**PHI4 model for the Q&A Earnings Transcripts**

1. **Data extraction**

Earnings transcript data were extracted using the Ninja API. The dataset includes transcripts from several major banks: Barclays, Credit Suisse, Deutsche Bank, HSBC, JPMorgan Chase, Santander, and UniCredit. The standard extraction period spans from 2023 Q1 to 2025 Q1. An exception applies to Credit Suisse, which was acquired by UBS. As a result, transcript data for Credit Suisse are available only up to 2024 Q1. To ensure sufficient coverage, data for this bank were extracted starting from 2020 Q1.

1. **Data input**

After extracting the data for each bank, the transcripts were consolidated into a single dataset. The Ninja API provides two formats: the full transcript text and a version segmented by speaker. Only the speaker-separated format was used, as it eliminates the need for further manual segmentation of the Q&A sections by speaker.

1. **Data processing and handling**

To ensure consistency, the data had to be harmonized both across institutions and within each institution. For example, speaker names were sometimes recorded in varying formats—for instance, with or without accents or hyphens—which required standardization. Special attention was given to Credit Suisse, as its acquisition by UBS introduced additional complexity. During this period, the CEO and CFO positions changed multiple times, with some roles taken over by executives from UBS, further necessitating careful alignment of speaker identities.

1. **Creation of message prompts**

To accurately detect and structure question-and-answer exchanges in earnings call transcripts, a detailed preprocessing pipeline is developed. The first step focuses on identifying analyst questions, initially by searching for paragraphs containing a "?" that are attributed to speakers labelled as "Analyst." If the speaker is not an analyst, then the paragraph is presumed to be a response. This rule helps distinguish between question and answer segments with minimal manual intervention. Interjections such as "Should we take the next question?" or calling on specific participants like “Pam?” or interventions done by Operator are also filtered out, as they serve only a functional role in the flow of the call.

Special handling is applied to cases where the operator, who usually opens and closes the Q&A section, is missing. When no operator is present, the Q&A section is inferred by using the first analyst question as the start point and extending it through the last presenter response. To ensure proper closure, the section is assumed to end when the final answer includes a typical ending phrase like “thank you,” which is commonly used to wrap up Q&A interactions. This logic helps capture the full exchange even in transcripts with inconsistent formatting or speaker tags.

The process accounts for edge cases as well, such as analyst questions that lack a question mark. In such scenarios, additional rules are applied: if a paragraph appears in the Q&A section, is from an analyst, longer than 16 tokens (to exclude closing remarks), and is flagged with flag\_question = 0, it may still be considered a valid query. This nuanced approach balances flexibility with precision, reducing false negatives in question detection.

To streamline the downstream analysis and generate model prompts, the data is enriched with a set of binary flags: flag\_sum\_question\_qa identifies whether a paragraph is a question in the Q&A section; flag\_analyst marks analyst contributions; flag\_operator identifies operational text; and flag\_presenter tags speakers who are neither analysts nor operators (typically management). Based on combinations of these flags, a new type column is created, classifying each entry as a 'query', 'answer', or null. Short segments under 20 words—often just transitions or acknowledgments—are filtered out, ensuring that only substantive content is retained.

Finally, a new dataframe is constructed to support the automation of message prompts for the Phi-4 model. This structure includes fields such as type (query or answer), person\_type (participant or presenter), name, job, and text. A dictionary is created for each exchange, pairing queries with their respective answers. This dictionary is then wrapped in a standardized system prompt to produce message\_final, the input structure required by the Phi-4 model.

This automated setup enables large-scale, consistent extraction of insights from financial Q&A sessions, identify discussed financial metrics, their level, mentioning of their trajectory, and whether the analyst’s question received a response.

1. **Run Phi4 model**

The next step, is to run the Phi-4 model on the processed question-and-answer exchanges stored in the df\_messages dataframe. This dataframe is structured to support model input seamlessly: each row corresponds to either a query or an answer, with accompanying metadata such as type (query or answer), person\_type (participant or presenter), name (speaker's name), and job (speaker's role, relevant especially for participants). The text column holds the original transcript content, while the message\_final column contains the full prompt—combining a standardized system instruction with the structured message\_user dictionary. This prompt is fed into the Phi-4 model to assess whether financial metrics are mentioned, if a level or trajectory is provided, and whether the analyst's question is directly answered. The model's response is stored in a new column, model\_output, which captures structured insights and enables further interpretation at scale.

1. **Combining model output results with input file to create final results**

The output is collected in the column model\_output for downstream analysis. Once all entries have been processed through the Phi-4 model, the enriched df\_messages—containing both the original Q&A structure and the model’s insights—is concatenated back with the original transcript dataframe. This merged dataset preserves both raw and processed information, ensuring traceability and interpretability. The final version is then saved as a JSON file, providing a structured and portable format for further exploration.