On Finding Inconsistencies in Documents

Charles J. Lovering[1♢], **Seth Ebner**[1♢],
Brandon Smock[1], **Michael Krumdick**[1], **Saad Rabbani**[2],
Ahmed Muhammad[2], **Varshini Reddy**[1], **Chris Tanner**[1]

[1]Kensho Technologies [2]S&P Global
[♢]Core Contributor
{first}.{last}@kensho.com

Abstract

Professionals in academia, law, and finance audit their documents because inconsistencies can result in monetary, reputational, and scientific costs. Language models (LMs) have the potential to dramatically speed up this auditing process. To understand their abilities, we introduce a benchmark, **FIND** (**F**inding **IN**consistencies in **D**ocuments), where each example is a document with an inconsistency inserted manually by a domain expert. Despite the documents being long, technical, and complex, the best-performing model (`gpt-5`) recovered 64% of the inserted inconsistencies. Surprisingly, `gpt-5` also found inconsistencies already present in the original documents. For example, on 50 arXiv papers, we judged 136 out of 196 of the model's suggestions to be legitimate inconsistencies missed by the original authors. However, despite these findings, even the best models miss almost half of the inconsistencies in **FIND**, demonstrating that inconsistency detection is still a challenging task.

🤗 kensho/FIND

1 Introduction

Documents such as financial statements and scientific articles are audited for accuracy. Automated inconsistency detection could reduce the burden of auditing and improve document quality (YesNo-Error, 2024; Zhang and Abernethy, 2025; Wang et al., 2025; Xi et al., 2025). Inconsistency detectors could also be useful in automated workflows, such as for inspecting chain-of-thought traces (He et al., 2025) or reviewing generated reports.

As model capabilities improve, automated consistency checking could become ubiquitous, just as grammar and spelling checkers are today. Even models with low precision can be helpful because inconsistencies are often hard to find but easy to verify. Moreover, in part because the task requires processing long and complex documents, achieving high recall is difficult, suggesting significant room for future performance improvement.

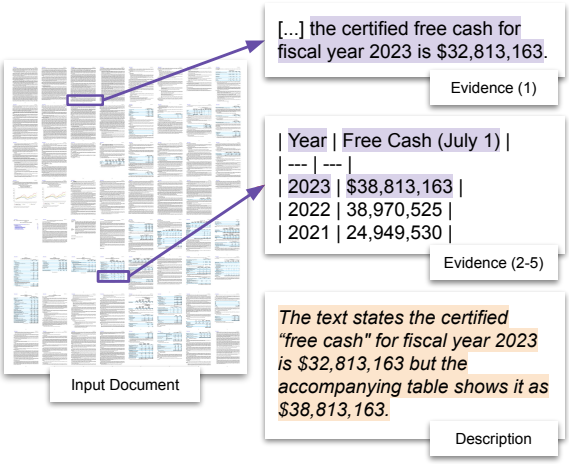


Figure 1: The task is to find inconsistencies within an input document. For each inconsistency, the model: (a) identifies the evidence spans that constitute the inconsistency and (b) generates a corresponding description.

Our Research Question

How effectively do current language models find inconsistencies within documents?

To approach this question, we introduce a benchmark, **FIND** (**F**inding **IN**consistencies in **D**ocuments), containing 375 test and 125 development problems. Each problem is a document with an inconsistency manually inserted by a financial expert. For each inconsistency the model identifies, it is expected to respond with *evidence spans* from the document that demonstrate the inconsistency and a natural language *description* of what makes those spans inconsistent. The documents are mostly financial reports, economic analyses, and general non-fiction. Documents in **FIND** are generally long: they have a median length of 35k tokens, and 25% of documents contain over 78k tokens.

Findings of the Association for Computational Linguistics: ACL 2026, pages 33523–33564
July 2-7, 2026

The length of the documents contributes to the challenge of the task. Auditing long documents is extremely resource intensive. The long context creates more opportunities for inconsistencies among related information. Experts interviewed in the course of this study noted that the review process to verify the consistency of tables in documents similar to those in **MFR** (Section 4) averaged four hours. The full review process and error checking when preparing a document like a 10-K takes even longer, with large teams spending weeks on a single document.[1] Our results show performance decreases as document length increases (Section 6).

Problem Scope **FIND** tests the ability of models to detect inconsistencies *within* documents, excluding both *resolving* inconsistencies and detecting inconsistencies that require *external validation* (such as cross-referencing values against a database).

Contributions We (1) contribute **FIND**, an expert-annotated benchmark for inconsistency detection. We (2) show that the best-performing models recall over 60% of inserted inconsistencies, and (3) find that, surprisingly, models find meaningful amounts of otherwise undiscovered inconsistencies with over 50% precision. *Models found five inconsistencies in drafts of this paper (see Appendix A).*

2 Related Work

Domains The ability to detect internal inconsistencies is useful in many domains, which prior work has explored. For example, Deußer et al. (2023) and Wang et al. (2025) examine inconsistencies in financial reports and filings. In the legal domain, Andow et al. (2019) consider privacy policies, while Mantravadi et al. (2025) and Choudhury et al. (2026) consider contracts and other legal documents. Masuda et al. (2016) consider system requirements and specifications. Sagimbayeva et al. (2025) construct a benchmark containing political statements. The Code.Debug task in ∞Bench (Zhang et al., 2024) tests models' abilities to detect inconsistencies in code. Concurrent work has also investigated finding inconsistent statements in academic publications, often through the lens of automating the review process (Son et al., 2025; Zhang and Abernethy, 2025; Dycke and Gurevych, 2025; Xi et al., 2025; Bianchi et al., 2025).

Broader domain datasets have also been constructed. ContraDoc (Li et al., 2024) considers news, stories, and Wikipedia articles. WikiContradiction (Hsu et al., 2021) is based on Wikipedia articles tagged by editors as self-contradictory.

Knowledge Conflicts Information can be in conflict in many ways. For example, question answering and retrieval augmented generation systems must contend with retrieved documents being in conflict with each other or with a model's parametric knowledge (Xu et al., 2024; Liu and Roth, 2025; Cattan et al., 2025). When conflicts occur, models perform worse (Chen et al., 2022; Hong et al., 2024; Hou et al., 2024; Liu et al., 2025; Zeng et al., 2025) possibly because models detect these conflicts poorly (Pham et al., 2024; Jiayang et al., 2024; Kurfali and Östling, 2025; Gokul et al., 2025). Relatedly, corpus-level inconsistencies in Wikipedia have been studied by Semnani et al. (2025). Tyen et al. (2024) and He et al. (2025) show that models struggle to detect errors within chain-of-thought streams and long reasoning chains. Fact checking, though also related, differs from our work in that the task of finding inconsistencies focuses on conflicts that arise *within* a document rather than on conflicts between information in the document and information from external sources.

Most similar to our work, Kurfali and Östling (2025) insert multiple conflicting "needles" into a document and ask a model a question involving the needles, finding that models frequently do not detect the conflict. As with other work based on targeted queries, Kurfali and Östling (2025) do not explore the more general problem of finding all inconsistencies in a document. Additionally, the inconsistencies in their dataset do not require domain knowledge or complex reasoning to uncover.

3 Experimental Design

Our dataset consists of professionally prepared documents with at least the one known inconsistency we inserted in them. We task models with finding all inconsistencies within a document, not just the one we inserted. If we instead required the model to return at most one inconsistency, it could return a naturally occurring one from the original document, which a metric comparing against the inserted one would not credit. Therefore, we formulate the task as returning a list of inconsistencies, which is also a more realistic and useful task. Our evaluation focuses on whether the model successfully returned

[1] Form 10-K is a report that a company files annually with the SEC in the United States.

the inserted inconsistency within that list; any annotated ground truth we do have is more useful for recall than for precision (Section 3.3). For precision, we view manual grading of predicted inconsistencies as the most reliable evaluation strategy (Section 7).

3.1 Inconsistency Structure

An *inconsistency* $\mathcal{I}$ has two parts (Figure 1). First, the evidence $\mathcal{E} = \{\varepsilon_1, \ldots, \varepsilon_n\}$ is a set of spans from the document $\mathcal{D}$. Second, the description δ is a natural language explanation of how the evidence spans are inconsistent.

There is a unique span in the document (by reading order) where an inconsistency can first be detected. We call this span the *trigger* (and include it in the evidence). The remaining evidence consists of the nearest set of spans preceding the trigger that are necessary to identify the inconsistency. Guidelines with examples of how to handle span selection edge cases are linked in Appendix C.

3.2 Task Definition

A model M is to return all inconsistencies within a document $\mathcal{D}$ from data source $\mathcal{S}$. Each predicted inconsistency $\hat{\mathcal{I}}$ must conform to the XML-like formatting specified in the prompt (see Appendix D.2) to be considered valid. Our task reflects the input-output interface of a tool, with extractive spans and explanatory descriptions. Other work has framed similar tasks using natural language inference (Hsu et al., 2021; Deußer et al., 2023; Song et al., 2025), natural logic (Krishna et al., 2022; Aly and Vlachos, 2024), and sheaves (Zadrozny and Garbayo, 2018; Huntsman et al., 2024).

3.3 Metric Definitions

We evaluate the evidence, description, and their combination (the full task) with different metrics. Because the model returns a list of candidate inconsistencies, we must compare each candidate to the expected inconsistency. For each document, we score each candidate against the expected inconsistency and keep the best match, and then average these per-document scores across the dataset:

$$\Lambda := \frac{1}{|\mathcal{S}|} \sum_{(\mathcal{D},\mathcal{I})\in\mathcal{S}} \max_{\hat{\mathcal{I}}\in M(\mathcal{D})} \lambda(\hat{\mathcal{I}}; \mathcal{I}, \mathcal{D}),$$

where λ is the underlying metric (evidence, description, or full task), $M(\mathcal{D})$ is the model's list of predicted inconsistencies for document $\mathcal{D}$ from source $\mathcal{S}$, and $\mathcal{I}$ is the expected inconsistency. $\hat{\mathcal{I}}$ is a predicted inconsistency (or answer). We call Λ *response-level*. In result tables, we report the response-level scores $\Lambda_\mathcal{E}$, Λ_δ, and $\Lambda_\mathcal{I}$ for evidence, description, and full task, respectively.

Evidence Metric Our evidence metric measures how well the predicted spans align with the expert annotated spans. This lexical method is fast and easily reproducible but sensitive to character level differences, offering a different set of tradeoffs compared to the neural metrics used below. It uses bipartite matching with weights based on longest common substrings to align the predicted and reference spans. The metric's output ranges from 0 to 1, where a score of 1 for a predicted inconsistency indicates an exact match against the expected inconsistency, and 0 indicates no alignment. Appendix E and Algorithm E.1 contain the design and algorithm details as well as discussion of a lenient version of the metric. We report the response-level score $\Lambda_\mathcal{E} \in [0, 1]$ in Table 5.

Description Metric To evaluate the predicted description against the reference description, we use BLEURT (Sellam et al., 2020), a trained neural text generation metric that evaluates how well the predicted text conveys the meaning of the reference text.[2] Though unbounded, its range is typically between 0 and 1.[3] These scores are best used to rank models on the same set of documents, rather than comparing a model's performance across different sets of documents. We report the response-level score Λ_δ in Table 5.

Task Metric We use LLM-as-a-judge (LMJ) to holistically compare a predicted inconsistency (answer) to the expected inconsistency. The LMJ score is conditioned on the predicted and expected inconsistencies (including all the evidence and descriptions) and the document. It returns a boolean (0 or 1). Informed by Wei et al. (2024); Arora et al. (2025), we found `gpt-4.1` an effective judge, reaching 0.96 Cohen's kappa with human judgments on 200 model responses sampled from the development set (some examples were answered by multiple models). These results suggest for our dataset that the LMJ is a useful proxy for a human

[2] We use the recommended `BLEURT-20` checkpoint from the https://github.com/lucadiliello/bleurt-pytorch library as the underlying model.

[3] https://github.com/google-research/bleurt/blob/cebe7e6/README.md#interpreting-bleurt-scores

Document (Input) and Evidence Spans (Annotation)

```
...
# History of Enrollment
Listed below are the District's fall enrollment figures for
the last five school years.

| | 2019-2020 | 2020-2021 | 2021-2022 | 2022-2023 |
| - | - | - | - | - |
| Pre-K | 180 | 176 | 204 | 201 |
| Elementary (K-3) | 1,507 | 1,431 | 1,419 | 1,400 |
| Middle (4-6) | 1,174 | 1,103 | 1,114 | 1,131 |
| Junior High (7-8) | 838 | 820 | 803 | 752 |
| High School (9-12) | 1,509 | 1,517 | 1,563 | 1,610 |
| Total | 5,155 | 5,047 | 5,103 | 5,094 |
...
```

Description (Annotation)

In the table, the total enrollment reported (5,155) does not match the sum of the listed rows (5,208).

Figure 2: *Numeric* inconsistency from **EMM**.

grader. Because the metric is boolean, the response level version of this metric, $\Lambda_{\mathcal{I}}$, functions like recall for the inserted inconsistency.[4] (In Section 7.1, we estimate precision by manually grading inconsistencies from a top-performing model.)

4 Dataset

To build **FIND**, we collected professional documents, described below, in the public domain (**BLS**, **SEC**, **EMM**, **PG**) or with direct usage permission (**PRE**). **FIND** has 125 development and 375 test examples each containing an annotated inconsistency. The documents in this dataset are unlikely to be in current model training corpora; most were released in 2025 (some in late 2024) and **PRE** is a private source. However, the auxiliary analysis data (**MFR**, **cs.CL**) are not date-protected (see Appendix G). Our goal was to create an informative benchmark with expert annotations—not necessarily an evergreen dataset. Guidelines used for the annotation and data creation process are linked in Appendix C.

The presence of unannotated, naturally occurring inconsistencies in the documents in **FIND** is expected. Exhaustive annotation is infeasible for documents of the length and complexity we consider. The professionals who prepared the original documents had every incentive to eliminate inconsistencies. Attempting exhaustive annotation would have required using much shorter documents, at the cost of the realism that makes this benchmark valuable. Additionally, we designed our recall metric to be robust to naturally occurring inconsistencies.

[4]Our setup allows multiple answers per response to be graded positively, but this is rare (Appendix H.3).

BLS	Bureau of Labor Statistics reports on U.S. employment and economic trends[5]
PRE	S&P presale reports analyzing pending bond deals, e.g., credit risk[6]
SEC	10-Q filings covering finances and operations of public U.S. companies[7]
EMM	Bond disclosures filed by U.S. municipalities when issuing debt[8]
PG	Non-fiction books from Project Gutenberg[9]

Table 1: Document sources in **FIND**.

4.1 Sources

We use English documents from professional sources and have financial experts insert inconsistencies for the purposes of the task. Document samples are in Appendix G (Figures G.2 to G.7), and Figures 2 to 4 show examples of inserted inconsistencies. The sources used in **FIND** are in Table 1.

We also examine *supplemental* documents with naturally occurring inconsistencies. These sources are excluded from the main **FIND** dataset and are used in additional experiments in Section 7.

cs.CL Research papers from the `cs.CL` category on arXiv.[10] Because arXiv provides version histories of papers, we were able to select documents where papers' authors found and fixed (at least) one inconsistency. We manually inspected the diffs between successive versions of papers to identify such corrections of inconsistencies. Note that this process is biased toward inconsistencies that humans (the papers' authors) were able to find and resolve. We also include "control" documents sampled from the `cs.CL` category without conditioning on the existence of a known inconsistency.

MFR Miscellaneous financial reports containing inconsistencies found in the wild by analysts, covering annual and ad-hoc financial reports. Most of the inconsistencies involve long tables.

4.2 Why Mostly Finance Sources?

The desiderata for our dataset were (1) permissible licenses, (2) high quality documents, and (3) diversity of content (tables and analyses, document lengths). The chosen finance and economic sources covered these needs. Also, prior work (Section 2)

[5]https://www.bls.gov/

[6]https://www.spglobal.com/ratings/en/regulatory/presale-reports

[7]https://www.sec.gov/edgar/search/

[8]https://emma.msrb.org/

[9]https://www.gutenberg.org/

[10]https://arxiv.org/list/cs.CL/recent

Document (Input) and Evidence Spans (Annotation)

```
Deferred revenue expected to be recognized as revenue more
than one year     subsequent  to  the  balance  sheet  date
is   included in short-term liabilities   on  the  condensed
consolidated balance sheets.
```

Description (Annotation)

The sentence incorrectly classifies deferred revenue expected after one year as a short-term liability. This creates a contradiction because such revenue should be reported as a long-term liability, based on standard accounting timelines.

Figure 3: *Non-numeric* inconsistency from **SEC**.

Document (Input) and Evidence Spans (Annotation)

```
These individuals would have preferred full-time employment
but were working part time because their hours had been
reduced or they were unable to find full-time jobs.
(See table 3.)  ...
###    Table 3  .       Employed people by disability status,
occupation, and sex, 2024 annual averages
| Occupation | People with a disability | People with no
disability | ...
```

Description (Annotation)

The paragraph gives information on the workers disability employed part time and full time and in the end of the paragraph it states "see Table 3" but Table 3 instead gives information on the "Employed people by disability status, occupation, and sex..."

Figure 4: *Structural* inconsistency from **BLS**.

suggests that models have low recall in identifying inconsistencies, but recall is critical in the finance domain, as even minor inconsistencies can result in penalties, making **FIND** an appropriate testbed for model development.

4.3 Annotators and Quality Control

To build a practical taxonomy of inconsistency types, we interviewed financial experts and then collaborated with a financial expert annotation team to create an annotation scheme. Then, the annotation team inserted inconsistencies into the documents paired with iterative review and validation. After annotation, we reviewed the dataset again. On average, creating, annotating, and reviewing each inconsistency took two hours. The full process is outlined in Figure G.1.

4.4 Inconsistency Types

The main types we consider are high-level distinctions of what type of information is in conflict. These types are listed below, with additional axes—modality (text, table, or both), scope (short or long range), and number of evidence spans—described in our guidelines. For each document, we provided a setting of the axes for the annotators to use when creating the inconsistency. Because creating inconsistencies is difficult, these settings served as suggestions, and the types were re-labeled after annotation based on the inconsistencies actually created. The distribution of inconsistency types is shown in Table 2 and includes:

		BLS	PRE	SEC	EMM	PG	Total
Content	Numeric	48	48	40	40	28	204
	Non-numeric	13	14	12	10	14	63
	Structural	14	13	23	25	33	108

Table 2: Distribution of inconsistency types in **FIND**.

(1) *Numeric* inconsistencies, e.g., incorrect sums or averages. See Figure 2.

(2) *Non-numeric*, conceptual, or logical inconsistencies, e.g., incorrect analysis or application of domain knowledge. See Figure 3.

(3) *Structural* inconsistencies, e.g., the absence of expected information or an incorrect reference to a table. See Figure 4.

4.5 Dataset Analysis

FIND consists of the first five sources listed in Section 4.1, each with 75 test documents. See Table 3 for dataset statistics. The average document length varies from 10k tokens (**BLS**) to 123k tokens (**EMM**). Evidence statistics are similar across sources (albeit with high variance): approximately 5 spans per document and 11 tokens per evidence span. Token counts were measured with `tiktoken` (cl100k_base).[11] Supplemental sources (**MFR** and **cs.CL**) are excluded from the main dataset and used in additional studies (Section 7) with manual grading of the models' findings. For expediency, the evidence in **cs.CL** was not annotated (explaining the empty cells in Table 3) but we do include reference descriptions. Lastly, **MFR** has a higher mean of 19 evidence spans per document but shorter average span length due to its focus on sums over long tables.

Figure 5 shows that most inconsistencies are located in the first half of the document, with the average position of the evidence being 30% through the document (Table 3), a limitation of our inconsistency distribution. Still, over 42% of the documents have evidence beyond 10k tokens (66% of documents when excluding documents from **PRE**

[11] https://github.com/openai/tiktoken

	$\mathcal{D}$	kTok/$\mathcal{D}$	$\varepsilon/\mathcal{D}$	Tok/ε	ε Pos%
BLS	75	$10_{\pm 14.8}$	$5_{\pm 3.3}$	$16_{\pm 60.9}$	$28_{\pm 23.4}$
PRE	75	$12_{\pm 5.5}$	$7_{\pm 10.6}$	$9_{\pm 7.9}$	$37_{\pm 23.1}$
SEC	75	$63_{\pm 102.8}$	$5_{\pm 4.6}$	$11_{\pm 8.6}$	$37_{\pm 22.3}$
EMM	75	$123_{\pm 82.7}$	$6_{\pm 8.0}$	$11_{\pm 9.8}$	$24_{\pm 20.8}$
PG	75	$109_{\pm 106.6}$	$4_{\pm 2.6}$	$8_{\pm 6.0}$	$22_{\pm 24.1}$
FIND	375	$63_{\pm 89.6}$	$5_{\pm 6.7}$	$11_{\pm 28.3}$	$30_{\pm 23.6}$
MFR	45	$48_{\pm 33.6}$	$19_{\pm 9.0}$	$4_{\pm 1.3}$	$37_{\pm 26.8}$
cs.CL Idnt	25	$20_{\pm 10.1}$	—	—	—
cs.CL Ctrl	25	$25_{\pm 25.5}$	—	—	—

Table 3: **Statistics by document source**: number of documents ($\mathcal{D}$), mean tokens per document (in thousands) (kTok/$\mathcal{D}$), mean evidence spans per document ($\varepsilon/\mathcal{D}$), mean tokens per evidence span (Tok/ε), and mean relative position of evidence within documents (ε Pos%). Subscripts denote standard deviations. **FIND** row reports statistics pooled over the sources above; bottom section reports supplemental sources used in Section 7.

