CAM_DS_401_Employer Project_P3_2025

Surfacing Supervisory Signals From Quarterly Bank Disclosures

Project Scope and Plan

Prepared by team JALCO Insights, October 2025

Word count: 1626

Contents

1. Introduction

The Prudential Regulation Authority (PRA) supervises UK financial institutions to ensure safety, soundness, and market stability. One challenge is extracting early warning signals from unstructured public disclosures, particularly quarterly earnings call transcripts. These contain rich qualitative data, but manual review is time-consuming and inconsistent.

This project explores how GenAI and NLP can surface supervisory indicators – such as liquidity concerns, governance tone, and strategic confidence – from these transcripts. Our goal was to develop a modular, regulator-ready pipeline that extracts supervisory-relevant insights, enables earlier risk detection, and packages outputs into formats that support supervisory workflows.

2. Methodology

A. Overall Approach

We built the model using HSBC, a Global Systemically Important Bank (G-SIB) with detailed disclosures and broad risk exposure: ideal for validating early-warning signal detection. Transcripts spanned Q3 2015 to Q1 2025.

Through iterative optimization, we developed a modular pipeline combining BERTopic, FinBERT, LLaMA 3.1, and Retrieval-Augmented Generation (RAG). These models were selected for their complementary strengths:

- **BERTopic** uncovers latent risk themes aligned with PRA frameworks, tracks their evolution, and distinguishes substantive discourse from boilerplate. Unlike keyword or LDA methods, it handles fragmented, conversational text: critical for mapping supervisory signals.
- **FinBERT** identifies finance-specific sentiment, with strong performance in domain language. When paired with contextual models, it helps surface hedging, uncertainty, and guarded tone that may precede quantitative deterioration.
- **LLaMA 3.1** offers long-context reasoning, linking statements across transcripts and mapping findings to PRA categories. It infers unstated implications and adapts to evolving taxonomies: ideal for extracting structured insights from unstructured disclosures.
- **RAG** retrieves and injects the most relevant transcript passages, ensuring grounded, citation-linked outputs. It improves recall and precision, reduces hallucinations, and synthesizes schema-correct insights without retraining.

Together, these models form a complementary, regulator-ready pipeline for surfacing granular, explainable risk signals (see Figure 1).

Figure 1: summary of models selected

DIMENSION	BERTOPIC ALONE	FINBERT ALONE	LLAMA 3.1 ALONE	ENSEMBLE
Coverage	Topic-level only	Document-level only	Chunk-level	✓ multi-granularity
Specificity	Generic themes	Binary sentiment	Risk-specific	✓ PRA-aligned taxonomy
Temporal	✓ Tracks trends	✗ No memory	✗ Single-shot	✓ BERTopic + time index
Implicit risks	✗ Surface keywords	✗ Misses context	✓ Reasoning	✓ LLaMA + FinBERT tone
Validation	Clustering metrics	Correlation w/ outcomes	✗ Subjective	✓ Multi-metric

B. Step-by-step process with *selected code snippets introduced in italics.*

00. Generating database

We built a structured transcript database from quarterly earnings calls stored in Google Drive. After authenticating access and downloading .docx files, transcripts were cleaned and serialized into JSONL format. A regex-based splitter segmented text into speaker-attributed blocks, classifying operators, presenters, and analysts while enriching metadata with firm, role, and position details. The final dataset is modular, speaker-level, and traceable: ready for topic modelling, sentiment scoring, and signal extraction.

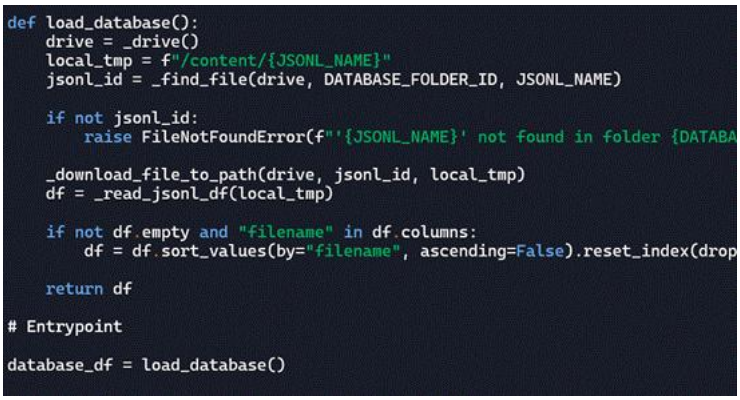
Figure 2: Segmenting transcripts into speaker-attributed blocks

```
def split_raw_blocks(text: str):
    blocks = []
    cur_name, cur_body = None, []
    lines = _clean_text_loose_for_split(text).split("\n")

    for ln in lines:
        s = ln.strip()
        if NAME_COLON_RE.match(s):
            flush()
            cur_name = NAME_COLON_RE.match(s).group(1).strip()
            cur_body = []
        elif EMDASH_INLINE_RE.match(s):
            flush()
            m = EMDASH_INLINE_RE.match(s)
            cur_name = m.group(1).strip()
            cur_body = [m.group(2).strip()]
        elif cur_name is not None:
            cur_body.append(s)

    flush()
    return [b for b in blocks if b.get("name") and b.get("text")]
```

01. Loading the Database

Structured transcript data was loaded into a pandas DataFrame to support downstream NLP workflows. This ensured consistency across transcripts and provided a clean foundation for topic modelling, sentiment analysis, and RAG integration.

Figure 3: Loading structured transcript data from Google Drive into a pandas DataFrame



```
def load_database():
    drive = _drive()
    local_tmp = f"/content/{JSONL_NAME}"
    jsonl_id = _find_file(drive, DATABASE_FOLDER_ID, JSONL_NAME)

    if not jsonl_id:
        raise FileNotFoundError(f"'{JSONL_NAME}' not found in folder {DATABA

    _download_file_to_path(drive, jsonl_id, local_tmp)
    df = _read_jsonl_df(local_tmp)

    if not df.empty and "filename" in df.columns:
        df = df.sort_values(by="filename", ascending=False).reset_index(drop

    return df

# Entrypoint

database_df = load_database()
```

02. Model Testing

We applied PRA-aligned regex patterns to label transcript blocks by supervisory risk category, then detected adverse signals using explicit phrases and proximity-based cues. Severity scores were calculated based on speaker role, numeric intensity, and multi-flag boosts. Summarization logic tracked novel spikes across quarters and selected high-severity blocks for manual verification.

