Alejandro Rodriguez

UCF ID: 5361434

GitHub: https://github.com/arodriguezb52/Projects/tree/main

LLM 10-K Product Extraction Report

Introduction

The objective of this assignment is to build a pipeline capable of extracting information about newly announced or updated products directly from SEC Form 10-K filings. The extraction process leverages a local Large Language Model (LLM)—specifically, the DeepSeek R1 model with 1.5 billion parameters to parse extensive and unstructured financial documents and transform them into structured summaries. Through carefully designed prompts and clear instructions, the LLM identifies and records new product information accurately. Once identified, the pipeline organizes these extracted details into a CSV file, which includes essential data such as the company name, ticker symbol, filing date, new product name, and a concise product description.

Methodology

The methodology for this assignment began by automating the retrieval of the latest 10-K filings from selected companies. Initially, the pipeline prompts the user to input a ticker symbol, which it then converts into the corresponding Central Index Key (CIK) using a predefined mapping dictionary. This conversion is essential because the SEC's EDGAR system primarily identifies companies through their CIK rather than ticker symbols. Once the correct filing is identified, the pipeline queries the SEC's API to retrieve JSON-formatted data regarding the company's most recent 10-K submission. The pipeline then downloads the corresponding filing document using Python's requests library. Simultaneously, BeautifulSoup is used to parse and extract the entire text content from the HTML file, which is then saved locally for convenient reuse and further analysis.

Once the raw 10-K text has been stored, a keyword-based filtering step isolates passages most likely relevant to new products. The system scans txt files line by line, searching for words such as "new product," "launch," "introduced," "announced," or "released." If it locates lines containing these terms, it consolidates them into a smaller "product-related" section; if no lines meet the criteria, it defaults to using the entire filing. This approach substantially reduces the text length passed to the LLM, lowering the computational needs of subsequent operations, and improving the LLMs capacity of retrieving the necessary information. This is one of the many solutions to fix the maximum capacity of tokens that an LLM can process at once.

Once a relevant text subset was isolated, the DeepSeek R1 LLM was prompted with carefully engineered instructions. These instructions explicitly asked the model to identify any new or updated products, including expansions, next-generation releases, and also guided it to recognize synonyms such as "unveiled," "redesigned," or "enhanced." The model was directed to return the output strictly in JSON format, specifically excluding chain-of-thought explanations or additional commentary to maintain structured consistency. Post-processing involved parsing the model’s JSON response, with a fallback regex-based extraction step ready in case direct parsing failed. Finally, any correctly structured product announcements were recorded into a CSV file, detailing the company name, ticker symbol, filing date, product name, and brief description for each new product extracted.

Shortcomings

The most significant shortcoming that the pipeline has is that it processes each company ticker individually, which is manageable for a small set of companies but not ideal for large-scale analysis. Due to limited computing resources, each session is executed one ticker at a time, prompting the user for a new ticker after completing a pipeline run. A more scalable solution would employ either parallel processing or a batch approach capable of handling multiple tickers without separate user interaction. However, within current resource constraints, the system demonstrates a reliable means of summarizing new product announcements from 10-K filings.

Conclusion

In conclusion, the pipeline successfully demonstrated the increased productivity by using LLMs to extract structured information about new products from lengthy and complex 10-K fillings. With the combination of automated retrieval, parsing techniques, keyword-based filtering, and prompt engineering the pipeline was able to identify product announcements efficiently. Even though, the pipeline has limitations in terms of scalability the methodology employed gives it a strong foundation that could be adapted for bigger tasks.