Model	Valid	kTok/$\mathcal{D}$	$\hat{\mathcal{I}}/\mathcal{D}$	$\varepsilon/\hat{\mathcal{I}}$	Tok/ε
gemma-3-12b-it	99.8	0.6	3.4	2.4	16.8
gemma-3-27b-it	99.8	0.9	4.7	2.5	21.7
gpt-oss-20b	96.0	5.5	2.3	5.9	8.0
gpt-oss-120b	99.5	2.7	1.9	3.2	16.9
sonnet-v4	97.6	4.9	1.8	4.9	8.3
gpt-5-mini	100.0	5.9	2.6	2.6	31.9
gpt-5	99.0	11	3.2	2.5	25.7
o3-mini	99.5	2.9	1.4	5.0	10.3
o3	99.5	5.4	1.9	2.3	23.2
gemini-2.5-flash	94.3	12	4.3	4.1	27.5
gemini-2.5-pro	99.5	13	5.1	3.7	17.8

Table 4: **Model response statistics.** Format validity rate (%), mean tokens per document (in thousands) (kTok/$\mathcal{D}$), mean answers per response ($\hat{\mathcal{I}}/\mathcal{D}$), mean evidence spans per answer ($\varepsilon/\hat{\mathcal{I}}$), and mean tokens per evidence span (Tok/ε).

and **BLS**), with the 75th percentile being at 17k tokens (25k tokens when **PRE** and **BLS** are excluded). While much of the evidence appears relatively early within the documents, a large portion appears later.

5 Methods

We simply prompt an LLM to find inconsistencies (as do Deußer et al. (2023); Li et al. (2024); Choudhury et al. (2026); Xi et al. (2025)), and leave model development to future work.

Open Source Dataset and annotation information are in Appendix C. Prompts are in Appendix I.

Models We test open-weight models `gpt-oss-20b`, `gpt-oss-120b` (OpenAI, 2025a), `gemma-3-12b-it`, and `gemma-3-27b-it` (Gemma, 2025); and closed-source models `claude-v4-sonnet` (Anthropic, 2025), `gpt-5-mini`, `gpt-5` (OpenAI, 2025b), `o3-mini`, `o3` (OpenAI, 2025c), `gemini-2.5-flash` (Google DeepMind, 2025a), and `gemini-2.5-pro` (Google DeepMind, 2025b).

Context Length Limits Because of hardware limitations and compute costs, we set upper limits on the number of input tokens (105k) and the number of output tokens (20k). In the test set 83 out of 375 documents (at most, depending on the model and tokenizer) had their text truncated. However, only for the `gemma-3` models was any evidence included in the truncated text, and even then only for two documents. Moreover, because no model approaches a saturated score on this benchmark, we consider this truncation to be a reasonable tradeoff in terms of cost.

6 Results and Analysis

Below, we present results on **FIND**. In Table 5, we bold the top-performing model and models with no statistically significant difference from that model score ($\alpha = 0.05$) via a paired t-test. For the **AVG** column we use a clustered test (Miller, 2024) over the sources. Standard errors are in Table H.4.

Model Response Statistics Table 4 covers basic statistics of how different models responded. All models consistently generated output in the valid format (8 of 11 models gave valid output over 99% of the time, and the worst case was still over 94%). No model consistently saturated the available output token budget of 20k tokens. However, the top performing models, `gpt-5` and `gemini-2.5-pro`, used among the most tokens and generated among more answers per response compared to other closed-source models.

Overall Performance The response-level metrics tell two basic stories (Table 5). First, the open-weight models underperform closed-source models: besides `o3-mini`, all closed-source models outperform all open-weight models. Second, larger models outperform the smaller models within a series, with `gpt-5` and `gemini-2.5-pro` performing best overall, each with over 60% recall. The description score averages fall in a narrow range (27–46, with 7 of 11 models scoring between 36–42), suggesting that the descriptions are of similar quality across models. The rankings formed by the three metrics over the model and source scores show strong agreement: pairwise Spearman rank correlations range from 0.86 to 0.92 (all $p < 0.001$).

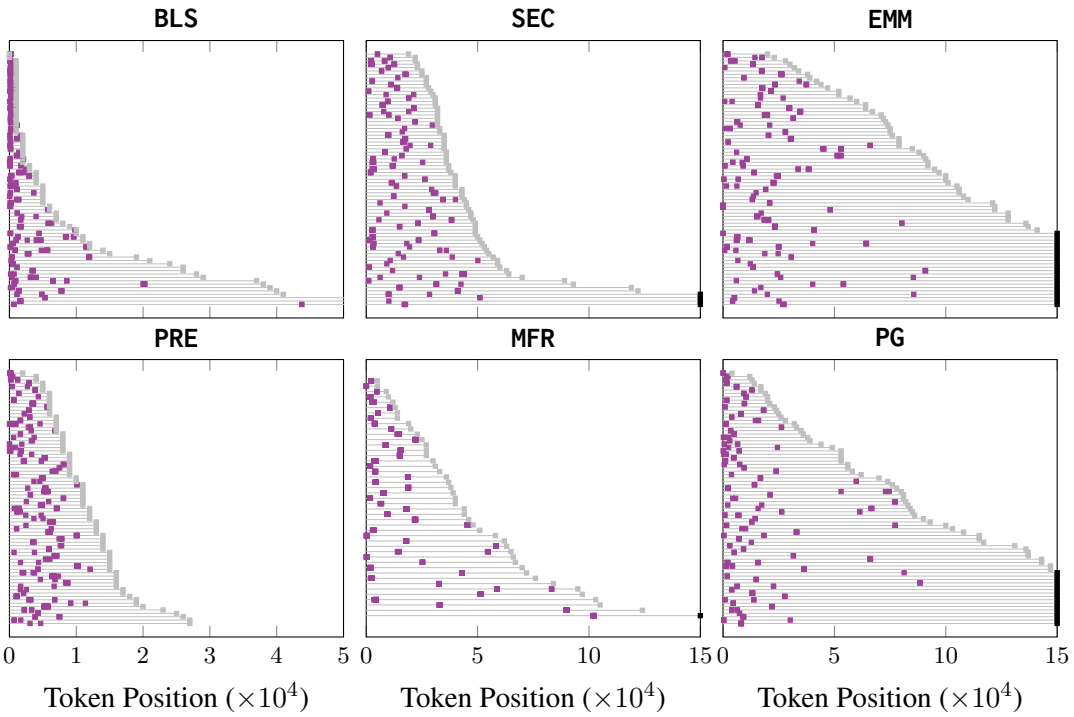


Figure 5: **Absolute Evidence Locations and Test Document Lengths.** Each row corresponds to a document, with each row ending at a gray square indicating document length. Purple squares mark evidence locations. Black squares indicate that the document exceeds the displayed length. (For **BLS** and **PRE** the axis ends at 5×10^4.)

	Evidence Score ($\Lambda_{\mathcal{E}}$)						Description Score (Λ_{δ})						Task Score ($\Lambda_{\mathcal{I}}$)					
Model	BLS	PRE	SEC	EMM	PG	AVG	BLS	PRE	SEC	EMM	PG	AVG	BLS	PRE	SEC	EMM	PG	AVG
gemma-3-12b-it	18	5	6	3	4	7	43	37	35	34	33	36	57	25	7	3	9	20
gemma-3-27b-it	23	8	4	1	0	7	45	38	36	34	34	37	61	27	7	8	12	23
gpt-oss-20b	30	11	3	2	2	10	45	32	27	28	28	32	21	3	1	0	3	6
gpt-oss-120b	35	14	7	5	4	13	47	36	34	34	32	37	28	3	0	0	1	6
sonnet-v4	**44**	**35**	13	14	9	**23**	**53**	43	30	33	32	**38**	**83**	49	20	21	23	39
gpt-5-mini	37	27	25	17	9	**23**	46	43	39	38	38	**41**	80	53	47	**37**	39	51
gpt-5	**43**	**36**	**42**	**27**	**18**	**33**	50	**46**	**44**	**41**	**43**	**45**	**87**	**67**	**73**	**43**	**52**	**64**
o3-mini	23	5	1	0	1	6	39	31	19	23	22	27	56	17	8	3	12	19
o3	**40**	29	25	17	10	**24**	49	42	39	35	37	**40**	**85**	49	57	31	40	53
gemini-2.5-flash	**42**	25	19	13	5	**21**	**53**	45	38	39	36	**42**	75	48	36	25	31	43
gemini-2.5-pro	**45**	**38**	**36**	**27**	12	**32**	**54**	**48**	**42**	**43**	**44**	**46**	**88**	**71**	61	**40**	**45**	**61**

Table 5: Evidence, description, and task scores across the **FIND** test set (response-level metrics). For all three scores, higher is better. Scores are presented as percents (out of 100).

Performance Per Data Source The order of performance across sources for `gpt-5` and `gemini-2.5-pro` (the best models) largely matches the order of average document length, with worse performance on longer documents. Models did not saturate their output token budgets, indicating that lower recall on longer documents results from missed inconsistencies rather than output length constraints. For the tested models, **BLS** was the easiest task (highest model scores), whereas **EMM** and **PG** were the hardest (lowest model scores). Qualitatively, inconsistencies in **BLS** are simpler because the documents tend to comprise a few tables and straightforward analysis.

Performance vs. Input Length and Inconsistency Type Figure 6 shows that input length correlates with task score, based on a linear regression conditioned on the data sources and inconsistency types as control variables (all effects are significant). For clarity, the figure displays only some of the models. See Table H.5 for regression coefficients. The inconsistency types also account

Model	Task Score ($\Lambda_{\mathcal{I}}$) Num	Non	Struc
gemma-3-12b-it	0.06	0.10	0.02
gemma-3-27b-it	0.10	0.03	0.02
gpt-oss-20b	0.26	0.25	0.06
gpt-oss-120b	0.27	0.38	0.06
sonnet-v4	0.45	0.49	0.23
gpt-5-mini	0.60	0.56	0.31
gpt-5	0.68	0.70	0.54
o3-mini	0.24	0.27	0.06
o3	0.58	0.59	0.39
gemini-2.5-flash	0.50	0.46	0.28
gemini-2.5-pro	0.67	0.65	0.48

Table 6: Task score by inconsistency type.

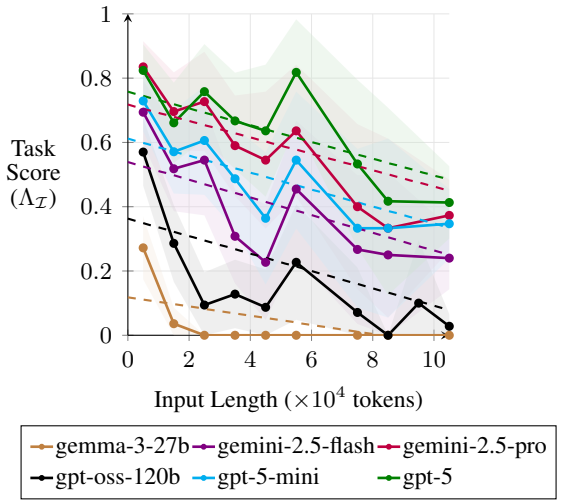


Figure 6: **Task Score ($\Lambda_{\mathcal{I}}$) vs. Input Length**. Dashed lines correspond to linear regression coefficients ($p < 0.05$); See Table H.5 for the coefficients.

for some variation in performance; models perform worse on structural examples compared to other types (Table 6). Both top-performing models' task scores degrade 2.6 percent for every 10k token increase in input length. Across models, the uneven counts of documents per length bin and performance on **PG** explain the jump at 50k tokens (Table H.6). Note that the association between performance and input length is only correlative.

Benchmark Difficulty The two best models find about 60% of the inserted inconsistencies. While there are easier subsets of data where performance generally is high (**BLS** and **PRE**), other subsets prove more difficult. In the main **FIND** dataset, there is a set of 78 out of 375 (or 20%) documents for which all models failed to find the inserted inconsistency. The sources **PG** and **EMM** have the highest rates of documents for which all models are wrong (35% and 47%, respectively).

Feature	All wrong	Some right
Documents	78	297
Tokens (median)	76,198	26,860
ε Pos% (median)	27%	16%
Numeric	37%	59%
Structural	45%	25%
Non-numeric	18%	16%

Table 7: Comparison of items all 7 closed-source models failed vs. some solved in **FIND**.

Table 7 shows that the documents which all models fail on are longer (almost 3×) than the documents which some models get right, and their evidence is located further into the document. The plurality of inserted inconsistencies that are missed by all models is structural. Long documents, distantly positioned evidence, and structural inconsistencies all present challenges for current models.

7 Additional Experiments

7.1 Models Find Otherwise Undiscovered Inconsistencies

We examine predicted inconsistencies unmatched to expected inconsistencies: many are otherwise undetected inconsistencies in the original documents.

Methods These inconsistencies, sometimes nuanced, required expert evaluation. To make this tractable, we limit this analysis only to outputs from the best model (`gpt-5` according to Table 5), and we manually grade the outputs on only 25 documents sampled from each source (200 total), including **MFR** and both subsets of **cs.CL**. Including **cs.CL** expands the domains evaluated, providing a broader look at model capability in practice.

For documents in **FIND** and **MFR**, the annotators of **FIND** were the graders, and for **cs.CL**, two authors of this work served as the graders. The three grading categories are whether the predicted inconsistency is (1) real, (2) not an inconsistency, or (3) the grader is unsure. Our financial experts had relatively low agreement scores over the three grading categories. Cohen's kappa was 0.31 with agreement on 58.8% of items. Because of the low agreement scores and the complexity of the task, the experts subsequently deliberated together to arrive at an agreement for each model prediction. For **cs.CL**, similarly, Cohen's kappa was 0.38 with agreement on 67.8% of items, and the disagreements were resolved through a deliberation phase.

	BLS	PRE	SEC	EMM	PG	AVG	cs.CL MFR	cs.CL Ctrl	cs.CL Idnt
$\hat{\mathcal{I}}/\mathcal{D}$	1.7	2.6	2.4	4.0	2.4	2.6	4.0	4.4	3.9
P (%)	65.1	46.2	42.4	65.7	43.3	52.5	50.5	72.1	70.1
U (%)	69.8	70.8	64.4	74.7	53.3	66.6	56.6	91.9	90.7

Table 8: $\hat{\mathcal{I}}/\mathcal{D}$ denotes the number of predicted answers per document. Precision (P) denotes the percent of answers that were judged as inconsistent. Useful (U) denotes the percent of answers that were judged as either inconsistent or helpful for a user. Predicted answers come from `gpt-5`. For **cs.CL**, scores for the control/identified dataset are denoted "Ctrl"/"Idnt".

Results On **FIND** (Table 8), the precision of `gpt-5`'s predictions averages above 50%. We also report the usefulness of the answers, including cases where the annotator found answer helpful (if not a strict inconsistency). This averaged almost 70%. For the supplemental sources, **MFR** and **cs.CL**, performance was also high: on both subsets of **cs.CL**, precision exceeded 70% and usefulness exceeded 90%, indicating only a small number of false positives. These high precision scores, paired with the number of inconsistencies predicted per response being 1.7 to 4.4, suggest that **models produce useful suggestions at meaningful rates.** Examples are shown in Figures B.1 to B.4.

The relatively high precision (52.5%) that `gpt-5` achieves here suggests that it is not gaming the multiple-response aspect of our evaluation (Section 3.3) to achieve its high overall recall (64%). We intend **FIND** to be used for evaluation only; however, if used for training, models might be incentivized to game recall by overgenerating responses.

7.2 Models Find Known Naturally Occurring Inconsistencies

Here, we focus on known naturally occurring inconsistencies—ones that either external experts or the original document authors discovered. We find that models recover some of them.

MFR has mostly numeric inconsistencies in long tables found by financial experts in public financial documents. While the precision for the suggested inconsistencies was high ($> 50\%$) (see Table 8), the highest recall scores range from 7 to 11 percent, with the open-weight models finding only 0 to 4 percent of the inconsistencies (Table H.2). All three metrics are lower than for data in **FIND** (Table 5).[12]

[12]The standard errors are high (Table H.2) in part because the total number of documents in this source is 45.

For the identified subset of **cs.CL**, those with a known inconsistency, we manually grade the model responses for recall. The known inconsistency was recovered in 12 of 25 documents. The papers' authors fixed these inconsistencies in subsequent drafts, suggesting they were to some degree meaningful. These results suggest a mixed outlook: current models find inconsistencies with relatively high precision, making them useful, but depending on the document type, still miss other inconsistencies, meaning that alone they are not fully reliable.

Lastly, we inspected where predicted inconsistencies occur within **FIND**'s documents. The model exhibits an "early-document bias" as the mean evidence position falls in the first half of the document for approximately 75% of predictions. However, mean evidence position does not predict whether a prediction is correct (point-biserial $r = -0.0004$, $p = 0.99$), with correctness essentially flat across position quartiles (0.49–0.59).

8 Discussion

Models are useful tools for inconsistency detection. This task is a practical, more complex version of a needle-in-a-haystack task, and the models find the inserted inconsistencies at a high enough rate to indicate that they are useful. Moreover, **the models find naturally occurring inconsistencies in published documents**. The observations in Section 7 suggest that using models in this way during document preparation can be helpful. In fact, we did so on a draft of this paper and found five novel inconsistencies, which are documented in Appendix A. (None changed the conclusions of our work.) Despite the promising performance, the recall on this long-context task is about 60% for the best models, which means that **the model misses almost half the inserted inconsistencies within a set of documents**, suggesting more research is needed to make these capabilities fully reliable.

Our results show that current models have difficulty with long contexts and understanding document information at a structural level. For example, Table 6 shows that models tend to perform about equally on numeric and non-numeric inconsistencies but perform much worse on structural inconsistencies. Future work could explore: 1) studies on how to fix inconsistencies, which could include human-in-the-loop and user studies, 2) understanding the significance of found inconsistencies, and 3) including modalities like figures and charts.

9 Limitations

Our task formulation focuses on finding inconsistencies, not on fixing them. Because the inconsistencies in FIND are based on information internal to the document, there is no fact checking or data verification component to the task, which would involve access to ground truth information in external sources such as databases or other documents. For example, for a given inconsistency where two spans disagree, choosing which span (if either) is true may require institutional knowledge (the author's intent, a database value). There are many different ways one could "fix" the inconsistency: perhaps editing a particular span would make the information consistent, even if it results in a false statement. Determining which edit is the right one to make involves knowing what the correct state of the world is, and that knowledge required to disambiguate which edit to make might be available only through trusted knowledge bases, institutional knowledge, or other sources outside the document. Overall, including this aspect of making corrections to a document to resolve the inconsistencies would have significantly complicated the scope of our work, making it more difficult to determine model performance on the specific task of finding inconsistencies.

The inconsistencies found are meaningful but most would not change the documents' conclusions. Our interviews with subject matter experts, largely in the finance domain, highlighted how even simple inconsistencies can have significant monetary penalties. However, within the domain of cs.CL papers, the range of inconsistencies found by gpt-5 can be understood in a different way. First, some of the papers were not intended to be the "camera-ready" version. Moreover, while some inconsistencies cast their results in a better light, there were no examples where the inconsistency changed the conclusions of the work.

Evidence selection is extractive but we use (generative) LLMs. Because the models do not provide span offsets into the document, disambiguation of text mentions is made more difficult, complicating evaluation and downstream applications.

FIND has distributional limitations. The documents are all in English due to the language knowledge of the annotators and researchers. Each example considers a single document, but multi-document use cases would also be interesting (Semnani et al., 2025). The locations of the inserted inconsistencies are biased toward the start of the documents (see Table 3 and Figure 5). Lastly, the inconsistencies don't concern visual information, which presents an exciting avenue for future work.

Span boundaries can be subjective, leading to ambiguity in what the correct span(s) should be. For example, consider: "the scores are always positive and we show that in Table 2." If this were evidence, then it could be captured with one span or two, such as "scores are always positive" and "show that in Table 2". To help mitigate this limitation, we encourage our annotators to prefer shorter spans, and use both strict and lenient measures of success during evaluation.

Potential Risks Our work provides evidence that models are (perhaps surprisingly) effective at finding meaningful inconsistencies within documents. However, there is the risk that using models to help prepare and audit documents could lead to overall less sound documents if users over-rely on the models. As our results demonstrate, models still do not find all errors (at least not in a single pass). Our understanding given the results is that models are a useful tool, but one that still makes mistakes.