Figure 4: Specifying risk categories and associated keywords

```
PRA_GLOSSARY = {
    "Credit risk": {
        "explanation": "Borrowers/counterparties fail to meet obligations.",
        "why_pra_cares": "Impairments & capital; loss absorbency.",
        "keywords": [
            r"\bcredit (?:risk|quality|loss(?:es)?)\b", r"\bimpairment(?:s)?
            r"\bprovision(?:s|ing)?\b", r"\bwrite[- ]?off[s]?\b",
            r"\bECLs?\b", r"\bIFRS ?9\b", r"\bSICR\b", r"\bstage ?[123]\b",
            r"\bNPLs?\b", r"\bn(?:on)?[- ]?performing loans?\b", r"\bdelinq
            r"\bforbearance\b", r"\bLGD\b", r"\bPD\b", r"\bEAD\b", r"\bcover
            r"\bLTV\b", r"\bcredit downgrade\b", r"\bCRE\b", r"\bcommercial
        ]
    },
    "Market risk": {
        "explanation": "Losses from rates, FX, spreads, equities, commoditi
        "why_pra_cares": "Trading book capital (VaR/SVaR/IRC) & PnL.",
        "keywords": [
            r"\bmarket (?:risk|volatility)\b", r"\bVaR\b", r"\bstressed VaR\
            r"\bIRC\b", r"\btrading book\b", r"\bFX\b", r"\bderivative[s]?\b
            r"\bhedg(?:e|ing)\b", r"\bCVA\b", r"\bFVA\b", r"\bDVA\b",
            r"\binterest[- ]?rate (?:move|shock|risk|increase|cut)s?\b",
            r"\bcredit spread(?:s)?\b", r"\bspread (?:widening|tightening)\b
        ]
    },
    "IRRBB (interest rate risk in banking book)": {
        "explanation": "Rate sensitivity of banking book (NII/EVE/CSRBB).",
        "why_pra_cares": "Earnings & value; structural hedge governance.",
        "keywords": [
            r"\bIRRBB\b", r"\bCSRBB\b", r"\bstructural hedge\b", r"\bduratio
            r"\bsensitivity\b", r"\bEVE\b", r"\bNII\b", r"\bgap sensitivity\
            r"\b\d+(\.\d+)?\s*(bp|bps|basis points?)\b", r"\b\d+(\.\d+)?\s*%
        ]
    },
    "Liquidity & Funding": {
        "explanation": "Ability to meet obligations; stable funding.",
        "why_pra_cares": "LCR/NSFR, ILAAP; outflow resilience.",
        "keywords": [
            r"\bLCR\b", r"\bNSFR\b", r"\bHQLA\b", r"\bliquidit(?:y|ies)\b",
            r"\bfunding (?:gap|mix|plan|costs?)\b", r"\bwholesale funding\b
            r"\bdeposit (?:flows|outflows|migration|beta|betas|mix)\b",
            r"\bILAAP\b", r"\bcontingent funding\b", r"\bstress outflow\b"
        ]
    },
    "Capital & Leverage": {
        "explanation": "Loss-absorbing capacity; constraints.",
        "why_pra_cares": "Solvency; distributions; MDA headroom.",
        "keywords": [
            r"\bCET1\b", r"\bAT1\b", r"\bTier ?1\b", r"\bTier ?2\b", r"\bRWA
            r"\bMREL\b", r"\bPillar ?2[AB]?\b", r"\bSREP\b",
            r"\bMDA\b", r"\bMDA (?:headroom|trigger)\b",
            r"\b(?:leverage|LR) ratio\b", r"\bbuffer[s]?\b", r"\bCCyB\b", r"
            r"\bICAAP\b", r"\bcapital (?:raise|return|shortfall|headroom)\b"
            r"\bbuy[- ]?back[s]?\b"
        ]
    },
    # ... [remaining clusters omitted for brevity, but available on request]
}
```

Figure 5: Labelling transcript blocks by supervisory risk category

```
def assign_labels(text):
    labels = []
    for category, patterns in PRA_GLOSSARY.items():
        for pat in patterns:
            if re.search(pat, text, re.IGNORECASE):
                labels.append(category)
                break  # avoid duplicate matches

    if not labels:
        labels = ["No signal"]
    return labels

# Apply to transcript blocks

records = []
for _, row in database_df.iterrows():
    fname = row["filename"]
    for block in row["preprocessed"]:
        text = block.get("text", "")
        labels = assign_labels(text)
        records.append({
            "filename": fname,
            "speaker_name": block.get("name"),
            "speaker_role": block.get("person_type"),
            "block_type": block.get("type"),
            "text": text,
            "labels": labels
        })

labeled_df = pd.DataFrame(records)
```

