Core Earnings Explorers

30135 – Winter 2025

Sandesh Puligundla

Pramod Nayak

Pramod Bykhovsky

Bruno Graca Coelho

CHICAGO BOOTH

Challenge, context and objective

Problem statement/ challenge

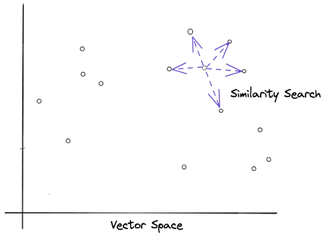
- **Estimation of core earnings**, i.e., a firm's persistent profitability from its core business activities is central to investors' assessments of economic performance and valuations.
- Quantifying core earnings requires judgment and integration of information scattered throughout financial disclosures contextualized with general industry knowledge. This has become increasingly difficult as financial disclosures have become more "bloated" and accounting standards have increased non-recurring impacts on GAAP net income.
- The chasm between GAAP earnings and what investors consider "core" earnings has widened, and bridging it has become more challenging

Goals:

Use LLMs, with their ability to process unstructured text, incorporate general knowledge, and mimic human reasoning, to calculate/ estimate core earnings

Approach – End-to-end solution overview

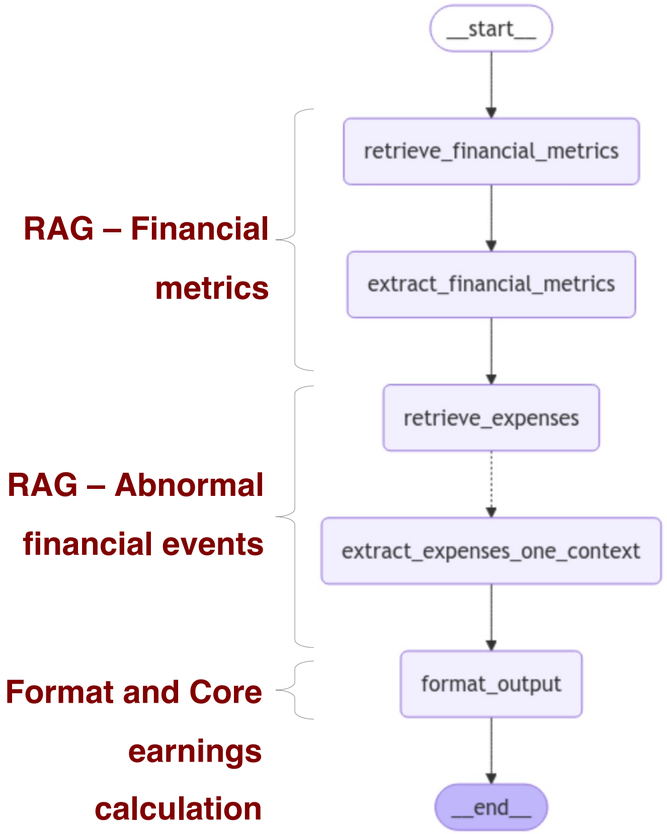
RAG

1 Collecting/ extracting the 10-K reports	2 Chunking the 10-K text using 'token text splitter'	3 Chunk embedding	4 Load embedded vector	5 Retrieve (Similarity search)	6 Augment (retrieved chunks)	7 Generate (responses)
					Prompt + Context	Gemini OpenAI
• Used sec_edgar_ downloader python module to get text extracts of 10-K filings	• Chunking of 10-Ks text documents using 'token text splitter' from Lang chain. • Applied a fixed length chunking of **1024 tokens** with **overlap of 100 tokens between consecutive chunks**	• Embed each chunk using 'text ada embedding small' model	• Load embedded vector into vector store - **pinecone index** of dimension 1,536	• Search and fetch relevant information from the vector-DB using similarity search	• Augment the retrieved chunks with prompt into LLM (4o or 4o-Mini or Gemini)	• Generate appropriate response using an LLM (4o, 4o-Mini, Gemini) based on the augmented user question and retrieved information

CHICAGO BOOTH

Approach – Workflow implementation overview

Workflow overview



Details/ reasoning

- Pinecone setup and model specific global variable for vector-based retrieval and language model processing
- First, using vector search, through *retrieve_financial_metrics* function we retrieve relevant documents that are after used to extract financial metrics (*extract_financial_metrics*) necessary for core earnings calculation (e.g. net income, #shares outstanding, effective tax rate, stock split ratio)
- After, replicate a similar process but focusing on the identification of abnormal expenses or revenues to be considered in core earnings calculation. For each relevant financial item, function *Extract_expenses_one_context* returns financial item description, amount, fiscal year and date
- Finally, format responses, review duplicates, calculate core earnings and extract results to excel