Acknowledgments

We thank Blake MacDonald, Craig Schmidt, and Adam Wiemerslage for their helpful discussions and advice. Thank you also to Sudhker Gundlapally, Shruti Hajirnis, and Akshata Joshi for help with understanding inconsistencies in financial documents. Thanks to arXiv for use of its open access interoperability. We appreciate all the dataset sources for their services.

References

Rami Aly and Andreas Vlachos. 2024. TabVer: Tabular fact verification with natural logic. Transactions of the Association for Computational Linguistics, 12:1648–1671.

Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. 2019. Policylint: investigating internal privacy policy contradictions on google play. SEC'19, page 585–602, USA. USENIX Association.

Anthropic. 2025. Claude sonnet 4. https://www-cdn.anthropic.com/6d8a8055020700718b0c49369f60816ba2a7c285.pdf. Accessed: 2026-04-16.

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. 2025. Healthbench: Evaluating large language models towards improved human health. Preprint, arXiv:2505.08775.

Federico Bianchi, Yongchan Kwon, Zachary Izzo, Linjun Zhang, and James Zou. 2025. To err is human: Systematic quantification of errors in published ai papers via llm analysis. Preprint, arXiv:2512.05925.

Arie Cattan, Alon Jacovi, Ori Ram, Jonathan Herzig, Roee Aharoni, Sasha Goldshtein, Eran Ofek, Idan Szpektor, and Avi Caciularu. 2025. Dragged into conflicts: Detecting and addressing conflicting sources in search-augmented llms. Preprint, arXiv:2506.08500.

Hung-Ting Chen, Michael Zhang, and Eunsol Choi. 2022. Rich knowledge sources bring complex knowledge conflicts: Recalibrating models to reflect conflicting evidence. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 2292–2307, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Manan Roy Choudhury, Adithya Chandramouli, Mannan Anand, and Vivek Gupta. 2026. Better call CLAUSE: A discrepancy benchmark for auditing LLMs legal reasoning capabilities. In Findings of the Association for Computational Linguistics: EACL 2026, pages 5776–5818, Rabat, Morocco. Association for Computational Linguistics.

Tobias Deußer, Maren Pielka, Lisa Pucknat, Basil Jacob, Tim Dilmaghani, Mahdis Nourimand, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2023. Contradiction detection in financial reports. In Proceedings of the Northern Lights Deep Learning Workshop, volume 4.

Tobias Deußer, David Leonhard, Lars Hillebrand, Armin Berger, Mohamed Khaled, Sarah Heiden, Tim Dilmaghani, Bernd Kliem, Rüdiger Loitz, Christian Bauckhage, and Rafet Sifa. 2023. Uncovering inconsistencies and contradictions in financial reports using large language models. In 2023 IEEE International Conference on Big Data (BigData), pages 2814–2822.

Nils Dycke and Iryna Gurevych. 2025. Automatic reviewers fail to detect faulty reasoning in research papers: A new counterfactual evaluation framework. Preprint, arXiv:2508.21422.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. 2020. The pile: An 800gb dataset of diverse text for language modeling. Preprint, arXiv:2101.00027.

Team Gemma. 2025. Gemma 3.

Vignesh Gokul, Srikanth Tenneti, and Alwarappan Nakkiran. 2025. Contradiction detection in rag systems: Evaluating llms as context validators for improved information consistency. Preprint, arXiv:2504.00180.

Google DeepMind. 2025a. Gemini 2.5 flash. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf. Accessed: 2026-04-16.

Google DeepMind. 2025b. Gemini 2.5 pro. https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Pro-Model-Card.pdf. Accessed: 2026-04-16.

Yancheng He, Shilong Li, Jiaheng Liu, Weixun Wang, Xingyuan Bu, Ge Zhang, Z.y. Peng, Zhaoxiang Zhang, Zhicheng Zheng, Wenbo Su, and Bo Zheng. 2025. Can large language models detect errors in long chain-of-thought reasoning? In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 18468–18489, Vienna, Austria. Association for Computational Linguistics.

Giwon Hong, Jeonghwan Kim, Junmo Kang, Sung-Hyon Myaeng, and Joyce Jiyoung Whang. 2024. Why so gullible? enhancing the robustness of retrieval-augmented models against counterfactual noise. In Findings of the Association for Computational Linguistics: NAACL 2024, pages 2474–2495, Mexico City, Mexico. Association for Computational Linguistics.

Yufang Hou, Alessandra Pascale, Javier Carnerero-Cano, Tigran Tchrakian, Radu Marinescu, Elizabeth Daly, Inkit Padhi, and Prasanna Sattigeri. 2024. Wikicontradict: A benchmark for evaluating llms on real-world knowledge conflicts from wikipedia. In Advances in Neural Information Processing Systems, volume 37, pages 109701–109747. Curran Associates, Inc.

Cheng Hsu, Cheng-Te Li, Diego Saez-Trumper, and Yi-Zhan Hsu. 2021. Wikicontradiction: Detecting self-contradiction articles on wikipedia. In 2021 IEEE International Conference on Big Data (Big Data), pages 427–436.

Yu-Shiang Huang, Yun-Yu Lee, Tzu-Hsin Chou, Che Lin, and Chuan-Ju Wang. 2025. Finnue: Exposing the risks of using bertscore for numerical semantic evaluation in finance. Preprint, arXiv:2511.09997.

Steve Huntsman, Michael Robinson, and Ludmilla Huntsman. 2024. Prospects for inconsistency detection using large language models and sheaves. Preprint, arXiv:2401.16713.

Cheng Jiayang, Chunkit Chan, Qianqian Zhuang, Lin Qiu, Tianhang Zhang, Tengxiao Liu, Yangqiu Song, Yue Zhang, Pengfei Liu, and Zheng Zhang. 2024. ECON: On the detection and resolution of evidence conflicts. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing,

pages 7816–7844, Miami, Florida, USA. Association for Computational Linguistics.

Zdeněk Kasner, Vilém Zouhar, Patrícia Schmidtová, Ivan Kartáč, Kristýna Onderková, Ondřej Plátek, Dimitra Gkatzia, Saad Mahamood, Ondřej Dušek, and Simone Balloccu. 2025. Large language models as span annotators. Preprint, arXiv:2504.08697.

Amrith Krishna, Sebastian Riedel, and Andreas Vlachos. 2022. Proofver: Natural logic theorem proving for fact verification. Transactions of the Association for Computational Linguistics, 10:1013–1030.

Murathan Kurfali and Robert Östling. 2025. Conflicting needles in a haystack: How LLMs behave when faced with contradictory information. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34349–34364, Suzhou, China. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles.

Jierui Li, Vipul Raheja, and Dhruv Kumar. 2024. ContraDoc: Understanding self-contradictions in documents with large language models. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6509–6523, Mexico City, Mexico. Association for Computational Linguistics.

Jiacheng Liu, Sewon Min, Luke Zettlemoyer, Yejin Choi, and Hannaneh Hajishirzi. 2024. Infini-gram: Scaling unbounded n-gram language models to a trillion tokens. In First Conference on Language Modeling.

Siyi Liu, Qiang Ning, Kishaloy Halder, Zheng Qi, Wei Xiao, Phu Mon Htut, Yi Zhang, Neha Anna John, Bonan Min, Yassine Benajiba, and Dan Roth. 2025. Open domain question answering with conflicting contexts. In Findings of the Association for Computational Linguistics: NAACL 2025, pages 1838–1854, Albuquerque, New Mexico. Association for Computational Linguistics.

Siyi Liu and Dan Roth. 2025. Conflicts in texts: Data, implications and challenges. In Findings of the Association for Computational Linguistics: EMNLP 2025, pages 10073–10091, Suzhou, China. Association for Computational Linguistics.

Ananya Mantravadi, Shivali Dalmia, Olga Pospelova, Abhishek Mukherji, Nand Dave, and Anudha Mittal. 2025. Legalwiz: A multi-agent generation framework for contradiction detection in legal documents. Preprint, arXiv:2510.03418.

Marc Marone and Benjamin Van Durme. 2023. Data portraits: Recording foundation model training data. In Advances in Neural Information Processing Systems, volume 36, pages 15121–15135. Curran Associates, Inc.

Satoshi Masuda, Tohru Matsuodani, and Kazuhiko Tsuda. 2016. Detecting logical inconsistencies by clustering technique in natural language requirements. IEICE TRANSACTIONS on Information and Systems, 99(9):2210–2218.

Evan Miller. 2024. Adding error bars to evals: A statistical approach to language model evaluations. Preprint, arXiv:2411.00640.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Allyson Ettinger, Michal Guerquin, David Heineman, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Jake Poznanski, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2025. 2 olmo 2 furious. Preprint, arXiv:2501.00656.

OpenAI. 2025a. gpt-oss-120b & gpt-oss-20b Model Card. Preprint, arXiv:2508.10925.

OpenAI. 2025b. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/. Accessed: 2026-04-16.

OpenAI. 2025c. Openai o3. https://openai.com/index/introducing-o3-and-o4-mini/. Accessed: 2026-04-16.

Quang Hieu Pham, Hoang Ngo, Anh Tuan Luu, and Dat Quoc Nguyen. 2024. Who's who: Large language models meet knowledge conflicts in practice. In Findings of the Association for Computational Linguistics: EMNLP 2024, pages 10142–10151, Miami, Florida, USA. Association for Computational Linguistics.

Nursulu Sagimbayeva, Ruveyda Betül Bahçeci, and Ingmar Weber. 2025. Misleading through inconsistency: A benchmark for political inconsistencies detection. In Workshop Proceedings of the 19th International AAAI Conference on Web and Social Media – MisD 2025: 1st Workshop on Misinformation Detection in the Era of LLMs.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 7881–7892, Online. Association for Computational Linguistics.

Sina Semnani, Jirayu Burapacheep, Arpandeep Khatua, Thanawan Atchariyachanvanit, Zheng Wang, and Monica Lam. 2025. Detecting corpus-level knowledge inconsistencies in Wikipedia with large language models. In Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 34827–34854, Suzhou, China. Association for Computational Linguistics.

Luca Soldaini, Rodney Kinney, Akshita Bhagia, Dustin Schwenk, David Atkinson, Russell Authur, Ben Bogin, Khyathi Chandu, Jennifer Dumas, Yanai Elazar, Valentin Hofmann, Ananya Jha, Sachin Kumar, Li Lucy, Xinxi Lyu, Nathan Lambert, Ian Magnusson, Jacob Morrison, Niklas Muennighoff, Aakanksha Naik, Crystal Nam, Matthew Peters, Abhilasha Ravichander, Kyle Richardson, Zejiang Shen, Emma Strubell, Nishant Subramani, Oyvind Tafjord, Evan Walsh, Luke Zettlemoyer, Noah Smith, Hannaneh Hajishirzi, Iz Beltagy, Dirk Groeneveld, Jesse Dodge, and Kyle Lo. 2024. Dolma: an open corpus of three trillion tokens for language model pretraining research. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15725–15788, Bangkok, Thailand. Association for Computational Linguistics.

Guijin Son, Jiwoo Hong, Honglu Fan, Heejeong Nam, Hyunwoo Ko, Seungwon Lim, Jinyeop Song, Jinha Choi, Gonçalo Paulo, Youngjae Yu, and Stella Biderman. 2025. When ai co-scientists fail: Spot-a benchmark for automated verification of scientific research. Preprint, arXiv:2505.11855.

Mooho Song, Hye Ryung Son, and Jay-Yoon Lee. 2025. Introducing verification task of set consistency with set-consistency energy networks. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 33346–33366, Vienna, Austria. Association for Computational Linguistics.

Gladys Tyen, Hassan Mansoor, Victor Carbune, Peter Chen, and Tony Mak. 2024. LLMs cannot find reasoning errors, but can correct them given the error location. In Findings of the Association for Computational Linguistics: ACL 2024, pages 13894–13908, Bangkok, Thailand. Association for Computational Linguistics.

Yan Wang, Keyi Wang, Shanshan Yang, Jaisal Patel, Jeff Zhao, Fengran Mo, Xueqing Peng, Lingfei Qian, Jimin Huang, Guojun Xiong, Xiao-Yang Liu, and Jian-Yun Nie. 2025. Finauditing: A financial taxonomy-structured multi-document benchmark for evaluating llms. Preprint, arXiv:2510.08886.

Maurice Weber, Daniel Y. Fu, Quentin Anthony, Yonatan Oren, Shane Adams, Anton Alexandrov, Xiaozhong Lyu, Huu Nguyen, Xiaozhe Yao, Virginia Adams, Ben Athiwaratkun, Rahul Chalamala, Kezhen Chen, Max Ryabinin, Tri Dao, Percy Liang, Christopher Ré, Irina Rish, and Ce Zhang. 2024. Redpajama: an open dataset for training large language models. In Advances in Neural Information Processing Systems, volume 37, pages 116462–116492. Curran Associates, Inc.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. 2024. Measuring short-form factuality in large language models. Preprint, arXiv:2411.04368.

Sarina Xi, Vishisht Rao, Justin Payan, and Nihar B. Shah. 2025. Flaws: A benchmark for error identification and localization in scientific papers. Preprint, arXiv:2511.21843.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. 2024. Knowledge conflicts for LLMs: A survey. In Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, pages 8541–8565, Miami, Florida, USA. Association for Computational Linguistics.

YesNoError. 2024. Yesnoerror (yne) whitepaper v1.0.

Wlodek Zadrozny and Luciana Garbayo. 2018. A sheaf model of contradictions and disagreements. preliminary report and discussion. Preprint, arXiv:1801.09036.

Linda Zeng, Rithwik Gupta, Divij Motwani, Yi Zhang, and Diji Yang. 2025. Worse than zero-shot? a fact-checking dataset for evaluating the robustness of RAG against misleading retrievals. In The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track.

Tianmai M. Zhang and Neil F. Abernethy. 2025. Reviewing scientific papers for critical problems with reasoning LLMs: Baseline approaches and automatic evaluation. In NeurIPS 2025 AI for Science Workshop.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In International Conference on Learning Representations.

Xinrong Zhang, Yingfa Chen, Shengding Hu, Zihang Xu, Junhao Chen, Moo Hao, Xu Han, Zhen Thai, Shuo Wang, Zhiyuan Liu, and Maosong Sun. 2024. ∞Bench: Extending long context evaluation beyond 100K tokens. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15262–15277, Bangkok, Thailand. Association for Computational Linguistics.

Housekeeping

1. The potential inconsistencies identified in public documents were graded to the best of our abilities (Section 7.1, Section 7.2). However, they were *not* confirmed by the original documents' authors. Our findings are for research purposes without warranty. It is possible that some external or additional context could explain some of these issues.

2. This appendix has a large number of figures, so we provide a table of contents below to help navigate the sections.

3. Claude Code (`sonnet/v4.5`) was used to help generate the code for the tables and figures in this paper (especially with `pgfplots`). We also used `gpt-5` and `gemini-2.5-pro` to find inconsistencies within this draft: Appendix A.

4. The data annotation required for this project was done by paid members of our organization (also authors on this paper). We chose to collaborate rather than out-source this process in part because the task difficulty.

A Examples of Undiscovered Inconsistencies from a Draft of This Paper

We ran `gpt-5` and `gemini-2.5-pro` on a draft of this work. We put one intentional inconsistency within the work, which `gemini-2.5-pro` found. Together, the two models found five meaningful and previously undiscovered issues and recovered the intentional error. Overall, this anecdotal case study suggests that models are helpful for real-world use.

`gpt-5` found three real issues (Figures A.1 to A.3). One of those issues would have been found anyway because it was a placeholder for this analysis. The model did not find the inconsistency we intentionally inserted. `gemini-2.5-pro` made one incorrect suggestion, two helpful suggestions (Figures A.4 and A.5), and found four real issues (Figures A.6 to A.9). One of the real issues was the inconsistency we intentionally inserted. Another of the raised issues overlapped with one found by `gpt-5`. After fixing these inconsistencies in our draft and running the models again, `gpt-5` found one more issue (Figure A.10), and `gemini-2.5-pro` found an interesting apparent mismatch between the abstract and the analysis section (Figure A.11), which arises because we do not fully describe the experiment in the abstract.

Document (Input) and Evidence Spans (Annotation)

```
containing 375 test and 125 development problems.
In the test set 83 out of 420 documents (at most, depending on the model and tokenizer) had their text truncated.
```

Description (Annotation)

The document defines the test set as 375 items, but later refers to the “test set” as having 420 documents when discussing truncation, creating a contradiction about test set size.

Figure A.1: **Real** inconsistency found by `gpt-5` in an earlier draft. The 420 value includes the **MFR** data.

Document (Input) and Evidence Spans (Annotation)

```
we did so on a late version of our draft and found \\textcolor{orange}{three} inconsistencies, which are documented in
\\Cref{app:sec:paper:inconsistencies}.
\\section{Inconsistencies Found in Earlier Versions of This Work}\\label{app:sec:paper:inconsistencies}
```

Description (Annotation)

The text claims three inconsistencies are documented in the cited appendix section, but the referenced section appears without the promised documentation, indicating missing expected content.

Figure A.2: **Real** inconsistency found by `gpt-5` in an earlier draft. `gemini-2.5-pro` found this same issue. We note that this would have been fixed anyway, as it was a placeholder value for this analysis. *We do not count this toward the number of novel inconsistencies found.*

Document (Input) and Evidence Spans (Annotation)

```
gpt-5-mini
\\model{gpt-5-nano}
```

Description (Annotation)

The Methods list “gpt-5-nano” as a tested model, whereas results tables and later sections report “gpt-5-mini” creating an inconsistency in the model variant named as evaluated.

Figure A.3: **Real** inconsistency found by `gpt-5` in an earlier draft.

Document (Input) and Evidence Spans (Annotation)

```
the best-performing model (\\model{gpt-5}) recovered 64\\\% of the inserted inconsistencies.
even the best models miss almost half of the inconsistencies in \\datasetname{}, demonstrating that inconsistency detection
is still a challenging task.
```

Description (Annotation)

The abstract states that the best model, gpt-5, recovered 64% of inconsistencies, but concludes by stating that even the best models miss “almost half” (50%) of them. A 64% recovery rate means a 36% miss rate, which is substantially different from “almost half”. This mischaracterization is repeated in the Discussion section.

Figure A.4: **Helpful** comment made by `gemini-2.5-pro` on an earlier draft. We would argue that “almost half” is a fair characterization.

Document (Input) and Evidence Spans (Annotation)

```
\\PG{} & 75 & 109$_{\\pm106.6}$ & 4$_{\\pm2.6}$ & 8$_{\\pm6.0}$ & 22$_{\\pm24.1}$
\\PGS{} & 25 & 146$_{\\pm111.8}$ & 5$_{\\pm3.7}$ & 7$_{\\pm4.8}$ & 22$_{\\pm17.9}$
```

Description (Annotation)

The paper uses the abbreviation 'PG' for "Project Gutenberg" documents in the test set tables (e.g., Table 2). However, in the development set tables (e.g., Table A.8), it uses 'PGS' for what appears to be the same source. The dataset statistics for 'PG' (avg. 109k tokens) and 'PGS' (avg. 146k tokens) are different, but the 'PGS' abbreviation is never defined, creating ambiguity.

Figure A.5: **Helpful** comment made by `gemini-2.5-pro` on an earlier draft. We did not define PGS in the development section (an oversight). (PGS stands for "Project Gutenberg Seen", as the **PG** development data may be leaked to models.)

Document (Input) and Evidence Spans (Annotation)

```
the recall on this long-context task is about 55\\\% for the best models, which means that \\textbf\{the model misses almost
half the expected inconsistencies within a set of documents,\}
gpt-5 & \\textbf{87} & \\textbf{67} & \\textbf{73} & \\textbf{43} & \\textbf{52} & \\textbf{64}
gemini-2.5-pro & \\textbf{88} & \\textbf{71} & 61 & \\textbf{40} & \\textbf{45} & \\textbf{61}
```

Description (Annotation)

The Discussion section claims that recall for the best models is "about 55%". However, Table 4 shows the two best models, gpt-5 and gemini-2.5-pro, achieve average recall scores (Task Score) of 64% and 61% respectively, which are notably higher than the cited 55%.

Figure A.6: **Real** inconsistency found by `gemini-2.5-pro`. We intentionally placed this inconsistency into a late draft of the paper. *We do not count this toward the number of novel inconsistencies found.*

Document (Input) and Evidence Spans (Annotation)

```
While the precision for the suggested errors was high ($>50$\%) (see \\Cref{tbl:analysis:precision}), the highest recall
scores range from 7 to 11 percent, [...]
```

Model	$\Lambda_{\mathcal{E}}$	Λ_{δ}	Λ_I
[...]			
sonnet-v4	**12 (3.1)**	33 (1.8)	**2 (2.2)**
gpt-5-mini	10 (1.3)	35 (1.0)	**2 (2.2)**
gpt-5	**12 (2.2)**	**36 (1.5)**	**7 (3.7)**
o3-mini	0 (0.1)	24 (2.3)	**9 (2.2)**
o3	9 (2.1)	33 (1.5)	**9 (4.2)**
gemini-2.5-flash	**12 (2.6)**	**34 (2.5)**	**9 (4.2)**
gemini-2.5-pro	**16 (2.4)**	**38 (1.0)**	**9 (4.2)**

Description (Annotation)

Section 7.2 states that for the MFR dataset, "the highest recall scores range from 7 to 11 percent". However, the data in Table A.7 shows that the highest recall score achieved by any model is 9%, making the stated range incorrect.