Figure 6: Detecting negative sentiment cues in proximity to specified risk terms

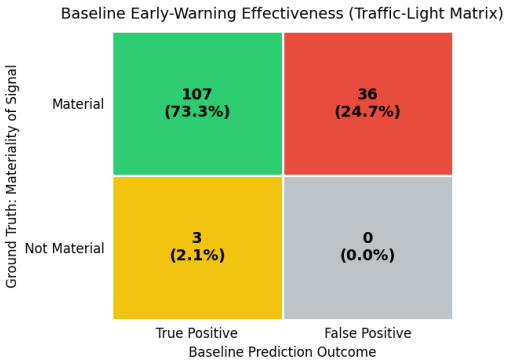
```
def is_adverse_for_category(text, category):
    text = text or ""
    # 1) Explicit adverse phrases

    if any_regex(ADVERSE_EXPLICIT.get(category, []), text):
        return True
    # 2) Proximity: anchors near negative cues or numeric deltas

    anchors = PRA_GLOSSARY.get(category, [])
    generic_cues = GENERIC_DOWN + GENERIC_UP + NUMERIC_DELTA + COST_PRESSURE
    if proximity_match(text, anchors, generic_cues, window_tokens=6):
        return True
    return False

def flag_adverse_categories(text, labels):
    flags = []
    for lab in labels:
        if lab not in PRA_GLOSSARY:
            continue
        if is_adverse_for_category(text, lab):
            flags.append(lab)
    return flags

labeled_df["early_warning_flags"] = labeled_df.apply(
    lambda r: flag_adverse_categories(r["text"], r["labels"]), axis=1
)
```

Figure 7: Establishing severity scoring for prioritization

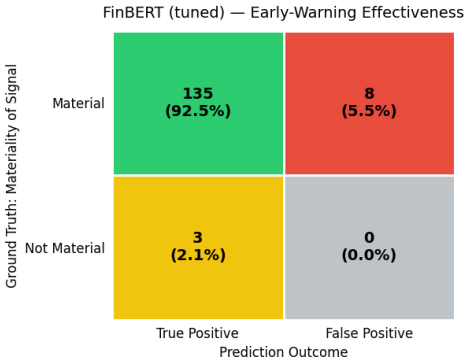
```python
def role_weight(role: str):
    role = (role or "").lower()
    if "presenter" in role:
        return 1.20
    if "participant" in role:
        return 0.95
    return 1.00

def numeric_severity(text: str):
    t = text or ""
    s = 1.0
    if re.search(r"\b\d{1,3}\s?bps\b", t, re.I): s *= 2.0
    if re.search(r"\b\d+(\.\d+)?\s?%|\bpercent\b", t, re.I): s *= 1.5
    if re.search(r"[$£€]\s?\d", t): s *= 1.4
    if re.search(r"\b\d+(\.\d+)?\s?(bn|billion|mn|million|m)\b", t, re.I):
    elif re.search(r"\b\d{1,3}(\.\d+)?\b", t): s *= 1.1
    if re.search(INTENSIFIERS, t, re.I): s *= 1.2
    if re.search(HEDGES, t, re.I): s *= 0.9
    return float(np.clip(s, 1.0, 5.0))

def multi_flag_boost(early_flags):
    n = len(early_flags) if isinstance(early_flags, list) else 0
    return 1.0 + min(0.1 * max(0, n-1), 0.3)

def compute_severity(text, role, early_flags):
    if not early_flags or len(early_flags) == 0:
        return 0.0
    return numeric_severity(text) * role_weight(role) * multi_flag_boost(ea

labeled_df["severity_score"] = labeled_df.apply(
    lambda r: compute_severity(r["text"], r.get("speaker_role",""), r.get("
    axis=1
)
```

We also benchmarked performance metrics – precision, recall, and F1 – against a manually validated golden set (see Figure 8). FinBERT reached 94.4% recall, and LLaMA 3.1 achieved 100%, validating our choice to center the model around this pairing.

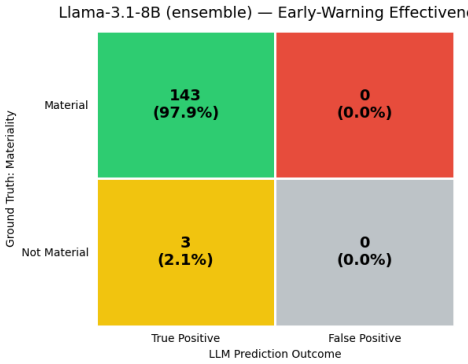
Figure 8. Performance by model



Precision	73.3%
Recall	74.8%
F1	74.0%



Precision	92.5%
Recall	94.4%
F1	93.4%



Precision	97.9%
Recall	100.0%
F1	99.0%

03. Topic Extraction, Sentiment & Summarization

We paired analyst questions with management responses and segmented text using a sliding four-sentence window, yielding 2,737 analytical chunks. BERTopic was applied using transformer embeddings, UMAP, and HDBSCAN to uncover domain-specific themes. Seed topics guided clustering, and low-frequency terms were pruned to improve coherence. Topics were auto-labeled using our PRA-aligned glossary.

FinBERT generated sentiment probabilities (positive, neutral, negative) for each chunk, aggregated by topic and time. To mitigate false neutrals, LLaMA 3.1 re-scored ambiguous chunks, identifying implicit negatives and producing concise topic summaries and stance rationales. This enriched the analysis with interpretable tone insights across both thematic and sentiment dimensions.

Figure 9. Pairing analyst questions with management responses

```
def pair_qa_ignore_leading_answers(turns_df: pd.DataFrame) → pd.DataFrame:
    pairs = []
    for fname, g in turns_df.groupby("filename", sort=False):
        g = g.sort_values("seg_idx").reset_index(drop=True)
        i = 0
        while i < len(g) and not (g.at[i,"person_type"]=="participant" and g
            i += 1
        while i < len(g):
            row = g.loc[i]
            if row["person_type"]=="participant" and row["turn_type"]=="quer
                q_idx = int(row["seg_idx"]); q_speaker=row["speaker"]; q_tex
                i += 1
                a_texts, a_speakers, a_indices = [], [], []
                while i < len(g) and g.at[i,"person_type"]=="presenter" and
                    nxt = g.loc[i]
                    if nxt["text"]: a_texts.append(nxt["text"])
                    if nxt["speaker"]: a_speakers.append(nxt["speaker"])
                    a_indices.append(int(nxt["seg_idx"]))
                    i += 1
                answer = ("\n\n".join(a_texts)).strip() or pd.NA
                pairs.append({
                    "bank_id": row["bank_id"],
                    "filename": fname,
                    "q_seg_idx": q_idx,
                    "q_speaker": q_speaker,
                    "query": q_text,
                    "a_seg_idx": (min(a_indices) if a_indices else pd.NA),
                    "a_speakers": ("; ".join(a_speakers) if a_speakers else
                    "answer": answer,
                })
            else:
                i += 1
    return pd.DataFrame(pairs).sort_values(["filename","q_seg_idx"]).reset_i
```

