CAM DS 401 Employer Project: Solution Pitch

The Alchemists

Data Preparation

- Data selection
- Scraping
- Correcting
- Formatting
- Preprocessing

Data Selection

Financial Results Press Releases

Earnings Calls Presentations

Earnings Calls Transcripts → Specifically Q & A Section

Conference Presentation Transcripts

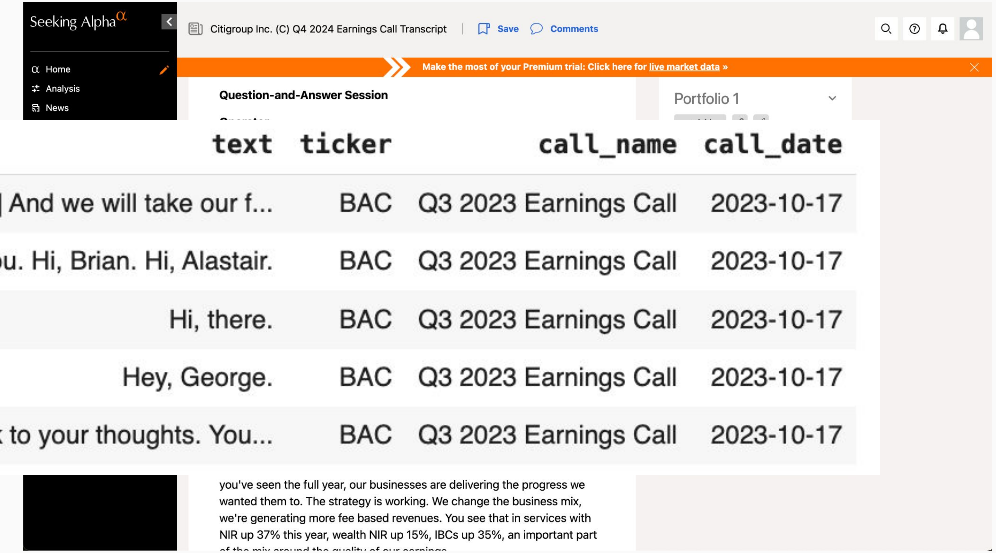
Data Selection

Bucket	G-SIB
5 (3.50%)	(Empty)
4 (2.50%)	JP Morgan Chase ←
3 (2.0%)	Citigroup ←
	HSBC ←
2 (1.50%)	Agricultural Bank of China
	Bank of America ←
	Bank of China
	Barclays ←

Scraping

```
(function() {
    let data = [];
    let foundQASession = false;
    let l
    let l
speak
    doc
        /
        i

p.que
Answ
            foundQASession = true;
        }
        return;
```

	person	role	text	ticker	call_name	call_date
0	Operator	operator	[Operator Instructions] And we will take our f...	BAC	Q3 2023 Earnings Call	2023-10-17
1	Gerard Cassidy	questioner	Thank you. Hi, Brian. Hi, Alastair.	BAC	Q3 2023 Earnings Call	2023-10-17
2	Brian Moynihan	answerer	Hi, there.	BAC	Q3 2023 Earnings Call	2023-10-17
3	Alastair Borthwick	answerer	Hey, George.	BAC	Q3 2023 Earnings Call	2023-10-17
4	Gerard Cassidy	questioner	Brian, can you come back to your thoughts. You...	BAC	Q3 2023 Earnings Call	2023-10-17



Correcting & Formatting

```
A – Adeel Khan: BCS
A – Andrew Coombs: BCS
A – Angela Cross: BCS
A – Ann
A – C.S
A – Den
A – Ric
A – Tus
A – Vim
Adam Te
Adeel K
Alastai
Alastai
Alvaro
Alvaro
Aman Ra
Amandee
Amit Goel: HSBC, BCS
Andrew Coombs: HSBC, BCS
Andrew Lim: C, GS, JPM
AndrewLim: JPM
Angela Cross: BCS
```

```python
# Remove rows where 'text' is 'Chart' or 'You can now buy 1 year of Seeking Alpha Premium for whomever you like.'
combined_df = combined_df[~combined_df['text'].isin(['Chart', 'You can now buy 1 year of Seeking Alpha Premium for whomever you l
```

	person	role	text	ticker	call_name	call_date	company
6193	Noel Quinn	answerer	Ewen?	HSBC	Q1 2021 Earnings Call	2021-04-27	HSBC
621	Mark Mason	answerer	Those things matter and impact the SCB. We tal...	C	Q3 2023 Earnings Call	2023-10-13	Citigroup
6865	Brian Moynihan	answerer	So, it's relentless and sustainable, you know,...	BAC	Q4 2024 Earnings Call	2025-01-16	Bank of America
1500	Kenneth Usdin	questioner	So it's a little -- so is it straight line or ...	BAC	Q4 2023 Earnings Call	2024-01-12	Jefferies
5836	Tushar Morzaria	answerer	Yes. Thanks, Omar. Again, why don't we do it t...	BCS	Q2 2021 Earnings Call	2021-07-28	Barclays

```python
'Michael Mayo': 'Michael Mayo',
'Mike Mayo': 'Michael Mayo',
'MikeMayo': 'Michael Mayo',
'Rob Noble': 'Robert Noble'
```

84	Unknown	Unknown
85	Vim Maru	Barclays
86	Vivek Juneja	JPMorgan

Preprocessing

NLTK stopwords library

Additional stopwords

Remove non-alphabetic chars.

Lowercase all

Lemmatisation

```python
stop_words = set(stopwords.words('english'))
additional_stopwords = {
    'weve', 'well', 'thats', 'look', 'going', 'think', 'would', 'give', 'obviously', 'expect', 'could', 'should',
    'year', 'chief', 'officer',
    'bank', 'financial', 'earnings', 'quarter', 'million', 'percent', 'revenue', 'guidance',
    'question', 'answer', 'operator', 'conference', 'call', 'analyst', 'presentation', 'discuss', 'discussion', 'look', 'going', 'ahead',
    'company', 'business', 'market', 'performance', 'customers', 'clients', 'global', 'team', 'industry',
    'good morning', 'morning', 'afternoon', 'evening', 'welcome', 'thanks', 'thank', 'appreciate', 'everyone', 'joining', 'talk', 'today', 'yesterday', 'tomorrow', 'time', 'end', 'start',
    'latest', 'update', 'updated', 'last', 'next', 'current', 'previous', 'ongoing',
    'executive', 'management', 'leadership', 'board', 'investors', 'shareholders',
    'cost', 'expense', 'profit', 'loss', 'capital', 'return', 'equity', 'assets', 'liabilities',
    'net', 'gross', 'adjusted', 'margin', 'tax', 'credit', 'debt', 'liquidity', 'dividend', 'investment', 'stock', 'valuation', 'price',
    "actual", "actually", "additionally", "afterwards", "almost", "already", "also", "although", "always", "amazing",
    "anew", "anyhow", "anymore", "anyway", "anywhere", "approximately", "aptly", "arbitrarily", "arguably", "around",
    "as", "aside", "assuredly", "at", "attentively", "automatically", "basically", "barely", "becoming", "before",
    "behind", "below", "beneath", "besides", "best", "better", "beyond", "briefly", "certainly", "clearly",
    "closely", "coincidentally", "collectively", "completely", "conclusively", "consequently", "considerably", "consistently", "conspicuously", "constantly",
    "conversely", "correctly", "currently", "daringly", "dearly", "definitely", "deliberately", "deservedly", "despite", "deterministically",
    "differently", "doubtfully", "dramatically", "easily", "effectively", "effortlessly", "elsewhere", "eminently", "enormously", "entirely",
    "equally", "especially", "essentially", "even", "evenly", "eventually", "exclusively", "explicitly", "extensively", "extremely",
    "fairly", "faintly", "faithfully", "famously", "far", "fast", "fatally", "feasibly", "finally", "first",
    "fluently", "forcibly", "fortunately", "frankly", "frequently", "fully", "generally", "genuinely", "globally", "good",
    "gradually", "greatly", "hard", "hardly", "headlong", "here", "hereafter", "hereby", "herein", "hereupon",
    "highly", "honestly", "hopefully", "hugely", "immediately", "implicitly", "in", "inadvertently", "incessantly", "indeed",
    "indefinitely", "independently", "indirectly", "increasingly", "inclusively", "incontrovertibly", "infinitely", "infrequently", "initially",
    "inordinately", "inside", "instantly", "instead", "intensely", "intently", "interestingly", "internally", "invariably", "inwardly",
    "just", "largely", "later", "lately", "least", "leisurely", "likely", "limitedly", "literally", "loosely", "like", "growth"
    "long", "longingly", "loyally", "mainly", "marginally", "maybe", "meagerly", "merely", "methodically", "mightily",
    "more", "moreover", "mostly", "much", "namely", "naturally", "nearly", "necessarily", "needfully", "next",
    "nicely", "noisily", "notably", "notwithstanding", "obviously", "occasionally", "oddly", "of", "off", "often",
    "only", "openly", "optimally", "or", "orderly", "ordinarily", "otherwise", "ought", "out", "outside",
    "overall", "overwhelmingly", "partially", "particularly", "perhaps", "per", "perpetually", "periodically", "pervasively", "phenomenally",
    "plausibly", "politely", "possibly", "positively", "powerfully", "precisely", "presently", "primarily", "probably", "promptly",
    "properly", "punctually", "purely", "quite", "rapidly", "readily", "really", "reasonably", "recently", "regularly",
    "relatively", "remarkably", "repeatedly", "respectively", "right", "roughly", "scarcely", "second", "secondly", "secretly",
    "seemingly", "seldom", "self", "selfishly", "sensibly", "seriously", "severely", "sharply", "shortly", "significantly",
    "similarly", "simply", "slowly", "smoothly", "somehow", "somewhat", "somewhere", "soon", "specifically", "steadily",
    "billion", "rate", "term", "youre", "make", "come", "continue", "deposit", "back", "theres", "sort"
    "strangely", "sufficiently", "surprisingly"
}

name_stopwords = {'adam', 'terelak', 'adeel', 'khan', 'alastair', 'borthwick', 'alastair', 'warr', 'alvaro', 'serrano', 'amandeep', 'rakkar', 'amit', 'goel',
                  'andrew', 'coombs', 'andrew', 'lim', 'anna', 'cross', 'benjamin', 'toms', 'betsy', 'graseck', 'brennan', 'hawken', 'brian', 'kleinhanzl', 'brian',
                   'moynihan', 'c.', 's.', 'venkatakrishnan', 'carey', 'halio', 'charles', 'peabody', 'charmsol', 'yoon', 'chris', 'cant', 'chris', 'hallam', 'christian',
                   'bolu', 'christopher', 'cant', 'christopher', 'kotowski', 'daniel', 'fannon', 'david', 'solomon', 'dennis', 'coleman', 'denny', 'nealon', 'devin', 'ryan',
                   'ebrahim', 'poonawala', 'edward', 'firth', 'erika', 'najarian', 'ewen', 'stevenson', 'fahed', 'kunwar', 'georges', 'elhedery', 'gerard', 'cassidy', 'glenn',
                   'schorr', 'gurpreet', 'singh', 'sahi', 'guy', 'stebbings', 'jamie', 'dimon', 'jane', 'fraser', 'jason', 'napier', 'jeff', 'harte', 'jennifer', 'landis', 'jennifer',
                   'piepszak', 'jeremy', 'barnum', 'jeremy', 'hugh', 'jeremy', 'sigee', 'jes', 'staley', 'jim', 'mitchell', 'john', 'mcdonald', 'jon', 'bingham', 'jonathan',
                   'pierce', 'joseph', 'dickerson', 'katherine', 'lei', 'kenneth', 'usdin', 'kian', 'abouhossein', 'lee', 'mcentire', 'manan', 'gosalia', 'manus', 'costello', 'mark',
                   'mason', 'martin', 'leitgeb', 'matthew', "o'connor", 'michael', 'mayo', 'mike', 'carrier', 'mike', "o'connor", 'nicholas', 'lord', 'noel', 'quinn', 'omar', 'keenan',
                   'paul', 'donofrio', 'perlie', 'mong', 'raul', 'sinha', 'richard', "o'connor", 'robert', 'noble', 'robin', 'down', 'rohith', 'chandra-rajan', 'ryan', 'kenny', 'saul',
                   'martinez', 'scott', 'siefers', 'sheng', 'wang', 'stephen', 'scherr', 'steven', 'chubak', 'thomas', 'rayner', 'tushar', 'morzaria', 'unknown', 'vim', 'maru', 'vivek',
                   'juneja', 'yafei', 'tian', 'unidentified', 'company', 'representative'}
```

