February, 2025

**APM_5AI29_TP**

# Language Models and Structured Data

Final Project Report

Acronym of the Team: albapati
Name: Alexandros Slavoudis, Baptiste Geisenberger, Pablo Bertaud-Velten, Tim Luka Horstmann

---

# Table Fact-Checking

**Our custom Table Fact-Checking Explorer:**
https://horstmann.tech/table-fact-checking-project/

## 1 Abstract

Table fact-checking is a critical task in Natural Language Processing (NLP) that requires reasoning across structured and unstructured data. This study explores the effectiveness of various deep learning approaches for table-based fact-checking, using the TabFact dataset as a benchmark. We evaluate existing models such as the Latent Program Algorithm (LPA) and Table-BERT while also introducing custom methodologies based on state-of-the-art large language models (LLMs). These include prompt engineering in zero-shot and chain-of-thought settings, code generation, and retrieval-augmented generation (RAG) techniques. Our experiments demonstrate that modern generative models, particularly reasoning-focused LLMs, achieve near-human performance with an accuracy of close to 90%. These approaches are further analyzed, yielding insights into the challenges encountered by these models. In addition to model development and evaluation, we identify limitations in the TabFact dataset's gold standard annotations, highlighting inconsistencies in label assignments. To enhance transparency and accessibility, we furthermore present the interactive Table Fact-Checking Explorer — a publicly available tool that offers a clear visualization of the dataset, provides claim-level insights into our results, and includes a feature for live table fact-checking with state-of-the-art LLMs.

# Contents


**1 Abstract**

**2 Problem Statement**

**3 Related Work**

**4 Dataset & Dataset Statistics**

**5 Methodology and Model Architectures**

5.1 Latent Program Algorithm (LPA) and Table-BERT . . . . . . . . . . . .

5.2 Prompt Engineering . . . . . . . . . . . . . . . . . . . . . . . . . . . .

5.3 Code Generation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

5.4 Retrieval-Augmented Generation . . . . . . . . . . . . . . . . . . . . .

**6 Experimentation**

6.1 Analysis of Existing Approaches . . . . . . . . . . . . . . . . . . . . . .

6.2 Custom Approaches . . . . . . . . . . . . . . . . . . . . . . . . . . . .

6.2.1 Prompt Engineering . . . . . . . . . . . . . . . . . . . . . . . .

6.2.2 Code Generation . . . . . . . . . . . . . . . . . . . . . . . . . .

6.2.3 Retrieval Augmented Generation . . . . . . . . . . . . . . . . .

6.3 Further Error Analysis . . . . . . . . . . . . . . . . . . . . . . . . . . .

6.4 The Custom Table Fact-Checking Explorer . . . . . . . . . . . . . . . .

**7 Critique of the Gold Standard**

**8 Discussion and Conclusion**

**A Compilation of Results from TabFact Studies**

**B Evaluation of the Table-BERT and LPA models**

**C Error Analysis per Category**

C.1 Explanation of Initial Categories . . . . . . . . . . . . . . . . . . . . . . . .

**D Examples of Gold Standard Errors**

D.1 Negation and Ambiguity Errors . . . . . . . . . . . . . . . . . . . . . . . .

D.2 Domain-Specific Misinterpretations . . . . . . . . . . . . . . . . . . . . . .

D.3 Irrelevant or Misaligned Claims . . . . . . . . . . . . . . . . . . . . . . . .

D.4 Phrasing and Numerical Inconsistencies . . . . . . . . . . . . . . . . . . . .

## 2 Problem Statement

Table fact-checking is a critical task in Natural Language Processing (NLP) that involves verifying textual claims by cross-referencing them with tabular data [2, 18, 9]. This task requires reasoning across both unstructured and structured data, encompassing linguistic reasoning (e.g., semantic understanding) and symbolic reasoning (e.g., numerical and logical operations) [2, 18, 36]. The challenges in this task arise from the inherent differences between unstructured textual claims, which are often open to interpretation, and structured tabular data, which require precise alignment and reasoning [2, 35].

Table fact-checking plays a critical role in combating misinformation, mitigating fake news, and automating fact verification across various industries [19]. However, existing ML models often struggle with the complex reasoning required for this task [19, 9], underscoring the urgent need for robust and scalable solutions [14, 27].

Many (structured) fact-checking approaches are limited by the lack of available data [2, 9]. Hence, a significant milestone in this field was the introduction of the large-scale dataset *TabFact* by Chen et al. [2] in 2020. Sourced from Wikipedia, it serves as a benchmark for table fact-checking, with the authors themselves proposing two models for table fact-checking: *Latent Programming Algorithm (LPA)* and *Table-BERT* [2].

The remainder of this report is structured as follows: Section 3 reviews related work, while Section 4 introduces the TabFact dataset. Section 5 describes both the LPA and Table-BERT models, as well as our own custom approaches leveraging modern generative ML models. In Section 6, we present experimental results, error analysis, and our interactive Table Fact-Checking Explorer. Section 7 provides a critical assessment of the TabFact dataset's gold standard annotations. Finally, we summarize our findings and discuss broader implications in Section 8.

## 3 Related Work

Advances in information extraction (IE) have enabled the derivation of structured and meaningful insights from text, driving the development of table-focused tasks in NLP to address the growing need for reasoning across textual and structured data [8, 20, 28].

Since 2015, several benchmarks have been introduced to evaluate table-focused models. These include WikiTableQuestions [16] (22k questions), FEVER [23] (185k claims), HiTab [4] (10k hierarchical tables), InfoTab [10] (29k claims), and TabFact [2] (118k claims). The latter has become a prominent benchmark for table fact-checking [15].

To address the challenges posed by these datasets, a variety of models have been proposed. In 2019, NumNet [17] was introduced, enhancing numerical reasoning through graph-based encodings. In 2020, Chen et al. [2] proposed two foundational methods: LPA, which employs symbolic reasoning, and Table-BERT, which utilizes a BERT model [6] for joint text-table encoding (Section 5.1). TAPEX [13], learning a neural SQL executor over a synthetic corpus, and DecompTAPAS [31], simplifying complex claims into subclaims for improved accuracy, were both developed in 2021. More recently, in 2022, PASTA [9] applied graph neural networks to model relationships within tables. Today, DATER [33] achieves state-of-the-art performance and exceeds human performance for the first time, leveraging LLMs as decomposers to simplify tables and claims. Many additional methods have been developed. We provide a comprehensive overview of the performance of these methods in Table 5 (Appendix A).

# 4 Dataset & Dataset Statistics

The TabFact dataset created by Chen et al. [2] includes 16k Wikipedia tables filtered to remain relatively simple in content and small in size, as well as 118k human-annotated statements labeled *entailed* or *refuted* by the corresponding tables[1]. It is the first dataset to evaluate language inference on structured data, which involves mixed reasoning skills in both symbolic and linguistic aspects [2]. TabFact includes simple and complex claims, where simple claims reference single table records, while complex claims require multi-row reasoning, higher-order operations (e.g., *argmax*, *count*, *average*), and deeper semantic understanding. Table 1 summarizes the basic statistics of the TabFact dataset.

| Channel | #Sent. | #Table | Len(Ent) | Len(Ref) | Split | #Sent. | #Table | Row | Col |
|---|---|---|---|---|---|---|---|---|---|
| **Simple** | 50,244 | 9,189 | 13.2 | 13.1 | **Train** | 92,283 | 13,182 | 14.1 | 5.5 |
| **Complex** | 68,031 | 7,392 | 14.2 | 14.2 | **Val** | 12,792 | 1,696 | 14.0 | 5.4 |
| **Total** | 118,275 | 16,573 | 13.8 | 13.8 | **Test** | 12,779 | 1,695 | 14.2 | 5.4 |

Table 1: Basic statistics of the TabFact dataset (reproduced from Chen et al. [2]).

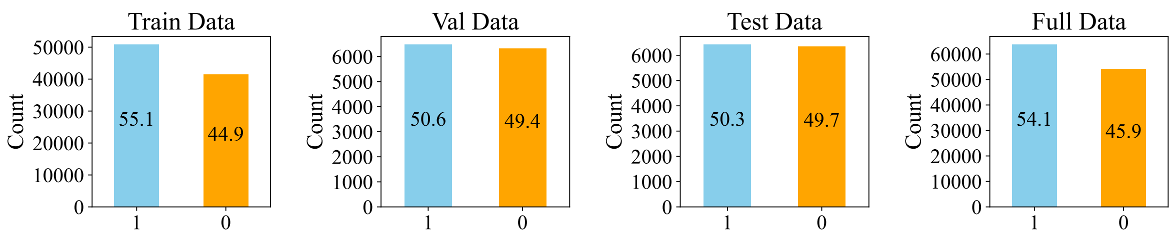



Figure 1: Label distribution across train, validation, test, and full datasets, showing class counts and relative distribution. Train and full datasets exhibit slight class imbalance.

Our data exploration phase confirms a fairly balanced distribution of entailed (1) and refuted (0) labels, with a slight skew towards entailed sentences, primarily noticeable in the train split (Figure 1). To prevent shallow models (e.g. bag-of-words) from achieving high performance, Chen et al. [2] ensured similar average lengths for simple and complex samples and removed obvious linguistic patterns for negations (Table 1). The authors further claim the data to have strong inter-annotator agreement (Fleiss $\kappa = 0.75$) and high data quality [2]. We provide an additional analysis of statement length and unique word counts in Figure 2, which further confirms a well-balanced dataset.

[1]An interface to explore the data is provided at `https://tabfact.github.io/explore.html`.