Figure 10. Flagging risk categories in management responses, alongside negation cues

```
def scan_text_with_glossary_fast(text: str):
    hits = []
    tokens, starts = pretokenize(text)
    for risk, meta in COMPILED_GLOSSARY.items():
        for rgx in meta["patterns"]:
            for m in rgx.finditer(text):
                tok_idx = token_index_for_char(starts, m.start())
                neg = is_negated_tokens(tokens, tok_idx)
                snippet = re.sub(r"\s+", " ", text[max(0, m.start()-80): m.e
                hits.append({
                    "risk": risk,
                    "match": m.group(0),
                    "negated": bool(neg),
                    "snippet": snippet,
                    "explanation": meta["explanation"],
                    "why_pra_cares": meta["why_pra_cares"],
                })
    return hits
```

Figure 11. Combining FinBERT sentiment scoring with LLaMA overrides to surface hidden negatives and explain tone.

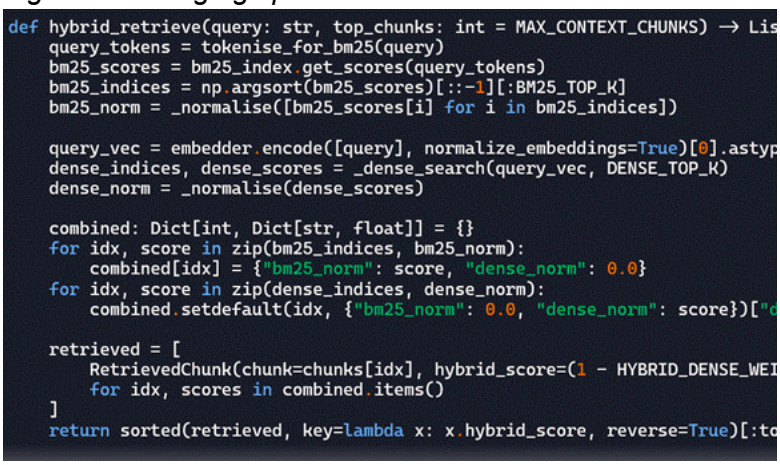
```
METRIC_NEARBY = re.compile(
    r"\b(CET1(?:\s*ratio)?|Tier ?1|AT1|Tier ?2|RWA|LCR|NSFR|NPLs?|ECLs?|NII|
    r"(?:[^0-9%\-\u2212]{0,20})"
    r"([\--\u2212]?\d+(?:\.\d+)?)?"
    r"\s*(%|bp|bps)?",
    re.I
)

metric_rows = []
for r in work.itertuples(index=False):
    t = r.text or ""
    for m in METRIC_NEARBY.finditer(t):
        raw_term, num, unit = m.group(1), m.group(2), m.group(3)
        if isinstance(num, str):
            num = num.replace("-", "-").replace("\u2212", "-")
        metric_rows.append({
            "bank_id": r.bank_id,
            "filename": r.filename,
            "q_seg_idx": r.q_seg_idx,
            "a_seg_idx": r.a_seg_idx,
            "term_raw": raw_term,
            "term": canon_term(raw_term),
            "value": (float(num) if num not in (None, "") else np.nan),
            "unit": (unit or "").lower(),
            "snippet": re.sub(r"\s+", " ", t[max(0, m.start()-60): m.end()+6
        })
metric_hits_df = pd.DataFrame(metric_rows)
```

04. RAG Chatbot

We implemented a RAG chatbot for PRA supervisors to explore risk signals in transcripts. It integrates sparse retrieval (BM25) for keyword precision and dense semantic retrieval (BGE embeddings) for contextual relevance. Retrieved chunks are ranked using hybrid scoring and passed to a grounded LLM prompt that generates concise, citation-linked bullet-point answers.

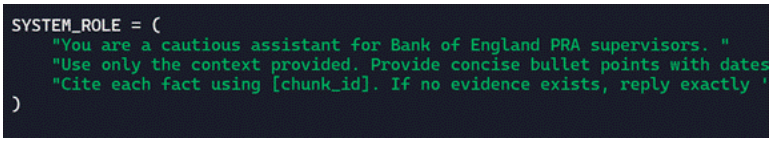
Each response is transparently supported by verifiable source text, enabling supervisors to query by risk category, quarter, institution, or speaker with full traceability. The chatbot transforms static transcript corpora into an interactive, regulator-grade Q&A environment.

Figure 12. Merging sparse and dense scores to surface relevant transcript chunks



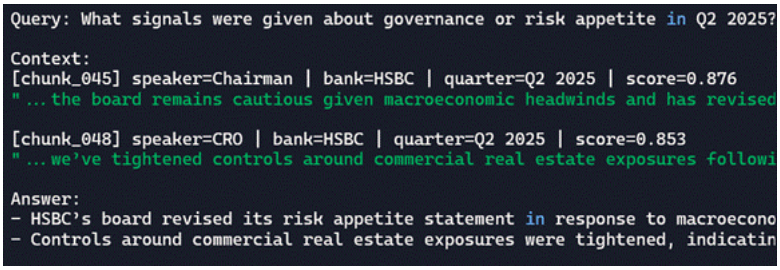
```
def hybrid_retrieve(query: str, top_chunks: int = MAX_CONTEXT_CHUNKS) -> Lis
    query_tokens = tokenise_for_bm25(query)
    bm25_scores = bm25_index.get_scores(query_tokens)
    bm25_indices = np.argsort(bm25_scores)[::-1][:BM25_TOP_K]
    bm25_norm = _normalise([bm25_scores[i] for i in bm25_indices])

    query_vec = embedder.encode([query], normalize_embeddings=True)[0].astyp
    dense_indices, dense_scores = _dense_search(query_vec, DENSE_TOP_K)
    dense_norm = _normalise(dense_scores)

    combined: Dict[int, Dict[str, float]] = {}
    for idx, score in zip(bm25_indices, bm25_norm):
        combined[idx] = {"bm25_norm": score, "dense_norm": 0.0}
    for idx, score in zip(dense_indices, dense_norm):
        combined.setdefault(idx, {"bm25_norm": 0.0, "dense_norm": score})["d

    retrieved = [
        RetrievedChunk(chunk=chunks[idx], hybrid_score=(1 - HYBRID_DENSE_WEI
        for idx, scores in combined.items()
    ]
    return sorted(retrieved, key=lambda x: x.hybrid_score, reverse=True)[:to
```

