Sheetpedia: A 300K-Spreadsheet Corpus for Spreadsheet Intelligence and LLM Fine-Tuning

Zailong Tian[1,2] **Zhuoheng Han**[1] **Houfeng Wang**[1]* **Lizi Liao**[2]
[1]State Key Laboratory for Multimedia Information Processing, Peking University
[2]School of Computing and Information Systems, Singapore Management University
wanghf@pku.edu.cn, {zltian, lzliao}@smu.edu.sg

Abstract

Spreadsheets are widely used for data analysis and reporting, yet their complex structure and formula logic pose significant challenges for AI systems. We introduce Sheetpedia, a large-scale corpus of over 290,000 diverse spreadsheets (from 324,000+ workbooks) compiled from enterprise email archives and online forums. We detail a rigorous collection and preprocessing pipeline (integrating the Enron email spreadsheet archive and the Fuse web corpus, plus a new crawl of Excel forums) to standardize formats, filter languages, and remove duplicates. Sheetpedia provides extensive coverage of real formulas and annotations – addressing a gap left by prior table datasets (e.g. web tables used in TURL or Text-to-SQL in Spider) which often lack formula semantics. We present comprehensive corpus statistics, highlighting rich formula diversity and a majority (78%+) of English content. To demonstrate the corpus's utility, we fine-tune large language models on Sheetpedia for two novel spreadsheet understanding tasks: Natural Language to Semantic Range (NL2SR) and Natural Language to Formula (NL2Formula). Using a rejection-sampling data generation strategy, our fine-tuned models achieve up to 97.5% accuracy on NL2SR and 71.7% on NL2Formula – substantially outperforming baseline approaches. Sheetpedia (to be released publicly) fills a crucial need for a large, high-quality spreadsheet benchmark, enabling more effective spreadsheet intelligence and natural language interfaces for spreadsheet tools.

1 Introduction

Spreadsheets are a ubiquitous tool for data-driven decision making, used by hundreds of millions of people worldwide. They combine tabular data with formula-based computations, allowing end-users to perform complex analyses without traditional programming. However, the semi-structured nature of spreadsheets – including free-form text, numeric data, and executable formulas intermingled in cells – makes automated understanding by AI systems extremely challenging. Unlike database tables or CSV files, spreadsheets often lack a fixed schema and rely on implicit contexts (cell references, formulas, and layout) to convey meaning. Natural language processing and machine learning methods struggle with this rich but irregular format, which calls for specialized resources and models to advance spreadsheet understanding.

Despite the widespread use of spreadsheets, large-scale corpora for spreadsheet AI research remain scarce. Existing datasets, such as EUSES (~4,000 files) [1], Enron (~15,000 files) [2], and Fuse (~249,000 files) [3], are limited by small scale, narrow domains, or a lack of formula-rich content. Many contain static data or simple tables, and accessing their raw data can be challenging. Sheetpedia addresses this gap by curating a comprehensive collection of spreadsheets from multiple sources, yielding a corpus of ~290,000 unique worksheets – to our knowledge, the largest publicly available

[*] Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.

spreadsheet corpus to date. Our dataset encompasses varied domains (financial models, inventory lists, academic data, etc.), and through careful preprocessing we ensure a high quality of content (e.g. language filtering yields 78% English sheets, and deduplication cuts 48% of near-duplicates). This diverse, large-scale corpus provides a rich foundation for training and evaluating AI models on real spreadsheet structures and formulas.