Figure A.7: **Real** inconsistency found by `gemini-2.5-pro` in an earlier draft. The issue arose due to an out-of-date table in the appendix.

Document (Input) and Evidence Spans (Annotation)

```
we did so on a late version of our draft and found \\textcolor{orange}{three} inconsistencies, which are documented in
\\Cref{app:sec:paper:inconsistencies}.
\\section{Inconsistencies Found in Earlier Versions of This Work}\\label{app:sec:paper:inconsistencies}
```

Description (Annotation)

The Discussion section states that three inconsistencies found in the paper's own draft are documented in Appendix I. However, Appendix I, while present, is empty and does not contain the promised documentation.

Figure A.8: **Real** inconsistency found by `gemini-2.5-pro` in an earlier draft. `gpt-5` found this same issue. We note that this would have been fixed anyway, as it was a placeholder value for this analysis. *We do not count this toward the number of novel inconsistencies found.*

Document (Input) and Evidence Spans (Annotation)

```
gemini-2.5-flash & 6h & 2.1m
The times above are reported for the 420 items
```

Description (Annotation)

In Table A.1, the total compute time for ‘gemini-2.5-flash‘ is listed as “6h” and the per-document time is “2.1m”. These two figures are contradictory for the stated 420 documents. 6 hours is 360 minutes, which averages to 0.86 minutes per document, not 2.1. Conversely, 2.1 minutes per document totals 882 minutes (14.7 hours), not 6 hours.

Figure A.9: **Real** inconsistency found by `gemini-2.5-pro` in an earlier draft.

Document (Input) and Evidence Spans (Annotation)

```
\\footnote{\\url{https://emma.msrb.org/}}, \\url{EMMA.msrp.org}
```

Description (Annotation)

The **EMM** source is given as ‘https://emma.msrb.org/‘ in the text, but later the figure caption uses ‘EMMA.msrp.org‘, changing ‘msrb‘ to ‘msrp‘. This mismatch in the domain name is an inconsistency in the referenced source.

Figure A.10: **Real** inconsistency found by `gpt-5` after the first round of fixes to model-identified inconsistencies.

Document (Input) and Evidence Spans (Annotation)

```
on 50 arXiv papers, we judged 136 out of 196 of the model's suggestions to be legitimate inconsistencies missed by the original
authors.
we manually grade the outputs on only 25 documents sampled from each source (200 total), including \\MFR{} and both subsets
of \\ARXIV{}.
1.7 & 2.6 & 2.4 & 4.0 & 2.4 & 2.6 & 4.0 & 4.4 & 3.9
\\BLS{} & \\PRE{} & \\SEC{} & \\EMM{} & \\PG{} & \\AVG{} & \\textcolor{ANALYSIS}{\\MFR{}} &
\\textcolor{ANALYSIS}{\\footnotesize{Ctrl}} & \\textcolor{ANALYSIS}{\\footnotesize{Idnt}}
```

Description (Annotation)

The abstract claims that on 50 arXiv papers, the model made 196 suggestions of which 136 were legitimate. However, the data in Table 5 implies different numbers. With 25 papers in each of the two arXiv subsets (‘Ctrl‘ and ‘Idnt‘), the total number of suggestions would be (25 * 4.4) + (25 * 3.9) = 207.5 (approx. 208), not 196. The number of legitimate suggestions would be (110 * 0.721) + (97.5 * 0.701) = 147.66 (approx. 148), not 136.

Figure A.11: **Noteworthy** comment made by `gemini-2.5-pro`. This apparent mismatch is intentional, as we exclude the 12 known inconsistencies that were found (Section 7.2) from the 208 total predictions when describing the analysis in the abstract.

B Examples of Undiscovered Inconsistencies in `cs.CL` and `PG`

See Figures B.1 to B.4. To improve readability, we made minor formatting changes to the text and removed long sections of text, replaced by "[...]".

Document (Input) and Evidence Spans (Annotation)

Model	14LAP			14RES			15RES		
	P	R	F1	P	R	F1	P	R	F1
CMLA+~\cite {wang2017multi}	30.10	36.90	33.20	39.20	39.80	37.00	41.30	42.10	41.70
RINANTE+ ~\cite {peng2020knowing}	21.70	18.70	20.10	29.90	30.10	30.00	25.70	22.30	23.90
WhatHowWhy~\cite {peng2020knowing}	37.38	50.38	42.87	43.24	48.07	63.66	51.46	57.51	52.32

Description (Annotation)

In the 14RES column, the F1 score (63.66) exceeds both precision (43.24) and recall (48.07), which is impossible since F1 must lie between P and R.

Figure B.1: **Real** inconsistency found by gpt-5 from **cs.CL** with type *Numeric*.

Document (Input) and Evidence Spans (Annotation)

```
[...]
\begin {figure}[t]
\centering  \includegraphics [width=\linewidth ]{figs/fig4.pdf}
\caption {Speedup comparison.}
[...]
Our hypothesis is that larger training datasets will lead to improved model performance, as indicated by reductions in
cross-entropy loss. Figure~\ref {fig:fig4} depicts these scaling curves for the different model sizes.
```

Description (Annotation)

Figure 4 is introduced earlier as a speedup comparison, but later it is referenced as showing scaling curves. This misreference creates an inconsistency between the figures stated content (speedup) and the later claim (scaling), making the pointer to Figure 4 for scaling incorrect.

Figure B.2: **Real** inconsistency found by gpt-5 from **cs.CL** with type *Structural*.

Document (Input) and Evidence Spans (Annotation)

```
These include [...] HellaSwag\~\cite {zellers2019hellaswag}, [...] and WinoGrande~\cite {sakaguchi2019winogrande}.
We demonstrate the results of this evaluation suite in~\Cref {tab:downstream}.
[...]

\caption {Zero-shot benchmark accuracy on downstream datasets.} \label {tab:downstream}
  Size    Model    Train. toks.    ARC-e    ARC-c    BoolQ    PIQA    OBQA    WinoG.    Avg.
```

Description (Annotation)

The text claims HellaSwag is part of the reported evaluation suite and that the results of this suite are shown in the referenced table. However, the table that follows does not report HellaSwag results, so an expected metric is missing from the presented results.

Figure B.3: **Real** inconsistency found by gpt-5 from **cs.CL** with type *Structural*.

Document (Input) and Evidence Spans (Annotation)

```
  Flour, per bag of 24.5 pounds    1.721    1.790
  Corn meal, per pound             .055     .067
  Potatoes, per peck               .579     .645
  Sugar, per pound                 .122     .110
The factors as worked out in Table 1 are ratios between yearly average retail prices and yearly
average Dun's index numbers.  Even retail prices, however, have some seasonal swing.  [...]
In the main, however, Table 2 is fairly accurate as it stands. It will be noted that with the exception of hog products, wheat
and potatoes, retail prices in September of 1919 tended to be lower than their normal ratio to Dun's index number.
```

Description (Annotation)

The text claims only hog products, wheat, and potatoes were above their index prices, but corn meal's actual price (0.067) exceeded its index price (0.055). This contradicts the stated set of exceptions.

Figure B.4: **Real** inconsistency found by gpt-5 from **PG** with type *Numeric*.

C Open-Source and Links

The dataset and terms of usage are available at: https://huggingface.co/datasets/kensho/FIND. Note that documents in the dataset have been modified to include inconsistencies. The guides used for annotation and dataset creation are available at: https://drive.google.com/drive/folders/18qOjCICcMPL_NSzJjirpTxx6G4ecwlfB?usp=drive_link.

D Reproducing this Work and Implementation Details

D.1 Language Model Software and Hardware

For the open-weight models we use huggingface transformer[13] weights and VLLM (Kwon et al., 2023) for inference. For hardware, we use a `P4de-24, 8 x nvidia A100` AWS instance. For the closed-source models, we use the `LiteLLM`[14] interface to interact with their endpoints. For the closed-source models, all items were run synchronously in an online manner.

Runtimes are presented in Table D.1. The times are reported for the 420 items in the **FIND** test set and **MFR**. Open-weight models were run on a `P4de-24, 8 x nvidia A100` AWS instance using VLLM. The closed-source models were run by the respective model providers' hardware via API. Because these values are based on single runs, we expect there to be some variance over compute times, especially for the closed-source models.

For each model's outputs on **FIND**, the grader (`gpt-4.1`) completed in 5 minutes to 30 minutes. Larger models tended to output more inconsistencies and thus took longer to grade because each answer was graded independently.

D.2 Handling Span Generation

Kasner et al. (2025) discuss options for span generation. Our prompts (and grading code) instruct the models to present the answer in a tagged format:

```
<answer>
    <evidence>ε_1</evidence>
    ...
    <evidence>ε_n</evidence>
    <description>δ</description>
</answer>
```

We follow the recommendation of Kasner et al. (2025) to "[list] textual content of the spans", and expect the generated evidence spans are to appear verbatim within the original document.[15] In pilot testing, we found that models were generally successful at producing answers in this format, especially compared to other options (such as JSON and Markdown). For responses indicating no inconsistency was present, we specify that the response should be:

```
<answer></answer>
```

Model	Time Total	Time Per Doc
gemma-3-12b-it	20m	3s
gemma-3-27b-it	30m	5s
gpt-oss-20b	1h 5m	10s
gpt-oss-120b	1h 5m	10s
sonnet-v4	10h	1.5m
gpt-5-mini	10h	1.5m
gpt-5	10h	1.5m
o3-mini	3h	0.4m
o3	9h	1.5m
gemini-2.5-flash	6h	0.9m
gemini-2.5-pro	15h	2.1m

Table D.1: Compute time per document across **FIND** (and **MFR**). All times are rounded up for legibility.

E Evidence Metric

We support two versions of the lexical evidence metric: one strict and one lenient, each making different decisions about certain conditions on the predictions. Only the strict version is reported. For a concrete illustration of how the computation of the strict and lenient metrics differ see Figures E.2 and E.3, and for pseudocode see Algorithm E.1.

The first decision point is whether the metric should be aware of span boundaries. The strict metric considers each piece of evidence ε as a separate span, while the lenient metric considers the totality of the evidence ε as a sequence of words.[16] The lenient approach allows for disagreements on seg-

[13] https://huggingface.co/docs/transformers/index

[14] https://github.com/BerriAI/litellm

[15] Note that the textual contents of many evidence spans are likely to occur multiple times in the documents in **FIND**, especially for evidence such as row/column labels or table cell values, so there is ambiguity in which particular mention of the evidence text is the intended span in the document. Additionally, because the text may appear multiple times, text matching heuristics to identify the particular mention will not be able to correctly disambiguate all mentions. However, the approaches of XML-tagging the entire document or having the model generate span offsets have more severe drawbacks (Kasner et al., 2025), and so we do not use them.

[16] For example, the strict metric views $\mathcal{E}$ = {"this is evidence", "this is more evidence"} as two spans, and the lenient metric views it as $\mathcal{E}$ = {"this", "is", "evidence", "this", "is", "more", "evidence"}.

mentation that do not affect the overall information conveyed, such as whether a piece of text is one long span or two shorter spans.

The second decision point is whether the amount of string overlap (and ultimately string similarity, see below) between predicted and reference evidence text should be based on their longest common substring (LCStr) or longest common subsequence (LCSeq).[17] LCStr (strict) more harshly penalizes predictions that differ from the reference even by one character, whereas LCSeq (lenient) allows for small differences (due to miscopying or using a slightly different piece of evidence than the reference does).

The third decision point is whether the predicted evidence should appear verbatim in the document. The strict approach requires that the predicted evidence occurs in the document and marks any predicted evidence that does not appear in the document as ineligible for matching to reference evidence. The lenient approach allows for mistakes in copying the evidence text.

We evaluate the predicted evidence against the reference evidence using a bipartite matching approach (discussed next), which aligns the predicted and reference evidence text. From the matching, we calculate the true positive count as the total amount of string overlap in the matching pairs of text, total predicted count as the amount of evidence text predicted by the model, and the total reference count as the amount of evidence text given in the reference answer.[18]

To obtain the matching between predicted evidence and reference evidence, we first compute similarity scores between each predicted element and reference element.[19] The score is based on the string overlap between the two elements. Specifically, the similarity score is the F1 based on string overlap between the predicted element x and reference element y, where w is the sequence of overlapping characters (computed by LCStr or LCSeq), precision is $|w|/|x|$, recall is $|w|/|y|$, and F1 is the harmonic mean of precision and recall. The matching is then obtained from the similarity scores with a linear sum assignment algorithm to globally maximize the total score among matched predicted elements and reference elements.

The total amount of character overlap among the matched elements is the true positive count achieved by the model on the example, which is then normalized by the character length of all the predicted evidence text to get precision or by the character length of all the reference evidence text to get recall. Intuitively, the precision is the proportion of predicted evidence text that appears in the reference evidence, and the recall is the proportion of reference evidence text that is covered by the predicted evidence (both conditioned on the alignment between predicted and reference evidence).[20]

To determine the ability of the evidence metrics to distinguish correct from incorrect responses, we compared their scores against manual judgments of evidence quality. We manually graded 202 responses with binary labels for whether the predicted evidence was similar enough to the reference evidence to correctly identify the inconsistency. The 202 responses were predictions on 101 internal development examples from two models, `Qwen3-30B-A3B-Instruct-2507` and `openai gpt-oss-120b`. Of the 202 responses, 61 were auto-gradable, and the remaining 141 were scored by the evidence metric.

One could classify correct and incorrect evidence predictions by imposing a threshold on the scalar metric scores. Figure E.1 shows ROC curves that consider all such thresholds as well as the areas under the curve (AUC) for the strict and lenient metrics, based on the 141 non-autograded examples. Both metrics achieve high AUCs of 0.884 and 0.909, respectively, indicating that they are effective at distinguishing good predicted evidence from bad predicted evidence.

F Description Metric

To determine the ability of the description metric to distinguish correct from incorrect responses, we use the same procedure as in Appendix E. We manually graded with binary labels the same 202 responses for whether the predicted description was similar enough to the reference description to

[17] The longest common substring of x and y is the longest *contiguous* string that appears both in x and in y. The longest common subsequence removes the contiguity constraint, allowing other text to be interspersed within the common subsequence.

[18] All quantities are computed at the character level.

[19] For the strict metric, each element is a span. For the lenient metric, each element is a word, since we have removed the span boundaries from the computation.

[20] Certain cases can be scored automatically instead, without considering the content of the predicted evidence, such as when the model response is not well-formed, when the predicted and reference evidence are both empty (such as for consistent documents), and when either the reference or the predicted answer is empty and the other is not.

Algorithm E.1 Evidence Metric Computation

function OVERLAPSCOREFN(x, y, overlap_fn)
  if overlap_fn $=$ LCStr **then**
    overlap_block $\leftarrow$ longest contiguous matching block between x and y
  else
    overlap_blocks $\leftarrow$ all matching blocks (allowing gaps) between x and y
  end if
  overlap_amount $\leftarrow$ total characters in matching block(s)
  precision $\leftarrow$ overlap_amount / $|x|$
  recall $\leftarrow$ overlap_amount / $|y|$
  F1 $\leftarrow 2 \times$ precision $\times$ recall / (precision + recall)
  return F1, overlap_amount
end function

Input: predicted_spans, reference_spans, mode $\in$ {strict, lenient}, document_text
▷ Configure based on mode
if mode $=$ strict **then**
  predictions $\leftarrow$ predicted_spans
  references $\leftarrow$ reference_spans
  overlap_fn $\leftarrow$ LCStr
  require_verbatim $\leftarrow$ true
else
  predictions $\leftarrow$ flatten(predicted_spans)
  references $\leftarrow$ flatten(reference_spans)
  overlap_fn $\leftarrow$ LCSeq
  require_verbatim $\leftarrow$ false
end if

▷ Build similarity matrix S and character overlap matrix C
for all $(\text{pred}_i, \text{ref}_j) \in$ predictions $\times$ references **do**
  if require_verbatim and $\text{pred}_i \notin$ document_text **then**
    $S[i, j] \leftarrow -\infty$
    $C[i, j] \leftarrow 0$
  else
    similarity_score, overlap_amount $\leftarrow$ OVERLAPSCOREFN($\text{pred}_i, \text{ref}_j$, overlap_fn)
    $S[i, j] \leftarrow$ similarity_score
    $C[i, j] \leftarrow$ overlap_amount
  end if
end for

▷ Get matched pairs of evidence
matches $\leftarrow$ linear_sum_assignment(S, maximize $=$ true)
valid_matches $\leftarrow$ filter(matches, lambda i, j: $S[i, j] \neq -\infty$) ▷ Remove spurious matches

▷ Compute metrics
total_char_overlap $\leftarrow \sum_{(i,j) \in \text{valid\_matches}} C[i, j]$
total_char_predicted $\leftarrow \sum_{p \in \text{predictions}} |p|$ ▷ Includes non-verbatim and unmatched predictions
total_char_reference $\leftarrow \sum_{r \in \text{references}} |r|$ ▷ Includes unmatched references
precision $\leftarrow$ total_char_overlap / total_char_predicted
recall $\leftarrow$ total_char_overlap / total_char_reference
F1 $\leftarrow 2 \times$ precision $\times$ recall / (precision + recall)
Output: precision, recall, F1

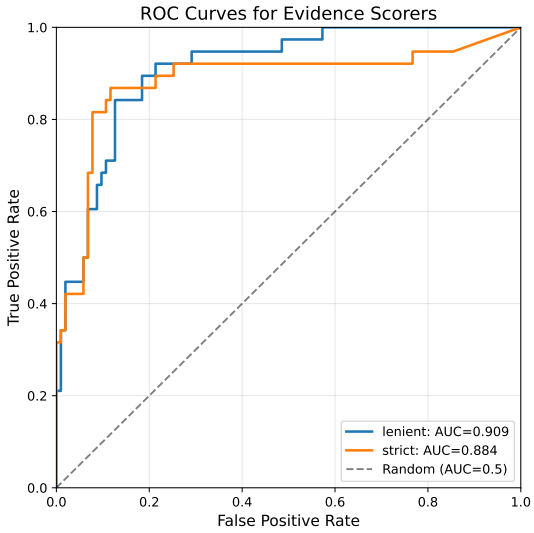


Figure E.1: ROC curves for evidence scorers based on 141 non-autograded examples from internal development data. Both the strict and lenient metrics can distinguish correct from incorrect evidence sets.

correctly characterize the inconsistency. 61 of the responses were auto-gradable and 141 were scored by the description metric.

We considered a range of potential description metrics, some lexical and some neural, each of which provides a scalar score for the predicted description. Figure F.1 shows ROC curves for the description metrics we considered, based on the 141 non-autograded examples. Despite neural metrics like BERTScore (Zhang et al., 2020) exhibiting low sensitivity to numerical differences (Huang et al., 2025), the neural metric BLEURT obtains the highest AUC (0.888). ROUGE-2 obtains the second best AUC (0.862), and BLEU obtains the lowest (0.777). Based on these findings, we use BLEURT as the description metric.

G Dataset

Documents from each data source in **FIND** were converted into Markdown documents using https://extract.kensho.com/app and https://github.com/kensho-technologies/kenverters, except for a subset of **BLS** documents which were already in plain text. Most of the conversions were successful and high fidelity. Some **BLS** documents required manual editing. For other document sources, when the conversion process failed we replaced the document with an alternative sample. Auxiliary files accompanying the **cs.CL** documents were combined into a single plain LaTeX document with no additional markup. The auxiliary **MFR** documents come from a diversely formatted set of PDF documents. Some of these documents were many hundreds of pages long. Because the formatting was non-uniform (and non-standard) we manually edited the converted outputs, and because of the length, in some cases, we trimmed the documents to the relevant sections that contained the inconsistency identified by the expert. Note that these sections are still long, averaging 48k tokens.

Figure G.1 covers the flow of annotation and dataset creation.

Figures G.2 to G.7 show examples of documents from the sources in **FIND** and **MFR**.

Additional information about data from the sources is:

BLS 91 documents were released between January and April 2025. 9 documents were released between September and December 2025. Charts, which are out of scope for this work, were removed from the document.

PRE These documents are not generally public and were published 2020 or before. Because we required explicit permission to access these documents, which we received through direct communication, we expect most models are not trained on them. We removed the names of the primary stakeholders of these documents as a condition of use.

SEC We selected documents filed in January and February 2025.

EMM These reports show the financial health of local governments and provide ongoing updates to investors who bought their bonds. We selected documents released May 2025 and later.