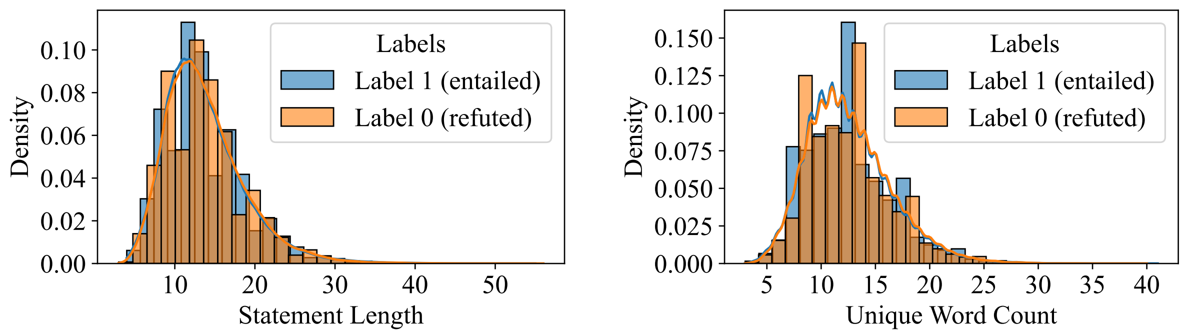



Figure 2: Comparison of distributions for (a) statement lengths and (b) unique word counts by label in the TabFact dataset. The classes show a balanced distribution.

# 5 Methodology and Model Architectures

Building on the TabFact dataset presented in Section 4, this section briefly describes the two models for table fact-checking presented by Chen et al. [2] (Section 5.1) as well as our own LLM-based approaches to table fact-checking (Section 5.2 - Section 5.4). All our models and experiments were run on NVIDIA RTX A5000 GPUs with 24 GB of RAM.

## 5.1 Latent Program Algorithm (LPA) and Table-BERT

The LPA model formulates table fact-checking as a weakly supervised program synthesis task [2]. It comprises a latent program search phase, which constructs candidate programs using predefined APIs (e.g., count, max, filter), and a discriminator that ranks the most consistent program. While LPA provides strong symbolic reasoning and interpretability, it depends on accurate entity linking and may yield spurious programs. In contrast, Table-BERT treats the task as a Natural Language Inference (NLI) problem, encoding both the statement and table as sequences using BERT [2, 6]. It employs two encoding strategies: direct concatenation with [SEP] tokens and natural language templates, i.e. *naturalization*. The Table-BERT approach improves linguistic reasoning but struggles with symbolic tasks and is constrained by BERT's sequence length [6]. Both models score similarly, with LPA achieving 65.2% accuracy and Table-BERT 66.1% [2].

## 5.2 Prompt Engineering

Building on recent advancements and the strong performance of modern generative models [21], our first custom approach leverages prompt engineering to address table fact-checking with LLMs. Specifically, we employ state-of-the-art LLMs available via *Ollama*[2], selected to align with our hardware constraints: *Llama3.2 (3B)*, *Mistral (7B)*, *Phi4 (14B)*, *DeepSeek-R1 (14B)*, and *DeepSeek-R1 (32B)*. This selection enables us to

[2] `https://ollama.com/`

assess the performance of modern LLMs while also comparing direct inference models with the speculative decoding architecture of the DeepSeek series [5].

To evaluate the impact of different table encoding formats on model performance, we follow prior research in providing the LLMs with tables either in *HTML*, *JSON*, *markdown*, or *naturalized* form, where tables are described in natural language by converting each row into a descriptive sentence that outlines the column names and corresponding values [11, 1]. Furthermore, we implement different learning/prompt strategies, *zero shot* and *chain of thought* (CoT), where the latter encourages the LLMs to "think" before delivering its response [26]. To streamline the extraction and evaluation process, our approach guides the model to output a strictly formatted JSON object containing the final verdict of the claim verification against the table, i.e. *TRUE* in case of an entailed claim, *FALSE* otherwise, along with the relevant cell locations in the table. Robust post-processing is introduced to ensure accurate answer extraction.

## 5.3 Code Generation

Our second approach seeks to fact-check the claim using code generated by an LLM. We explicitly prompt the LLM to write code, providing it with the claim and table (in JSON format), as well as the column data types. Subsequently, we extract the code from the LLM's output, and execute it. The generated code simply outputs a boolean, i.e. the verdict of the fact-checking process, for the given claim. Similar to the prompt engineering approach, we employ *Llama3.2 (3B)*, *Mistral (7B)*, *Phi4 (14B)*.

Our code generation approach aims to answer the question of whether the LLM might be more effective at generating code — particularly in well-established languages like Python's *Panda*s library or SQL (using Python's *SQLite* library) — rather than directly reasoning over a table, having to identify relevant rows and columns, and handling different table formats. We also seek to evaluate the performance of the code generation in Python compared to SQL.

## 5.4 Retrieval-Augmented Generation

Our final approach leverages retrieval-augmented generation (RAG) techniques to enhance table fact-checking. In this method, we embed both the claim and the table using *nomic-embed-text* embeddings[3] and perform a similarity search based on cosine similarity. We evaluate three different strategies for embedding the table: (1) embedding each row individually, (2) embedding both rows and columns, and (3) embedding each cell separately. In a sense, we treat the table as a collection of smaller units, retrieving the most relevant segments rather than considering the table as a whole. Once retrieved, these segments are naturalized and passed to the language model for the final fact-verification. Similar to the other approaches, we employ *Llama3.2 (3B)*, *Mistral (7B)*, and *Phi4 (14B)*.

[3] `https://ollama.com/library/nomic-embed-text`

# 6 Experimentation

This section evaluates table fact-checking approaches on the test set of the TabFact dataset, which includes 1,695 tables and 12,779 claims, nearly 70% of which are considered complex. We first analyze the performance of the models from Chen et al. [2] (Section 6.1) before presenting results from our custom approaches (Section 6.2). A comprehensive comparison of TabFact studies, including our findings, is provided in Table 5 (Appendix A).

## 6.1 Analysis of Existing Approaches

To establish a baseline, we evaluated the two primary models introduced by Chen et al. [2]: LPA and Table-BERT. Using the authors' checkpoints[4], we achieved accuracy scores on the test set similar to the reported ones (65.1% for LPA and 64.8% for Table-BERT).

**Performance Trade-offs**
Going beyond the original work by Chen et al. [2], we analyzed additional metrics during model evaluation to offer deeper insights into the performance of their models. Our analysis highlights the complementary strengths of both models. Table-BERT excels in recall (75.1%), making it more effective for identifying valid claims, while LPA maintains a slightly higher precision (63%), making it more reliable in rejecting false claims. This trade-off is further illustrated in Appendix B in the Precision-Recall (PR) curve (Figure 5a) and the ROC curve (Figure 5b), where Table-BERT shows superior overall classification performance with an AUC of 0.73 vs. 0.58 for LPA.

**Threshold-Based Evaluation**
Further analysis of threshold adjustments (Figure 6, Appendix B) reveals how performance metrics shift. Table-BERT benefits from lower thresholds, improving recall at the cost of precision, while LPA remains conservative, sacrificing recall for fewer false positives. These insights indicate the potential for ensemble strategies or task-specific threshold tuning to balance the strengths of both approaches.

Overall, our experiments reinforce the findings by Chen et al. [2] regarding the complementary nature of the two approaches. Table-BERT's semantic reasoning framework is ideal for complex linguistic claims, while LPA's symbolic reasoning offers reliability in precision-critical scenarios. These complementary strengths suggest that combining the models or tuning their thresholds could optimize performance for specific use cases [2].

## 6.2 Custom Approaches

In this section, we evaluate the performance of our custom approaches — prompt engineering (Section 6.2.1), code generation (Section 6.2.2), and RAG (Section 6.2.3) — by analyzing their accuracy, strengths, and limitations.

[4] `https://github.com/wenhuchen/Table-Fact-Checking`

### 6.2.1 Prompt Engineering

To assess our prompt engineering approach (Section 5.2), we evaluated the four implemented table formatting settings across the five introduced LLMs, yielding 20 distinct test settings. In contrast to the other models, DeepSeek-R1 32B was tested on about half of the test set due to slower inference speeds. In total, we evaluated over 240,000 claims/prompts for the prompt engineering approach alone. The results of these experiments are summarized in Table 2.

Our analysis reveals three key findings. First, there is a clear performance hierarchy: general-purpose models like Llama3.2 and Mistral achieved accuracies between 0.56 and 0.63, whereas reasoning-focused models such as DeepSeek and Phi4 achieved significantly higher scores, with Phi4 achieving up to 0.89 accuracy (close to human performance of 0.92) using the naturalized format — a nearly 60% improvement over Llama3.2's results.

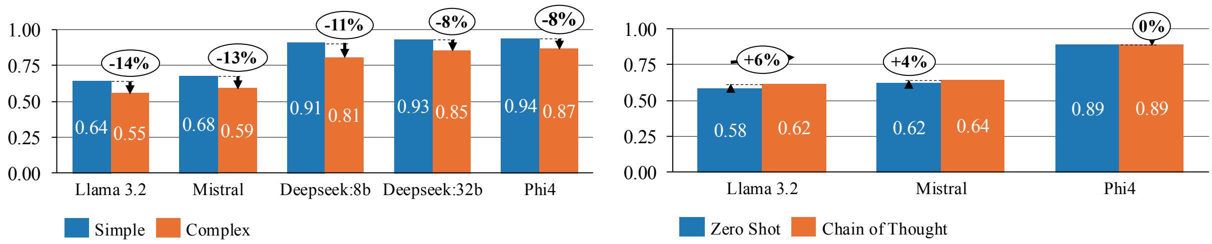
Second, the table format influences performance differently across models. For DeepSeek and Phi4, markdown and naturalized formats consistently yielded higher scores compared to HTML and JSON, suggesting that these representations provide clearer structural cues. In contrast, Llama3.2 and Mistral show less variation across formats.