Another key contribution of Sheetpedia is demonstrating how such data can be leveraged to improve model performance on novel spreadsheet-related tasks. While general table understanding has seen progress (e.g. representation learning on Wikipedia tables in TURL [4], or semantic parsing to SQL queries in Spider [5]), spreadsheets pose unique challenges that require new problem formulations. We focus on two tasks that highlight natural language interaction with spreadsheets: (1) Natural Language to Semantic Range (NL2SR) – mapping a user's plain-language request to the correct cell range or region in a spreadsheet; and (2) Natural Language to Formula (NL2Formula) – generating a valid Excel formula that fulfills a given natural language description. These tasks simulate practical scenarios, such as a user asking "What is the total sales for Q1?" (NL2SR would identify the relevant cells to sum, and NL2Formula would produce the SUM formula over that range). Both tasks are challenging because they require understanding the semantics of the spreadsheet content (schema, headings, data distributions) as well as the natural language query. They go beyond standard table QA: NL2Formula in particular is a code-generation problem (synthesizing formula code), which has only recently begun to be studied in NLP.

A major hurdle in tackling NL2SR and NL2Formula is the lack of large annotated training datasets. Recent work by [6] introduced an NL2Formula benchmark of 70K NL-formula pairs by converting existing text-to-SQL examples into spreadsheet context. This conversion approach, while clever, is limited by the coverage of SQL patterns and does not address the NL2SR task. In our work, we propose a complementary data generation strategy leveraging LLMs themselves to bootstrap training data. We employ a rejection sampling framework, utilizing a specialized judge model [7] to curate high-quality synthetic training data at scale. Specifically, we prompt a LLM to generate candidate formulas or ranges for novel natural language queries. These candidates are then evaluated by the judge model, which filters out incorrect or inconsistent outputs, ensuring a robust and reliable dataset. This approach is inspired by successes in code generation, where sampling multiple outputs and selecting correct ones can dramatically improve accuracy. For example, OpenAI's Codex model [8] (a GPT-3 [9] variant for code) solved 70% of programming tasks with 100 samples vs. only 28% with one try. By applying iterative self-refinement via rejection sampling, we build sizeable training sets for NL2SR and NL2Formula without extensive manual labeling.

We fine-tune state-of-the-art LLMs on Sheetpedia for these tasks and find that spreadsheet-specific training yields substantial gains. Our NL2SR model accurately identifies the correct cell ranges for user queries in 97.5% of test cases, and our NL2Formula model achieves 71.7% exact formula generation accuracy. These results underscore the value of domain-specific data: general-purpose models like Codex or GPT-3, while powerful, benefit greatly from fine-tuning on targeted spreadsheet data and tasks. Indeed, specialized models for spreadsheet formulas (e.g. the 60M-parameter FLAME model [10]) have been shown to outperform much larger general code models on formula prediction tasks, reinforcing our findings.

In summary, our work makes the following contributions: (1) We introduce Sheetpedia, a large-scale, diverse, and formula-rich spreadsheet corpus comprising over 290,000 worksheets, addressing the limitations of prior datasets in scale, quality, and semantic richness; (2) We define and release benchmarks for two novel spreadsheet understanding tasks—Natural Language to Semantic Range (NL2SR) and Natural Language to Formula (NL2Formula)—grounded in realistic spreadsheet usage; and (3) We demonstrate that fine-tuning LLMs on Sheetpedia, combined with a rejection sampling-based data generation framework, significantly improves task performance, achieving up to 97.5% accuracy on NL2SR and 71.7% on NL2Formula.

2 Related Work

Spreadsheet and Table Datasets. Early spreadsheet datasets, such as the EUSES Spreadsheet Corpus, provided a collection of approximately 4,000 real-world spreadsheets sourced from the web. However, EUSES is limited in scale, and many files lack formulas, reducing their utility for training formula-centric models. The Enron spreadsheet dataset, derived from the Enron email release, offers around 15,000 spreadsheets with richer formula content (60% include formulas) but is confined to a single corporate domain.More recently, the Fuse dataset mined 249,000 spreadsheets from the Common Crawl, but only 7% contain formulas, and many files are incomplete or require reconstruction. The DECO dataset [11] provides 1,165 annotated spreadsheets from the Enron corpus, with cell-level layout annotations for table recognition tasks. The DeExcelerator framework[12], evaluated on datasets like DeEx, CIUS, and SAUS, extracts relational data from semi-structured spreadsheets and HTML tables. Beyond spreadsheets, large table corpora, such as Wikipedia tables used in TURL (1.6M tables) and Google's TaPas [13] (millions of tables), support tasks like table question-answering but lack the dynamic formulas and user-generated context inherent to spreadsheets. Sheetpedia addresses these limitations by combining web-crawled data (inspired by Fuse), enterprise data (like Enron), and a scrape of 147,000 user-contributed spreadsheets from ExcelForum.