PG We selected documents that were uploaded between September 2024 and June 2025 and which we predicted to not appear in pre-training corpora (see below). The items in the development set of this source in particular are not date-protected, with many uploaded in 2017 or before. Because we modified the book contents (via the inserted inconsistencies), we also removed the Project Gutenberg license and all references to Project Gutenberg from the documents per the requirement of their license.

To predict whether a candidate **PG** document has been seen during model pre-training, we

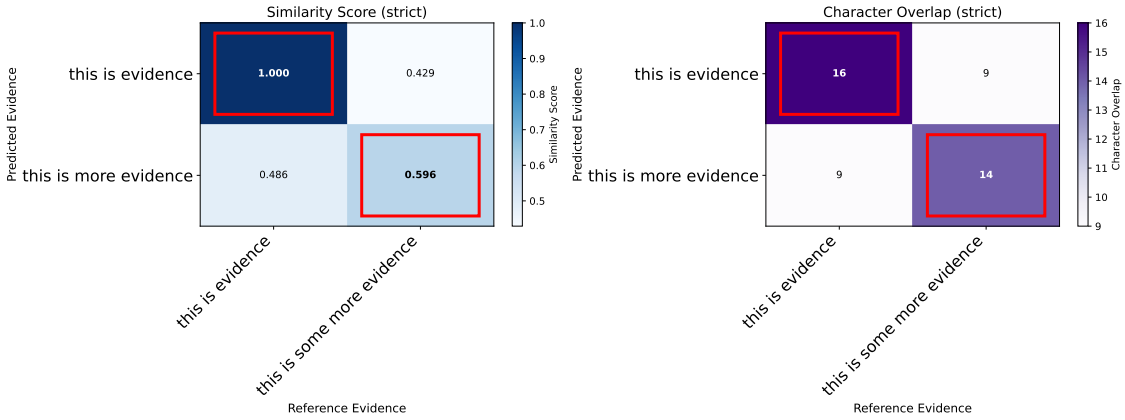


Figure E.2: Similarity scores and character overlaps computed for the strict evidence metric with predicted evidence $\hat{\mathcal{E}}$ = {"this is evidence", "this is more evidence"} and reference evidence $\mathcal{E}$ = {"this is evidence", "this is some more evidence"}. Red outlines indicate the matched evidence determined via the similarity scores and superimposed on the character overlap matrix. The score for this example is then computed from the total amount of character overlap in the matched evidence, the total length of the predicted evidence, and the total length of the reference evidence.

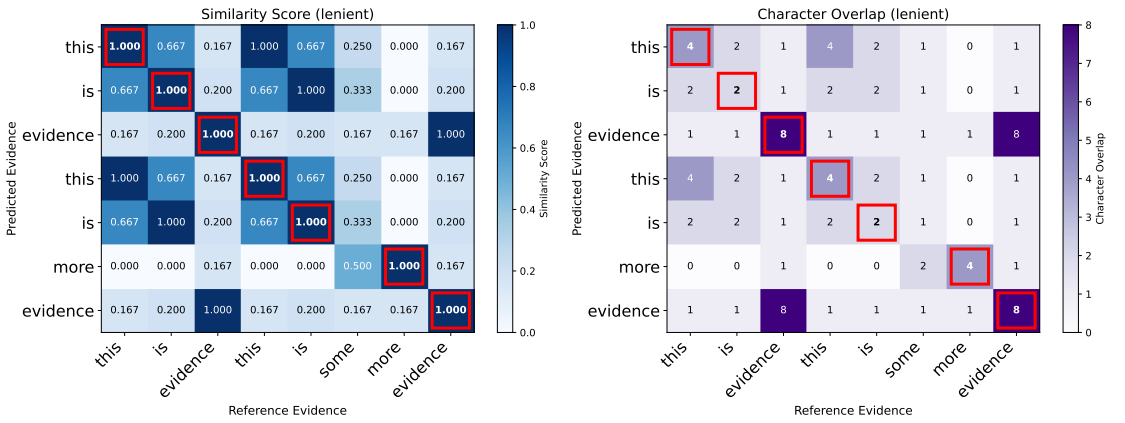


Figure E.3: Similarity scores and character overlaps computed for the lenient evidence metric with predicted evidence $\hat{\mathcal{E}}$ = {"this is evidence", "this is more evidence"} and reference evidence $\mathcal{E}$ = {"this is evidence", "this is some more evidence"}. Red outlines indicate the matched evidence determined via the similarity scores and superimposed on the character overlap matrix. The score for this example is then computed from the total amount of character overlap in the matched evidence, the total length of the predicted evidence, and the total length of the reference evidence.

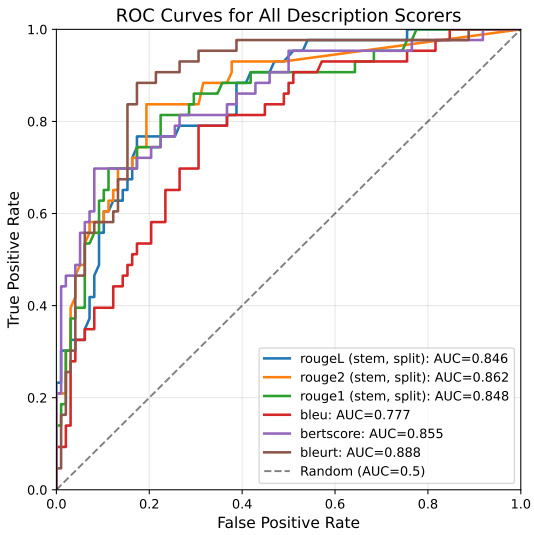


Figure F.1: ROC curves for all candidate description scorers based on 141 non-autograded examples from internal development data. All metrics can distinguish correct from incorrect descriptions, with BLEURT performing the best.

sample from the document 5 lines of more than 50 characters each and for each line obtain the number of occurrences it has within 3 corpora (OLMo-2-0325-32B-Instruct (OLMo et al., 2025), Dolma-v1.7 (Soldaini et al., 2024), and RedPajama (Weber et al., 2024)) using Infini-gram (Liu et al., 2024) and whether it appears in the Pile (Gao et al., 2020) using Data Portraits (Marone and Van Durme, 2023).[21] Documents with a line that appears in the searched corpora are considered to be potentially seen by the models.

Analysis sources.

MFR These documents range in release date, and we expect it is possible that models were exposed to these documents.

cs.CL These documents range in release date, and it is likely that models were exposed to them. These documents often have multiple versions, particularly the subset with identified inconsistencies, for which versions with and without the inconsistency are available by construction. We do not know if models were trained on multiple versions of these papers.

[21] Additional context before and after the sampled line was included in the input to Data Portraits to reduce false negatives.

H More Results

H.1 Development Data Results

See Table H.7, Table H.8, and Table H.9 for results on the development set. There are not many notable differences between development and test performances. The development test size is relatively small per source (25 vs. 75 in the test set), making statistical tests less informative.

That being said, the main area we were curious to observe potential differences in is performance on inconsistencies in **PG** between development and test settings. **PG** development instances were *not* date-protected and are possibly in the pre-training corpora because they were present in the corpora we searched (Appendix G). In the aforementioned tables, we denote **PG** as **PGS** (Project Gutenberg Seen) to make this distinction explicit. In general, however, the performance on average was lower for the development set versus the test set for **PG** and not statistically different under the LMJ task metric, $\Lambda_{\mathcal{I}}$. Even though most models performed worse on **PGS** compared to **PG**, we still consider it good practice to select documents unlikely to be in general pre-training data; larger datasets for more statistical power are needed to draw firmer conclusions.

We did not work to overly prompt tune the models with the development set. Instead, it was used to validate our grading model, which has high agreement with manual judgments (Cohen's kappa of 0.96).

H.2 Extended Results

Table H.2 reports all three metrics for the **MFR** source. Basic response statistics across sources and models are in Table H.3. Table H.4 reports the evidence, description, and task scores with standard errors. Table H.5 reports the linear regression coefficients between input length, inconsistency type, and task score, with each dataset source as the controls.

H.3 Duplicate Hits in Task Scores (Recall) Are Rare

Models could return the same answer more than once within a list of answers. Additionally, the model grader could grade more than once answer as correct within a list of answers because each answer is graded independently. If these events occurred at high frequency, it would suggest the grader had a general high false positive rate.

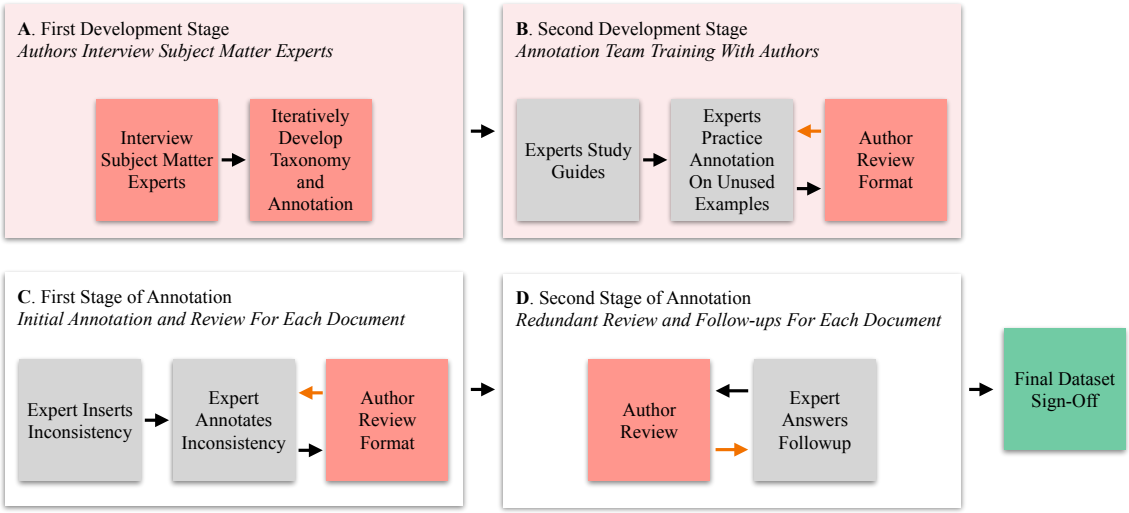


Figure G.1: Annotation and task quality workflow. The first two stages (A, B) are part of the dataset and task design. The next two stages (C, D) cover iterative annotation. First (**A**), we interviewed financial experts to help scope a useful taxonomy of inconsistency types. Next (**B**), our annotation team of financial experts reviewed the taxonomy of errors and annotation scheme. Then (**C**), the annotation team created and inserted inconsistencies into the documents. This was an iterative process of data review and validation which was followed by an additional review process (**D**). The complete annotation process required significant resources. On average, creating, annotating, and reviewing each inconsistency took two hours.

In Table H.1, we show the rate at which the LMJ grader returns True (or 1) for more than one answer in a single response. Across all the closed-source models, this occurs in 13/2940 generations (less than 0.05%). For both open-weight OpenAI models, it occurs in 3–6% of generations, and it does not occur at all for both Gemma models. Upon manual inspection of the responses from the closed-source models, we observed that 10/13 times were legitimate duplicates (the models truly did return the same answer more than once). For each of the other 3/13, one answer was correctly graded True (matching the expected inconsistency), and the other was a borderline case. The borderline prediction contained much of the content of the gold answer but missed some information or was the same issue present in different places of the document.

Overall, the occurrence of the grader model marking multiple answers in a single response as matching the expected inconsistency was rare enough that it does not seem to impact our results or indicate an unacceptable false positive rate.

	Duplicates
gemma-3-12b-it	0 / 420
gemma-3-27b-it	0 / 420
gpt-oss-20b	25 / 420
gpt-oss-120b	13 / 420
sonnet-v4	2 / 420
gpt-5-mini	2 / 420
gpt-5	0 / 420
o3-mini	0 / 420
o3	2 / 420
gemini-2.5-flash	2 / 420
gemini-2.5-pro	5 / 420

Table H.1: Number of times a model returned a response in which the grader model (`gpt-4.1`) found that more than one answer matched the expected inconsistency.

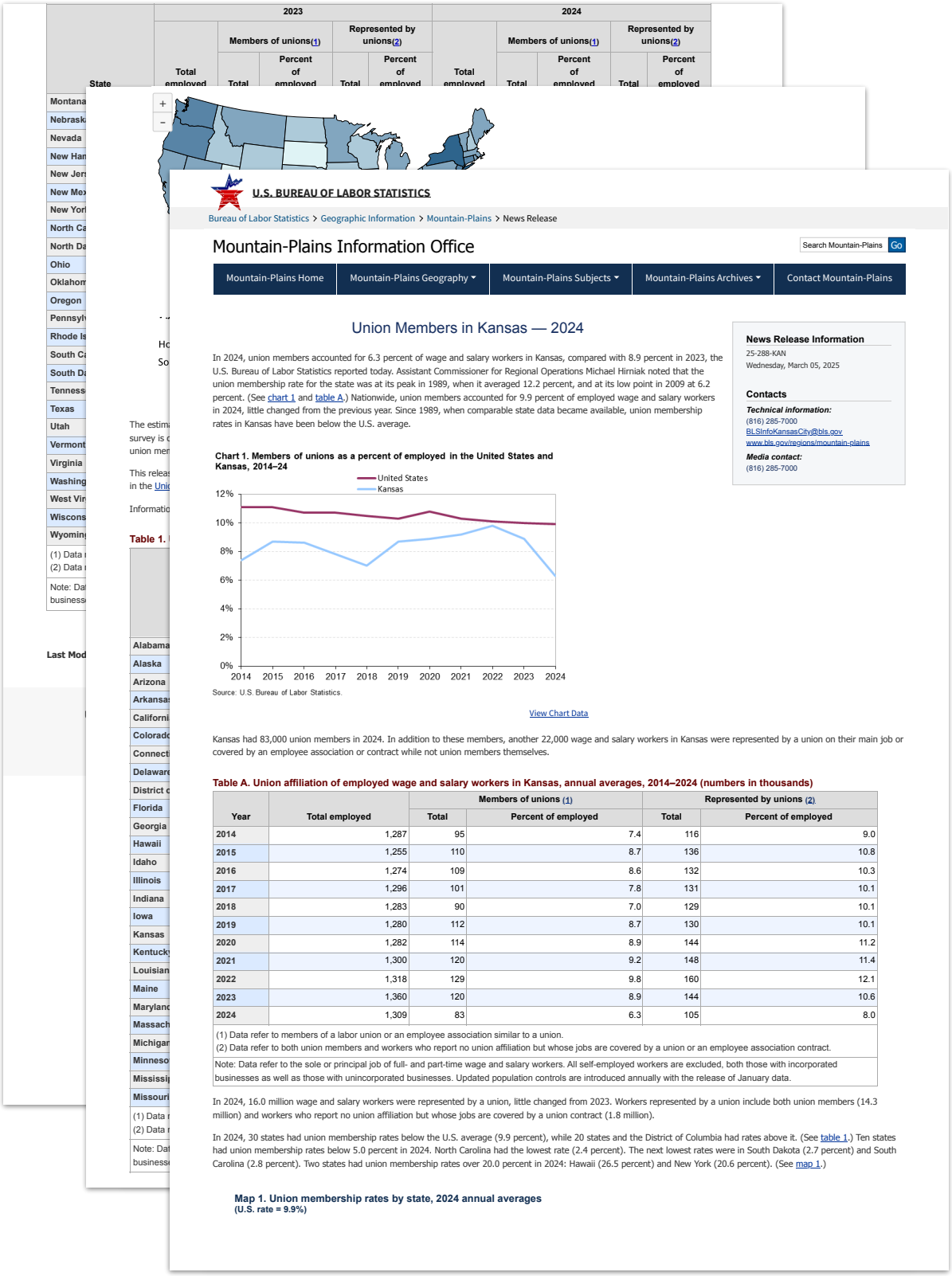
Model	$\Lambda_{\mathcal{E}}$	Λ_{δ}	$\Lambda_{\mathcal{I}}$
gemma-3-12b-it	4 (0.8)	36 (0.9)	**2 (2.2)**
gemma-3-27b-it	6 (1.3)	**38 (0.8)**	**2 (2.2)**
gpt-oss-20b	4 (1.4)	29 (1.9)	0 (0.0)
gpt-oss-120b	5 (1.1)	34 (1.4)	**4 (3.1)**
sonnet-v4	**15 (3.4)**	**35 (1.9)**	**11 (4.7)**
gpt-5-mini	10 (1.3)	35 (1.0)	2 (2.2)
gpt-5	**12 (2.2)**	**36 (1.5)**	**7 (3.7)**
o3-mini	0 (0.1)	24 (2.3)	2 (2.2)
o3	9 (2.1)	33 (1.5)	**9 (4.2)**
gemini-2.5-flash	**12 (2.6)**	**34 (2.5)**	**9 (4.2)**
gemini-2.5-pro	**16 (2.4)**	**38 (1.0)**	**9 (4.2)**

Table H.2: Evidence, description, and task scores (with standard errors) on **MFR** (response-level metrics). For all three scores, higher is better. Scores are presented as percents (out of 100). For results with the same mean, the variance in the paired differences can still differ.

	Valid					kTok/$\mathcal{D}$					$\hat{\mathcal{I}}/\mathcal{D}$				
Model	**BLS**	**PG**	**PRE**	**SEC**	**EMM**	**BLS**	**PG**	**PRE**	**SEC**	**EMM**	**BLS**	**PG**	**PRE**	**SEC**	**EMM**
gemma-3-12b-it	100.0	100.0	100.0	100.0	98.7	0.6	0.6	0.7	0.6	0.6	2.5	4.4	3.5	3.1	3.5
gemma-3-27b-it	100.0	100.0	100.0	100.0	98.7	0.8	0.8	0.9	0.9	0.9	4.1	5.3	4.4	4.3	4.6
gpt-oss-20b	100.0	94.7	98.7	93.3	96.0	4.8	4.3	6.4	6.9	4.8	2.7	2.3	2.4	2.2	2.1
gpt-oss-120b	100.0	98.7	100.0	100.0	98.7	2.6	2.5	2.7	2.9	2.7	2.2	1.5	1.9	1.8	1.9
sonnet-v4	100.0	100.0	100.0	92.0	96.0	3.2	4.8	4.5	6.5	4.8	1.5	1.4	2.2	1.3	1.8
gpt-5-mini	100.0	100.0	100.0	100.0	100.0	4.7	5.2	6.0	7.3	6.0	1.8	2.3	2.8	2.3	3.4
gpt-5	97.3	100.0	98.7	100.0	100.0	9.1	8.6	13.7	12.9	10.0	1.9	2.9	3.4	2.8	4.3
o3-mini	100.0	100.0	100.0	97.3	100.0	2.3	2.7	3.4	3.0	2.7	1.3	1.5	1.5	1.2	1.3
o3	100.0	100.0	100.0	98.7	98.7	4.8	4.1	7.2	6.9	3.8	1.4	1.7	2.2	1.7	2.2
gemini-2.5-flash	100.0	90.7	100.0	89.3	97.3	6.4	11.8	8.3	15.9	13.2	2.7	5.2	3.8	3.4	5.3
gemini-2.5-pro	100.0	100.0	100.0	97.3	100.0	8.7	10.6	11.1	17.2	16.0	2.7	5.1	5.3	4.5	6.9

	$\varepsilon/\hat{\mathcal{I}}$					Tok/ε				
Model	**BLS**	**PG**	**PRE**	**SEC**	**EMM**	**BLS**	**PG**	**PRE**	**SEC**	**EMM**
gemma-3-12b-it	2.9	2.0	2.4	2.5	2.3	12.1	14.1	18.6	19.4	18.9
gemma-3-27b-it	2.9	2.2	2.6	2.6	2.4	17.3	21.3	24.9	22.3	25.7
gpt-oss-20b	6.0	4.9	6.6	6.7	5.1	7.6	8.8	6.6	6.9	12.1
gpt-oss-120b	4.1	3.1	3.5	2.7	2.7	13.6	16.0	16.1	21.2	20.7
sonnet-v4	3.7	4.4	4.4	6.1	4.1	8.8	11.3	9.6	6.7	11.2
gpt-5-mini	2.6	2.5	2.7	2.6	2.3	33.8	29.4	36.3	29.5	32.8
gpt-5	2.6	2.3	2.5	2.6	2.2	22.7	24.1	28.4	23.3	30.8
o3-mini	4.4	5.4	3.1	7.7	5.1	8.5	10.3	17.0	6.2	12.2
o3	2.1	2.6	2.1	2.5	2.3	25.4	19.3	28.7	21.7	23.0
gemini-2.5-flash	6.2	2.7	4.1	4.8	3.4	13.4	47.3	25.1	22.2	39.6
gemini-2.5-pro	3.1	2.2	3.6	4.5	3.5	18.8	28.1	19.2	13.1	22.7

Table H.3: **Model response statistics across data sources.** Format validity rate (%), mean tokens per document (in thousands) (kTok/$\mathcal{D}$), mean answers per response ($\hat{\mathcal{I}}/\mathcal{D}$), mean evidence spans per answer ($\varepsilon/\hat{\mathcal{I}}$), and mean tokens per evidence span (Tok/ε).