Finally, it is worth noting that the higher-performing models exhibited a more balanced precision and recall, indicating both accurate and comprehensive fact-checking. Lower-performing models, however, tend to have larger discrepancies between these measures.

| Model | Table Format | Accuracy | Precision | Recall | F1 | # Tables | # Claims |
|---|---|---|---|---|---|---|---|
| Llama3.2 | html | 0.565 | 0.609 | 0.378 | 0.467 | 1,695 | 12,779 |
| | json | 0.556 | 0.601 | 0.349 | 0.442 | 1,695 | 12,779 |
| | markdown | 0.592 | 0.632 | 0.453 | 0.528 | 1,695 | 12,779 |
| | naturalized | 0.582 | 0.644 | 0.376 | 0.475 | 1,695 | 12,779 |
| Mistral | html | 0.629 | 0.712 | 0.439 | 0.543 | 1,695 | 12,779 |
| | json | 0.631 | 0.685 | 0.492 | 0.572 | 1,695 | 12,779 |
| | markdown | 0.630 | 0.697 | 0.468 | 0.560 | 1,695 | 12,779 |
| | naturalized | 0.620 | 0.718 | 0.401 | 0.515 | 1,695 | 12,779 |
| Deepseek-r1:8b | html | 0.785 | 0.844 | 0.701 | 0.766 | 1,695 | 12,779 |
| | json | 0.798 | 0.864 | 0.710 | 0.779 | 1,695 | 12,779 |
| | markdown | 0.831 | 0.886 | 0.761 | 0.819 | 1,695 | 12,779 |
| | naturalized | 0.840 | 0.887 | 0.781 | 0.831 | 1,695 | 12,779 |
| Deepseek-r1:32b | html | 0.806 | 0.883 | 0.706 | 0.784 | 951 | 9,019 |
| | json | 0.834 | 0.907 | 0.743 | 0.817 | 960 | 9,058 |
| | markdown | 0.856 | 0.913 | 0.788 | 0.846 | 967 | 9,109 |
| | naturalized | 0.858 | **0.916** | 0.790 | 0.848 | 973 | 9,138 |
| Phi4 | html | 0.863 | 0.885 | 0.836 | 0.860 | 1,695 | 12,779 |
| | json | 0.872 | 0.886 | 0.856 | 0.871 | 1,695 | 12,779 |
| | markdown | 0.886 | 0.906 | 0.862 | 0.884 | 1,695 | 12,779 |
| | naturalized | **0.889** | 0.909 | **0.866** | **0.887** | 1,695 | 12,779 |

Table 2: Results of our prompt engineering approaches by model and table input format. Phi4 emerged as the best performing model, *naturalized* as a good table input format.

Building on the insights from Table 2, we further analyzed the impact of table complexity and prompt strategies using the naturalized table format. Figure 3a reveals that performance dropped for all models when handling more complex verification tasks, but the extent varies. Llama3.2, Mistral, and DeepSeek-8b exhibited the most significant declines (-14% to -11%), whereas DeepSeek-32b and Phi4 were less affected (-8%). This suggests that advanced reasoning models are more resilient to increased complexity. Figure 3b compares zero shot vs. CoT prompting. While Llama3.2 and Mistral improved with CoT (+4% and +6%, respectively), Phi4's performance gain was negligible.




(a) Performance on simple vs. complex pairs of the test set as categorized by Chen et al. [2].

(b) Performance of the direct inference models in a zero shot vs. chain of thought setting.

Figure 3: Prompt engineering performance across simple/complex and zero shot/CoT settings. Phi4 exhibited robust performance while other LLMs were affected more strongly.

Our results align well with prior findings that CoT prompting can improve model performance but also that reasoning-oriented models seem to already excel in zero shot settings and do not benefit from additional reasoning steps [26, 5].

### 6.2.2 Code Generation

We evaluated the performance of our code generation approach by running various models on the two different languages generated: Python and SQL. The models tested on the entire dataset were Llama3.2 (3B), Mistral (7B) and Phi4 (14B). DeepSeek-R1 (14B) was tested on a subset of around 7% of the data. Results are shown in Table 3.

| Model | Language generated | Accuracy | Precision | Recall | F1 | # Claims |
|---|---|---|---|---|---|---|
| Phi4 | Python | **0.819** | **0.856** | **0.768** | **0.810** | 12779 |
| | SQL | 0.704 | 0.788 | 0.5617 | 0.656 | 12779 |
| DeepSeek-R1:14b | Python | - | - | - | - | - |
| | SQL | 0.681 | 0.809 | 0.458 | 0.585 | 1000 |
| Mistral | Python | 0.658 | 0.762 | 0.469 | 0.580 | 12779 |
| | SQL | 0.560 | 0.700 | 0.2204 | 0.335 | 12779 |
| Llama3.2 | Python | 0.610 | 0.672 | 0.452 | 0.541 | 12779 |
| | SQL | - | - | - | - | - |

Table 3: Performance of our code generation approaches across different models and languages. Python-based generation consistently outperforms SQL across all models, with Phi4 achieving the highest accuracy and the most balanced precision-recall tradeoff.

We note that recall tends to be lower than precision because when an LLM fails to generate a response, we default to *False*, leading to an overrepresentation of negative predictions. As a result, fewer entailed claims are correctly identified, lowering recall. Notably, Phi4 is the only model with a balance of precision and recall that is almost even. This indicates that Phi4 is not only capable of generating accurate code but also maintains a consistent ability to verify both entailed and refuted claims effectively.

Another significant finding is that Python code generation consistently outperforms SQL generation across all models. For instance, Phi4's accuracy drops by approximately 14% when generating SQL compared to Python. This discrepancy may stem from Python's greater flexibility — the *Pandas* library supports complex operations more effectively than SQL, which may struggle with intricate table structures and logical conditions. Additionally, since column data types were passed in Python, this may have further favored Python-based execution.

### 6.2.3 Retrieval Augmented Generation

We evaluated our three table embedding approaches using Llama3.2 (3B) and Mistral (7B) on 40% of the dataset and Phi4 (14B) on 16% of the dataset. As we can see in Table 4, Phi4, as a stronger model, scores the highest, while Mistral generally outperforms Llama3.2. Phi4's strong performance indicates that the retrieval of the relevant context of the table is accurate. This also suggests that the low scores of the general purpose models, Llama 3.2 and Mistral stem from their limited reasoning capacity.

We further observe that Mistral scores better with cell-wise embeddings, while Phi4 performs notably better with the row and row+column embeddings. Llama3.2 shows similar results across all embedding strategies.

| Model | Table Embeddings | Accuracy | Precision | Recall | F1 | #Claims |
|---|---|---|---|---|---|---|
| Llama3.2 | Rows | **0.591** | 0.613 | 0.492 | 0.546 | 4993 |
| | Rows and Columns | 0.589 | 0.608 | 0.501 | 0.549 | 4993 |
| | Cells | 0.580 | 0.604 | 0.458 | 0.521 | 4993 |
| Mistral | Rows | 0.607 | 0.606 | 0.605 | 0.606 | 4993 |
| | Rows and Columns | 0.596 | 0.587 | 0.640 | 0.613 | 4993 |
| | Cells | **0.623** | 0.685 | 0.455 | 0.547 | 4993 |
| Phi4 | Rows | 0.847 | 0.830 | 0.867 | 0.849 | 1998 |
| | Rows and Columns | **0.857** | 0.851 | 0.862 | 0.857 | 1998 |
| | Cells | 0.770 | 0.802 | 0.712 | 0.754 | 1998 |

Table 4: Results of our RAG approaches by model and embedding type. Phi4 achieves the highest accuracy, particularly with row and row+column embeddings, while Mistral performs best with cell-wise embeddings, and Llama3.2 shows consistent results across all strategies.

Comparing these results with the regular prompt engineering in Table 2, there is no significant evidence of performance gains through filtering the table through RAG. For Llama3.2, accuracy is slightly better while for Mistral and Phi4 it is slightly worse. This suggests that the LLMs have no trouble processing the entire table. Instead performance slightly drops after RAG-filtering — possibly because of the risk of not always retrieving the necessary, i.e. relevant, information.

## 6.3 Further Error Analysis

To go beyond a simple quantitative performance analysis and gain deeper insights into the limitations of our models, we analyzed failure cases to identify areas for improvement. We categorized claims using predefined criteria, leveraging Mistral (7B) to efficiently process and classify them. Custom prompts were designed for each claim-category pair, allowing our agent to accurately assign claims to their respective categories.

The initial set of categories, derived from Chen et al. [2], included *aggregation, negation, superlative, comparative, ordinal, unique, all,* and *none*, as detailed in C.1. While these categories were useful for comparing LPA and Table-Bert, they offered limited insights in our context, as the accuracy for each category closely mirrored the global accuracy of the models 8.

As a result, we refined the categorization through an extensive manual analysis and iterative testing, leading to the creation of new, more relevant categories based on the observed difficulties encountered by the LLMs. The categories identified were as follows:

- **Unclear or Noisy Language:** This category includes claims with noisy language. It spans from granular issues like spelling or heavy grammatical errors to broader contextual ambiguities where the claim itself is unclear in meaning.
- **Numerical Reasoning:** This category includes claims involving complex numerical operations, requiring precise or approximate numerical results.
- **Multistep Logic:** This category includes claims that require multiple logical steps or stages to reach a conclusion, often involving complex reasoning and integration of information across different parts of the claim.
- **Negation:** This category includes claims with negations that can lead to incorrect interpretations. Claims with double-negations are not uncommon and can complicate the model's understanding by flipping the meaning of statements.