Representative testing scenarios, approach and objectives

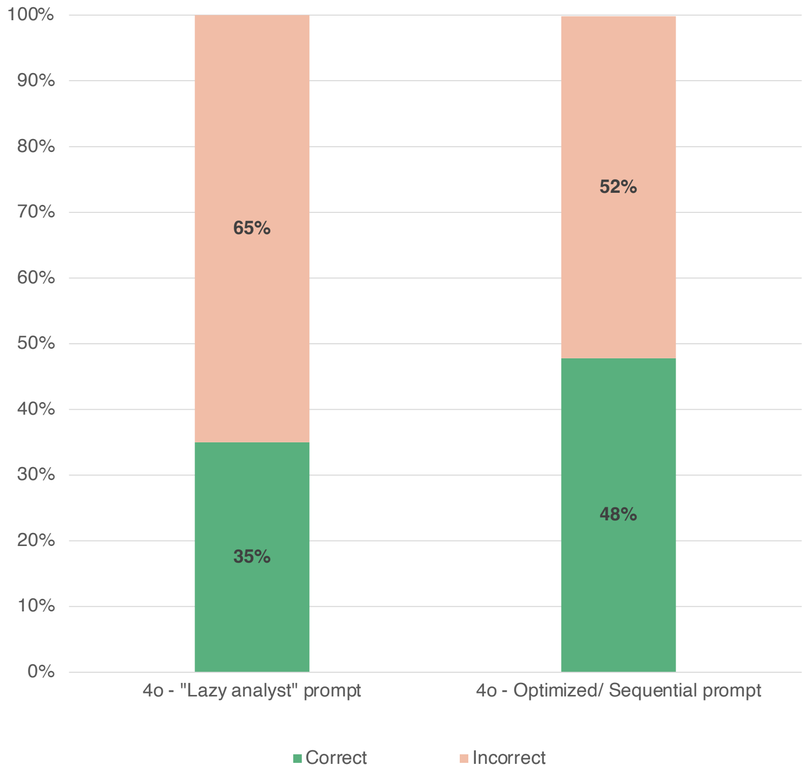
Testing scenarios	Testing approach	Objectives
1 Exhaustive abnormal adjustments analysis	• Our team used the 10-Ks information from 18 different companies. Executed the solution using the 2 different models (4o, 4o-Mini), using 2 different prompts. From **249 financial adjustments**, we **sampled and reviewed 100**	• Sample and review identified abnormal adjustments so we could achieve confidence level of around **85%**
2 Walkthrough/ Test of one	• Our team reviewed all financial adjustments identified by the 2 different models (4o, 4o-mini) making a financial analysis over all adjustments identified for one company	• Understand/ assess all abnormal financial events for a representative firm
3 Test of one with 20 years data (Apple)	• Our team executed the 2 different models using 20 years of financial information (10-Ks) identifying abnormal adjustments and calculating core earnings for each year. • Leveraged XGBoost and Random Forest to project 2024 core earnings per share	• Understand/ assess solution consistency over time