Additional: Adding Stock Price Movements

```python
# Dictionary to store DataFrames
stock_histories = {}

for ticker in unique_ tickers:
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={ticker}&outputsize=full&apikey=T3A3JGIPVTORDC5V"

    response = requests.get(url).json()
    history = response.get("Time Series (Daily)", {})

    if not history:
```

person
role
text
ticker
call_date
call_q
call_year
bank pre_earnings_price
post_earnings_price
earnings_period_price_change
call_day_open
call_day_close
call_day_price_change

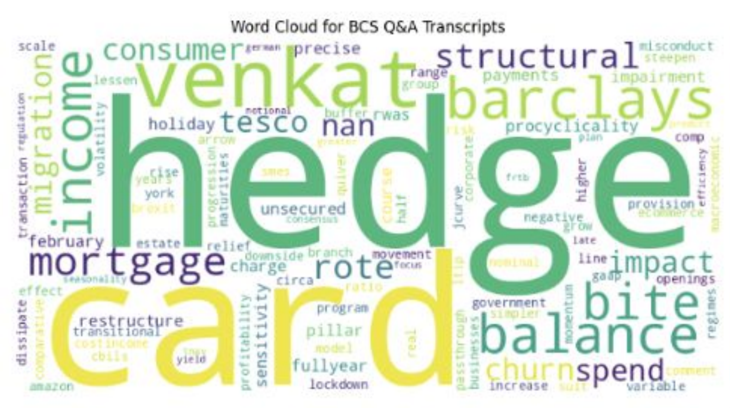
	ticker	call_date	call_day_open	call_day_close	pre_earnings_price	post_earnings_price
4	BAC	2023-10-17	27.95	27.62	26.90	26.31
1704	BAC	2024-01-12	32.25	32.80	34.43	31.80
6038	BAC	2021-07-14	39.11	38.86	40.04	36.93
6243	BAC	2022-01-17	33.36	33.62	29.86	34.95
5546	BAC	2022-04-18	37.42	38.85	38.82	37.56

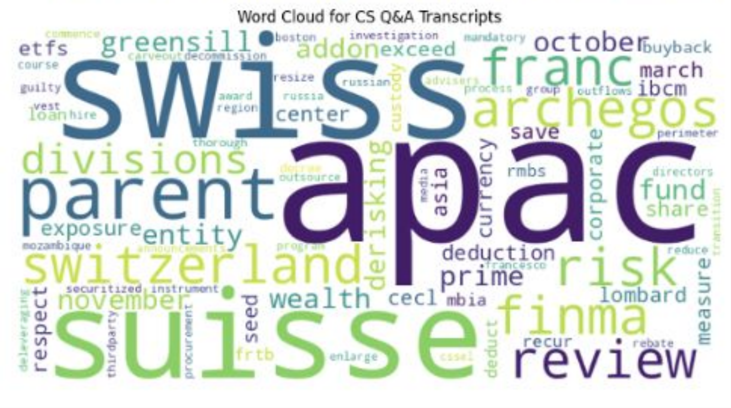
```python
    filename = f"{ticker}_stock_history.json"
    with open(filename, "w") as f:
        json.dump(formatted_data, f, indent=4)  # Pretty-print for readability

    stock_histories[ticker] = pd.DataFrame.from_dict(formatted_data, orient="index")

    print(f"Saved {filename}")
```

Initial Data Exploration

- Wordclouds
- Most common words
- BERTopic

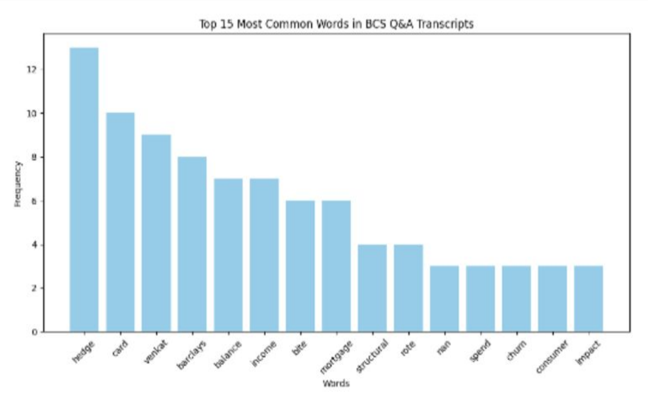
Wordclouds

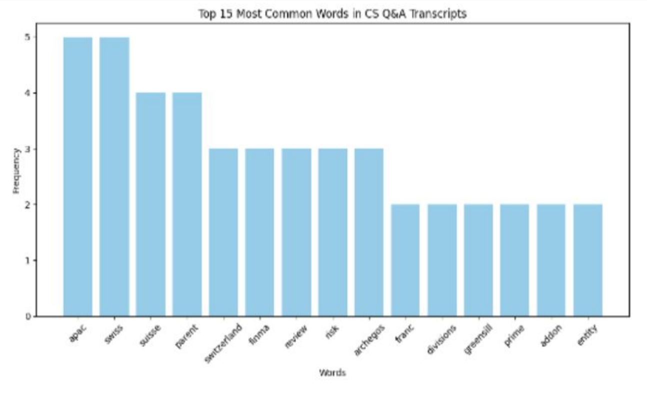

Word Cloud for BCS Q&A Transcripts
hedge
card
venkat
barclays
structural
income
mortgage
balance
bite
consumer
migration
tesco
nan
rote
impact
spend
churn
unsecured
holiday
precise
procyclicality
restructure
transitional
fullyear
pillar
sensitivity
february
government
lockdown
increase
variable
scale
amazon

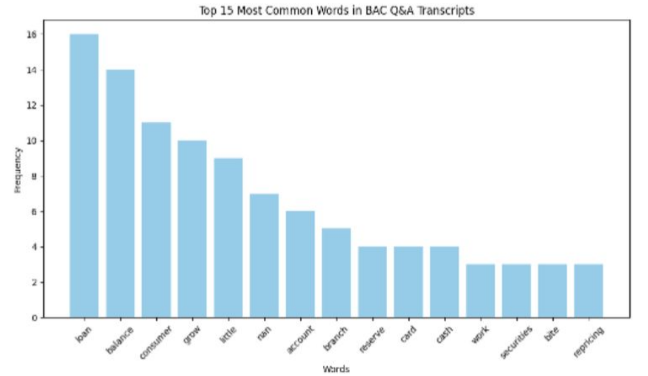

Word Cloud for CS Q&A Transcripts
swiss
apac
suisse
parent
switzerland
archegos
franc
greensill
divisions
finma
review
risk
october
addon
exceed
etfs
center
entity
exposure
november
wealth
cecl
prime
deduction
derisking
currency
corporate
asia
fund
share
lombard
march
ibcm
buyback
save
rmbs
measure
respect
seed
frtb
mbia
recur


Word Cloud for BAC Q&A Transcripts
loan
balance
consumer
grow
branch
account
little
reserve
nan
card
work
cash
bite
securities
repricing
commercial
stimulus
basis
merchant
pretty
chargeoffs
customer
deferrals
overdraft
rule
money
retirement
drive
forgive
operate
dollar
income
advantage
swap
invest
share
rotation
bucket
cloud
covid
ncos
merrill
press
sheer