These categories established, we computed the misclassification rates for each category using the model outputs on the categorized claims. Figure 4 presents the misclassification rates of the best-performing models across categories.

We first observe that claims falling under a category are more likely to be misclassified than those that do not, highlighting the importance of our categorization. Notably, Phi4 with a naturalized language table performs better in verifying claims with unclear or noisy language and negation when using zero-shot prompting compared to chain-of-thought prompting. Additionally, claims involving negation achieve the lowest misclassification rates with a markdown table format and zero-shot prompting. Although these differences remain small, Figure 9 highlights a more pronounced gap in the code generation approach: using DeepSeek with SQL generation, zero-shot prompting, and a JSON-formatted table reduces the error rate by nearly 3% compared to Phi4 with SQL generation, zero-shot prompting, and a markdown-formatted table — despite the latter achieving an overall accuracy more than 2% higher.

These findings suggest a potential approach where specific claims could be routed to the models that perform best for their respective categories. However, the fact that the best models identified through prompt engineering exhibit highly similar performance across all categories, coupled with the presence of randomness and occasional mislabeling

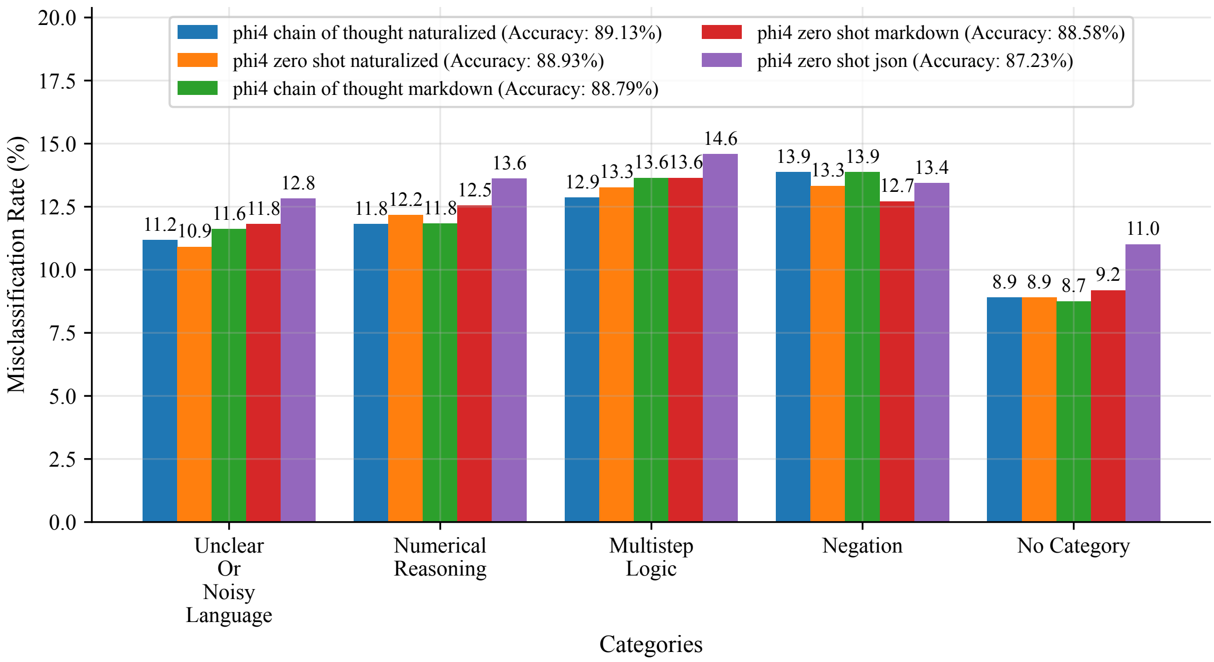



Figure 4: Misclassification rate of best performing models per category

in the ground truth data (see Section 7) raises concerns about the generalizability and applicability of this method in our case.

### 6.4 The Custom Table Fact-Checking Explorer

Alongside our model development, we created an interactive web tool — the *Table Fact-Checking Explorer* — publicly available at `https://horstmann.tech/table-fact-checking-project/`. This tool enables users to interactively explore model results, analyze predictions at the claim level, and perform live table fact-checking using state-of-the-art inference models like DeepSeek running locally. We designed it to enhance transparency, improve interpretability, and serve as a valuable resource for further analysis.

## 7 Critique of the Gold Standard

While the TabFact dataset by Chen et al. [2] represents a large-scale and generally robust benchmark that has significantly contributed to table fact-checking research, our analysis reveals multiple cases where the gold standard labels appear flawed. These inconsistencies raise concerns about the labelling process and its impact on model evaluation. We observed four recurring issues: (1) *Negation and ambiguity errors*, where incorrect handling of (double) negations led to false *refuted* labels; (2) *Domain-specific misinterpretations*, such as in golf, where lower scores indicate better performance but were mislabelled; (3) *Irrelevant or misaligned claims*, where some claims appear to be associated with incorrect tables; and (4) *Phrasing and numerical inconsistencies*, where minor linguistic changes or numerical misinterpretations caused incorrect gold labels.

These errors suggest that reported model accuracies — including our Phi-based prompt engineering approach, which achieved nearly 90% — may in fact be underestimated, as some flagged "errors" may be correct predictions. It is worth noting that Chen et al. [2] employed multiple quality control measures. However, our findings indicate that additional expert review and cross-validation could further enhance the dataset's reliability. Since our error identification was manual, we cannot rule out the existence of further mislabeled cases or additional error categories. We provide concrete examples for the above error categories in Appendix D.

# 8 Discussion and Conclusion

In this study, we explored the NLP task of table fact-checking. We find this area of research compelling due to the challenges of integrating unstructured and structured data for prediction, with a wide range of impactful applications.

We first navigated the complex landscape of table fact-checking, building on the TabFact dataset and the models developed by Chen et al. [2]. After having examined the multitude of techniques refined since then, we derived our custom comprehensive LLM-based set of approaches, presenting three main strategies: prompt engineering, code generation, and RAG. These approaches were evaluated through multiple variations in LLM-choice, table formatting, and prompting techniques. Our prompt engineering method, using Phi4 with chain-of-thought prompting and a naturalized table format, achieved the highest accuracy of 89.13% on the TabFact test set. It is also important to note that all three directions demonstrated strong performance. We further deepened our comprehension of the model's performance through an extensive category-based error analysis.

This project yielded valuable insights into both the potential and challenges of NLP models. We demonstrated the impressive capabilities of state-of-the-art generative models like Phi4 and DeepSeek, which achieved near-human performance (∼92%) in these kind of reasoning tasks. However, during this work, we also faced significant challenges, including outdated repositories, irreproducible results, and hardware limitations, highlighting the need for adaptability and collaboration in modern ML research. For example, we rebuilt the LPA and Table-BERT pipeline to replicate Chen et al. [2] results and gain additional insights. Attempts to reproduce *TAPEX* [13] and *PASTA* [9] were hindered by GPU constraints and corrupted data. Moreover, our evaluation of the TabFact dataset's gold standard annotations uncovered inconsistencies, emphasizing the critical importance of rigorous annotation processes as well as questioning gold standards.

We note several avenues for future work. One is fine-tuning LLMs specifically for this task to improve their management of tabular data. Another involves implementing claim decomposition to better guide the models in their reasoning, possibily combined with a hybrid multi-agent approach that builds on our category-based analysis to identify strengths and delegate tasks accordingly. Finally, our Custom Table Fact-Checking Explorer emphasizes the importance of promoting collaboration and popularizing these techniques to the public. Leveraging the efficiency and lightweight nature of these models for widespread fact-checking offers a valuable and impactful application.

# References


[1] Wenhu Chen. *wenhuchen/Table-Fact-Checking*. original-date: 2019-03-13T06:14:32Z. Jan. 2025. URL: https://github.com/wenhuchen/Table-Fact-Checking (visited on 01/05/2025).

[2] Wenhu Chen et al. "TabFact: A Large-scale Dataset for Table-based Fact Verification". In: Addis Ababa, Ethiopia, Apr. 2020. URL: https://openreview.net/forum?id=rkeJRhNYDH (visited on 12/01/2024).

[3] Zhoujun Cheng et al. *Binding Language Models in Symbolic Languages*. Mar. 2023. DOI: 10.48550/arXiv.2210.02875. (Visited on 01/05/2025).

[4] Zhoujun Cheng et al. *HiTab: A Hierarchical Table Dataset for Question Answering and Natural Language Generation*. arXiv:2108.06712 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2108.06712. URL: http://arxiv.org/abs/2108.06712 (visited on 01/05/2025).

[5] DeepSeek-AI et al. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. arXiv:2501.12948 [cs]. Jan. 2025. DOI: 10.48550/arXiv.2501.12948. URL: http://arxiv.org/abs/2501.12948 (visited on 02/17/2025).

[6] Jacob Devlin et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805 [cs]. May 2019. DOI: 10.48550/arXiv.1810.04805. URL: http://arxiv.org/abs/1810.04805 (visited on 01/05/2025).

[7] Julian Martin Eisenschlos, Syrine Krichene, and Thomas Müller. *Understanding tables with intermediate pre-training*. arXiv:2010.00571 [cs] version: 2. Oct. 2020. DOI: 10.48550/arXiv.2010.00571. URL: http://arxiv.org/abs/2010.00571 (visited on 01/05/2025).

