Replication of de Kok, T. (2025).

ChatGPT for textual analysis? How to use generative LLMs in accounting research.

Group 7 | CrossBay

Members | Shutong Dong, Minghao Yan

Contents

01

Motivation

The Motivation: Why "Non-Answers" Matter?

01

The Context

Q&A sessions are the most direct channel for analysts to probe for details.

02

The Problem

Managers use "Strategic Obfuscation"—subtle statements of inability or unwillingness to answer—to conceal bad news.

03

The Consequence

These signals increase Information Asymmetry and hinder market pricing efficiency.

Research subject: Definition of Non-Answers

Lable 1

Strategic Non-Answers

> Specific Scenarios:

“We don’t provide that”

“I can’t comment.”

“Not in a position to disclose.”

Lable 0

Valid Responses

> Specific Scenarios:

Normal anticipated uncertainty

Situations where a substantive answer is given immediately after a disclaimer

02

Background

Background

01

Manual coding

rich but not scalable; expensive and slow

02

Rule-based / regex

scalable but brittle;

misses indirect refusals & context

Why it's hard and LLMs help

"non-answer" is pragmatic → meaning

depends on intent and what's omitted;

stronger semantic + pragmatic understanding

(refusal/deflection/hedging)

03

Design choice

two-stage (keyword → LLM) reduces

cost + false alarms

04

03

Data & Method

Data

Firm-level filtering → final transcript sample (Table 1):

- Start: 12,614 unique firms (CIKs) in CapitalIQ (2013–2022)
- Exclude: missing industry + finance & utilities (GICS 40 & 55) → −6,988
- Exclude: no earnings call Q&A transcripts OR < 5 Q&A exchanges per call → −69
- Exclude: no Q&A pairs meeting min length (Q ≥30 chars, A ≥10 chars, total ≥75 chars) → −86
- Final: 5,471 firms and 166,848 Q&A pairs
- For analysis: 100 manually labeled pairs + 1,000 random pairs

Manual Benchmark

Manual label rule (positive class: non-answer):

We define **non-answer** as: a response is a non-answer if it includes a statement, explanation, or justification indicating an inability or unwillingness to answer the question.

Benchmark set:

- 100 randomly drawn Q&A pairs were manually labeled.
- Class balance: 81 answers vs 19 non-answers (19% non-answers).
- We treat “non-answer” as the positive class in evaluation.

Manual Benchmark

Methods

Four approaches we compare

We implement four approaches:

1) Gow et al. (2021) — rules

Regex categories (REFUSE, UNABLE, AFTERCALL, …)
Flag non-answers by pattern match.

2) Spark Pro — zero-shot GLLM

Structured prompt → JSON output
Binary “noanswer” label + explanation.

3) Spark Max — stronger zero-shot GLLM

Same prompt as 2), more advanced model
Expect higher recall / better reasoning.

4) Keyword + Spark Max — hybrid

Keyword pre-filter first
Only call LLM when keywords hit
Goal: fewer false positives + lower cost.

Evaluation

We report the same metrics as de Kok (2024):

Confusion matrix (positive = non-answer)

	Pred = Non-answer	Pred = Answer
True = Non-answer	TP	FN
True = Answer	FP	TN

Metrics

Accuracy = (TP + TN) / N

Type I error = FP / (FP + TN)

Type II error = FN / (FN + TP)

Precision = TP / (TP + FP)

Recall = TP / (TP + FN)

F1 = 2TP / (2TP + FP + FN)

04

Results and Conclusion

Results

Table 2 — Performance on 100 labeled Q&As

Key metrics for the non-answer class:

Method	Precision	Recall	F1	Accuracy	Type I	Type II
Gow et al. (2021)	0.56	0.26	0.36	0.82	0.05	0.74
Spark Pro (zero-shot)	0.56	0.47	0.51	0.83	0.09	0.53
Spark Max (zero-shot)	0.67	0.53	0.59	0.86	0.06	0.47
Keyword + Spark Max	**0.82**	**0.47**	**0.60**	**0.88**	**0.02**	**0.53**

Benchmark: 81 answers, 19 non-answers (N=100)

Headline: LLMs increase performance (e.g. Precision, Recall, F1, and Accuracy); the keyword+LLM hybrid cuts false positives and achieves the highest F1 (0.60).

Interpretation

What do these patterns mean?