Most Common Words







BERTopic

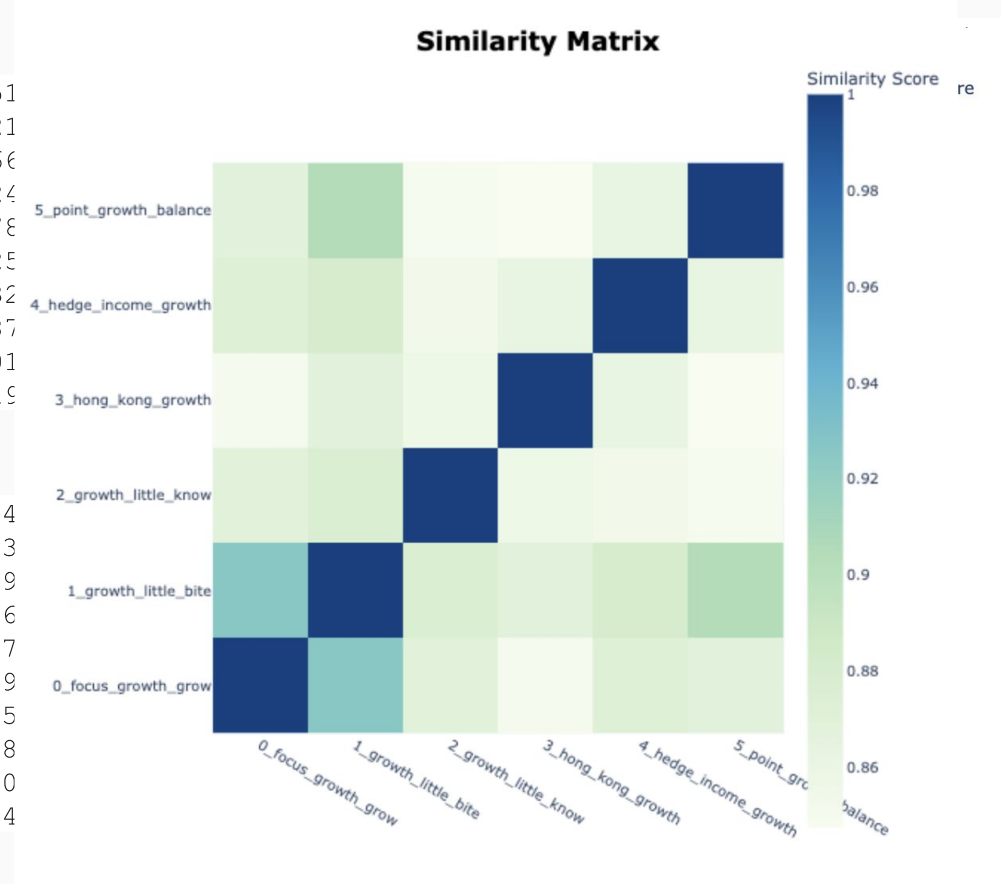
```python
topic_texts = {}

# Loop over each ticker
for ticker in ticker_mapping.keys():
    # Filter for the given ticker and sort by call_date (chronologically)
    df_ticker = df_processed[df_processed['ticker'] == ticker].sort_values('call_date')

    # Group by call_name; note: if multiple transcripts per call exist, they'll be aggregated
    groups = df_ticker.groupby('call_name', sort=False)

    for call, group in groups:
        # Extract quarter and year from call_name (assumes it starts like "Q1 2021")
        tokens = call.split()
        if len(tokens) >= 2:
            quarter = tokens[0].lower()  # e.g., 'q1'
            year = tokens[1]             # e.g., '2021'
        else:
            quarter = 'unknown'
            year = 'unknown'

        # Create a key in the format: ticker_text_quarter_year (e.g., c_text_q1_2021)
        key = f"{ticker.lower()}_text_{quarter}_{year}"

        # Join all text from the same call into a single string
        aggregated_text = " ".join(group['text'].tolist())

        # Store the aggregated text in the dictionary
        topic_texts[key] = aggregated_text

print(topic_texts.keys())
```

```
dict_keys(['c_text_q1_2021', 'c_text_q2_2021', 'c_text_q3_2021', 'c_text_q4_2021', 'c_text_q1_2022',
```

BERTopic

```
Topic 0:
  focus: 0.0258938851
  growth: 0.025850321
  grow: 0.0233939246(
  point: 0.0226260224
  things: 0.021120678
  businesses: 0.02025
  activity: 0.0198032
  target: 0.019408187
  want: 0.01914254701
  firm: 0.0190807241(
```

```
Topic 3:
  hong: 0.0386117594
  kong: 0.0384126723
  growth: 0.03505249
  point: 0.030042656
  basis: 0.028036917
  interest: 0.027859
  book: 0.0250392445
  income: 0.02498818
  want: 0.0248333540
  higher: 0.02454444
```




```
 2:
vth: 0.041743507898423474
:le: 0.039828533786521O8
v: 0.03857982811478l216
e: 0.0338026418O667305
nt: 0.028716464523471017
ngs: 0.024614302599257242
n: 0.024368629383905084
:: 0.022618249597546877
v: 0.021898512048l504
:: 0.021693591556769927
```

```
: 5:
int: 0.030051058446434045
owth: 0.029377001616071353
lance: 0.024870973393984308
an: 0.022314144800856854
nt: 0.021813003916306185
nt: 0.021136948319434582
rd: 0.020987591578943744
te: 0.020632413900954293
ings: 0.019637795246953155
el: 0.018948619364869926
```

FinBERT for Sentiment Analysis

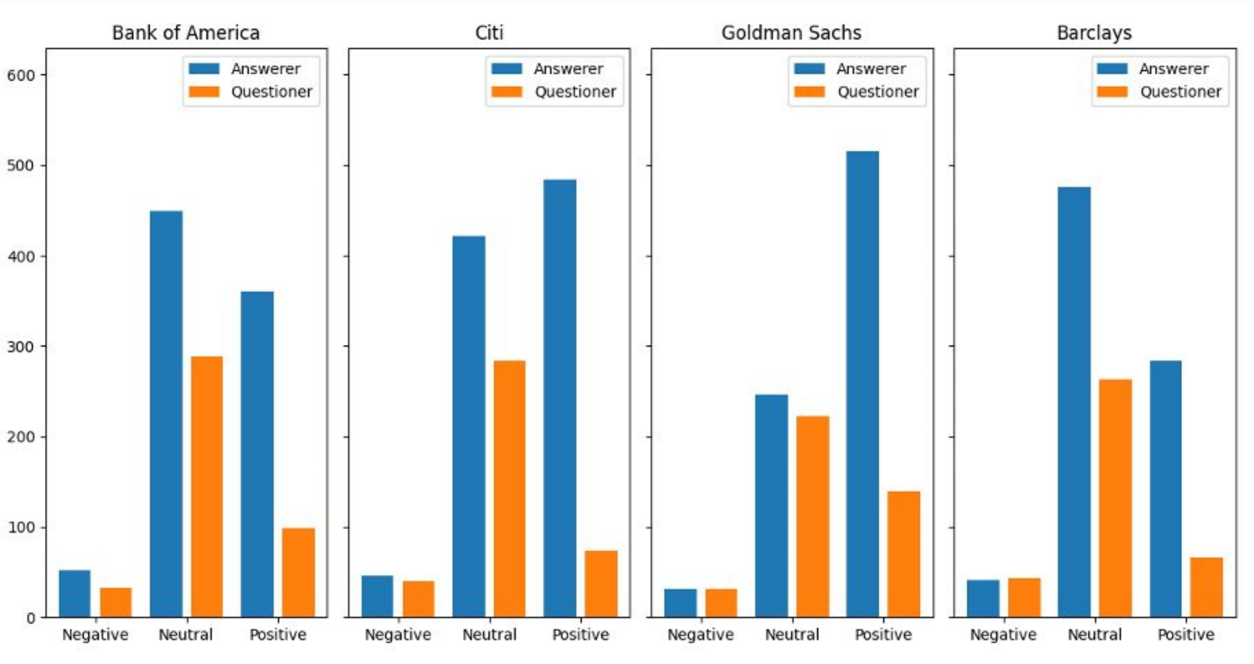
Untuned FinBERT

```
from transformers import BertTokenizer, BertForSequenceClassification, pipeline

finbert = BertForSequenceClassification.from_pretrained('yiyanghkust/finbert-tone', num_labels=3)
tokenizer = BertTokenizer.from_pretrained('yiyanghkust/finbert-tone')
nlp = pipeline("text-classification", model=finbert, tokenizer=tokenizer)
```

preprocess_text	finbert_call_sentiment	finbert_call_sentiment_score
come back thoughts youre talking consumer spending holding right obviously strong levels two years ago look mentioned guys thinking economy troughs middle next could hold consumer spend consumers hold spending number could actually deteriorate	Negative	0.999771
couple things one theres obviously external events could change situation globe dramatically given pathway doesnt kind event doesnt take place rate theyre spending consistent lower inflation embedded teams indiscernible teams economic projections return inflation target end rate structure comes beginning middle next still stays around end given economy inflation coming economies would still growing getting back towards trend growth would hold steady	Neutral	0.928081
pretty steady month august september october level thats kind people get paid spend little bit pricing goes ebbs flows within spend right youve kind seen adjustments came pandemic last couple years sort adjust system mean lot goods purchase lot travel lot returntooffice spending track people buying stuff thats kind leveled system including drop fuel prices increase basically relatively bounce around theyre spending amount money gas spent last	Neutral	0.999715
big aggregate numbers keep bumping along level consistent low inflation low growth economy effectively shows consumer brought line scenario fed reaching target thats see well take time work system retail sales number seems stronger today shake moment	Positive	1.000000