[8] Vidhya Govindaraju, Ce Zhang, and Christopher Ré. "Understanding Tables in Context Using Standard NLP Toolkits". In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Ed. by Hinrich Schuetze, Pascale Fung, and Massimo Poesio. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 658–664. URL: https://aclanthology.org/P13-2116/ (visited on 01/06/2025).

[9] Zihui Gu et al. "PASTA: Table-Operations Aware Fact Verification via Sentence-Table Cloze Pre-training". In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. Ed. by Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics, Dec. 2022, pp. 4971–4983. DOI: 10.18653/v1/2022.emnlp-main.331. URL: https://aclanthology.org/2022.emnlp-main.331 (visited on 12/29/2024).

[10] Vivek Gupta et al. "INFOTABS: Inference on Tables as Semi-structured Data". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 2309–2324. DOI: 10.18653/v1/2020.acl-main.210. URL: https://aclanthology.org/2020.acl-main.210/ (visited on 01/05/2025).

[11] Stefan Hegselmann et al. "TabLLM: Few-shot Classification of Tabular Data with Large Language Models". en. In: *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*. ISSN: 2640-3498. PMLR, Apr. 2023, pp. 5549–5581. URL: https://proceedings.mlr.press/v206/hegselmann23a.html (visited on 01/06/2025).

[12] Atin Sakkeer Hussain. *Advanced Reasoning and Transformation Engine for Multi-Step Insight Synthesis in Data Analytics with Large Language Models*. arXiv:2412.14146 [cs] version: 2. Dec. 2024. DOI: 10.48550/arXiv.2412.14146. (Visited on 01/05/2025).

[13] Qian Liu et al. *TAPEX: Table Pre-training via Learning a Neural SQL Executor*. arXiv:2107.07653 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2107.07653. (Visited on 12/26/2024).

[14] Mark Stencel Luther Joel. *Fact-checking count tops 300 for the first time*. en-US. Oct. 2020. URL: https://reporterslab.org/fact-checking-count-tops-300-for-the-first-time/ (visited on 01/05/2025).

[15] Meta AI. *TabFact Benchmark (Table-based Fact Verification)*. URL: https://paperswithcode.com/sota/table-based-fact-verification-on-tabfact?p=pasta-table-operations-aware-fact (visited on 01/03/2025).

[16] Panupong Pasupat and Percy Liang. *Compositional Semantic Parsing on Semi-Structured Tables*. arXiv:1508.00305 [cs] version: 1. Aug. 2015. DOI: 10.48550/arXiv.1508.00305. URL: http://arxiv.org/abs/1508.00305 (visited on 01/05/2025).

[17] Qiu Ran et al. *NumNet: Machine Reading Comprehension with Numerical Reasoning*. arXiv:1910.06701 [cs]. Oct. 2019. DOI: 10.48550/arXiv.1910.06701. (Visited on 01/05/2025).

[18] Qi Shi et al. "Learn to Combine Linguistic and Symbolic Information for Table-based Fact Verification". In: *Proceedings of the 28th International Conference on Computational Linguistics*. Ed. by Donia Scott, Nuria Bel, and Chengqing Zong. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5335–5346. DOI: 10.18653/v1/2020.coling-main.466. URL: https://aclanthology.org/2020.coling-main.466/ (visited on 01/05/2025).

[19] Kai Shu et al. "Fake News Detection on Social Media: A Data Mining Perspective". In: *SIGKDD Explor. Newsl.* 19.1 (Sept. 2017), pp. 22–36. ISSN: 1931-0145. DOI: 10.1145/3137597.3137600. URL: https://doi.org/10.1145/3137597.3137600 (visited on 01/05/2025).

[20] Sonit Singh. *Natural Language Processing for Information Extraction*. arXiv:1807.02383 [cs]. July 2018. DOI: 10.48550/arXiv.1807.02383. URL: http://arxiv.org/abs/1807.02383 (visited on 01/06/2025).

[21] Stanford University Human-Centered Artificial Intelligence. *Artificial Intelligence Index Report 2024*. Tech. rep. URL: https://aiindex.stanford.edu/wp-content/uploads/2024/05/HAI_AI-Index-Report-2024.pdf (visited on 09/28/2024).

[22] Yi Tay et al. *UL2: Unifying Language Learning Paradigms*. arXiv:2205.05131 [cs]. Feb. 2023. DOI: 10.48550/arXiv.2205.05131. URL: http://arxiv.org/abs/2205.05131 (visited on 01/05/2025).

[23] James Thorne et al. "FEVER: a Large-scale Dataset for Fact Extraction and VERification". In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Ed. by Marilyn Walker, Heng Ji, and Amanda Stent. New Orleans, Louisiana: Association for Computational Linguistics, June 2018, pp. 809–819. DOI: 10.18653/v1/N18-1074. URL: https://aclanthology.org/N18-1074/ (visited on 01/05/2025).

[24] Fei Wang et al. *Table-based Fact Verification with Salience-aware Learning*. Sept. 2021. DOI: 10.48550/arXiv.2109.04053. (Visited on 01/05/2025).

[25] Zilong Wang et al. *Chain-of-Table: Evolving Tables in the Reasoning Chain for Table Understanding*. arXiv:2401.04398 [cs] version: 2. Jan. 2024. DOI: 10.48550/arXiv.2401.04398. URL: http://arxiv.org/abs/2401.04398 (visited on 01/05/2025).

[26] Jason Wei et al. *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv:2201.11903 [cs]. Jan. 2023. DOI: 10.48550/arXiv.2201.11903. URL: http://arxiv.org/abs/2201.11903 (visited on 02/17/2025).

[27] World Economic Forum. *The Global Risks Report 2024*. Tech. rep. 19. Jan. 2024. URL: https://edge.sitecorecloud.io/zurichinsur6934-zwpcorp-prod-ae5e/media/project/zurich/dotcom/industry-knowledge/global-risks/docs/the-global-risks-report-2024.pdf (visited on 01/05/2025).

[28] Xueqing Wu, Jiacheng Zhang, and Hang Li. *Text-to-Table: A New Way of Information Extraction*. arXiv:2109.02707 [cs]. Mar. 2022. DOI: 10.48550/arXiv.2109.02707. URL: http://arxiv.org/abs/2109.02707 (visited on 01/06/2025).

[29] Bin Xiao et al. "Efficient Prompting for LLM-based Generative Internet of Things". In: *IEEE Internet of Things Journal* (2024). arXiv:2406.10382 [cs], pp. 1–1. ISSN: 2327-4662, 2372-2541. DOI: 10.1109/JIOT.2024.3470210. URL: http://arxiv.org/abs/2406.10382 (visited on 01/05/2025).

[30] Tianbao Xie et al. *UnifiedSKG: Unifying and Multi-Tasking Structured Knowledge Grounding with Text-to-Text Language Models*. arXiv:2201.05966 [cs] version: 3. Oct. 2022. DOI: 10.48550/arXiv.2201.05966. URL: http://arxiv.org/abs/2201.05966 (visited on 01/05/2025).

[31] Xiaoyu Yang and Xiaodan Zhu. "Exploring Decomposition for Table-based Fact Verification". In: *Findings of the Association for Computational Linguistics: EMNLP 2021*. Ed. by Marie-Francine Moens et al. Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1045–1052. DOI: 10.18653/v1/2021.findings-emnlp.90. URL: https://aclanthology.org/2021.findings-emnlp.90/ (visited on 01/05/2025).

[32] Xiaoyu Yang et al. *Program Enhanced Fact Verification with Verbalization and Graph Attention Network*. arXiv:2010.03084 [cs]. Sept. 2021. DOI: 10.48550/arXiv.2010.03084. URL: http://arxiv.org/abs/2010.03084 (visited on 01/05/2025).

[33] Yunhu Ye et al. *Large Language Models are Versatile Decomposers: Decompose Evidence and Questions for Table-based Reasoning*. arXiv:2301.13808 [cs] version: 3. Apr. 2023. DOI: 10.48550/arXiv.2301.13808. URL: http://arxiv.org/abs/2301.13808 (visited on 01/05/2025).

[34] Hongzhi Zhang et al. "Table Fact Verification with Structure-Aware Transformer". In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Bonnie Webber et al. Online: Association for Computational Linguistics, Nov. 2020, pp. 1624–1629. DOI: 10.18653/v1/2020.emnlp-main.126. URL: https://aclanthology.org/2020.emnlp-main.126/ (visited on 01/05/2025).

[35] Yilun Zhao et al. *ReasTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples*. arXiv:2210.12374 [cs] version: 1. Oct. 2022. DOI: 10.48550/arXiv.2210.12374. URL: http://arxiv.org/abs/2210.12374 (visited on 01/05/2025).

[36] Wanjun Zhong et al. "LogicalFactChecker: Leveraging Logical Operations for Fact Checking with Graph Module Network". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Ed. by Dan Jurafsky et al. Online: Association for Computational Linguistics, July 2020, pp. 6053–6065. DOI: 10.18653/v1/2020.acl-main.539. URL: https://aclanthology.org/2020.acl-main.539/ (visited on 01/05/2025).