Figure 13. Prompting the assistant to behave as a cautious analyst responding with grounded evidence



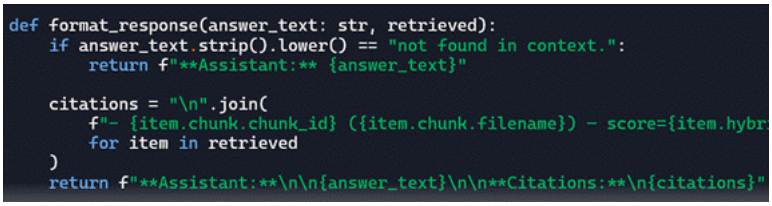
```
SYSTEM_ROLE = (
    "You are a cautious assistant for Bank of England PRA supervisors. "
    "Use only the context provided. Provide concise bullet points with dates
    "Cite each fact using [chunk_id]. If no evidence exists, reply exactly '
)
```

Figure 14. Chatbot prompts querying supervisory signals by risk category and quarter



```
Query: What signals were given about governance or risk appetite in Q2 2025?

Context:
[chunk_045] speaker=Chairman | bank=HSBC | quarter=Q2 2025 | score=0.876
"...the board remains cautious given macroeconomic headwinds and has revised

[chunk_048] speaker=CRO | bank=HSBC | quarter=Q2 2025 | score=0.853
"...we've tightened controls around commercial real estate exposures followi

Answer:
- HSBC's board revised its risk appetite statement in response to macroecono
- Controls around commercial real estate exposures were tightened, indicatin
```

Figure 15. Outputting responses and citation metadata



```
def format_response(answer_text: str, retrieved):
    if answer_text.strip().lower() == "not found in context.":
        return f"**Assistant:** {answer_text}"

    citations = "\n".join(
        f"- {item.chunk.chunk_id} ({item.chunk.filename}) - score={item.hybr
        for item in retrieved
    )
    return f"**Assistant:**\n\n{answer_text}\n\n**Citations:**\n{citations}"
```

05. Interactive dashboard

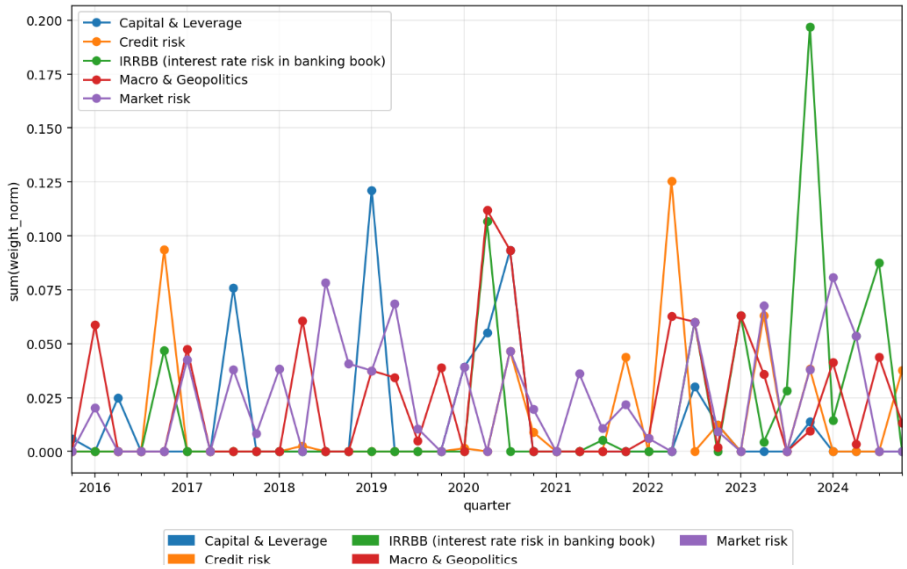
Finally, we prototyped an Azure SQL dashboard for supervisory colleagues to access and interact with the data. Here is a prototype, which is being developed further for our presentation: https://supervisorinsightclient-g7enhkgyfydhcdac.westus3-01.azurewebsites.net/.

3. Results

A. Spotting long-range trends

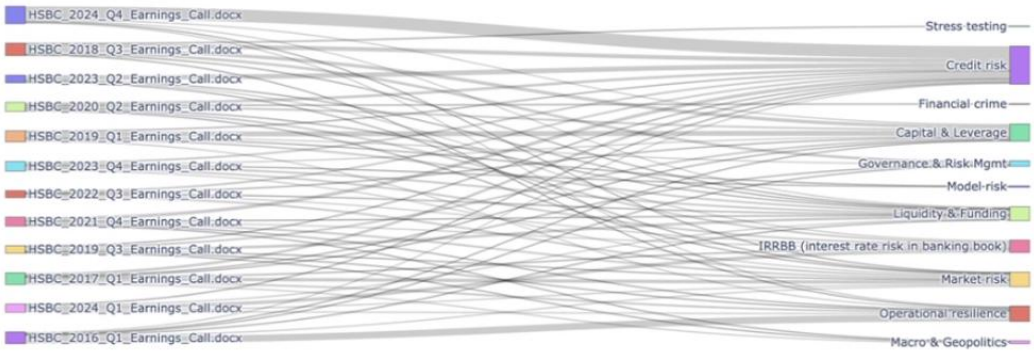
Analysis showed that the top 5 areas in terms of risk signals from 2015 to 2025 were: Capital & Leverage, Credit Risk, IRRBB (interest rate risk in banking book), Macro & Geopolitics and Market Risk. The biggest spikes related to IRRBB, Credit Risk, Capital & Leverage.