Untuned FinBERT

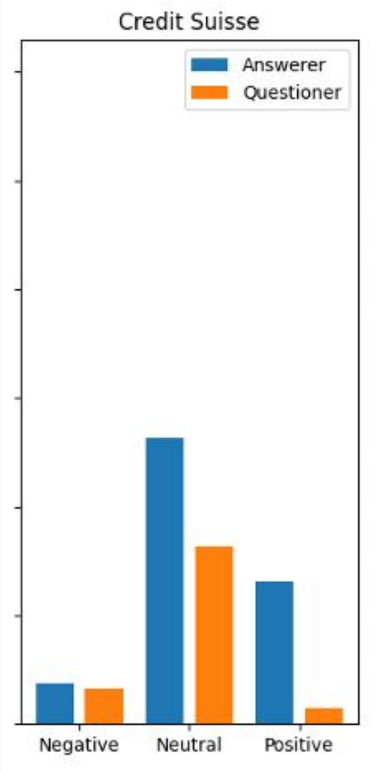
- Very neutral
- Few negative comments, no negative calls




```
Mostly Positive Calls:
    c_text_q2_2021
    c_text_q3_2021
    c_text_q2_2022
    c_text_q2_2023
    c_text_q1_2024
    c_text_q3_2024
    c_text_q4_2024
    bac_text_q1_2024
    bac_text_q3_2024
    hsbc_text_q1_2021
    hsbc_text_q2_2024
    gs_text_q1_2021
    gs_text_q2_2021
    gs_text_q3_2021
    gs_text_q4_2021
    gs_text_q1_2022
    gs_text_q2_2022
    gs_text_q3_2022
    gs_text_q4_2022
    gs_text_q1_2023
    gs_text_q2_2023
    gs_text_q3_2023
    gs_text_q4_2023
    gs_text_q1_2024
    gs_text_q2_2024
    gs_text_q3_2024
    gs_text_q4_2024
    bcs_text_q2_2022

Mostly Negative Calls:

Mostly Neutral Calls:
    c_text_q1_2021
    c_text_q4_2021
    c_text_q1_2022
    c_text_q3_2022
    c_text_q4_2022
    c_text_q1_2023
    c_text_q3_2023
    c_text_q4_2023
    c_text_q2_2024
    bac_text_q1_2021
    bac_text_q2_2021
    bac_text_q3_2021
    bac_text_q4_2021
    bac_text_q1_2022
    bac_text_q2_2022
    bac_text_q3_2022
    bac_text_q4_2022
    bac_text_q1_2023
    bac_text_q2_2023
```

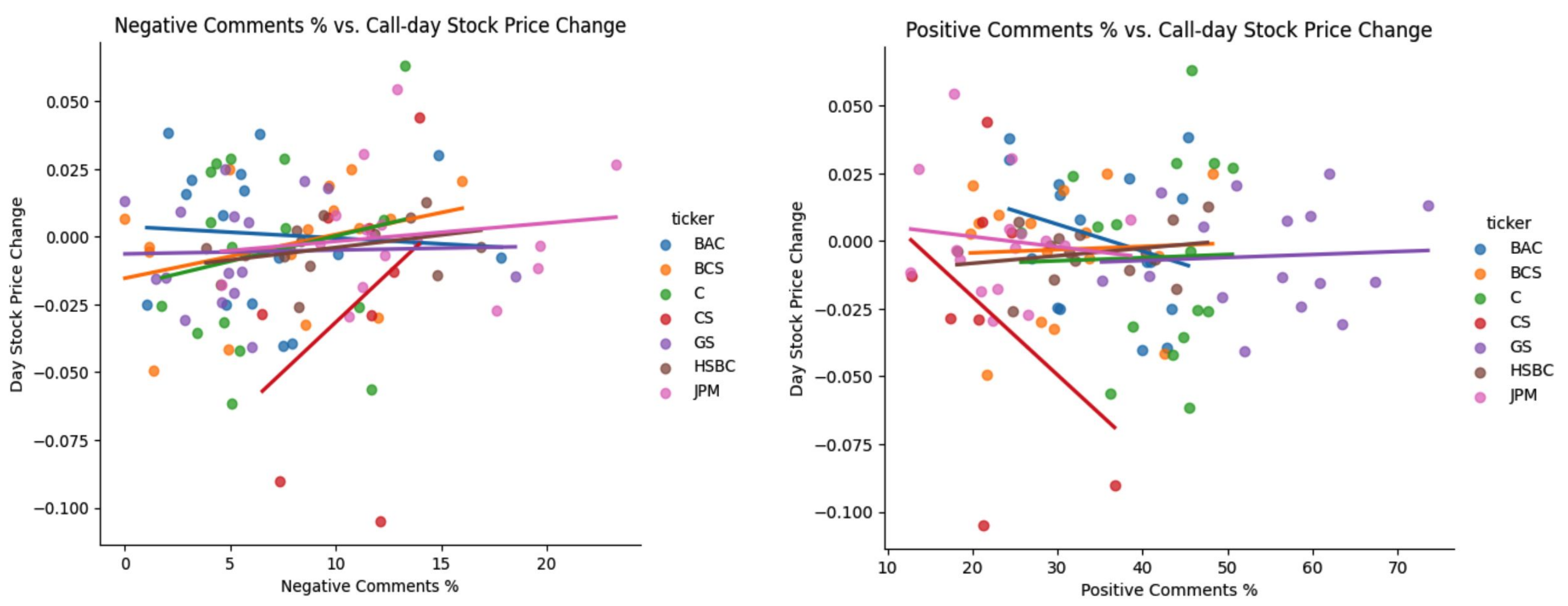
Benchmarking financial stress & sentiment

- Credit Suisse: not an outlier




```python
import json
import pandas as pd
import requests


# Dictionary to store DataFrames
stock_histories = {}

for ticker in unique_ tickers:
    url = f"https://www.alphavantage.co/query?function=TIME_SERIES_DAILY&symbol={ticker}&outputsize=full&apikey=

    response = requests.get(url).json()
    history = response.get("Time Series (Daily)", {})

    if not history:
        print(f"Warning: No data found for {ticker}")
        continue

    formatted_data = {
        date: {
            "open": float(values["1. open"]),
            "high": float(values["2. high"]),
            "low": float(values["3. low"]),
            "close": float(values["4. close"]),
            "volume": int(values["5. volume"])
        }
        for date, values in history.items()
    }


    filename = f"{ticker}_stock_history.json"
    with open(filename, "w") as f:
        json.dump(formatted_data, f, indent=4)  # Pretty-print for readability

    stock_histories[ticker] = pd.DataFrame.from_dict(formatted_data, orient="index")

    print(f"Saved {filename}")
```

FinBERT (pre-trained) Findings vs Stock Price moves




```
Ticker
BAC      0.194915
BCS     -0.004598
C        0.180323
CS      -0.662922
GS       0.033987
HSBC     0.432808
JPM     -0.161052
Name: POSITIVE COMMENTS VS EARNINGS PERIOD PRICE,
```

Fine-tuning BERT

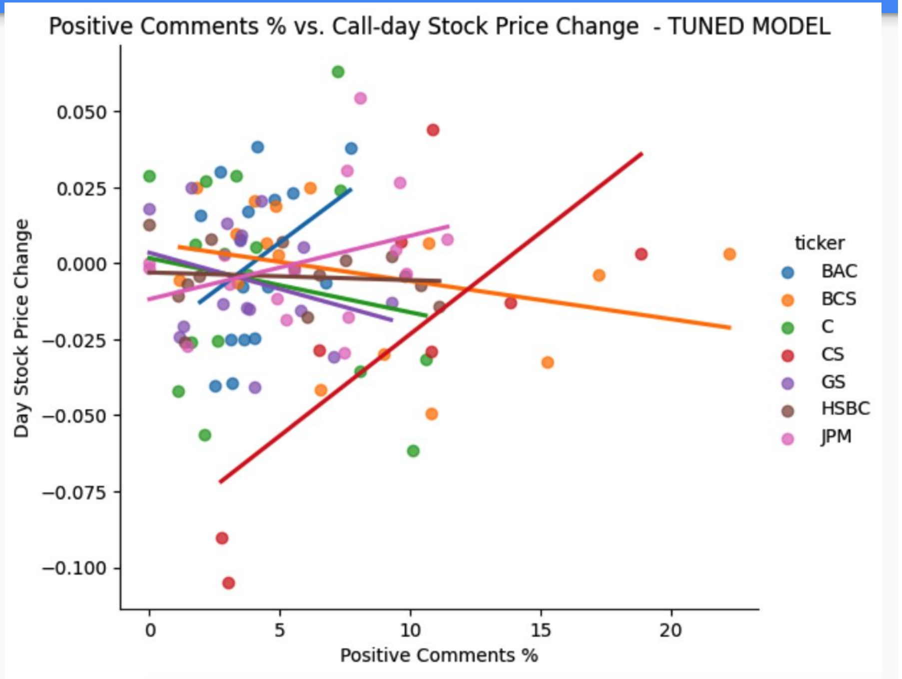
```python
# Create a DatasetDict to organize them
dataset = DatasetDict({
    "train": train_dataset,
    "validation": val_dataset,
    "test": test_dataset
})

# Verify structure
print(dataset)
```