CHICAGO BOOTH

Prompts and models impact greatly result's accuracy

1 Exhaustive abnormal adjustments analysis

Accuracy results



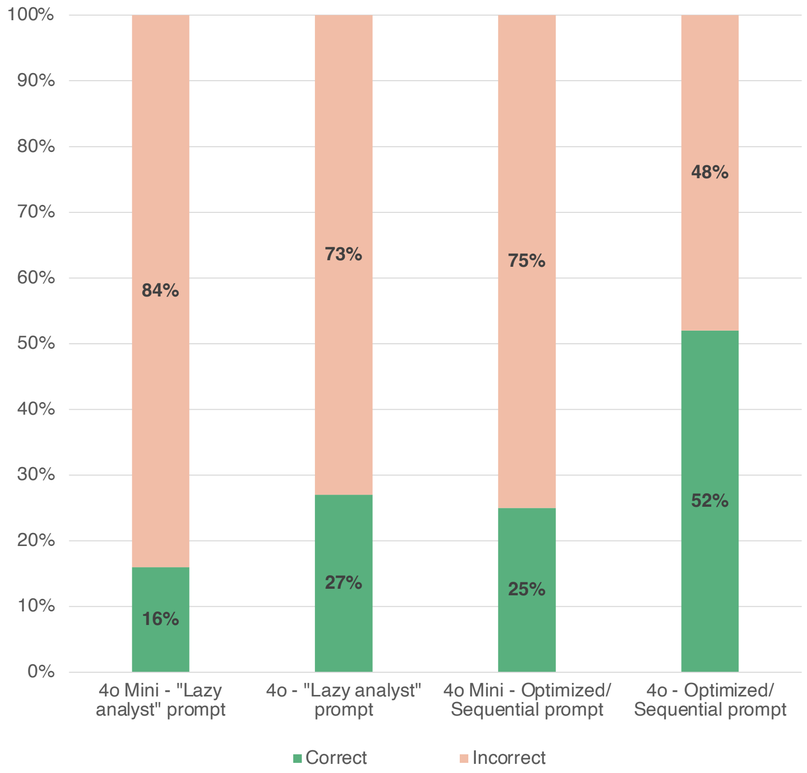
Details and conclusions

- Overall, running with an **optimized/ sequential prompt produces the best results** if compared with "lazy analyst" prompting approach
- **Optimized/ sequential prompting produced around 47.8% accuracy**, within 85% confidence level (from sampling 100 out of 249 identified adjustments from 18 companies)
- **Carefully designed/ optimized prompts can produce reasonable core earnings estimations** – prompting engineering and functional knowledge is key
- Adding supervisory function doesn't solve the identified errors. When chat GPT instructed to review the results, it doesn't catch these errors

CHICAGO BOOTH

Prompts and models impact greatly result's accuracy

Accuracy results



Details and conclusions

- "Lazy analyst" prompt approach makes more mistakes and misses more adjustments than the optimized/ sequential prompt – with 4o and 4o-Mini
- When reviewing this firm, we observed that model **4o** with **Optimized/ sequential performed better than average**
- On the other hand, **Optimized/ sequential prompting with 4o-Mini performed worse.** We observed one case of hallucination, and four reasoning mistakes
- Solution repeats mistakes, despite clear guidance to review and avoid (e.g. Other Comprehensive Income)
- LLM appears to be **insufficiently equipped to reason** whether the nature of financial event impacts Net Income

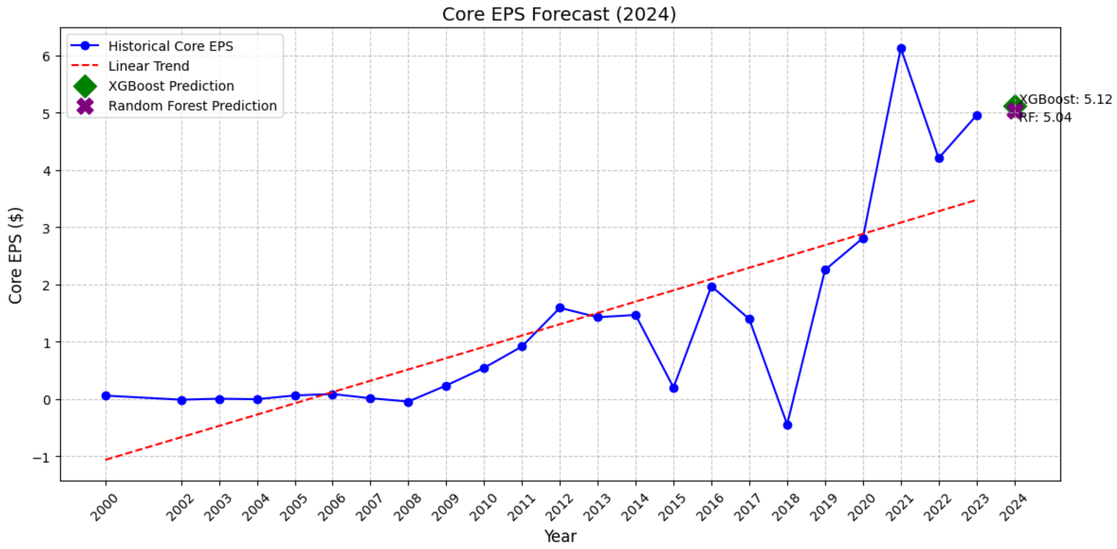
Workflow also projects core earnings, based on historica

3 Test of one with 20 years data (Apple)

Apple – Core earnings per share projection

- Apple is a simple and representative case, with “clean” and ”simple” adjustments. Our team used 20 years of Apple 10-Ks, making it an “easy” case for workflow
- To enable this last step, we changed the prompt, so the model roleplays as a skilled financial analyst, specialized in forecasting, using the following regression models:
    1. XGBoost regression model
    2. Random Forest regression model

Details and conclusions (4o)



Main conclusions and next steps

Conclusions

- **Carefully designed/ optimized sequential prompts can produce reasonable core earnings estimations** – Quality control at scale is challenging
- **Complexity matters**. With simple core earnings calculation the workflow performs well, but when complex and numerous adjustments exist the accuracy drops significantly
- **GPT 4o performs better than 4o-Mini**. We observed a significantly decrease in the quality of results when using 4o-mini for this specific task – more items identified, but more incorrect
- Additional prompt engineering and quality controls at scale are necessary
- LLM appears to be insufficiently equipped to reason whether the nature of financial event impacts Net Income

Next steps

- Develop a full-blown agentic solution to fully compare against workflow
- Conclude analysis of results using Gemini LLM and compare against 4o and 4o-Mini
- Cost/ Benefit analysis comparing different models