U.S. BUREAU OF LABOR STATISTICS

Bureau of Labor Statistics > Geographic Information > Mountain-Plains > News Release

Mountain-Plains Information Office

Search Mountain-Plains Go

Mountain-Plains Home | Mountain-Plains Geography | Mountain-Plains Subjects | Mountain-Plains Archives | Contact Mountain-Plains

Union Members in Kansas — 2024

In 2024, union members accounted for 6.3 percent of wage and salary workers in Kansas, compared with 8.9 percent in 2023, the U.S. Bureau of Labor Statistics reported today. Assistant Commissioner for Regional Operations Michael Hirniak noted that the union membership rate for the state was at its peak in 1989, when it averaged 12.2 percent, and at its low point in 2009 at 6.2 percent. (See chart 1 and table A.) Nationwide, union members accounted for 9.9 percent of employed wage and salary workers in 2024, little changed from the previous year. Since 1989, when comparable state data became available, union membership rates in Kansas have been below the U.S. average.

News Release Information
25-288-KAN
Wednesday, March 05, 2025

Contacts
Technical information:
(816) 285-7000
BLSInfoKansasCity@bls.gov
www.bls.gov/regions/mountain-plains
Media contact:
(816) 285-7000

Chart 1. Members of unions as a percent of employed in the United States and Kansas, 2014–24

Source: U.S. Bureau of Labor Statistics.

View Chart Data

Kansas had 83,000 union members in 2024. In addition to these members, another 22,000 wage and salary workers in Kansas were represented by a union on their main job or covered by an employee association or contract while not union members themselves.

Table A. Union affiliation of employed wage and salary workers in Kansas, annual averages, 2014–2024 (numbers in thousands)

Year	Total employed	Members of unions (1)		Represented by unions (2)	
		Total	Percent of employed	Total	Percent of employed
2014	1,287	95	7.4	116	9.0
2015	1,255	110	8.7	136	10.8
2016	1,274	109	8.6	132	10.3
2017	1,296	101	7.8	131	10.1
2018	1,283	90	7.0	129	10.1
2019	1,280	112	8.7	130	10.1
2020	1,282	114	8.9	144	11.2
2021	1,300	120	9.2	148	11.4
2022	1,318	129	9.8	160	12.1
2023	1,360	120	8.9	144	10.6
2024	1,309	83	6.3	105	8.0

(1) Data refer to members of a labor union or an employee association similar to a union.
(2) Data refer to both union members and workers who report no union affiliation but whose jobs are covered by a union or an employee association contract.

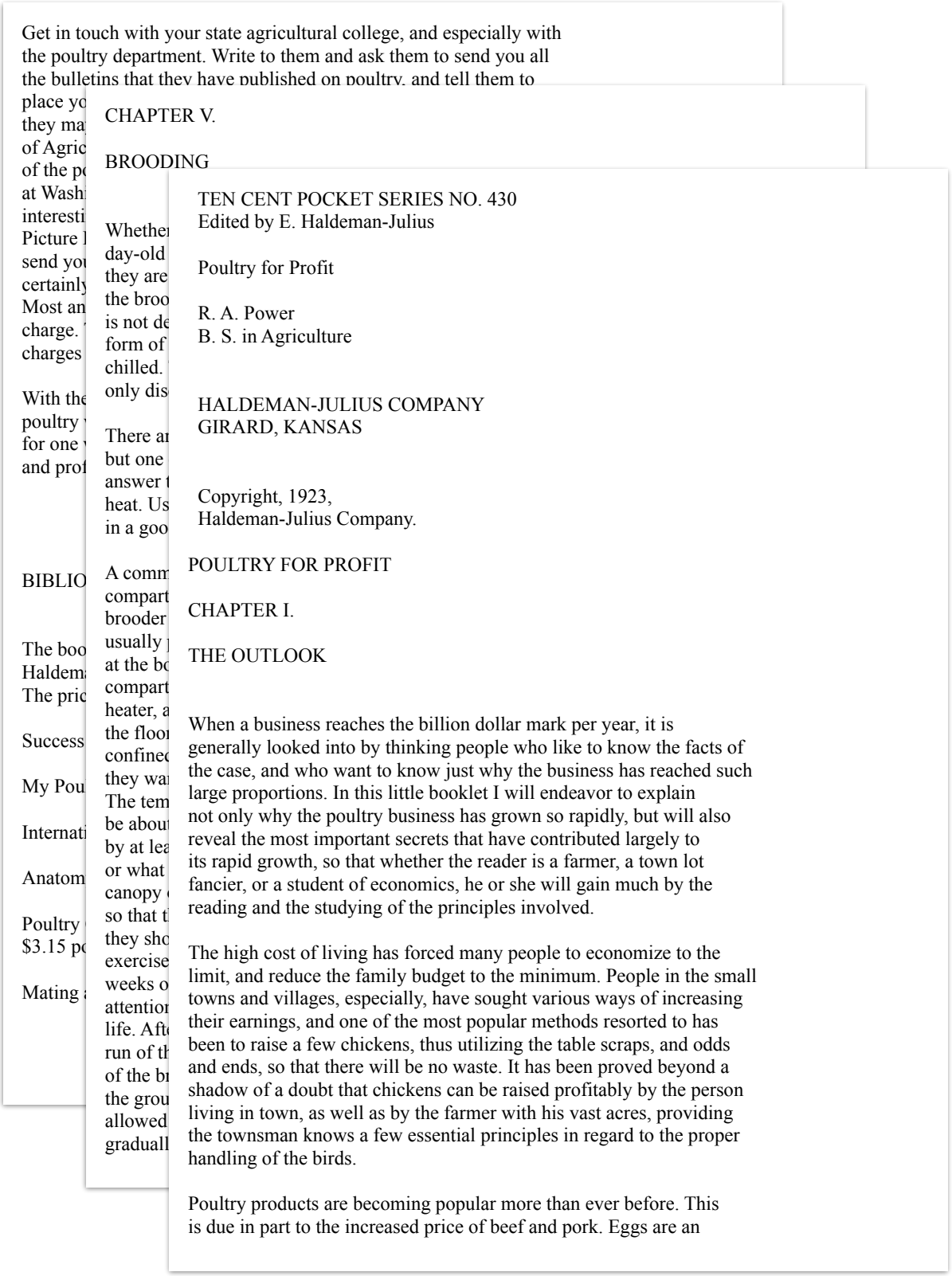
Note: Data refer to the sole or principal job of full- and part-time wage and salary workers. All self-employed workers are excluded, both those with incorporated businesses as well as those with unincorporated businesses. Updated population controls are introduced annually with the release of January data.

In 2024, 16.0 million wage and salary workers were represented by a union, little changed from 2023. Workers represented by a union include both union members (14.3 million) and workers who report no union affiliation but whose jobs are covered by a union contract (1.8 million).

In 2024, 30 states had union membership rates below the U.S. average (9.9 percent), while 20 states and the District of Columbia had rates above it. (See table 1.) Ten states had union membership rates below 5.0 percent in 2024. North Carolina had the lowest rate (2.4 percent). The next lowest rates were in South Dakota (2.7 percent) and South Carolina (2.8 percent). Two states had union membership rates over 20.0 percent in 2024: Hawaii (26.5 percent) and New York (20.6 percent). (See map 1.)

Map 1. Union membership rates by state, 2024 annual averages
(U.S. rate = 9.9%)

Figure G.2: Pages from an example from **BLS**. See original at https://www.bls.gov/regions/mountain-plains/news-release/2025/unionmembership_kansas_20250305.htm.



Get in touch with your state agricultural college, and especially with the poultry department. Write to them and ask them to send you all the bulletins that they have published on poultry, and tell them to

CHAPTER V.

BROODING

TEN CENT POCKET SERIES NO. 430
Edited by E. Haldeman-Julius

Poultry for Profit

R. A. Power
B. S. in Agriculture

HALDEMAN-JULIUS COMPANY
GIRARD, KANSAS

POULTRY FOR PROFIT

CHAPTER I.

THE OUTLOOK

When a business reaches the billion dollar mark per year, it is generally looked into by thinking people who like to know the facts of the case, and who want to know just why the business has reached such large proportions. In this little booklet I will endeavor to explain not only why the poultry business has grown so rapidly, but will also reveal the most important secrets that have contributed largely to its rapid growth, so that whether the reader is a farmer, a town lot fancier, or a student of economics, he or she will gain much by the reading and the studying of the principles involved.

The high cost of living has forced many people to economize to the limit, and reduce the family budget to the minimum. People in the small towns and villages, especially, have sought various ways of increasing their earnings, and one of the most popular methods resorted to has been to raise a few chickens, thus utilizing the table scraps, and odds and ends, so that there will be no waste. It has been proved beyond a shadow of a doubt that chickens can be raised profitably by the person living in town, as well as by the farmer with his vast acres, providing the townsman knows a few essential principles in regard to the proper handling of the birds.

Poultry products are becoming popular more than ever before. This is due in part to the increased price of beef and pork. Eggs are an

Figure G.3: Pages from an example from **PG**. See original at https://www.gutenberg.org/ebooks/75419.

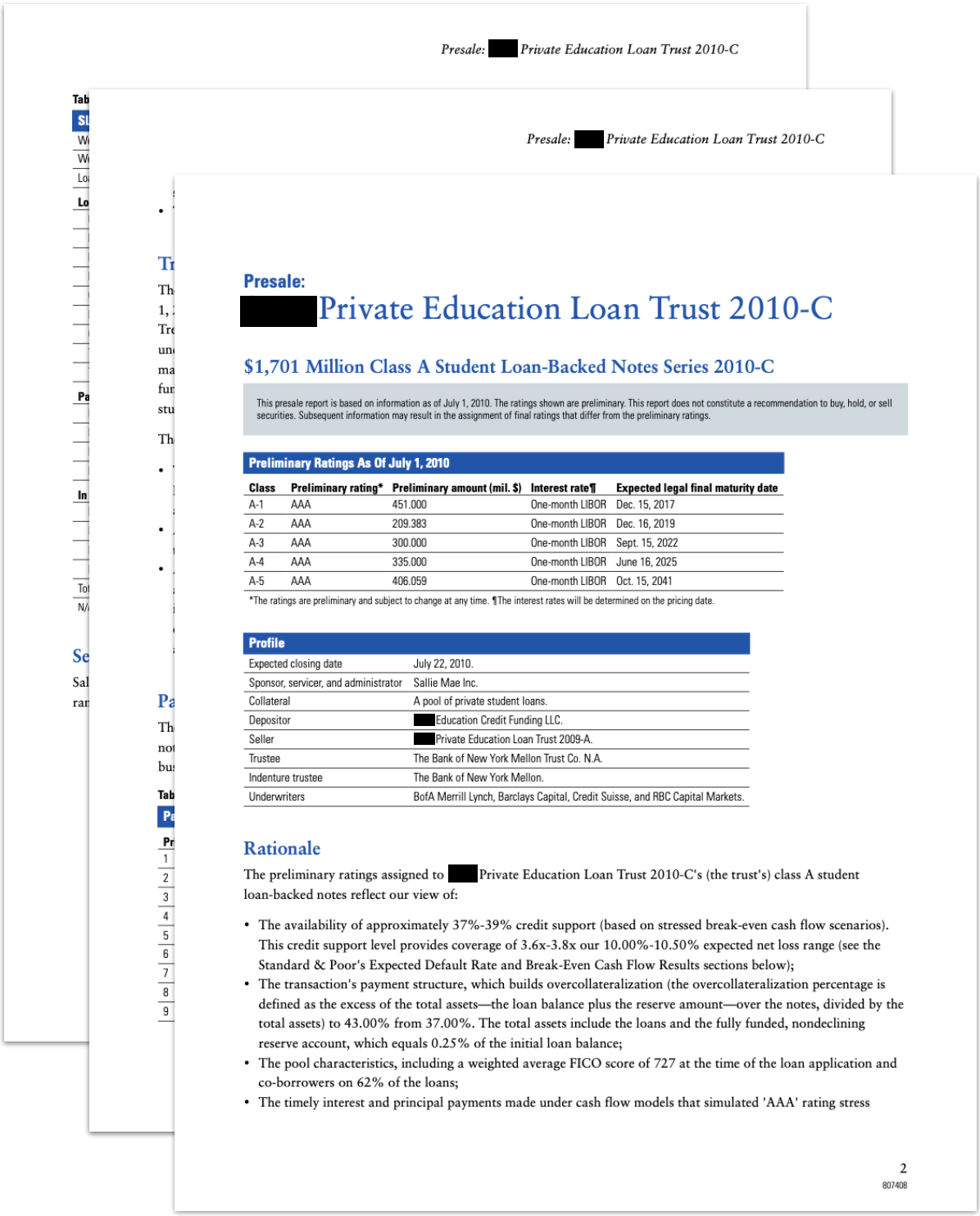
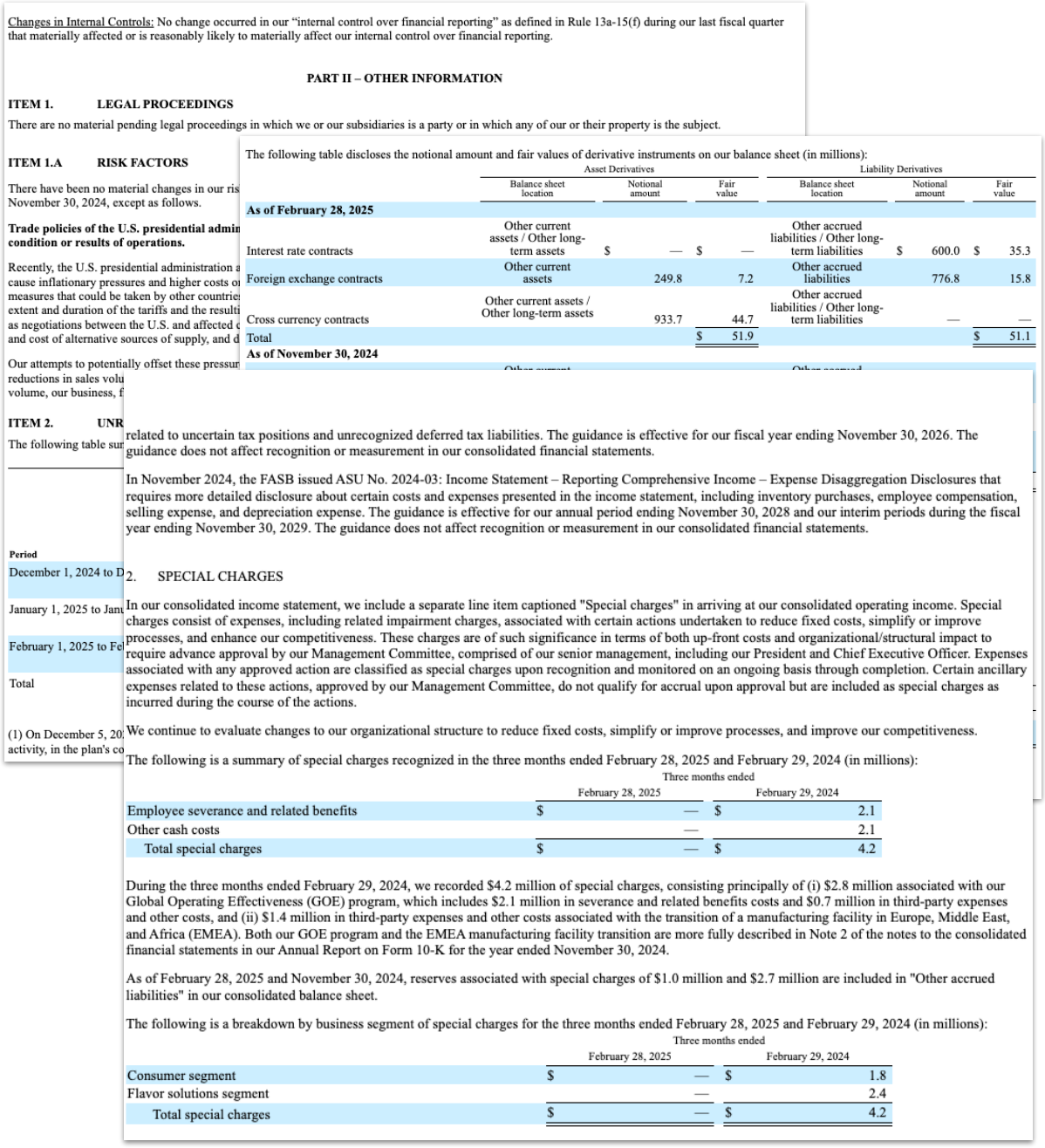


Figure G.4: Pages from an example from **PRE**. Documents were anonymized.



Changes in Internal Controls: No change occurred in our "internal control over financial reporting" as defined in Rule 13a-15(f) during our last fiscal quarter that materially affected or is reasonably likely to materially affect our internal control over financial reporting.

PART II – OTHER INFORMATION

ITEM 1. LEGAL PROCEEDINGS

There are no material pending legal proceedings in which we or our subsidiaries is a party or in which any of our or their property is the subject.

ITEM 1.A RISK FACTORS

There have been no material changes in our ris
November 30, 2024, except as follows.

**Trade policies of the U.S. presidential admin
condition or results of operations.**

Recently, the U.S. presidential administration a
cause inflationary pressures and higher costs o
measures that could be taken by other countrie
extent and duration of the tariffs and the resulti
as negotiations between the U.S. and affected c
and cost of alternative sources of supply, and d

Our attempts to potentially offset these pressur
reductions in sales volu
volume, our business, f

ITEM 2. UNR

The following table su

Period
December 1, 2024 to D
January 1, 2025 to Jan
February 1, 2025 to Fe
Total

(1) On December 5, 20
activity, in the plan's co

The following table discloses the notional amount and fair values of derivative instruments on our balance sheet (in millions):

	Asset Derivatives			Liability Derivatives		
	Balance sheet location	Notional amount	Fair value	Balance sheet location	Notional amount	Fair value
As of February 28, 2025						
Interest rate contracts	Other current assets / Other long-term assets	$ —	$ —	Other accrued liabilities / Other long-term liabilities	$ 600.0	$ 35.3
Foreign exchange contracts	Other current assets	249.8	7.2	Other accrued liabilities	776.8	15.8
Cross currency contracts	Other current assets / Other long-term assets	933.7	44.7	Other accrued liabilities / Other long-term liabilities	—	—
Total			$ 51.9			$ 51.1
As of November 30, 2024						

related to uncertain tax positions and unrecognized deferred tax liabilities. The guidance is effective for our fiscal year ending November 30, 2026. The guidance does not affect recognition or measurement in our consolidated financial statements.

In November 2024, the FASB issued ASU No. 2024-03: Income Statement – Reporting Comprehensive Income – Expense Disaggregation Disclosures that requires more detailed disclosure about certain costs and expenses presented in the income statement, including inventory purchases, employee compensation, selling expense, and depreciation expense. The guidance is effective for our annual period ending November 30, 2028 and our interim periods during the fiscal year ending November 30, 2029. The guidance does not affect recognition or measurement in our consolidated financial statements.

2. SPECIAL CHARGES

In our consolidated income statement, we include a separate line item captioned "Special charges" in arriving at our consolidated operating income. Special charges consist of expenses, including related impairment charges, associated with certain actions undertaken to reduce fixed costs, simplify or improve processes, and enhance our competitiveness. These charges are of such significance in terms of both up-front costs and organizational/structural impact to require advance approval by our Management Committee, comprised of our senior management, including our President and Chief Executive Officer. Expenses associated with any approved action are classified as special charges upon recognition and monitored on an ongoing basis through completion. Certain ancillary expenses related to these actions, approved by our Management Committee, do not qualify for accrual upon approval but are included as special charges as incurred during the course of the actions.

We continue to evaluate changes to our organizational structure to reduce fixed costs, simplify or improve processes, and improve our competitiveness.