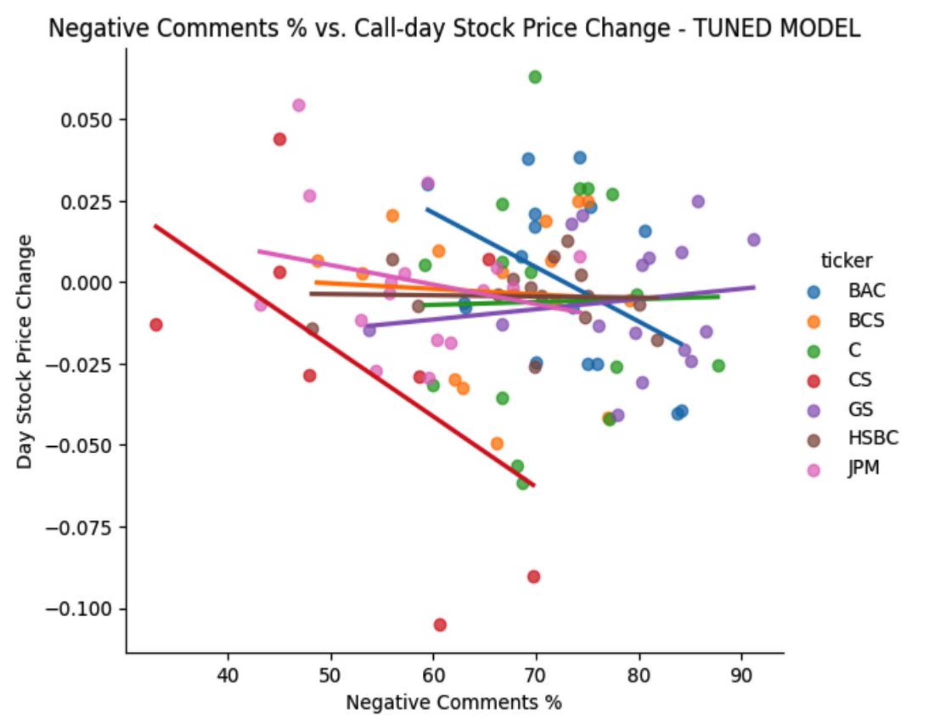
```
DatasetDict({
    train: Dataset({
        features: ['text', 'label', '__index_level_0__'],
        num_rows: 4673
    })
    validation: Dataset({
        features: ['text', 'label', '__index_level_0__'],
        num_rows: 585
    })
    test: Dataset({
        features: ['text', 'label', '__index_level_0__'],
        num_rows: 584
    })
})
```

```python
# Load pre-trained FinBERT model
model = BertForSequenceClassification.from_pretrained("yiyanghkust/finbert-tone", num_labels=3)

# Define accuracy metric
accuracy = evaluate.load("accuracy")
def compute_metrics(eval_pred):
    logits, labels = eval_pred
    predictions = np.argmax(logits, axis=-1)
    return accuracy.compute(predictions=predictions, references=labels)

# Set training arguments
training_args = TrainingArguments(
    output_dir="./finbert_finetuned",
    evaluation_strategy="epoch",
    save_strategy="epoch",
    per_device_train_batch_size=8,
    per_device_eval_batch_size=8,
    num_train_epochs=3,
    weight_decay=0.01,
    logging_dir="./logs",
    logging_steps=100,
    load_best_model_at_end=True,
    metric_for_best_model="accuracy"
)

# Initialize Trainer
trainer = Trainer(
    model=model,
    args=training_args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["validation"],
    tokenizer=tokenizer,
    compute_metrics=compute_metrics
)

# Train the model
trainer.train()
```

Epoch	Training Loss	Validation Loss	Accuracy
1	0.614600	0.578361	0.794872
2	0.330000	0.552333	0.803419
3	0.195900	0.699055	0.798291

FinBERT TUNED Findings (1)





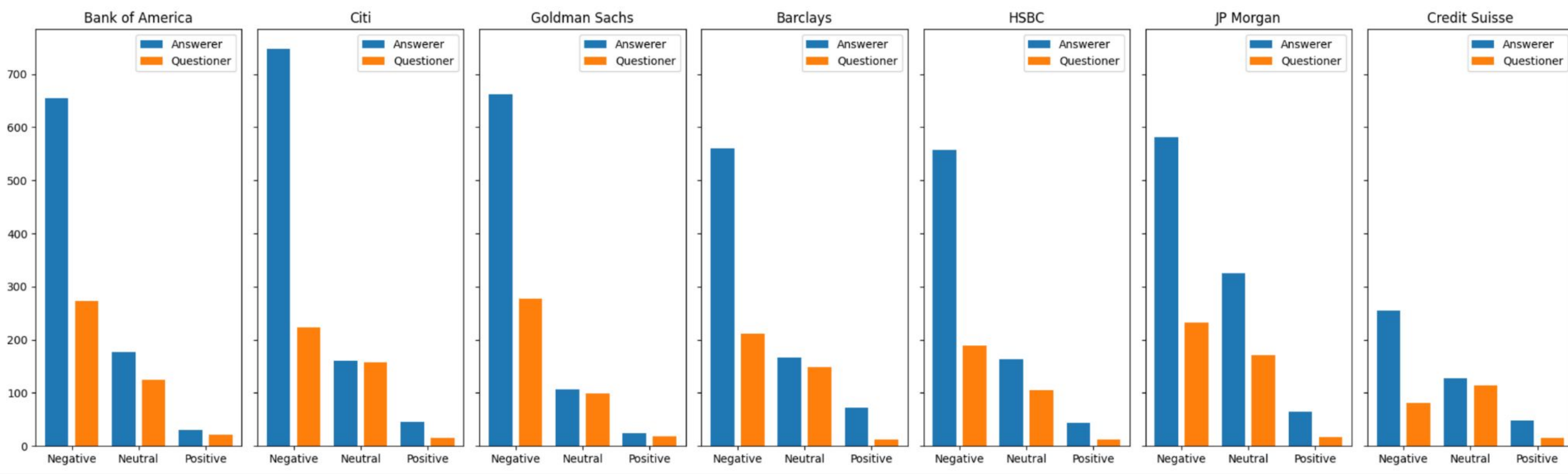
finbert_finetuned_fsn_sentiment

Negative	5509
Neutral	2148
Positive	443

Ticker	Positive	Positive TUNED	Negative	Negative TUNED
CS	-0.399754	0.7265	0.385926	-0.536967

FinBERT TUNED Findings (2)

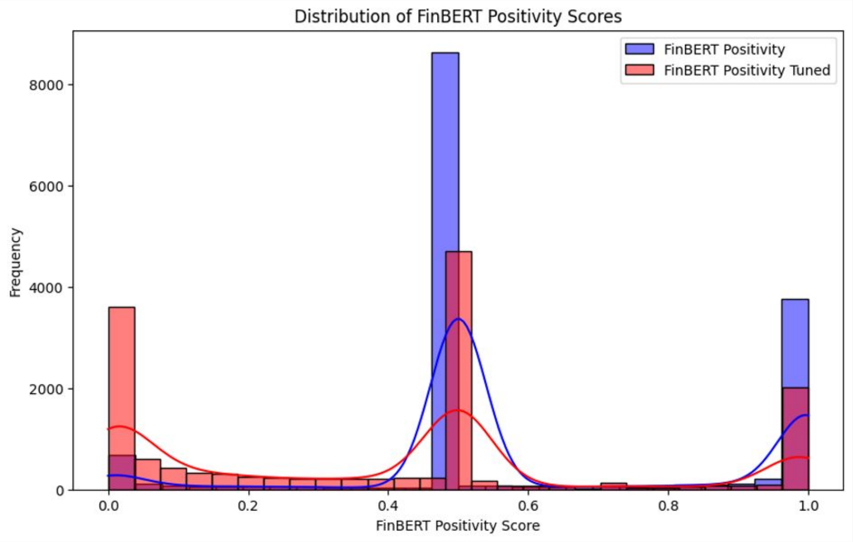
- CS now an outlier - FEWEST NEGATIVE COMMENTS!!
- Most markedly among bank's own management
- Reflective of management (and some analysts) trying to allay market fears?



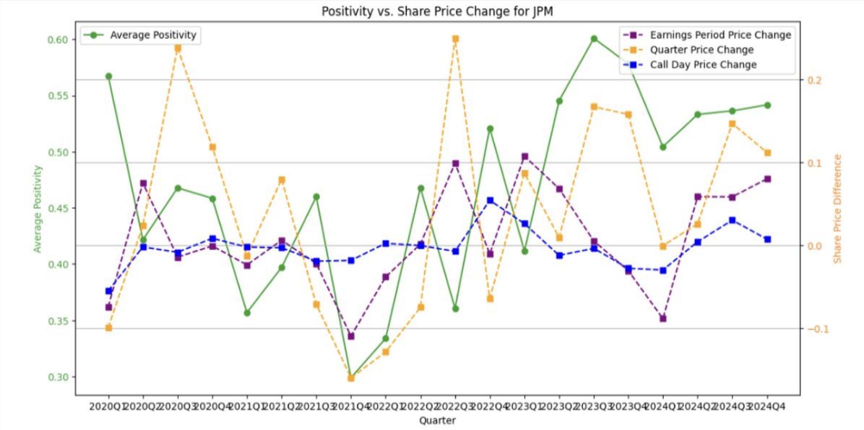
Positivity

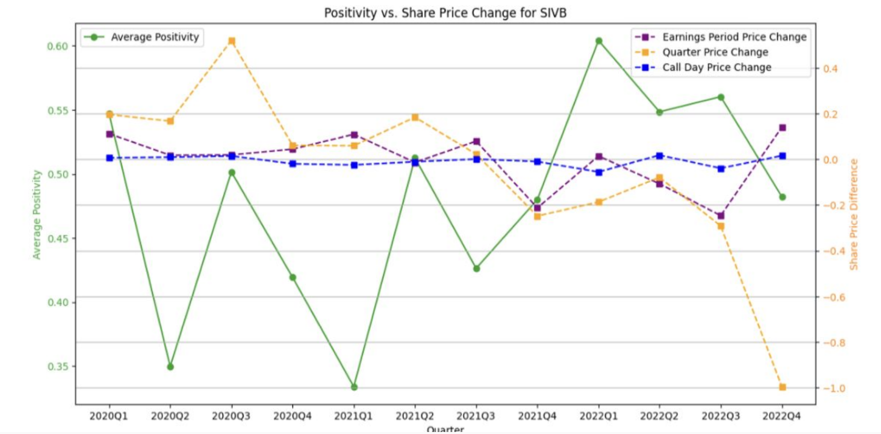
- New metric introduced
- Allowed us to measure how positivity changed over time
- Proposed client-facing output: comparing selected call's positivity compared to bank average, and also the average for that quarter

$$P = \begin{cases} S_p - S_n + 1, \text{if the sentiment is "positive"} \\ S_n - S_p + 1, \text{if the sentiment is "negative"} \\ 0.5, \text{if the sentiment is "neutral"} \end{cases}$$

