you a skeleton code because you should create this system however you think is best given the problem at hand.

The Problem You Need to Solve

10-Ks and 10-Qs are likely one of the most accurate sources of public company data on the internet, as they are reports from companies given to the government about their financials, growth, key developments, etc.

As a result, there are a surplus of things that someone may want to know about a company's 10-K or 10-Q - however, these documents are often 10s of pages long and finding specific information in them is a tedious, slow process. Having a reliable AI system to be able to take in queries in natural language, pull down the relevant 10-K or 10-Q filing, and correctly answer questions on them, would be incredibly useful in the day-to-day of a financial services professional.

For reference, 10-Ks are annual filings, and 10-Qs are quarterly filings for all quarters except Q4, which is covered by the annual 10-K. They both are similar types of documents and formatting, but the 10-Ks tend to have more robust and detailed information.

Considerations & Technical Goals

There are multiple aspects of this system to value when designing a solution, as they drastically affect the user experience:

1. **Latency** of inference time (chatbot query → response)
2. **Accuracy** of responses
3. **Tracking of sources** to audit your responses and guarantee accuracy

These should be the foundation of your decision making process when evaluating tradeoffs during system design.

To accomplish these values in a system, these are *examples* of core technical processes that could allow you to hit all three of those values:

1. A preprocessing step that digests 10-Ks and 10-Qs into whatever format you think is best for reliable, fast Q/A and that does not miss any information. Both are presented in HTML or PDF formats, so a likely candidate of this is PDF → text and / or some special augmentations for things like charts / tables.
2. A solution that takes a query, and determines what 10-Ks / 10-Qs it needs to look over - for example, “What was Google’s CAGR in 2024?” would pull the 2024 preprocessed 10-K and three 2024 10-Qs if available.
3. A reliable process for determining which parts of the pulled, processed 10-K / 10-Q are the most relevant to answer the question, and feeding that to an LLM to answer the question.

Technologies

Other than **access to an OpenAI** token that we will give you, you are on your own for what technologies you use for this project. Things like Pinecone / vector databases, document / data stores, etc. you should be able to use for free from different services on the internet. *However, if there is something that you want to use that is paywalled, please let us know and we may be able to accommodate it.*

OpenAI Key:

```
sk-proj-SuEjxGLYWhnTljC3fE3mcmeq3S8VE7UPWiUVAehb-
KAkHiqLR4vYfA74FGFI6ExmEcMP6zTC6jT3BlbkFJcOumVKu6NFgBdmP2SkUyIt6l8g70JS6VtgIkvf66J8ejo-
8As69nlw90Eqb0RiIEkJke0FxjYA
```

Where do I get the 10-Ks / 10-Qs from programmatically?

You should be getting your 10-Ks and 10-Qs from the EDGAR API that is published by the government - it is completely free. Please read the documentation for this API and reference the Developer FAQ to learn more about how to use this API to fetch the documents.

Deliverables

Below are the explicit technical deliverables that we would expect from one of our engineers for a project like this. To complete these, please take your time to plan out your project so that you can ensure that you have objective results to show about its performance:

1. A test suite of potential questions someone may ask and the answers for at least 5 different companies' recent 3 years of 10-K and 10-Qs. This test suite should be made before implementation. Questions should cover retrieving information across different modes of information, like tables, charts, and text, which cover all of the ways information is presented in these documents.
    a. This allows you to have an objective measurement of accuracy at the end of your project.
2. An API with two endpoints:
    a. **A processing endpoint:** a method that is able to, given a ticker and year, pull the relevant 10-K or 10-Q, process and store it in your system.
    b. **An inference endpoint**: given a natural language query, return a response in natural language that also cites what 10-K or 10-Q it got the information from and where in the it resides in the document.
3. Write a brief write-up about the design that you went with, what the tradeoffs were of that design compared to other designs that you tried, and what the results of the project were from an objective metric standpoint.
4. Your solution should be able to answer nearly any question that is asked of at least **5 different companies' last three years of 10-K / 10-Qs**. Please let us know which documents you have in your system when you return it.
5. Please give us the link to a GitHub repository at the end of this, and send over any secrets / installation instructions necessary to run the code.