The following is a summary of special charges recognized in the three months ended February 28, 2025 and February 29, 2024 (in millions):

	Three months ended	
	February 28, 2025	February 29, 2024
Employee severance and related benefits	$ —	$ 2.1
Other cash costs	—	2.1
Total special charges	$ —	$ 4.2

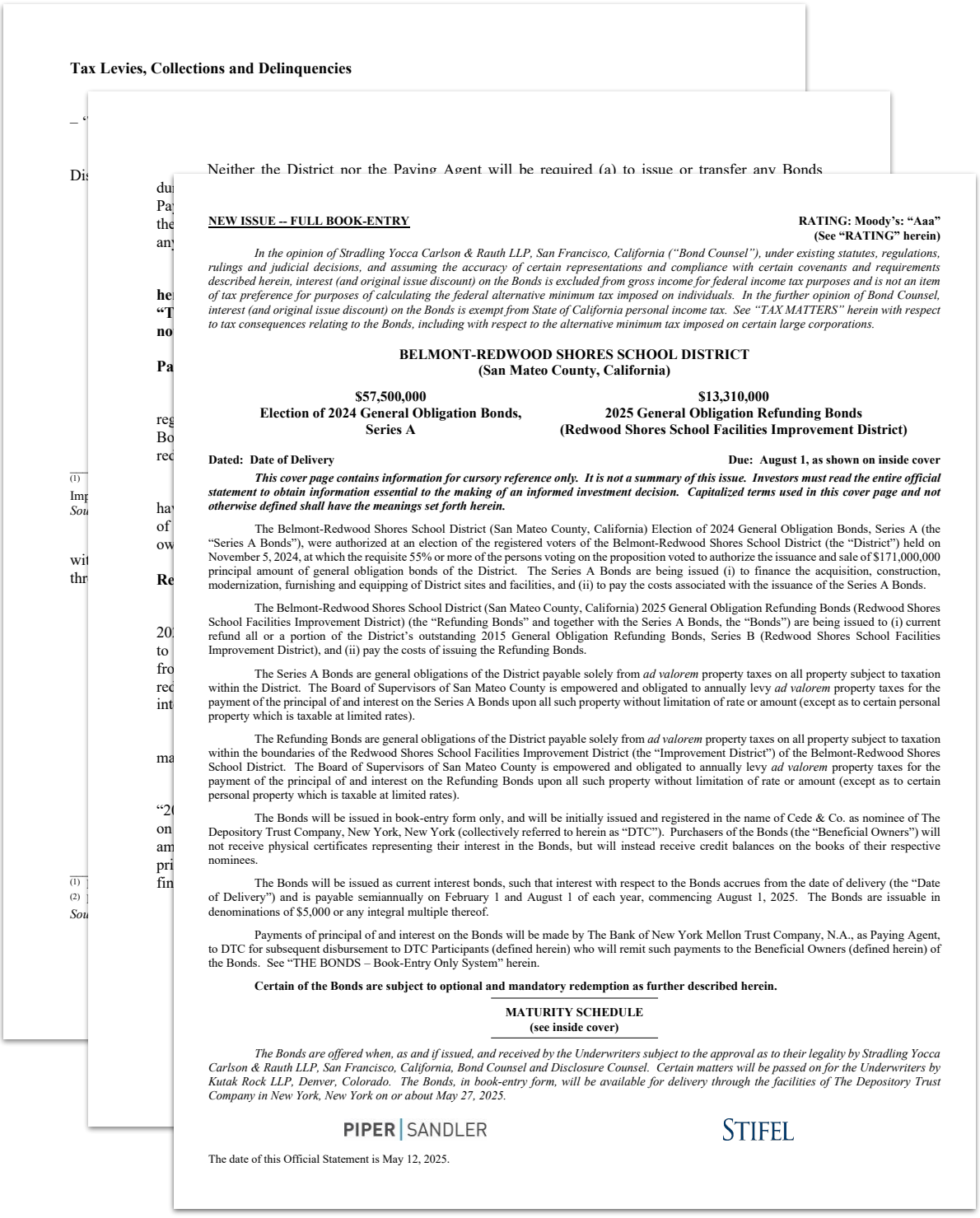
During the three months ended February 29, 2024, we recorded $4.2 million of special charges, consisting principally of (i) $2.8 million associated with our Global Operating Effectiveness (GOE) program, which includes $2.1 million in severance and related benefits costs and $0.7 million in third-party expenses and other costs, and (ii) $1.4 million in third-party expenses and other costs associated with the transition of a manufacturing facility in Europe, Middle East, and Africa (EMEA). Both our GOE program and the EMEA manufacturing facility transition are more fully described in Note 2 of the notes to the consolidated financial statements in our Annual Report on Form 10-K for the year ended November 30, 2024.

As of February 28, 2025 and November 30, 2024, reserves associated with special charges of $1.0 million and $2.7 million are included in "Other accrued liabilities" in our consolidated balance sheet.

The following is a breakdown by business segment of special charges for the three months ended February 28, 2025 and February 29, 2024 (in millions):

	Three months ended	
	February 28, 2025	February 29, 2024
Consumer segment	$ —	$ 1.8
Flavor solutions segment	—	2.4
Total special charges	$ —	$ 4.2

Figure G.5: Pages from an example from **SEC**. See original at https://www.sec.gov/ix?doc=/Archives/edgar/data/63754/000006375425000024/mkc-20250228.htm.



Tax Levies, Collections and Delinquencies

Neither the District nor the Paying Agent will be required (a) to issue or transfer any Bonds

NEW ISSUE -- FULL BOOK-ENTRY

RATING: Moody's: "Aaa"
(See "RATING" herein)

In the opinion of Stradling Yocca Carlson & Rauth LLP, San Francisco, California ("Bond Counsel"), under existing statutes, regulations, rulings and judicial decisions, and assuming the accuracy of certain representations and compliance with certain covenants and requirements described herein, interest (and original issue discount) on the Bonds is excluded from gross income for federal income tax purposes and is not an item of tax preference for purposes of calculating the federal alternative minimum tax imposed on individuals. In the further opinion of Bond Counsel, interest (and original issue discount) on the Bonds is exempt from State of California personal income tax. See "TAX MATTERS" herein with respect to tax consequences relating to the Bonds, including with respect to the alternative minimum tax imposed on certain large corporations.

BELMONT-REDWOOD SHORES SCHOOL DISTRICT
(San Mateo County, California)

$57,500,000
Election of 2024 General Obligation Bonds, Series A

$13,310,000
2025 General Obligation Refunding Bonds
(Redwood Shores School Facilities Improvement District)

Dated: Date of Delivery

Due: August 1, as shown on inside cover

This cover page contains information for cursory reference only. It is not a summary of this issue. Investors must read the entire official statement to obtain information essential to the making of an informed investment decision. Capitalized terms used in this cover page and not otherwise defined shall have the meanings set forth herein.

The Belmont-Redwood Shores School District (San Mateo County, California) Election of 2024 General Obligation Bonds, Series A (the "Series A Bonds"), were authorized at an election of the registered voters of the Belmont-Redwood Shores School District (the "District") held on November 5, 2024, at which the requisite 55% or more of the persons voting on the proposition voted to authorize the issuance and sale of $171,000,000 principal amount of general obligation bonds of the District. The Series A Bonds are being issued (i) to finance the acquisition, construction, modernization, furnishing and equipping of District sites and facilities, and (ii) to pay the costs associated with the issuance of the Series A Bonds.

The Belmont-Redwood Shores School District (San Mateo County, California) 2025 General Obligation Refunding Bonds (Redwood Shores School Facilities Improvement District) (the "Refunding Bonds" and together with the Series A Bonds, the "Bonds") are being issued to (i) current refund all or a portion of the District's outstanding 2015 General Obligation Refunding Bonds, Series B (Redwood Shores School Facilities Improvement District), and (ii) pay the costs of issuing the Refunding Bonds.

The Series A Bonds are general obligations of the District payable solely from *ad valorem* property taxes on all property subject to taxation within the District. The Board of Supervisors of San Mateo County is empowered and obligated to annually levy *ad valorem* property taxes for the payment of the principal of and interest on the Series A Bonds upon all such property without limitation of rate or amount (except as to certain personal property which is taxable at limited rates).

The Refunding Bonds are general obligations of the District payable solely from *ad valorem* property taxes on all property subject to taxation within the boundaries of the Redwood Shores School Facilities Improvement District (the "Improvement District") of the Belmont-Redwood Shores School District. The Board of Supervisors of San Mateo County is empowered and obligated to annually levy *ad valorem* property taxes for the payment of the principal of and interest on the Refunding Bonds upon all such property without limitation of rate or amount (except as to certain personal property which is taxable at limited rates).

The Bonds will be issued in book-entry form only, and will be initially issued and registered in the name of Cede & Co. as nominee of The Depository Trust Company, New York, New York (collectively referred to herein as "DTC"). Purchasers of the Bonds (the "Beneficial Owners") will not receive physical certificates representing their interest in the Bonds, but will instead receive credit balances on the books of their respective nominees.

The Bonds will be issued as current interest bonds, such that interest with respect to the Bonds accrues from the date of delivery (the "Date of Delivery") and is payable semiannually on February 1 and August 1 of each year, commencing August 1, 2025. The Bonds are issuable in denominations of $5,000 or any integral multiple thereof.

Payments of principal of and interest on the Bonds will be made by The Bank of New York Mellon Trust Company, N.A., as Paying Agent, to DTC for subsequent disbursement to DTC Participants (defined herein) who will remit such payments to the Beneficial Owners (defined herein) of the Bonds. See "THE BONDS – Book-Entry Only System" herein.

Certain of the Bonds are subject to optional and mandatory redemption as further described herein.

MATURITY SCHEDULE
(see inside cover)

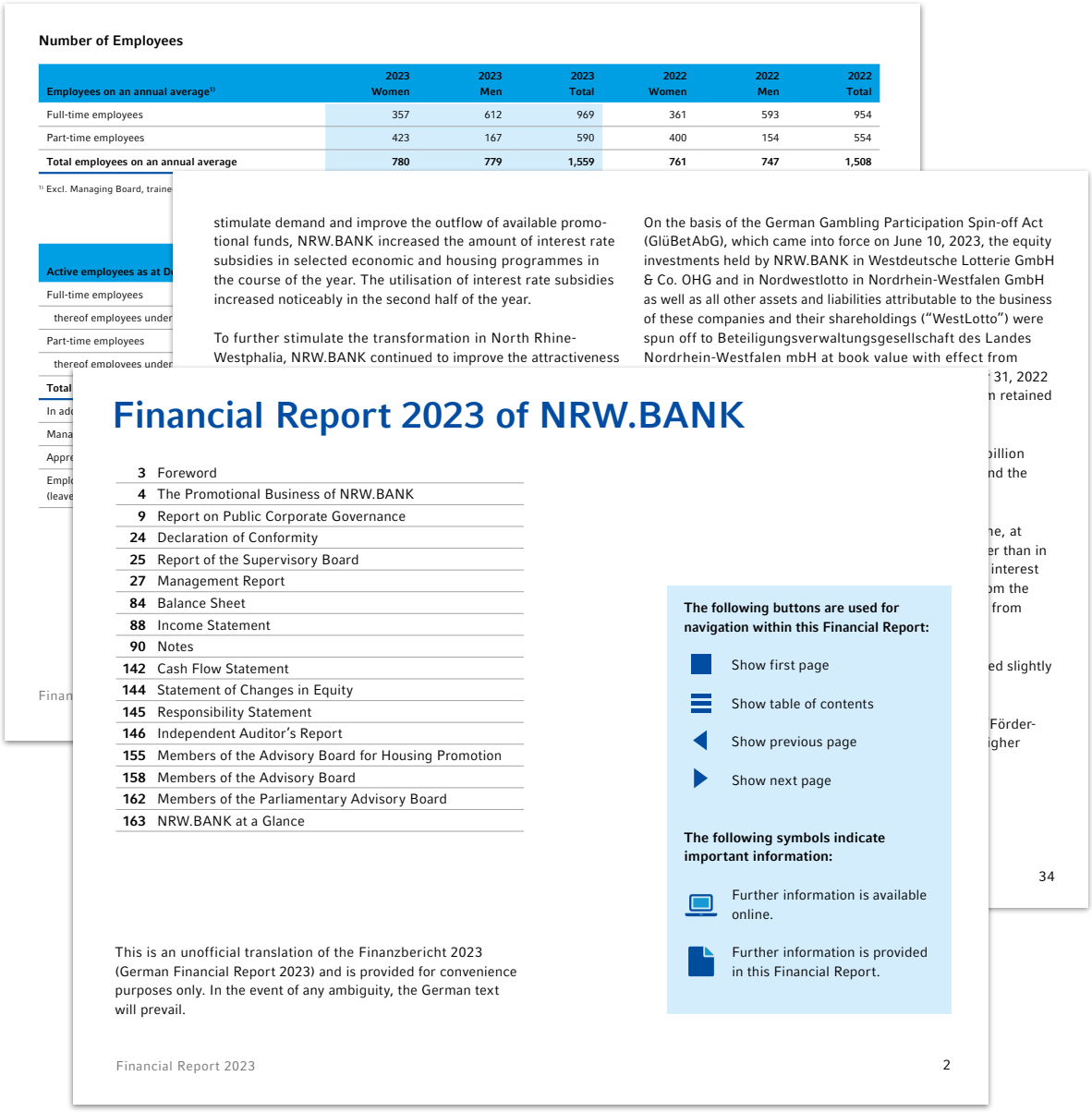
The Bonds are offered when, as and if issued, and received by the Underwriters subject to the approval as to their legality by Stradling Yocca Carlson & Rauth LLP, San Francisco, California, Bond Counsel and Disclosure Counsel. Certain matters will be passed on for the Underwriters by Kutak Rock LLP, Denver, Colorado. The Bonds, in book-entry form, will be available for delivery through the facilities of The Depository Trust Company in New York, New York on or about May 27, 2025.

PIPER | SANDLER

STIFEL

The date of this Official Statement is May 12, 2025.

Figure G.6: Pages from an example from **EMM**. See original at https://emma.msrb.org/P11854654-P11420157-P11863682.pdf.



Number of Employees

Employees on an annual average[1]	2023 Women	2023 Men	2023 Total	2022 Women	2022 Men	2022 Total
Full-time employees	357	612	969	361	593	954
Part-time employees	423	167	590	400	154	554
Total employees on an annual average	**780**	**779**	**1,559**	**761**	**747**	**1,508**

[1] Excl. Managing Board, traine

Active employees as at D

Full-time employees
thereof employees under
Part-time employees
thereof employees under
Total
In ad
Mana
Appre
Emplo
(leave

stimulate demand and improve the outflow of available promotional funds, NRW.BANK increased the amount of interest rate subsidies in selected economic and housing programmes in the course of the year. The utilisation of interest rate subsidies increased noticeably in the second half of the year.

To further stimulate the transformation in North Rhine-Westphalia, NRW.BANK continued to improve the attractiveness

On the basis of the German Gambling Participation Spin-off Act (GlüBetAbG), which came into force on June 10, 2023, the equity investments held by NRW.BANK in Westdeutsche Lotterie GmbH & Co. OHG and in Nordwestlotto in Nordrhein-Westfalen GmbH as well as all other assets and liabilities attributable to the business of these companies and their shareholdings ("WestLotto") were spun off to Beteiligungsverwaltungsgesellschaft des Landes Nordrhein-Westfalen mbH at book value with effect from

31, 2022
n retained

illion
nd the

ne, at
er than in
interest
m the
from

ed slightly

Förder-
igher

34

Finan

Financial Report 2023 of NRW.BANK

The following buttons are used for navigation within this Financial Report:

- Show first page
- Show table of contents
- Show previous page
- Show next page

The following symbols indicate important information:

- Further information is available online.
- Further information is provided in this Financial Report.

This is an unofficial translation of the Finanzbericht 2023 (German Financial Report 2023) and is provided for convenience purposes only. In the event of any ambiguity, the German text will prevail.

Figure G.7: Pages from an example from **MFR**. See original at https://www.nrwbank.de/export/.galleries/downloads/Info-und-Service/Finanzberichte/nrwbank-financial-report-2023.pdf.

	Evidence Score ($\Lambda_{\mathcal{E}}$)						Description Score (Λ_{δ})						Task Score ($\Lambda_{\mathcal{I}}$)					
Model	BLS	PRE	SEC	EMM	PG	AVG	BLS	PRE	SEC	EMM	PG	AVG	BLS	PRE	SEC	EMM	PG	AVG
gemma-3-12b-it	18 (2.4)	5 (0.6)	6 (1.2)	3 (0.5)	4 (1.0)	7 (2.5)	43 (1.4)	37 (0.8)	35 (0.8)	34 (0.8)	33 (0.9)	36 (1.6)	57 (5.7)	25 (5.0)	7 (2.9)	3 (1.9)	9 (3.4)	20 (9.0)
gemma-3-27b-it	23 (2.7)	8 (0.9)	4 (0.7)	1 (0.4)	0 (0.2)	7 (3.7)	45 (1.3)	38 (0.7)	36 (0.7)	34 (0.8)	34 (0.8)	37 (1.9)	61 (5.6)	27 (5.1)	7 (2.9)	8 (3.1)	12 (3.8)	23 (9.2)
gpt-oss-20b	30 (3.4)	11 (2.2)	3 (0.6)	2 (0.9)	2 (1.3)	10 (4.8)	45 (1.9)	32 (1.9)	27 (1.7)	28 (1.4)	28 (1.6)	32 (3.0)	21 (4.7)	3 (1.9)	1 (1.3)	0 (0.0)	3 (1.9)	6 (3.5)
gpt-oss-120b	35 (3.5)	14 (2.2)	7 (1.7)	5 (1.3)	4 (1.5)	13 (5.1)	47 (2.0)	36 (1.5)	34 (0.9)	34 (1.0)	32 (1.5)	37 (2.4)	28 (5.2)	3 (1.9)	0 (0.0)	0 (0.0)	1 (1.3)	6 (4.9)
sonnet-v4	**44 (3.3)**	**35 (3.6)**	13 (2.3)	14 (2.9)	9 (2.3)	**23 (6.2)**	**53 (1.5)**	43 (1.4)	30 (2.0)	33 (1.7)	32 (2.1)	**38 (3.9)**	**83 (4.4)**	49 (5.8)	20 (4.6)	21 (4.7)	23 (4.8)	39 (10.9)
gpt-5-mini	37 (3.0)	27 (2.9)	25 (2.9)	17 (2.3)	9 (1.9)	**23 (4.2)**	46 (1.2)	43 (1.0)	39 (0.9)	38 (1.1)	38 (1.5)	**41 (1.3)**	80 (4.6)	53 (5.8)	47 (5.8)	**37 (5.6)**	39 (5.6)	51 (6.9)
gpt-5	**43 (3.2)**	**36 (3.2)**	**42 (3.2)**	**27 (3.4)**	**18 (2.9)**	**33 (4.2)**	50 (1.5)	**46 (1.4)**	**44 (1.1)**	**41 (1.2)**	**43 (1.5)**	**45 (1.4)**	**87 (3.9)**	**67 (5.4)**	**73 (5.1)**	**43 (5.7)**	**52 (5.8)**	**64 (6.9)**
o3-mini	23 (3.6)	5 (1.4)	1 (0.3)	0 (0.2)	1 (1.3)	6 (3.8)	39 (2.7)	31 (1.8)	19 (2.1)	23 (1.5)	22 (2.0)	27 (3.3)	56 (5.7)	17 (4.4)	8 (3.1)	3 (1.9)	12 (3.8)	19 (8.5)
o3	**40 (3.0)**	29 (3.2)	25 (2.9)	17 (2.9)	10 (2.3)	**24 (4.7)**	49 (1.1)	42 (1.4)	39 (1.5)	35 (1.4)	37 (1.9)	**40 (2.2)**	**85 (4.1)**	49 (5.8)	57 (5.7)	31 (5.3)	40 (5.7)	53 (8.4)
gemini-2.5-flash	**42 (3.5)**	25 (2.9)	19 (2.9)	13 (2.6)	5 (1.0)	**21 (5.5)**	**53 (1.4)**	45 (1.0)	38 (1.7)	39 (1.3)	36 (1.7)	**42 (2.7)**	75 (5.0)	48 (5.8)	36 (5.5)	25 (5.0)	31 (5.3)	43 (7.9)
gemini-2.5-pro	**45 (3.6)**	**38 (3.1)**	**36 (3.3)**	**27 (3.3)**	12 (2.1)	**32 (5.1)**	**54 (1.2)**	**48 (1.2)**	**42 (1.3)**	**43 (1.2)**	**44 (1.5)**	**46 (1.9)**	**88 (3.8)**	**71 (5.3)**	61 (5.6)	**40 (5.7)**	**45 (5.7)**	**61 (7.8)**

Table H.4: Evidence, description, and task scores (with standard errors) across the **FIND** test set (response-level metrics). For all three scores, higher is better. Scores are presented as percents (out of 100). These results are an extension of those shown in the main body. For the average column, we use a clustered test to account for information correlated within a data source.

	Task Score ($\Lambda_{\mathcal{I}}$)			Coefficients (pp)			
Model	Num	Non	Struc	Len	Num	Non	R^2
gemma-3-12b-it	0.06	0.10	0.02	-1.0*	+2.5	+5.7	0.14
gemma-3-27b-it	0.10	0.03	0.02	-1.4**	+5.1	-1.4	0.23
gpt-oss-20b	0.26	0.25	0.06	-2.6***	+12.8**	+12.4*	0.29
gpt-oss-120b	0.27	0.38	0.06	-2.7***	+13.9**	+25.3***	0.30
sonnet-v4	0.45	0.49	0.23	-4.0***	+12.4*	+17.4**	0.30
gpt-5-mini	0.60	0.56	0.31	-2.6*	+23.9***	+19.3*	0.15
gpt-5	0.68	0.70	0.54	-2.6*	+9.5	+11.1	0.13
o3-mini	0.24	0.27	0.06	-3.5***	+11.8**	+14.8**	0.30
o3	0.58	0.59	0.39	-4.3***	+13.5*	+13.8	0.19
gemini-2.5-flash	0.50	0.46	0.28	-2.8**	+16.2**	+12.3	0.16
gemini-2.5-pro	0.67	0.65	0.48	-2.6*	+11.7*	+10.4	0.15

Table H.5: Task score by inconsistency type with regression coefficients for the types and document length. The average task score is shown for each type. Coefficients (pp): Len = change in task score ($\Lambda_{\mathcal{I}}$) per 10k tokens; Num/Non (which refers to Numeric/Non-numeric inconsistency types) columns indicate the effect compared to the structural inconsistency type (as a reference). Significance: *** $p < 0.001$, ** $p < 0.01$, * $p < 0.05$.