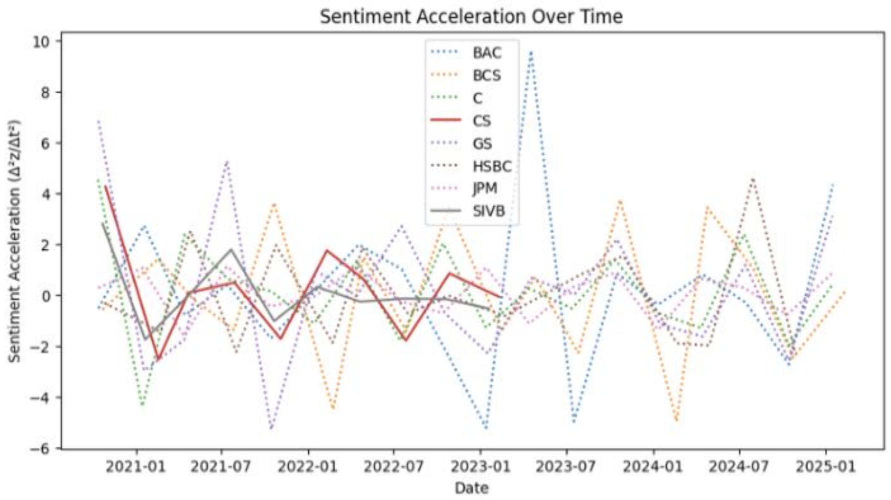


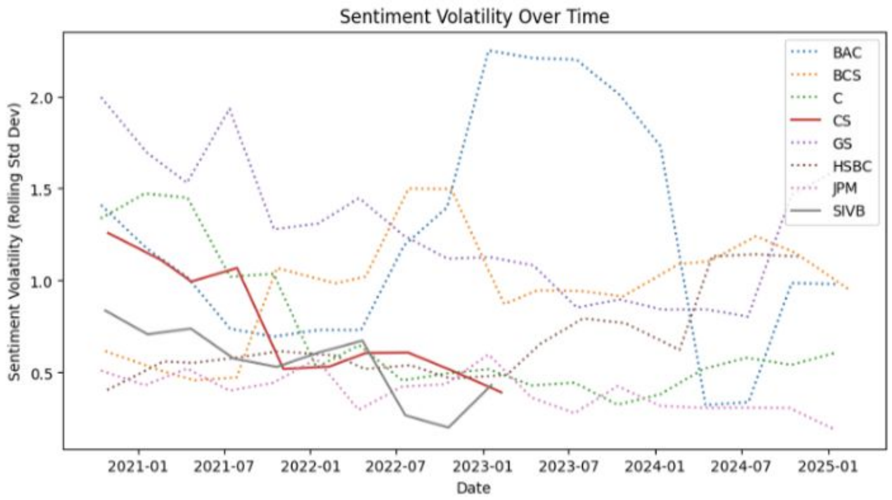
FinBERT Positivity vs Share Price





Sentiment Acceleration and Volatility over Time





LLMs

- Methodology
- Summaries
- Topics
- Takeaways
- Scores

LLM Methodology

- Comparison of quarterly earnings reports for 2 companies – JPMorgan and Citibank
- Comparison of outputs from multiple prompts
- Comparison of outputs with and without stopwords
- Summaries, topics, takeaways and scores
- Further analysis - multiple LLM’s, more varied prompts, more banks

First Attempt at LLMs

```python
def chunk_text(text: str, chunk_size=1000) -> list:
    words = text.split()
    chunks = []
    start = 0
    while start < len(words):
        end = min(start + chunk_size, len(words))
        chunk = " ".join(words[start:end])
        chunks.append(chunk)
        start = end
    return chunks


def phi_chat(system_msg: str, user_msg: str, **gen_kwargs) -> str:
    """
    Format 'messages' for the Phi-3-mini-128k-instruct pipeline,
    then call pipe(...) with generation_args. Return the model's text output.
    """
    messages = [
        {"role": "system", "content": system_msg},
        {"role": "user", "content": user_msg},
    ]
    output_list = pipe(messages, **gen_kwargs)
    return output_list[0]['generated_text'].strip()
```

```python
def ask_llm_for_summary(text: str) -> str:
    """
    1) Chunk the text,
    2) Summarize each chunk separately,
    3) Combine partial summaries into one final summary via a new prompt.
    """
    system_message = "You are a helpful AI assistant."

    # Split the transcript into chunks
    chunks = chunk_text(text, chunk_size=1000)
    partial_summaries = []

    # Summarize each chunk separately
    for i, chunk in enumerate(chunks, start=1):
        user_message = (
            f"Please summarize the following portion of a bank earnings call transcript:\n\n{chunk}\n\n"
            "Provide a concise summary of the key points."
        )
        chunk_summary = phi_chat(system_message, user_message, **generation_args)
        partial_summaries.append(chunk_summary)

    # Combine all partial summaries into a single summary
    combined_text = "\n\n".join(partial_summaries)
    user_message_final = (
        "Below are partial summaries of each chunk of the transcript:\n\n"
        f"{combined_text}\n\n"
        "Please combine them into a single concise overall summary."
    )
    final_summary = phi_chat(system_message, user_message_final, **generation_args)
    return final_summary.strip()
```

First Attempt at LLMs

```python
def ask_llm_for_topics(text: str, num_topics=3, summary=summary) -> str:
    """
    1) Get a hierarchical summary of the entire text,
    2) Ask for the top N topics in bullet points.
    """
    system_message = "You are a helpful AI assistant."
    user_message = (
        f"Here is a summary of a bank earnings call transcript:\n\n{summary}\n\n"
        f"List the top {num_topics} topics discussed, in bullet points."
    )
    return phi_chat(system_message, user_message, **generation_args)


def ask_llm_for_takeaways(text: str, num_takeaways=3, summary=summary) -> str:
    """
    Similar approach: first get a hierarchical summary,
    then ask for the key takeaways.
    """
    system_message = "You are a helpful AI assistant."
    user_message = (
        f"Here is a summary of a bank earnings call transcript:\n\n{summary}\n\n"
        f"List the {num_takeaways} most important takeaways, in bullet points."
    )
    return phi_chat(system_message, user_message, **generation_args)


def ask_llm_for_score(text: str, summary=summary) -> str:
    """
    1) Summarize the entire text (hierarchical approach),
    2) Ask for an overall performance rating out of 100 and a brief reason.
    """
    system_message = "You are a helpful AI assistant."
    user_message = (
        f"Here is a summary of a bank earnings call transcript:\n\n{summary}\n\n"
        "Please provide an overall performance score for the quarter out of 100, "
        "and give a short reason explaining why you assigned that score."
    )
    return phi_chat(system_message, user_message, **generation_args)
```

```python
results_all_models = {}
model_label = "microsoft/Phi-3-mini-128k-instruct"

all_results = []

for i, text in enumerate(transcript_texts, start=1):
    print(f"\n=== Transcript #{i} ===")

    # 1) Summarize
    summary = ask_llm_for_summary(text)
    print("Summary:\n", summary)

    # 2) Top 3 Topics
    topics = ask_llm_for_topics(text, num_topics=3, summary=summary)
    print("Topics:\n", topics)

    # 3) Top 3 Takeaways
    takeaways = ask_llm_for_takeaways(text, num_takeaways=3, summary=summary)
    print("Takeaways:\n", takeaways)

    # 4) Overall Performance Score
    score = ask_llm_for_score(text, summary=summary)
    print("Performance Score:\n", score)

    # Store results
    result = {
        "transcript_index": i,
        "summary": summary,
        "topics": topics,
        "takeaways": takeaways,
        "score": score
    }
    all_results.append(result)

results_all_models[model_label] = all_results
```

Summaries

Example 1: During Citigroup's Q2 2024 earnings call, CEO Jane Fraser discussed the bank's regulatory actions, including simplification and risk management improvements, as part of its transformation efforts. The bank reported a net income of $3.2 billion, an ROTCE of 7.2%, and revenues up 4% year-over-year. Expenses were down 2%, with savings from organizational simplification expected to save $2 to $2.5 billion annually. Citigroup plans to increase its dividend and resume buybacks.

Example 2: In the Q4 2024 earnings call, JPMorganChase reported a net income of $14 billion, EPS of $4.81, and revenue of $43.7 billion, with a 10% year-on-year revenue increase. Net Interest Income (NII) excluding Markets decreased by $548 million, while Net Interest Revenue (NIR) excluding Markets increased by $3.1 billion, or 30%. Expenses were down by $1.7 billion, excluding the FDIC special assessment. The company is focused on efficiency, modernization, and internal efficiencies, with peak modernization spend reached.

- Detailed summaries that cover the main points, and are not too long.
- Covers some slightly different points making comparison more difficult
- Further analysis would require more detailed prompts to standardise output

Topics and Takeaways

Example **:** Topics

- JPMorgan Chase & Co.'s strong Q2 2024 earnings with a 20% YoY increase in net income and revenue

- The bank's focus on organic and inorganic growth, sustainable dividends, and capital management

- The record number of first-time investors in CCB and the 2% YoY increase in average loans

- Topics are short and to the point
- Takeaways are longwinded
- Further analysis would require more detailed prompts to standardise output

Scores

Call	JPMorgan	Citigroup
Q1 2024	82	75
Q2 2024	92	85
Q3 2024	85	85
Q4 2024	85	92

- Scores vary by quarter for JPMorgan
- Scores increase over time for Citigroup
- Sufficient reasoning for scores

Example -

Reason: JPMorgan Chase & Co. demonstrated a strong financial performance in Q1 2024, with a significant net income of $13.4 billion and a robust ROTCE of 21%. The bank's strategic investments and resilience in the face of economic, geopolitical, and regulatory uncertainties are commendable. However, the revenue decreases in Commercial Banking and Asset & Wealth Management, along with the discontinuation of standalone CIB earnings, indicate areas for improvement. The bank's confidence and commitment to investing through economic cycles, along with a strong capital position, contribute to a high performance score.