Figure 16. Risk weighting over time (top 5 areas)



Meanwhile, the Sankey diagram below shows that Credit Risk appeared in more individual quarters than any other category, followed by Capital & Leverage.

Figure 17. Mentions by quarter (all risk areas)

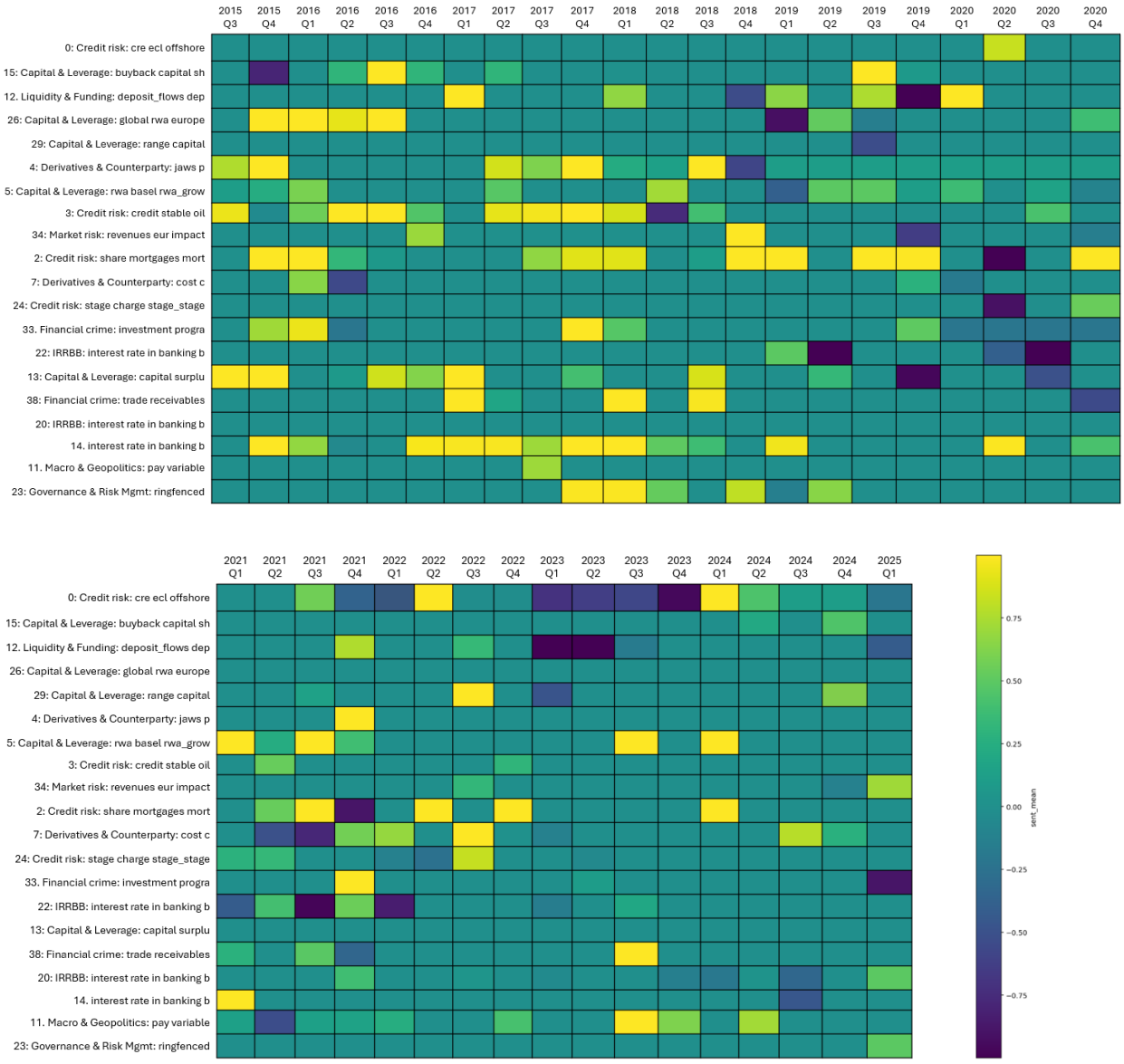


B. Heatmapping shifting sentiment by quarter and topic

To enable a more granular view, our model helps visualize trouble spots where a sharp shift in sentiment versus baseline may warrant particular attention. Areas of dark blue indicate greatest negativity. Looking at the past five years (2021-Q1 to 2025-Q1), notable instances relate to:

- Credit Risk: with negative sentiment throughout 2023 disclosures and in Q4 2021
- Liquidity & Funding: a negative-sentiment topic in Q1 and Q2 2023
- Derivatives & Counter Party: flagged in Q2-Q3 2021
- IRRBB: flagged in Q3 2021 and Q2 2022
- Financial Crime: flagged in Q1 2025

Figure 18. Heatmap of mean sentiment by topic and quarter (top 20 topics)