	Count					Task Score ($\Lambda_{\mathcal{I}}$) Across Models				
Length	BLS	PRE	SEC	EMM	PG	BLS	PRE	SEC	EMM	PG
0–10k	52	32	0	0	1	0.72	0.43	–	–	1.00
10–20k	9	38	1	1	7	0.63	0.33	0.00	0.67	0.41
20–30k	6	5	12	3	7	0.45	0.34	0.44	0.39	0.32
30–40k	3	0	26	4	6	0.54	–	0.26	0.31	0.30
40–50k	2	0	15	3	2	0.43	–	0.25	0.03	0.09
50–60k	2	0	9	4	7	0.44	–	0.34	0.30	0.36
60–70k	1	0	3	3	2	0.00	–	0.22	0.00	0.64
70–80k	0	0	1	10	4	0.00	–	0.23	0.22	0.18
80–90k	0	0	1	3	8	0.00	–	0.44	0.08	0.19
90–100k	0	0	1	5	1	–	–	0.18	0.21	0.06
100k	0	0	6	39	30	–	–	0.22	0.17	0.13

Table H.6: Data distribution and task score ($\Lambda_{\mathcal{I}}$) by document length and source. The document lengths are based on the `gpt-5` tokenizer. The task score is averaged across all models.

	$\mathcal{D}$	kTok/$\mathcal{D}$	$\varepsilon/\mathcal{D}$	Tok/ε	ε Pos%
BLS	25	$17_{\pm 21.1}$	$5_{\pm 3.1}$	$19_{\pm 44.1}$	$23_{\pm 18.8}$
PRE	25	$13_{\pm 6.9}$	$5_{\pm 3.8}$	$9_{\pm 9.5}$	$41_{\pm 20.7}$
SEC	26	$43_{\pm 22.4}$	$4_{\pm 2.5}$	$14_{\pm 11.5}$	$36_{\pm 26.5}$
EMM	24	$95_{\pm 57.2}$	$4_{\pm 3.0}$	$12_{\pm 11.1}$	$26_{\pm 19.4}$
PGS	25	$146_{\pm 111.8}$	$5_{\pm 3.7}$	$7_{\pm 4.8}$	$22_{\pm 17.9}$
FIND	125	$62_{\pm 76.8}$	$5_{\pm 3.3}$	$12_{\pm 21.9}$	$30_{\pm 22.2}$

Table H.7: **Statistics by document source (development set)**: number of documents ($\mathcal{D}$), mean tokens per document (in thousands) (kTok/$\mathcal{D}$), mean evidence spans per document ($\varepsilon/\mathcal{D}$), mean tokens per evidence span (Tok/ε), and mean relative position of evidence within documents (ε Pos%). Subscripts denote standard deviations. **FIND** row reports statistics pooled over the sources above.

Model	Valid	kTok/$\mathcal{D}$	$\hat{\mathcal{I}}/\mathcal{D}$	$\varepsilon/\hat{\mathcal{I}}$	Tok/ε
sonnet-v4	96.0	$5.0_{\pm 0.2}$	$1.50_{\pm 0.1}$	$4.33_{\pm 0.4}$	$10_{\pm 1.0}$
gpt-5-mini	100.0	$6.1_{\pm 0.2}$	$2.63_{\pm 0.1}$	$2.55_{\pm 0.1}$	$32_{\pm 2.0}$
gpt-5	99.2	$11_{\pm 0.3}$	$3.00_{\pm 0.1}$	$2.36_{\pm 0.1}$	$29_{\pm 1.3}$
o3-mini	100.0	$2.8_{\pm 0.1}$	$1.38_{\pm 0.0}$	$4.65_{\pm 0.5}$	$12_{\pm 1.7}$
o3	100.0	$5.4_{\pm 0.3}$	$1.77_{\pm 0.1}$	$2.57_{\pm 0.2}$	$22_{\pm 1.9}$
gemini-2.5-flash	95.2	$11_{\pm 0.4}$	$4.05_{\pm 0.3}$	$3.68_{\pm 0.3}$	$28_{\pm 2.7}$
gemini-2.5-pro	99.2	$13_{\pm 0.4}$	$5.18_{\pm 0.3}$	$3.06_{\pm 0.1}$	$22_{\pm 1.7}$

Table H.8: **Model response statistics over FIND development set.** Format validity rate (%), mean tokens per document (in thousands) (kTok/$\mathcal{D}$), mean answers per response ($\hat{\mathcal{I}}/\mathcal{D}$), mean evidence spans per answer ($\varepsilon/\hat{\mathcal{I}}$), and mean tokens per evidence span (Tok/ε). Subscripts denote standard deviations.

	Evidence Score ($\Lambda_{\mathcal{E}}$)						**Description Score (Λ_{δ})**						**Task Score ($\Lambda_{\mathcal{I}}$)**					
Model	**BLS**	**PRE**	**SEC**	**EMM**	**PGS**	**AVG**	**BLS**	**PRE**	**SEC**	**EMM**	**PGS**	**AVG**	**BLS**	**PRE**	**SEC**	**EMM**	**PGS**	**AVG**
sonnet-v4	**35** (**6.9**)	25 (5.4)	20 (4.5)	**19** (**6.0**)	3 (0.7)	**20** (**4.7**)	**44** (**4.1**)	40 (2.4)	34 (3.0)	34 (3.1)	31 (3.2)	36 (2.1)	52 (10.0)	36 (9.6)	38 (9.5)	29 (9.3)	24 (8.5)	36 (4.2)
gpt-5-mini	28 (5.0)	19 (4.3)	22 (4.2)	17 (5.0)	5 (1.8)	**18** (**3.4**)	41 (2.1)	37 (1.8)	38 (2.0)	37 (1.5)	34 (2.0)	38 (1.1)	56 (9.9)	32 (9.3)	58 (9.7)	25 (8.8)	28 (9.0)	40 (6.4)
gpt-5	**37** (**5.1**)	**32** (**6.2**)	**35** (**5.3**)	**24** (**5.5**)	8 (3.1)	**27** (**4.7**)	**46** (**2.3**)	44 (2.8)	**44** (**1.6**)	39 (1.7)	36 (1.8)	**42** (**1.7**)	**76** (**8.5**)	**68** (**9.3**)	**77** (**8.3**)	**38** (**9.9**)	32 (9.3)	**58** (**8.7**)
o3-mini	19 (6.6)	4 (1.0)	0 (0.2)	0 (0.1)	0 (0.1)	5 (3.2)	32 (5.0)	34 (3.1)	24 (3.1)	24 (2.8)	17 (2.7)	26 (2.6)	44 (9.9)	16 (7.3)	8 (5.2)	8 (5.6)	4 (3.9)	16 (6.5)
o3	**35** (**6.0**)	**25** (**5.3**)	17 (3.7)	15 (4.6)	3 (1.2)	**19** (**4.7**)	40 (3.6)	41 (3.2)	39 (1.5)	33 (2.5)	28 (2.1)	36 (2.2)	**68** (**9.3**)	**64** (**9.6**)	50 (9.8)	25 (8.8)	8 (5.4)	43 (10.3)
gemini-2.5-flash	**34** (**5.8**)	20 (5.2)	19 (4.4)	17 (4.8)	2 (0.8)	**18** (**4.5**)	**47** (**3.2**)	42 (2.1)	**38** (**3.0**)	**39** (**2.3**)	34 (2.5)	**40** (**1.9**)	56 (9.9)	40 (9.8)	50 (9.8)	21 (8.3)	16 (7.3)	37 (7.0)
gemini-2.5-pro	**40** (**6.2**)	**35** (**5.1**)	**31** (**4.9**)	**25** (**5.9**)	**18** (**4.3**)	**30** (**3.4**)	**49** (**2.7**)	**50** (**2.1**)	**41** (**2.2**)	**43** (**1.8**)	**44** (**1.8**)	**45** (**1.6**)	**68** (**9.3**)	**76** (**8.5**)	**65** (**9.3**)	**50** (**10.2**)	**56** (**9.9**)	**63** (**4.0**)

Table H.9: Evidence, description, and task scores (with standard errors) across the **FIND** development set. The task score functions like recall for the inserted inconsistency across the models' predicted inconsistencies. Scores are presented as percents (out of 100).

I Prompts

We list the user, system, and grading prompt below.

User Prompt

````
Here is a new example for you to help me with.
```
<document>
{{problem}}
</document>
```
Find any contradictions or inconsistencies. Start by reasoning within <think> and </think>
    ↪ tags, then return any answers within <answer> and </answer> tags.
````

System Prompt

````
I need to check this one document over and make sure there are no semantic errors like
    ↪ inconsistent information or contradicting facts. This really could be anything but not
    ↪ grammar. I can use a spellchecker for that later. For example, I want to make sure that
    ↪  the analysis is correct. The logic is correct. That type of thing.

Rules:
* Format your final answer within <answer> and </answer> tags
* Do NOT include comments within or outside of the tags
* Return the meaningful inconsistencies that matter and in the other that you find within the
    ↪ document. Track only the minimal information needed.
* Ignore spelling and grammar errors.
* Do not repeat the same basic inconsistency in your list of answers.
* Every answer within <answer> will be graded so only put your absolutely final and best
    ↪ results within those tags.

```
<think>...</think>
# first inconsistency
<answer>
<evidence>...</evidence>
<evidence>...</evidence>
<description>...</description>
</answer>
<think>...</think>
# second inconsistency
<answer>
<evidence>...</evidence>
<evidence>...</evidence>
<description>...</description>
</answer>
# etc.
```

First think through any potential error or things you need to check within <think> and </think
    ↪ > tags before giving your answer. Try to limit your thinking just to what you actually
    ↪ need. Format your outputs in the answer structure as shown in the examples below.

# Annotation Definitions

For each contradiction/inconsistency we want to collect the contradictory information present
    ↪ in the document, a description of the inconsistency, and the answers to two flags.

> trigger
Find the trigger first. The trigger is the first place top-to-bottom where the error becomes
    ↪ apparent. The rest of the needed evidence should appear in the document before the
    ↪ trigger. Note that the trigger is not itself always wrong, just the place where it is
    ↪ clear that there is now an error. (The trigger is part of the evidence, which we define
    ↪  next. You don't need to label it separately, it'll just always be the last piece of
    ↪ evidence.)

> evidence
Collect the minimal set of spans including the trigger that identify the error. This will
    ↪ generally be two spans, but could be many more, especially when we find errors in
````

```
    ↪ tables. Evidence spans must appear verbatim or else they will be wrong. Present the
    ↪ evidence in order of appearance within the document.

> description
Lastly, include a brief and complete description of why and how the spans form an
    ↪ inconsistency. The description should not include how to fix the contradiction. There
    ↪ will always be more than one way to fix the contradiction. However, showing exactly how
    ↪  and where the error arises would be helpful.

# Annotation Process

1. Find the trigger span. Call the position in the document of the trigger span *j*.
2. Now find the spans that the trigger contradicts, which we call the evidence. Include only
    ↪ evidence that is higher up in the document, with positions i < j.
3. Provide the description of why the spans are inconsistent.

# Some additional notes
- Within the description describe any key structural or background information needed to
    ↪ identify the error. Structural information could include patterns of bolding within a
    ↪ table (always max, or always best.) Background information could include some finance
    ↪ knowledge.

# What if there are two (or more) possible sets of evidence that end with the same trigger?
Pick the set of evidence closest to the trigger. Start by comparing spans of evidence from
    ↪ each candidate set furthest away (nearest the top of the document) and pick the set
    ↪ with the first span closer to the trigger.

- If candidates share the first k spans then compare the first differing span between sets and
    ↪  pick the set with the one closer to the trigger.
- If one set of evidence is a strict subset of another, choose the subset.

# What about meta errors?
- If information is expected to appear (for example in a footnote or appendix) but is missing,
    ↪  describe the problem. Do not attempt to show the absence of missing information. Only
    ↪ highlight the claim as evidence.
- If information is unexpected, then highlight why there is that expectation and the
    ↪ triggering span that violates the expectation.

# Example 1
<document>
Table 1
|  | $ |
|---|---|
| Val Ya | -1 |
| Val Yb | 0 |
| Val Yc | 3 |
| Val Y | 20 |

Val Y is the total of Ya Yb and Yc

{10 pages of text}

Table 2
|  | $ |
|---|---|
| Val X | 1 |
| Val Y | 2 |
| Val Z | 3 |
| Total | 6 |
</document>
<think>Note that the error is only within the first table. The trigger is the caption
    ↪ statement; from that point there is clearly a contradiction in the document. The values
    ↪  within the rows do not sum up to the total.</think>
<answer>
<evidence>Val Ya</evidence><evidence>-1</evidence>
<evidence>Val Yb</evidence><evidence>0</evidence>
<evidence>Val Yc</evidence><evidence>3</evidence>
<evidence>Val Y</evidence><evidence>20</evidence>
```

```
<evidence>Val Y is the total of Ya Yb and Yc</evidence>
<description>Val Y is reported to sum up to the value (-1) + 0 + 3 = 2 but is reported as
    ↪ having the value 20.</description>
</answer>


# Example 2
‘‘‘
<document>
\newcommand{ \yes}[0]{$ \color{violet} \checkmark$}
\newcommand{ \nah}[0]{$ \color{red}$}


\begin{table*}[t]
\small \centering
\begin{tabular}{l|rrr|ccc}
\toprule
{ \bf Dataset} & { \bf \#Docs} & { \bf \#QAs} & { \bf \#Words} & { \bf Multi-page} & { \bf
    ↪ Numeric} & { \bf Tabular} \ \
\midrule
NarrativeQA & 1,572 & 46,765 & 63,000 & \yes & - & - \\
QuALITY & 381 & 6,737 & 5,159 & \yes & - & - \\
PDFTriage & 82 & 908 & 12,000 & \yes & \yes & \yes \\
\midrule
TAT-QA & 2,757 & 16,552 & 260 & - & \yes & \yes \\
FinQA & 2,789 & 8281 & 687 & - & \yes & \yes \\
\midrule
DocFinQA & 801 & 7,437 & 123453 & \yes & \yes & \yes \\
\bottomrule
\end{tabular}
\caption{Comprison of DocFinQA and existing Finance QA and Long Document QA dataset. DocFinQA
    ↪ includes { \bf multi-page} documents with both { \bf numeric} and { \bf tabular} data.}
\label{tab:dataset}
\end{table*}
</document>
<think>There do not appear to be any errors. </think>
<answer></answer>
‘‘‘

# Example 3
‘‘‘
<document>
\begin{center}
\begin{tabular}{||c | c c c||}
 \hline
 Col1 & Col2 & Col2 & Col3 \\ [0.5ex]
 \hline\hline
 1 & 6 & 7837 & 787 \\
 \hline
 2 & 7 & 78 & 5415 \\
 \hline
 3 & \textbf{545} & 778 & \textbf{7507} \\
 \hline
 4 & \textbf{545} & \textbf{18744} & 7560 \\
 \hline
 5 & 88 & 788 & 9344 \\ [1ex]
 \hline
\end{tabular}
\end{center}
\label{table:example_9}
</document>
<think>
Each column appears to have the maximum value bolded but there is a mistake: the maximums
    ↪ respectively are 545, 18744, 9344. However, in Col3, 7507 is bolded but 7507 should not
    ↪  be since it is not the maximum in its column. Instead, 9344 should be bolded in Col3.
</think>
<answer>
<evidence>\textbf{7507}</evidence>
```

```
<evidence>9344</evidence>
<description>The wrong cell in Column 4 ('Col3') is bolded because it is not the largest value
    ↪  in that column: 9344 is larger than 7507.</description>
</answer>
```

```
# Example 4
```
<document>

The profit for selling D is low because the materials costs are so high. (See the table.)

Item	In Stock	Price
A	True	23.99
B	True	23.99
C	False	19.99
D	False	42.99

For the first time this year, Item C has the highest Price.

</document>
<answer>
<evidence>profit for selling D is low because the materials costs are so high</evidence>
<evidence>See the table.</evidence>
<description>The table is expected to report on profit or materials costs but this information
    ↪ is missing from the table.</description>
</answer>
<answer>
<evidence>C<evidence>
<evidence>19.99<evidence>
<evidence>D<evidence>
<evidence>42.99<evidence>
<evidence>For the first time this year, Item C has the highest Price.</evidence>
<description>Item C does not have the highest price. For example, Item D has a higher price.</
    ↪ description>
</answer>
```

```
# End of Examples.

Think hard and carefully and give the grade as directed in a compact manner. Feel free to plan
    ↪  and scope out key parts of the document to check and then provide each found
    ↪ inconsistency within answer tags.
Reasoning: high
```

Grading Prompt

```
Your job is to grade a set of results. The task was finding the error in a technical document.
    ↪  The student is meant to provide a clear description of the error and evidence.


Compare the ground truth Gold description and evidence compared to the student's output. Then
    ↪ assign a grade of either CORRECT or INCORRECT for whether the student found the error.
    ↪ If the suggested evidence is wrong or incomplete but still makes clear the student
    ↪ found the error then consider the output CORRECT. If the evidence of description
    ↪ pertains to something else in the document please mark the answer INCORRECT.


First, I will give examples of each grade, and then you will grade a new example.


The following are examples of CORRECT predicted answers.
```
Gold answer:
<evidence>\cellcolor[HTML]{FF6D01}\textbf{71.5}</evidence>
```

```
<description>The color of 71.5 does not match highlighting pattern of other cells; it should
    ↪ be cyan, not orange.</description>

Predicted answer 1:
<evidence>\cellcolor[HTML]{FF6D01}\textbf{71.5}</evidence>
<description>Highlighting 71.5 orange is wrong. It should be cyan.</description>

Predicted answer 2:
<evidence>"\cellcolor[HTML]{FF6D01}\textbf{71.5}</evidence>
<description>71.5 does not match the pattern of highlighting.</description>

Gold answer:
<evidence>\textbf{62.8}</evidence>
<evidence>61.6</evidence>
<description>In Col 6, 61.6 should be bolded, not 62.8.</description>

Predicted Answer 1:
<evidence>61.6</evidence>
<evidence>62.8</evidence>
<description>61.6 bolded, not 62.8.</description>

Predicted Answer 2:
<evidence>61.6</evidence>
<evidence>\textbf{62.8}</evidence>
<description>In Col 6  in Table {tab:rm_results}, 61.6 appears to be the minimum value and
    ↪ because the table is bolding minimums this value should be bolded for Col 6 and not the
    ↪  currently bolded number. Please change the bolding to fix this error.</description>

‘‘‘

These predicted answers are all CORRECT because:
    - They contain the important information in the gold target.
    - They do not contain any information that contradicts the gold target.
    - Only semantic meaning matters; capitalization, punctuation, grammar, and order don’t
    ↪ matter.
    - Hedging and guessing are permissible, provided that the gold target is fully included
    ↪ and the response contains no incorrect information or contradictions.
    - The predicted answers are all clearly talking about the correct target
    - The json formatting does not need to be perfect or compile for the answer to be correct


The following are examples of INCORRECT predicted answers.
‘‘‘
Gold answer:
<evidence>\cellcolor[HTML]{FF6D01}\textbf{71.5}</evidence>
<description>The color of 71.5 does not match highlighting pattern of other cells; it should
    ↪ be cyan, not orange.</description>

Predicted answer 1:
<description>There appears to be no semantic errors.</description>

Predicted answer 2:
<evidence>textbf</evidence>
<description>Color is wrong</description>

Predicted answer 3:
<evidence>\cellcolor[HTML]{FF6D01}\textbf{71.5}</evidence>
<description>Bolding seems wrong in Table {tab:dpo_vs_ppo}</description>

Predicted answer 4:
{}
‘‘‘
These predicted answers are all INCORRECT because:
    - The predicted answer and gold answer must refer to the same error; in the examples above
    ↪  they do not clearly match.
    - The important information from the gold description is not included in the answer
```

```
Here is a new example. Simply assign a grade of either CORRECT or INCORRECT.
‘‘‘
Original Document That Contains The Error from the Gold Answer:
<document>
{{problem_text}}
</document>

Gold answer:
{{gold_answer}}
Predicted answer:
{{model_answer}}
‘‘‘


Grade the predicted answer of this new question as one of:
A: CORRECT
B: INCORRECT


Just return the letter "[[A]]" or "[[B]]" (corresponding to CORRECT or INCORRECT, respectively
    ↪ ) within answer brackets.
```