LLM Review

- Models to large to run locally

  - Models too large to run locally
  - Poor results
  - Inconsistent format

  - Decided to look for pretrained solutions: Sonar LLM via the Perplexity API
  - Adjusted prompt engineering

Sonar LLM

1. Summarise the full Q&A transcript
2. Outline three heavily discussed topics

```python
def ask_llm_for_summary(text: str, api_key: str) -> str:
    """
    SUMMARY
    """
    user_message = (
        "Summarize the following bank earnings call transcript:\n\n"
        f"{text}\n\n"
        "**In your summary, be sure to cover:**\n"
        "- Overall financial performance (revenues, expenses, profit/loss, margins)\n"
        "- Key operational highlights (business segments, new initiatives, major deals)\n"
        "- Management's outlook or guidance for future quarters\n"
        "- Market conditions or external factors that impacted performance\n"
        "- Any risks, challenges, or concerns mentioned\n\n"
        "Keep your response concise, well-structured, and focused on the main points. "
        "Whenever possible, provide specific figures or examples mentioned in the transcript.\n"
    )
    return sonar_chat(SYSTEM_MESSAGE, user_message, api_key)


def ask_llm_for_topics(text: str, api_key: str) -> str:
    """
    TOP 3 TOPICS
    """
    user_message = (
        "Please read the following bank earnings call transcript:\n\n"
        f"{text}\n\n"
        "Identify the **top 3 topics** discussed and provide them in a consistent format.\n"
        "Use bullet points, and label them clearly as follows:\n\n"
        "TOPICS:\n"
        "1) [Topic One]\n"
        "2) [Topic Two]\n"
        "3) [Topic Three]\n\n"
        "For each topic:\n"
        "- Provide a short title.\n"
        "- Include a concise 1-2 sentence description.\n\n"
        "Do not include any additional commentary outside of these bullet points.\n"
        "Ensure the final output appears exactly in the 'TOPICS:' format described."
    )
    return sonar_chat(SYSTEM_MESSAGE, user_message, api_key)
```

Sonar LLM

3. Provide the BoE analyst with three takeaways for actioning or further analysis
4. Notify the BoE analyst of three potential causes of concern

```python
def ask_llm_for_takeaways(text: str, api_key: str) -> str:
    """
    TOP 3 TAKEAWAYS
    """
    user_message = (
        "Please read the following bank earnings call transcript:\n\n"
        f"{text}\n\n"
        "Identify the **top 3 takeaways** and present them in bullet-point format.\n"
        "Use the exact format:\n\n"
        "TAKEAWAYS:\n"
        "1) [Takeaway One]\n"
        "2) [Takeaway Two]\n"
        "3) [Takeaway Three]\n\n"
        "For each takeaway:\n"
        "- Provide a concise 1-2 sentence explanation.\n\n"
        "Do not include any additional commentary outside of these bullet points.\n"
        "Ensure the final output appears exactly in the 'TAKEAWAYS:' format described."
    )
    return sonar_chat(SYSTEM_MESSAGE, user_message, api_key)

def ask_llm_for_concerns(text: str, api_key: str) -> str:
    """
    TOP 3 CONCERNS
    """
    user_message = (
        "Please read the following bank earnings call transcript:\n\n"
        f"{text}\n\n"
        "Identify the **top 3 concerns or risks** discussed and present them in bullet-point format.\n"
        "Use the exact format:\n\n"
        "CONCERNS:\n"
        "1) [Concern One]\n"
        "2) [Concern Two]\n"
        "3) [Concern Three]\n\n"
        "For each concern:\n"
        "- Provide a concise 1-2 sentence explanation.\n\n"
        "Do not include any additional commentary outside of these bullet points.\n"
        "Ensure the final output appears exactly in the 'CONCERNS:' format described."
    )
    return sonar_chat(SYSTEM_MESSAGE, user_message, api_key)
```

Sonar LLM

5. Score the bank's performance in that financial quarter

```python
def ask_llm_for_score(text: str, api_key: str) -> str:
    """
    SCORE OUT OF 100
    """
    user_message = (
        "Please read the following bank earnings call transcript:\n\n"
        f"{text}\n\n"
        "Give an **overall performance score** for the quarter, on a scale of 0 to 100.\n"
        "Use the exact format:\n\n"
        "SCORE:\n"
        "[Score Value]\n\n"
        "REASON:\n"
        "[1-2 sentence reason explaining the assigned score]\n\n"
        "Do not include any additional commentary or text outside of the above two fields.\n"
        "Ensure the final output appears exactly as 'SCORE:' then 'REASON:'."
    )
    return sonar_chat(SYSTEM_MESSAGE, user_message, api_key)
```

XGBoost to Predict Price Change

- What output we're looking for
- Methodology
- Results

Formulating XGBOost

- Using TF-IDF on text data to predict changes on either
  - Earnings period price change
  - Call day price change
  - Quarter day price change

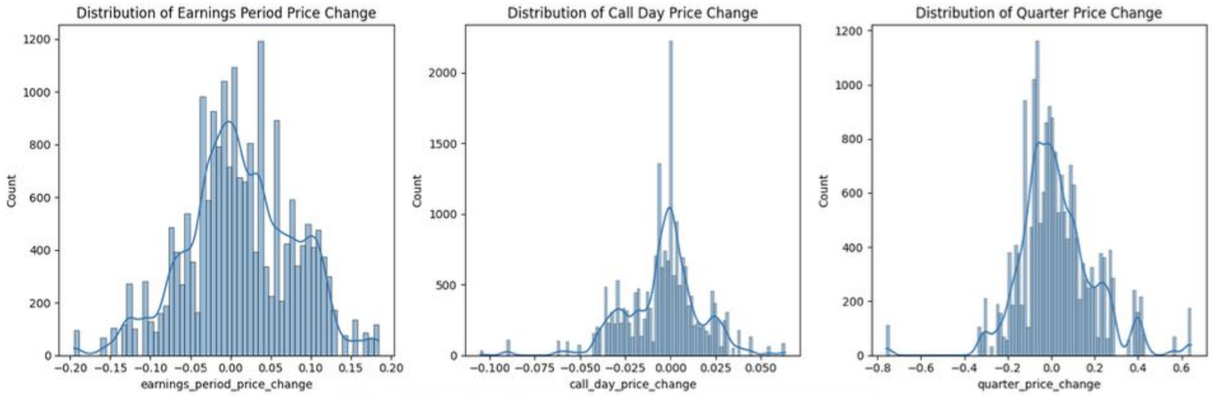
- Regression models ineffective so moved to a categorisation model

3 categories defined

Price Up

Price Stationary

Price Down




```python
# Define function to classify price changes into 3 categories
def categorize_price_change(series):
    """
    Categorizes price change values into 3 classes:
    - 0: Price Drop (bottom 33% of middle 90%)
    - 1: Stationary Price (middle 33% of middle 90%)
    - 2: Price Increase (top 33% of middle 90%)
    """
    lower_bound = series.quantile(0.05)  # Exclude bottom 5%
    upper_bound = series.quantile(0.95)  # Exclude top 5%

    filtered = series[(series >= lower_bound) & (series <= upper_bound)]  # Middle 90%
    q1, q2 = filtered.quantile([0.33, 0.66])  # Split into thirds

    # Assign categories
    def classify(val):
        if val < q1:
            return 0  # Price Drop
        elif val < q2:
            return 1  # Roughly Stationary
        else:
            return 2  # Price Increase

    return series.apply(classify)
```

XGBoost Results

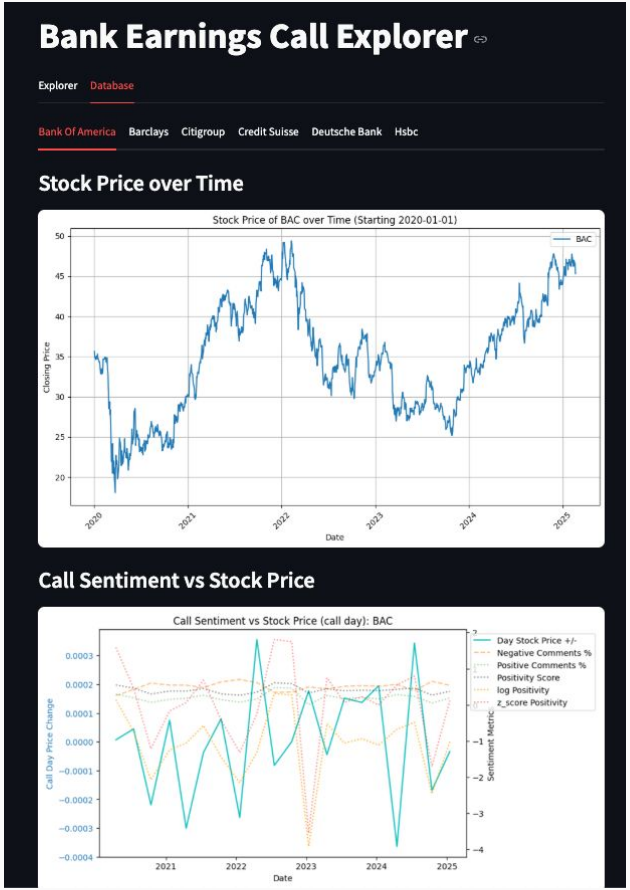
- For added accuracy added information from 7 more banks (~140 calls)
- Test/train split, testing on Q3,Q4 2024 data

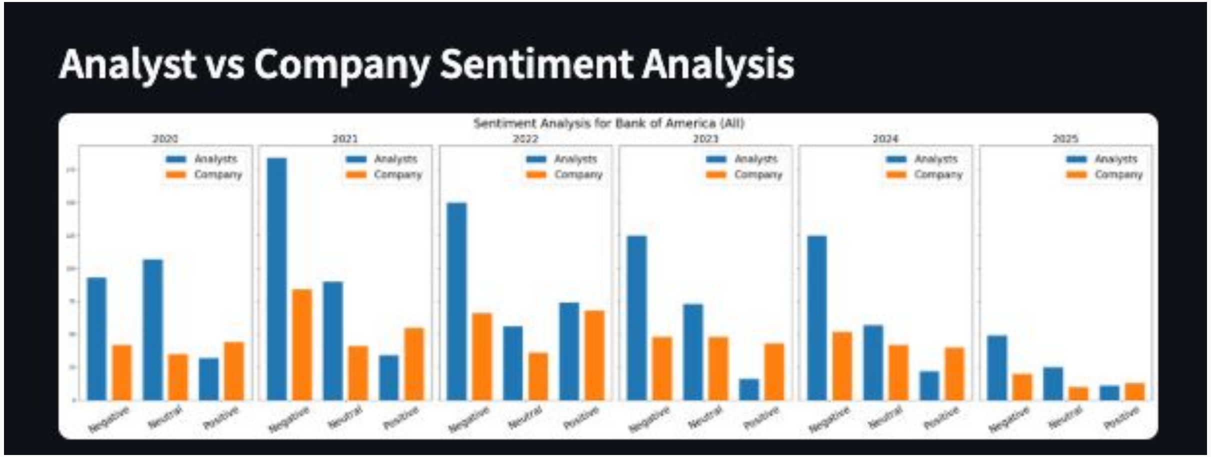
- 58.3% accuracy across the three possible predictions
  - **75.1% increase** from a 33% random selection -> acceptable for financial modelling