# Appendices

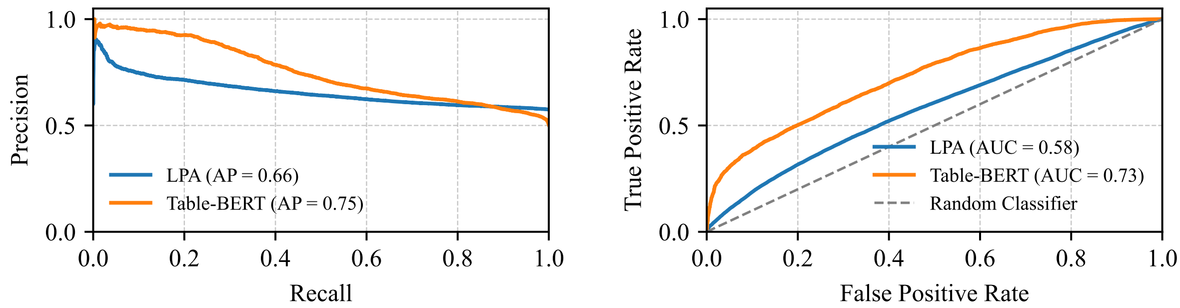
## A Compilation of Results from TabFact Studies

| | Model | Test Accuracy (%) | Validation Accuracy (%) | Year |
|---|---|---|---|---|
| | ARTEMIS-DA [12] | 93.1 | - | 2024 |
| | Dater [33] | 93.0 | - | 2023 |
| | *Human Performance: ≈ 92%* [2] | | | |
| | PASTA [9] | 89.3 | 89.2 | 2022 |
| **Ours*** | Phi4 (Zero Shot) (Section 5.2) | 88.9 | - | 2024 |
| | UL-20B [22] | 87.1 | | 2022 |
| | Chain-of-Table [25] | 86.6 | - | 2024 |
| | Binder [3] | 86.0 | - | 2022 |
| | Tab-PoT [29] | 85.8 | - | 2024 |
| **Ours*** | Phi4 (RAG Approach) (Section 5.4) | 85.7 | - | 2024 |
| | ReasTAP-Large [35] | 84.9 | 84.6 | 2022 |
| | TAPEX-Large [13] | 84.2 | 84.6 | 2021 |
| | T5-3b (UnifiedSKG) [30] | 83.7 | 84.0 | 2022 |
| | DecompTAPAS [31] | 82.7 | 82.7 | 2021 |
| | Salience-aware TAPAS [24] | 82.1 | 82.7 | 2021 |
| **Ours*** | Phi4 (Code Generation) (Section 5.3) | 81.9 | - | 2024 |
| | TAPAS-Large classifier with Counterfactual + Synthetic pre-training [7] | 81.0 | 81.0 | 2020 |
| | ProgVGAT [32] | 74.4 | 74.9 | 2020 |
| | SAT [34] | 73.2 | 73.3 | 2020 |
| | HeterTFV [18] | 72.3 | 72.5 | 2020 |
| | LFC (Seq2Action) [36] | 71.7 | 71.8 | 2020 |
| | LFC (LPA) [36] | 71.6 | 71.7 | 2020 |
| | Num-Net [17] | 72.1 | 72.1 | 2019 |
| **Studied here** | LPA-Ranking w/ Discriminator (Caption) [2] | 65.3 | 65.1 | 2020 |
| **Studied here** | Table-BERT-Horizontal-T+F-Template [2] | 65.1 | 66.1 | 2020 |
| **Studied here** | BERT classifier w/o Table [2] | 50.5 | 50.9 | 2020 |

*Best performing LLM in this approach; other models omitted for brevity

Table 5: Performance comparison of different models on the TabFact dataset as reported by Chen [1] and Meta AI [15] or evaluated in this work.

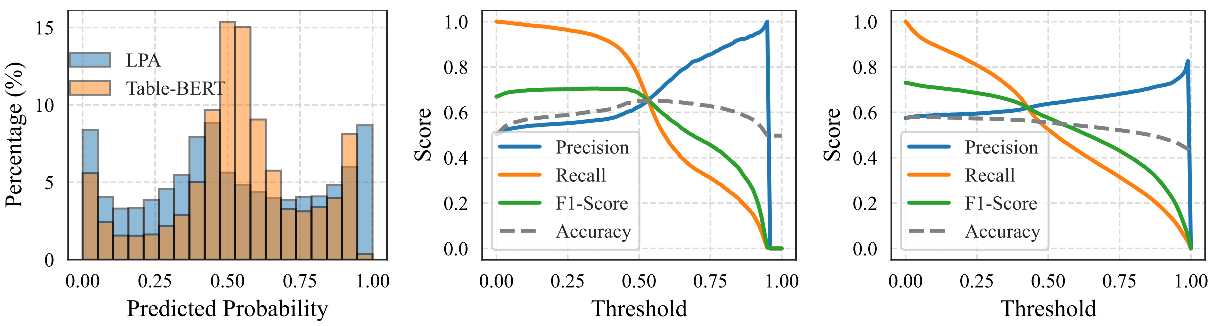
# B Evaluation of the Table-BERT and LPA models




(a) Precision-Recall curve

(b) ROC curve

Figure 5: Precision-Recall curve (a) and ROC curve (b) comparing Table-BERT and LPA. Despite similar accuracy, Table-BERT achieved higher AUC (0.73 vs. 0.58) and average precision (0.75 vs. 0.66), reflecting superior recall and average precision.




(a) Probability distribution of predicted probabilities for Table-BERT and LPA.

(b) Threshold-based performance metrics for the Table-BERT model.

(c) Threshold-based performance metrics for the LPA model.

Figure 6: The probability distribution (a) highlights Table-BERT's bias toward positive predictions and LPA's conservative tendencies. Threshold-based metrics (b & c) reveal Table-BERT's superior recall and F1-score at lower thresholds, while LPA maintains higher precision but sacrifices recall, demonstrating their complementary strengths.

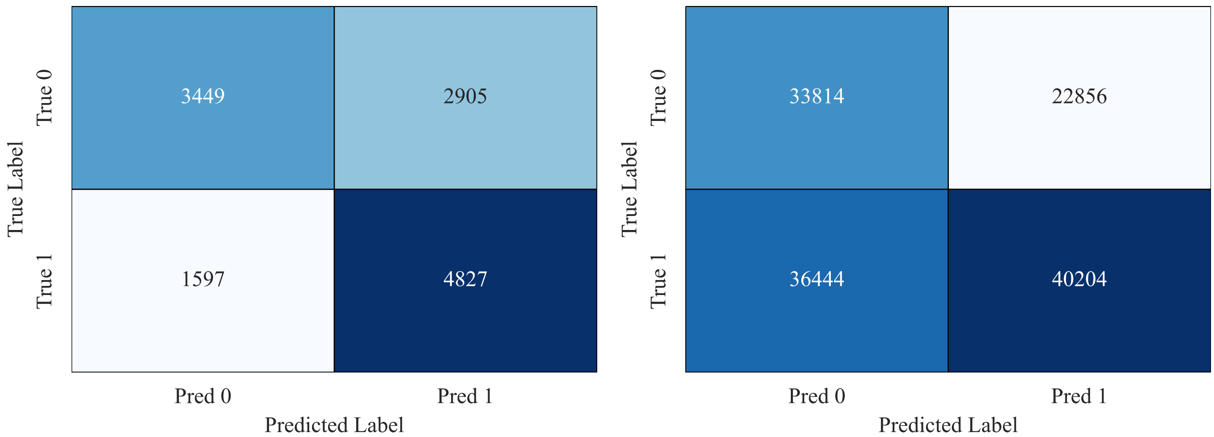



Figure 7: Confusion matrices comparing the performance of the Table-BERT model (left) and the LPA model (right) on the TabFact dataset. The Table-BERT model demonstrated a more balanced classification with fewer misclassifications, whereas the LPA model exhibited higher false positive and false negative rates despite identifying a larger number of true positives and negatives.

# C Error Analysis per Category

## C.1 Explanation of Initial Categories

The following categories were considered in the initial analysis, as derived from Chen et al. [2]. Each category describes a specific operation or relationship in the claims:

- **Aggregation:** Refers to sentences involving aggregation operations, such as “the averaged age of all ...”, “the total amount of scores obtained in ...”, etc.
- **Negation:** Refers to sentences with negation operations, such as “xxx did not get the best score”, “xxx has never obtained a score higher than 5”.
- **Superlative:** Refers to sentences involving superlative operations, such as “xxx achieves the highest score in ...”, “xxx is the lowest player in the team”.
- **Comparative:** Refers to sentences involving comparative operations, such as “xxx has a higher score than yyy”.
- **Ordinal:** Refers to sentences involving ordinal operations, such as “the first country to achieve xxx is xxx”, “xxx is the second oldest person in the country”.
- **Unique:** Refers to sentences involving unique operations, such as “there are 5 different nations in the tournament”, “there are no two different players from the U.S.”.
- **All:** Refers to sentences involving "for all" operations, such as “all of the trains are departing in the morning”, “none of the people are older than 25”.
- **None:** Refers to sentences that do not involve higher-order operations, such as “xxx achieves 2 points in xxx game”, “xxx player is from xxx country”.