C. Spotting guardedness, stress and uncertainty signals in management responses

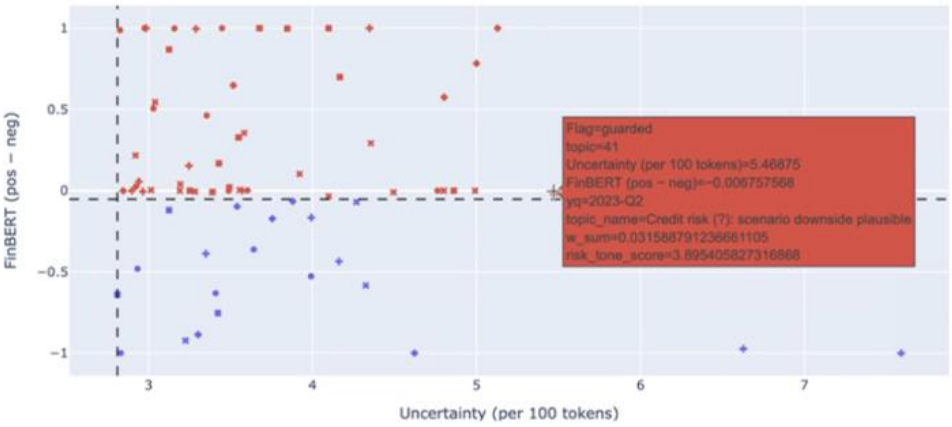
Beyond negative sentiment, our model was also able to flag moments of stress of guardedness in executive disclosures. The table below shows notable instances flagged during the past five years (2021-Q1 to 2025-Q1), with Liquidity & Funding, Credit Risk and IRRBB once again standing out as noteworthy topics for HSBC.

Figure 19. Instances of executive stress & guardedness

Area	Quarter	Topic	Tone	Confidence	Score	Evidence	Explanation
Liquidity & Funding	2023 – Q1	Liquidity & funding: dep	Guarded	0.25	1.65	1% per month migration into term deposits	Term deposit growth may decide liquidity
Credit risk	2023 – Q2	Liquidity & Funding: dep	Guarded	0.25	1.38	Subdued loan growth	Subdued loan growth may impact credit portfolio
Credit risk	2023 – Q2	Credit risk: cre ecl offsho	Guarded	0.35	1.42	Economic uncertainty, China's commercial real estate	Economic uncertainty impacts ECL
Credit risk	2023 – Q2	Credit risk: scenario	Guarded	0.45	1.31	Plausible downside scenario, $0.7 billion remaining	Potential loan impairment
Liquidity & Funding	2023 – Q2	Liquidity & Funding: dep	Guarded	0.35	1.41	Deposit costs, lagged migration to time deposits	Deposit costs and migration uncertainty impact funding conditions
Liquidity & Funding	2023 – Q3	Liquidity & Funding: dep	Guarded	0.35	1.34	Margin compression, deposit migration	Pressure on reposits may impact liquidity
Liquidity & Funding	2024 – Q3	IRRBB (interest rate ris	Stress	0.35	1.40	Argentina exclusion from NII guidance	Argentine-related uncertainty impacts NII

The chart below, meanwhile, surfaced multiple instances where there was negative sentiment regarding a topic alongside 'hedging' words from managers, reflecting uncertainty. The example shown calls out Credit Risk in Q2 2023, which corroborates with the insight shown above. Note, in our codebook, this chart is interactive, enabling further exploration of each instance surfaced. This will be showcased in our technical walkthrough.

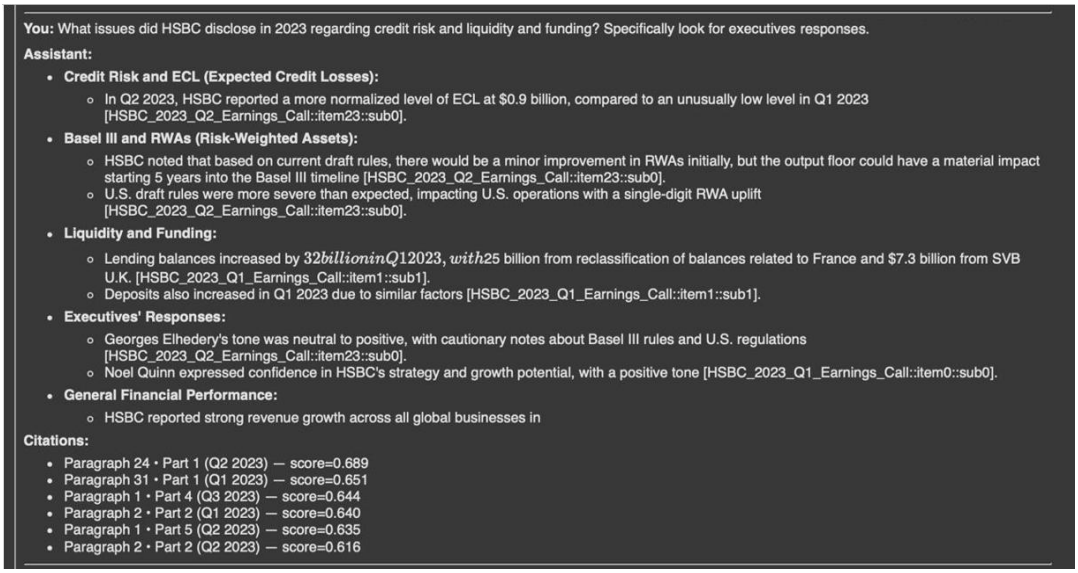
Figure 20. Instances of negative sentiment with uncertainty signals



D. Generating contextual summaries on demand

Finally, our RAG-powered chatbot helped drill into flagged topics to understand what was discussed and in which precise moments.

Figure 21. Example chatbot output



You: What issues did HSBC disclose in 2023 regarding credit risk and liquidity and funding? Specifically look for executives responses.