Takeaways from Table 2:

- Rule-based baseline is conservative → low false positives but many misses (high Type II error).
- Zero-shot LLMs act more “inclusive” → higher recall but more false positives.
- Keyword pre-filter shifts the trade-off: fewer false positives, higher precision.
- In practice, which is better depends on whether you fear FP or FN more.

Cost (reported in Table 2, RMB, for N=100):

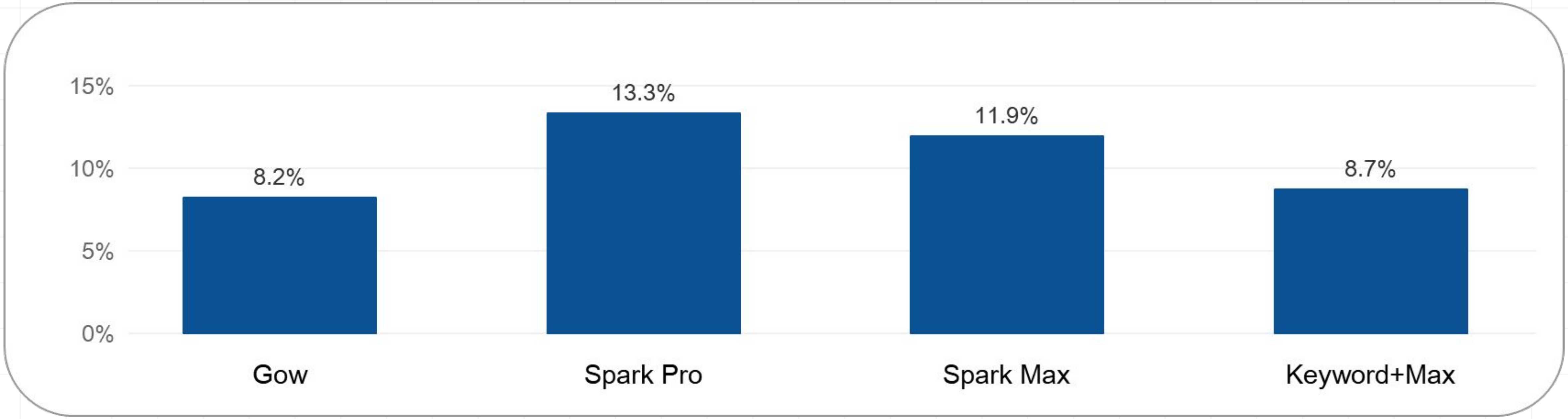
Spark Pro ≈ 0.361 • Spark Max ≈ 1.595 • Keyword + Spark Max ≈ 0.864

Keyword filtering reduces LLM calls → lower cost and fewer false positives.

Incidence

Table 3 — Non-answer rates in 1,000 Q&As

Estimated non-answer rate (mean over 1,000 random pairs):



- Rates vary from 8.2% (Gow) to 13.3% (Spark Pro).
- Hybrid keyword+LLM is close to Gow in incidence (8.7%), consistent with its higher precision.

Comparison

Why are we below the paper's best performance?

Paper vs replication (qualitative):

de Kok (2024)

- Fine-tuned, multi-step ChatGPT pipeline
- Evaluation set: 500 Q&A pairs
- Reported: accuracy ≈ 0.96, non-answer F1 ≈ 0.87

This replication

- Zero-shot Spark Pro / Spark Max
- Small benchmark: 100 labeled pairs
- Best: keyword + Spark Max → accuracy 0.88, F1 0.60

Likely reasons for the gap:

- Different data source + screening rules (CapitalIQ vs Finnhub).
- No fine-tuning; only zero-shot prompts.
- Much smaller evaluation set → more sampling variability.
- Model choice: Spark Pro/Max ≠ ChatGPT/ChatGPT-4 in the paper.

Conclusion

What we learn from this small replication:

- Directionally consistent with de Kok (2024): LLM-based methods improve non-answer detection.
- Pipeline design matters: keyword pre-filters can sharply reduce false positives and cost.
- Best result here (hybrid): non-answer F1 = 0.60 vs baseline F1 = 0.36.
- But performance is sensitive to data, model choice, and the size of the labeled benchmark.

Next steps (if we extend this project):

- Label a larger evaluation set (e.g., 500+) and check robustness.
- Try a multi-step pipeline (rationale extraction → decision), or fine-tune a small model.

Thank you!