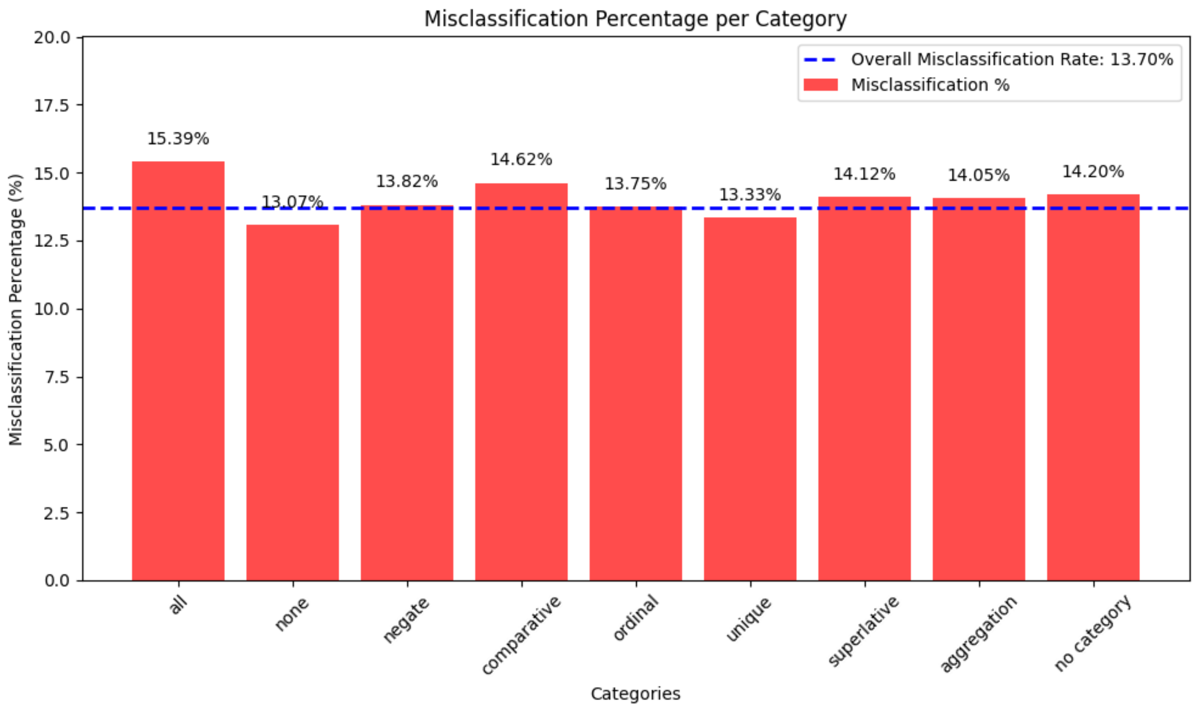



Figure 8: Misclassification Errors for the Initial Category Set in Our Best Approach

The histogram illustrates that the initial set of categories derived by Chen et al. [2] failed to provide meaningful insights, as the error rates for each category were similar to the overall misclassification rate.

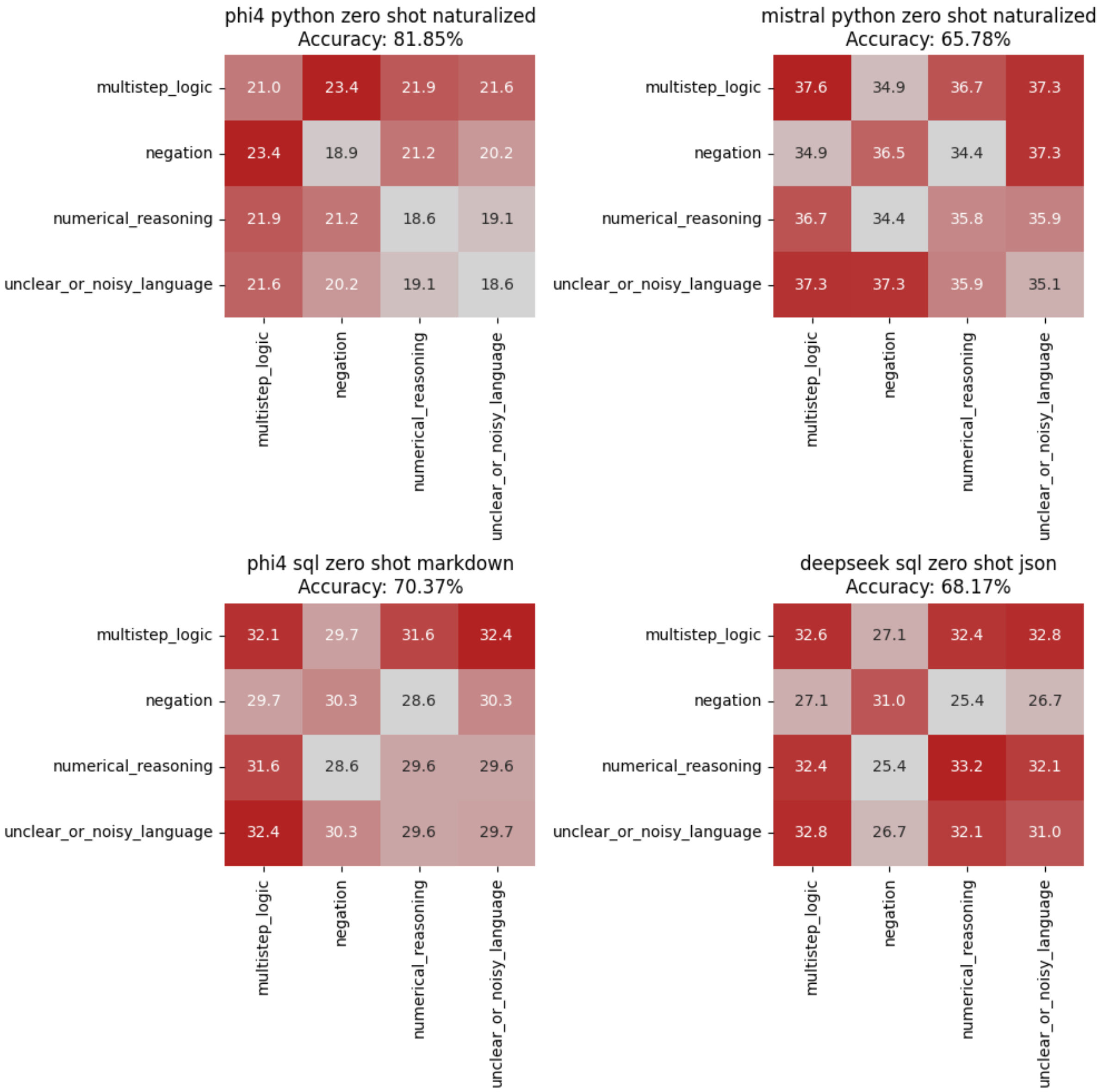



Figure 9: Classification Errors in Code Generation Models by categories.

# D Examples of Gold Standard Errors

In this appendix, we provide a concrete example for each category of gold standard errors identified in our analysis (Section 7). For each category, a brief description is given along with a screenshot from our *Table Fact-Checking Explorer* (available at `https://horstmann.tech/table-fact-checking-project/`) to illustrate the issue.

## D.1 Negation and Ambiguity Errors

Some claims are mislabelled due to incorrect handling of (double) negations, leading to false *refuted* labels.

**Claim:** tony lema do not win in the pga championship
**Predicted:** TRUE
**Raw Output:** To determine the truth value of the claim "Tony Lema did not win in the PGA Championship," we need to examine the relevant data from the table provided. 1. **Identify Relevant Information**: The claim specifically pertains to Tony Lema's performance in the PGA Championship, focusing on whether he achieved a win or not. 2. **Locate Relevant Data in Table**: - We need to find the row corresponding to the "PGA Championship." - Within that row, we should look at the column labeled "wins" to check if Tony Lema won any tournaments there. 3. **Examine Row 4 (PGA Championship)**: - The table's Row 4 corresponds to the PGA Championship. - In this row, under the "wins" column, it is recorded as 0. 4. **Conclusion**: - Since the number of wins in the PGA Championship for Tony Lema is 0, he did not win in that tournament. - Therefore, the claim "Tony Lema did not win in the PGA Championship" is supported by the data and is TRUE. 5. **Identify Relevant Cells**: - The relevant cell to verify this claim is found at Row 4 under the column named "wins." Thus, the JSON object representing the answer would be: ```json { "answer": "TRUE", "relevant_cells": [ {"row_index": 3, "column_name": "wins"} ] } ``` Note: The row index is adjusted to 0-based indexing from the naturalized format's 1-based index.
**Ground Truth:** FALSE

| tournament | wins | top - 5 | top - 10 | top - 25 | events | cuts made |
|---|---|---|---|---|---|---|
| masters tournament | 0 | 1 | 2 | 4 | 4 | 4 |
| us open | 0 | 2 | 3 | 4 | 6 | 5 |
| the open championship | 1 | 2 | 2 | 2 | 3 | 3 |
| pga championship | 0 | 0 | 1 | 2 | 5 | 4 |
| totals | 1 | 5 | 8 | 12 | 18 | 16 |

Model-highlighted cells (i.e. cells that the LLM considered relevant to the claim)
Entity-linked cells (i.e. cells that Wenhuchen et al. automatically identified as relevant to the claim)

Figure 10: Example of a negation and ambiguity error. The claim uses a negation, resulting in an incorrect label. The claim that Tony Lema did not win in the PGA championship should be *correct* since the table shows 0 wins for this tournament (double negation). The "raw output" of the model provides accurate reasoning for a correct classification.

## D.2 Domain-Specific Misinterpretations

Errors may arise from missing domain knowledge. For example, in golf, lower scores indicate better performance, yet some claims were mislabelled.