Assistant:

- **Credit Risk and ECL (Expected Credit Losses):**
  - In Q2 2023, HSBC reported a more normalized level of ECL at $0.9 billion, compared to an unusually low level in Q1 2023 [HSBC_2023_Q2_Earnings_Call::item23::sub0].
- **Basel III and RWAs (Risk-Weighted Assets):**
  - HSBC noted that based on current draft rules, there would be a minor improvement in RWAs initially, but the output floor could have a material impact starting 5 years into the Basel III timeline [HSBC_2023_Q2_Earnings_Call::item23::sub0].
  - U.S. draft rules were more severe than expected, impacting U.S. operations with a single-digit RWA uplift [HSBC_2023_Q2_Earnings_Call::item23::sub0].
- **Liquidity and Funding:**
  - Lending balances increased by $32 billion in Q1 2023, with 25 billion from reclassification of balances related to France and $7.3 billion from SVB U.K. [HSBC_2023_Q1_Earnings_Call::item1::sub1].
  - Deposits also increased in Q1 2023 due to similar factors [HSBC_2023_Q1_Earnings_Call::item1::sub1].
- **Executives' Responses:**
  - Georges Elhedery's tone was neutral to positive, with cautionary notes about Basel III rules and U.S. regulations [HSBC_2023_Q2_Earnings_Call::item23::sub0].
  - Noel Quinn expressed confidence in HSBC's strategy and growth potential, with a positive tone [HSBC_2023_Q1_Earnings_Call::item0::sub0].
- **General Financial Performance:**
  - HSBC reported strong revenue growth across all global businesses in

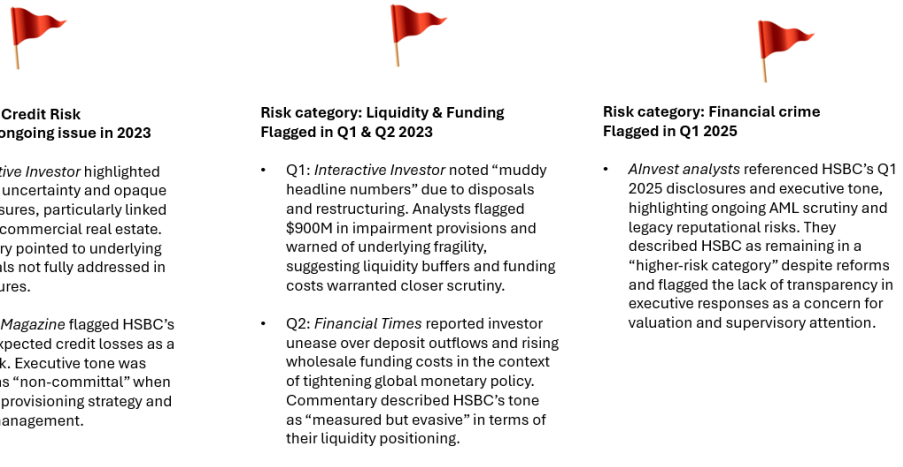
Citations:

- Paragraph 24 • Part 1 (Q2 2023) — score=0.689
- Paragraph 31 • Part 1 (Q1 2023) — score=0.651
- Paragraph 1 • Part 4 (Q3 2023) — score=0.644
- Paragraph 2 • Part 2 (Q1 2023) — score=0.640
- Paragraph 1 • Part 5 (Q2 2023) — score=0.635
- Paragraph 2 • Part 2 (Q2 2023) — score=0.616

E. Model performance

We cross-referenced the signals surfaced by our final model against real-world events to see what financial commentators wrote about them at the time. The chart below shows examples from the past 5 years where there were indeed concerns raised about the topics flagged, with comments even noting a sense of uncertainty or guarded executive behavior when challenged, supporting the validity of our model.

Figure 22. Flagged events in the news



Risk category: Credit Risk
Flagged as an ongoing issue in 2023

- Q2: *Interactive Investor* highlighted heightened uncertainty and opaque credit exposures, particularly linked to Chinese commercial real estate. Commentary pointed to underlying stress signals not fully addressed in the disclosures.
- Q3: *Shares Magazine* flagged HSBC's $1.1bn in expected credit losses as a material risk. Executive tone was described as "non-committal" when pressed on provisioning strategy and exposure management.

Risk category: Liquidity & Funding
Flagged in Q1 & Q2 2023

- Q1: *Interactive Investor* noted "muddy headline numbers" due to disposals and restructuring. Analysts flagged $900M in impairment provisions and warned of underlying fragility, suggesting liquidity buffers and funding costs warranted closer scrutiny.
- Q2: *Financial Times* reported investor unease over deposit outflows and rising wholesale funding costs in the context of tightening global monetary policy. Commentary described HSBC's tone as "measured but evasive" in terms of their liquidity positioning.

Risk category: Financial crime
Flagged in Q1 2025

- *AInvest analysts* referenced HSBC's Q1 2025 disclosures and executive tone, highlighting ongoing AML scrutiny and legacy reputational risks. They described HSBC as remaining in a "higher-risk category" despite reforms and flagged the lack of transparency in executive responses as a concern for valuation and supervisory attention.

4. Conclusion

Our pipeline successfully transformed raw earnings call transcripts into structured, supervisory-relevant warning signals. By combining PRA-aligned glossary scanning, adverse signal detection, severity scoring, and sentiment overrides, it surfaced early-warning indicators with traceable logic. The modular architecture enabled granular analysis – from block-level tagging to quarter-on-quarter trend visualization – while maintaining full auditability.

A key strength lies in its regulator-ready design: every output is grounded in transcript evidence, with citations, speaker roles, and temporal metadata preserved. The workflow supports manual verification, metric extraction, and interactive querying via a RAG chatbot, making it scalable and transparent.

To validate generalizability, the next step would be testing the pipeline on a second bank. This would confirm robustness across disclosure styles and speaker dynamics and support comparative supervisory analysis.

5. Recommended Next Steps

To embed the pipeline into supervisory practice, we recommend aligning outputs with existing review cycles – such as quarterly firm assessments, sector-wide stress reviews, or thematic deep-dives. Integrating it into analyst workflows would enable faster triage, consistent signal tracking, and richer evidence packs.

For scaling, the pipeline should be extended across multiple banks and reporting periods. Deployment options include secure hosting within PRA infrastructure, with the RAG chatbot available for real-time querying and traceable insight generation.

Further refinement should focus on expanding the glossary over time, improving sentiment calibration, and enhancing metric extraction to capture directional shifts. Embedding feedback loops from supervisory teams will help tune severity scoring and reduce noise. Ultimately, the goal is not just automation – but activation: turning transcript data into supervisory foresight, embedded into business-as-usual.