Results and Interface

- Client-facing product

Solution: Part 1





Section 1:

Bank information

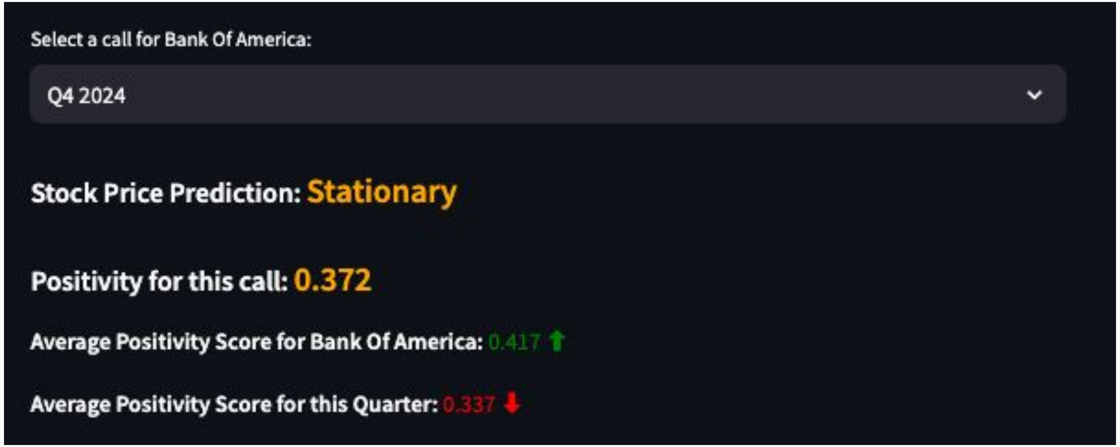
- Stock price
- Sentiment over time
- Sentiment broken down by analysts vs internal employees

Solution: Part 2

Section 2:

Call-specific information

- Stock price predictor (XGBoost model)
- Call positivity
- Average bank positivity, average quarter positivity for context



Solution: Part 3

Section 3:

LLM Output

- Summary
- Topics
- Takeaways
- Concerns
- Performance score

Slide Show ary

Here is a summary of the key points from the bank earnings call transcript:

Financial Performance:

- Net income increased 5.6% year-over-year to $268.2 billion in 2024
- Return on assets improved to 1.12%, up 3 basis points from 2023
- Net interest income (NII) growth is expected to accelerate to 6-7% for the full year 2025
- Efficiency ratio is projected to be around 60% in 2025

Operational Highlights:

- Deposit growth has stabilized and improved, with consumer deposits bottoming out in August and wealth deposits in July
- Loan growth has picked up, with $20 billion added in Q4 2024
- The wealth management business has seen 11 straight quarters of year-over-year growth
- The bank added 1 billion net new checking accounts, with 90-92% being primary accounts

Outlook:

- Management expects GDP growth in the low 2% range for 2025
- Unemployment is projected to stay between 4.1-4.3%
- The bank forecasts 2-3 fewer rate cuts than previously expected
- Deposit costs are estimated to be elevated at 2.03% in 2025

Market Conditions:

- Interest rates are expected to remain higher than initially anticipated
- The S&P 500 is projected to grow 5-6% in 2025
- Management noted increased optimism among small and medium-sized businesses

Risks and Challenges:

- Potential changes to capital requirements and stress testing methodologies
- Ongoing regulatory scrutiny, particularly in capital markets businesses
- Concerns about leverage building up outside the banking system
- Geopolitical risks, including wars and trade tensions

The bank appears to be performing well overall, with improving financial metrics and positive momentum across its business segments. Management is cautiously optimistic about the economic outlook while remaining vigilant about potential risks and regulatory changes.

Score (out of 100)

SCORE: 85

REASON: The bank reported strong deposit and loan growth, stable credit quality, and improving net interest income trends. Management expressed confidence in their strategic positioning and ability to capitalize on economic tailwinds, while acknowledging some ongoing challenges.

Topics

TOPICS:

1. [Deposit and Loan Growth]

- The bank is seeing positive momentum in deposit and loan growth. Deposits have grown for 6 straight quarters, while loans have increased for several quarters in a row, with expectations for continued growth in 2025.

2. [Net Interest Income (NII) Outlook]

- NII growth is expected to accelerate to 6% to 7% for the full year. This is driven by deposit growth, loan growth, and the repricing of fixed-rate assets as interest rates potentially decline.

3. [Regulatory Capital and Business Outlook]

- The bank is advocating for changes in regulatory capital requirements, which they believe are unnecessarily high. They expect potential relief could free up capital for growth, but are not constraining any business lines due to capital limitations currently.

Takeaways

TAKEAWAYS:

1. Deposit growth is returning to normal levels Deposit balances have stabilized and are now growing, with consumer deposits finding a floor in August and wealth deposits in July. This is supporting net interest income growth.
2. Loan growth is accelerating Loan growth has picked up in recent quarters, with $9billionaddedinQ2,$19 billion in Q3, and $20 billion in Q4. This indicates increased client optimism and demand.
3. Net interest income growth expected to accelerate to 6-7% for the full year The bank expects NII growth to accelerate due to deposit and loan growth, as well as fixed asset repricing from maturing loans and cash flow swaps.

Concerns

CONCERNS:

1. Economic uncertainty and potential slowdown
2. Regulatory changes and capital requirements
3. Deposit cost pressures

For each concern:

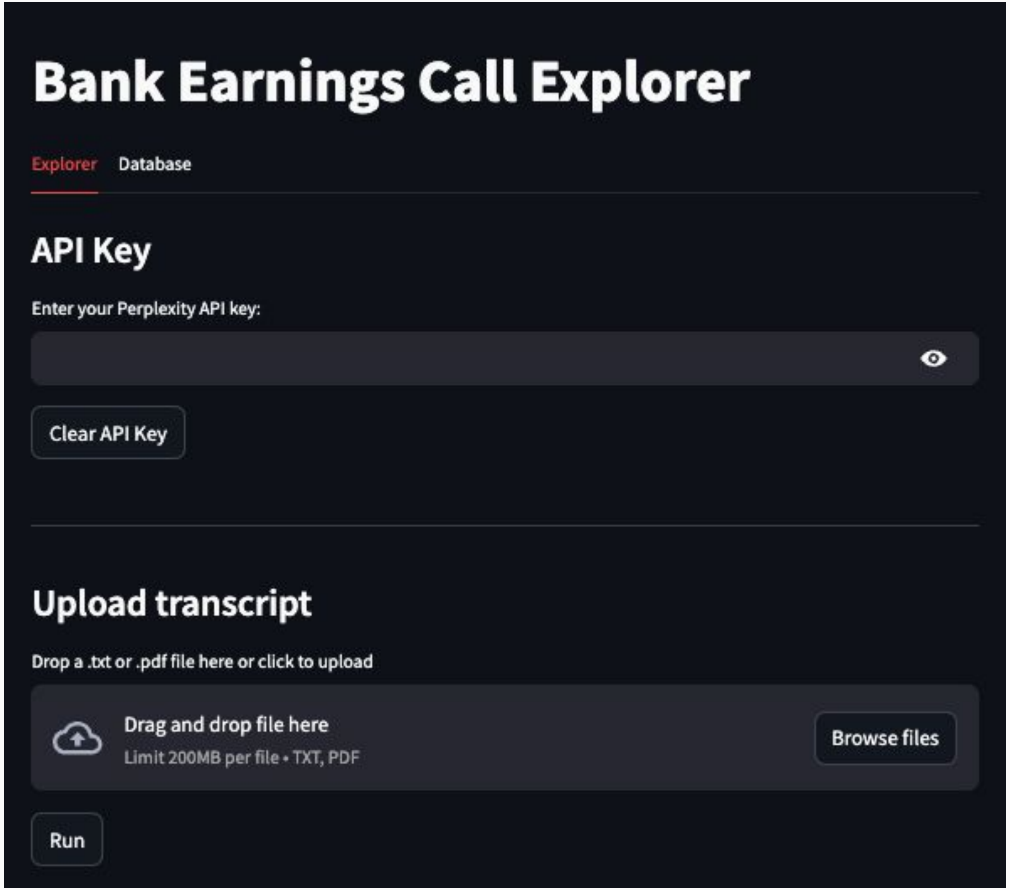
1. The bank is monitoring GDP growth, unemployment rates, and consumer spending patterns. There's uncertainty about how these factors may impact loan demand and credit quality.
2. The bank is anticipating changes to capital rules and stress testing methodologies. This could affect capital allocation, return on equity, and overall business strategy.
3. Despite expected interest rate cuts, deposit costs may remain elevated. The bank is working to optimize its deposit mix and pricing to manage net interest margins.

Solution: Part 4

Section 4:

Applying model to any transcript

- Allows BoE employee to create summaries and gather data from calls outside of the corpus



Summary and Conclusions

Successfully Achieved

- Evaluating call-specific sentiment
  - Contextualising it with quarter-specific and bank-specific values
- Summary of call
- Topic analysis
- Potential concerns flagged
- Stock price change predictor
- Banks in danger of failing got flagged as outliers (Credit Suisse, Silicon Valley)\
- Creating an interactive product

Further Improvements

- Assessing accuracy in sentiment of key individual analysts and/or banks and risk management organisations, when compared to financial changes to see whether there are specific participants in Q&As are markers to track
- Adding more data around Q&A transcripts, such as financial earnings statistics, general market data, to enhance the XGBoost model and contextualise the LLM output
- Expand the XGBoost model to look at price changes for the earning period and call date specifically, tuning it to provide better results
    - Increase the number of categories so BoE can distinguish between predicting a drop in stock price within a normal range, and predicting a bank in danger of failing
- Liaising with BoE to assess which graphics and metrics are most of use and iterate accordingly.

Thank You