**Claim:** steve stricker have the best score out of all of the player from the united state
**Predicted:** TRUE
**Raw Output:** To determine the truth of the claim "Steve Stricker has the best score out of all of the players from the United States," we need to verify if Steve Stricker indeed has the lowest score among U.S. players listed in the table. 1. **Identify Relevant Players:** - We are interested only in players from the United States, as indicated by their "country" being "united states." 2. **Extract Scores for U.S. Players:** - Look at each row and extract information relevant to U.S. players: - Row 1: Steve Stricker has a score of 139. - Row 5: Jim Furyk has a score of 142. - Row 7: Jason Dufner has a score of 143. - Row 9: Phil Mickelson has a score of 143. - Row 10: Arron Oberholser has a score of 143. 3. **Compare Scores:** - Steve Stricker's score is 139. - Jim Furyk's, Jason Dufner's, Phil Mickelson's, and Arron Oberholser's scores are all higher than 139 (142 for Jim Furyk; 143 for the others). 4. **Conclusion:** - Since Steve Stricker has a lower score (139) compared to all other U.S. players listed, he indeed has the best score. The claim is TRUE based on the data provided in the table. The relevant cells are those containing scores and countries of interest: ```json { "answer": "TRUE", "relevant_cells": [ {"row_index": 0, "column_name": "country"}, {"row_index": 0, "column_name": "score"}, {"row_index": 4, "column_name": "country"}, {"row_index": 4, "column_name": "score"}, {"row_index": 6, "column_name": "country"}, {"row_index": 6, "column_name": "score"}, {"row_index": 8, "column_name": "country"}, {"row_index": 8, "column_name": "score"}, {"row_index": 9, "column_name": "country"}, {"row_index": 9, "column_name": "score"} ] } ```
**Ground Truth:** FALSE

| place | player | country | score | to par |
|---|---|---|---|---|
| 1 | steve stricker | united states | 70 + 69 = 139 | - 1 |
| 2 | colin montgomerie | scotland | 69 + 71 = 140 | e |
| t3 | kenneth ferrie | england | 71 + 70 = 141 | + 1 |
| t3 | geoff ogilvy | australia | 71 + 70 = 141 | + 1 |
| t5 | jim furyk | united states | 70 + 72 = 142 | + 2 |
| t5 | pádraig harrington | ireland | 70 + 72 = 142 | + 2 |
| t7 | jason dufner | united states | 72 + 71 = 143 | + 3 |
| t7 | graeme mcdowell | northern ireland | 71 + 72 = 143 | + 3 |
| t7 | phil mickelson | united states | 70 + 73 = 143 | + 3 |
| t7 | arron oberholser | united states | 75 + 68 = 143 | + 3 |

Model-highlighted cells (i.e. cells that the LLM considered relevant to the claim)
Entity-linked cells (i.e. cells that Wenhuchen et al. automatically identified as relevant to the claim)

Figure 11: Example of a domain-specific misinterpretation. The table concerns golf scores where the conventional interpretation of higher scores being better is reversed, resulting in an incorrect label. The model correctly argues that Steve Stricker has the best score out of all players from the US.

## D.3 Irrelevant or Misaligned Claims

In some cases, claims appear to be erroneously associated with the wrong tables.

**Table Title:** catholic church by country
**Wikipedia Link:** https://en.wikipedia.org/wiki/Catholic_Church_by_country

× **Claim #1: it be also the largest non - government provider of education and health care in the world , while the diplomatic status of the holy see facilitate the access of its vast international network of charity**

× Claim #2: the church be also know as the people of god , the body of christ , the temple of the holy spirit , among other name

× Claim #3: accord to vatican ii 's gaudium et spes , the church have but 1 sole purpose -- that the kingdom of god may come and the salvation of the human race may be accomplish

× Claim #4: this communion of church comprise the latin church (or the roman or western church) as well as 23 eastern catholic church , canonically call sui juris church , each lead by either a patriarch or a major archbishop in full communion with the holy see

× Claim #5: archbishop and bishop administer individual diocese as successor of the 12 apostle

✓ Claim #6: it be also the smallestt non - government provider of education and health care in the world , while the diplomatic status of the holy see facilitate the access of its vast international network of charity

✓ Claim #7: the church be also know as the people of god , the body of christ , the temple of the holy spirit , among other name

✓ Claim #8: this communion of church comprise the latin church (or the roman or western church) as well as 23 eastern catholic church , canonically call sui juris church , each lead by either a patriarch or a major archbishop in full communion with the holy see

✓ Claim #9: accord to vatican ii 's gaudium et spes , the church have but 1 sole purpose -- that the kingdom of god may come and the salvation of the human race may be accomplish

✓ Claim #10: archbishop and bishop administer individual diocese as successor of the 12 apostle

**Claim:** it be also the largest non - government provider of education and health care in the world , while the diplomatic status of the holy see facilitate the access of its vast international network of charity
**Predicted:** FALSE
**Raw Output:** To determine the truth of the claim based on the provided table, we need to analyze the relevant aspects: ### Analyzing the Claim 1. **Largest Non-Government Provider**: - The claim mentions being the largest non-government provider of education and health care in the world. - This part of the claim is not about specific numbers or percentages but rather a qualitative statement about an entity's status. 2. **Diplomatic Status of the Holy See**: - The claim refers to the diplomatic status of the Holy See, which facilitates access to its international network of charity. - Again, this is more about a qualitative aspect of the Holy See's role and influence rather than specific numerical data from the table. ### Analyzing the Table The table provides statistical information related to Catholic populations in various regions: - **Regions**: Central Asia, East Asia, South Asia, Southeast Asia - **Total Population** for each region and globally. - **Catholic Population** for each region and globally. - **Percentage of Catholics** within each region's population. - **Global Percentage Contribution** of each region to the total Catholic population. ### Relevance of Table Data The table data focuses on demographic statistics about Catholic populations. It does not provide information regarding: - The Holy See's role as a non-government provider of education and health care. - The diplomatic status of the Holy See or its international network of charity. Thus, none of the columns in the table (total population, catholic population, % catholic, % of global catholic pop) are directly relevant to verifying the claim about being the largest non-government provider or the diplomatic status facilitating access to charity networks. ### Conclusion Since the table does not contain any data that can verify or refute the qualitative aspects of the claim regarding the Holy See's role in education, health care, and diplomacy, we cannot determine the truth of the claim based on this table alone. ```json { "answer": "FALSE", "relevant_cells": [] } ```
**Ground Truth:** TRUE

| region | total population | catholic | % catholic | % of global catholic pop |
|---|---|---|---|---|
| central asia | 92019166 | 199086 | 1.23% | 0.01% |
| east asia | 1528384440 | 13853142 | 0.90% | 1.28% |
| south asia | 1437326682 | 20107050 | 1.39% | 1.87% |
| southeast asia | 571337070 | 86701421 | 15.17% | 8.06% |
| total | 3629067358 | 120860699 | 3.33% | 11.24% |

Figure 12: Example of irrelevant or misaligned claims. Claims #1 to #6 do not correspond to the table content shown (Catholic Church by Country). This makes it impossible for the model to classify any of these claims as entailed, although they are according to the TabFact dataset.

## D.4 Phrasing and Numerical Inconsistencies

Minor linguistic or numerical misinterpretations have led to incorrect gold labels.

**Claim:** manfred kokot be from berlin
**Predicted:** FALSE
**Raw Output:** To evaluate the claim "Manfred Kokot be from Berlin," we need to examine the relevant data in the table to determine if this statement can be verified as true. 1. **Identify Relevant Data**: - We must find information about Manfred Kokot, specifically focusing on his place of achievement. - The columns pertinent to verifying this claim are "athlete" and "place." 2. **Locate Manfred Kokot in the Table**: - In the table provided, locate the row where the athlete is "Manfred Kokot." - According to the table, Manfred Kokot appears at row index 3. 3. **Check the Place Column for Manfred Kokot's Entry**: - For the entry corresponding to Manfred Kokot (row index 3), check the value in the "place" column. - The place listed for Manfred Kokot is "berlin." 4. **Evaluate the Claim**: - The claim states that "Manfred Kokot be from Berlin." Based on the table, his achievement was indeed recorded in Berlin (row index 3, place: berlin). - However, the statement "be from" implies origin or birthplace rather than the location of an event. The table only provides information about where Manfred Kokot performed and not where he is originally from. 5. **Conclusion**: - The table does not provide any information about Manfred Kokot's place of origin; it only indicates that his record was set in Berlin. - Therefore, the claim cannot be fully supported as true based on the data provided because "be from" requires knowledge about his birthplace or hometown, which is not available. Based on this analysis: ```json { "answer": "FALSE", "relevant_cells": [ {"row_index": 3, "column_name": "athlete"}, {"row_index": 3, "column_name": "place"} ] } ```
**Ground Truth:** TRUE

| rank | time | athlete | date | place |
|---|---|---|---|---|
| 1 | 5.56 b | donovan bailey | 9 february 1996 | reno |
| 1 | 5.56 b | maurice greene | 13 february 1999 | los angeles |
| 3 | 5.58 | leonard scott | 26 february 2005 | liévin |
| 4 | 5.61 | manfred kokot | 4 february 1973 | berlin |
| 4 | 5.61 | james sanford | 20 february 1981 | san diego |
| 4 | 5.61 | michael green | 16 february 1997 | liévin |
| 4 | 5.61 | deji aliu | 21 february 1999 | liévin |
| 4 | 5.61 | jason gardener | 16 february 2000 | madrid |
| 4 | 5.61 | freddy mayola | 16 february 2000 | madrid |
| 10 | 5.62 | emmit king | 5 march 1986 | kobe |
| 10 | 5.62 | andre cason | 15 february 1992 | los angeles |
| 10 | 5.62 | eric nkansah | 21 february 1999 | liévin |
| 10 | 5.62 | morne nagel | 24 february 2002 | liévin |

Model-highlighted cells (i.e. cells that the LLM considered relevant to the claim)
Entity-linked cells (i.e. cells that Wenhuchen et al. automatically identified as relevant to the claim)

Figure 13: Example of a phrasing inconsistency/ambiguity. As the model (correctly) states, the statement "Manfred Kokot *is* from Berlin" implies origin or birthplace rather than the location of an event (which is covered in the table). This phrasing leads to an incorrect ground truth label.

**Note:** Screenshots are captured from our interactive Table Fact-Checking Explorer to provide visual evidence of these errors. Each example represents one of the recurring error categories discussed in Section 7.