Spreadsheet Intelligence and Formula Synthesis. Research on intelligent spreadsheet assistance has focused on formula prediction and synthesis. Microsoft's FlashFill [14] infers string transformation formulas from user examples, a form of programming by example. More advanced approaches, such as SpreadsheetCoder [15], use neural networks to predict formulas based on cell context, treating the task as code completion. In the NLP community, semantic parsing techniques inspired by Text-to-SQL datasets like WikiSQL [16] and Spider have been adapted to spreadsheets. The NL2Formula task, for instance, involves generating Excel formulas from English queries, with a dataset of 70,000 query-formula pairs derived from Text-to-SQL problems. Complementary to this, we introduce NL2SemanticRange (NL2SR), a novel task for predicting cell range addresses from natural language queries, extending prior work on natural language information retrieval in spreadsheets, such as NLP-SIR [17]. Together, NL2Formula and NL2SR formalize core aspects of spreadsheet interaction—data selection and formula generation—enabling comprehensive intelligent assistance.

Large Language Models for Tables and Spreadsheets. The rise of LLMs has spurred interest in their application to structured data. Models like TURL and TaPas pre-train transformers on millions of relational tables to enhance tasks such as column type annotation and table question-answering. However, these models focus on static tables and lack spreadsheet-specific grounding, such as understanding cell ranges or formulas. To bridge this gap, FORTAP [18] incorporates formulas during pre-training to make the model aware of numerical reasoning over tables. In contrast, Codex, a GPT-3-based code model, can generate Excel formulas from natural language but struggles with spreadsheet layouts without examples. Specialized models like FLAME , a 60M-parameter transformer trained on Excel formulas, outperform larger general-purpose models on formula repair and completion by leveraging domain-specific data. Our work aligns with this trend, demonstrating that fine-tuning LLMs on Sheetpedia's rich spreadsheet corpus significantly improves performance on spreadsheet tasks compared to generic models. Additionally, our rejection sampling methodology for data generation draws inspiration from program synthesis techniques, where multiple candidate outputs are tested for correctness.

3 Sheetpedia

To build a high-quality spreadsheet corpus for NLP tasks, we integrate diverse sources, including public datasets and user-uploaded content from professional forums. This section outlines the workflow, which involves sourcing data from public datasets and ExcelForum, followed by a preprocessing pipeline that standardizes formats, cleans and filters content, and deduplicates worksheets to create a robust corpus of 290,509 unique worksheets. The following subsections introduce these in detail.

3.1 Data Sources

The corpus draws from two primary sources: existing public datasets and ExcelForum (`https://www.excelforum.com/`), a leading platform for technical spreadsheet discussions. These sources span applications like financial modeling and data analysis, ensuring a broad representation of spreadsheet use cases.

Public Datasets. It includes two public datasets. The Fuse dataset contains 161,323 workbooks (182,784 worksheets[1]) sourced from the internet, offering diverse content. The Enron dataset provides 15,927 workbooks (62,612 worksheets[1]) from corporate emails, reflecting enterprise contexts.

ExcelForum. A custom crawler collected 147,738 workbooks (320,489 worksheets) from ExcelForum, a platform with over 1 million threads and 5 million posts as of May 2024. Targeting the "Excel General" and "Excel Formulas & Functions" subforums, the crawler adhered to `robots.txt`, avoiding restricted directories (e.g., `search.php`) and bot-related files (e.g., Bytespider).

The raw corpus totals 324,988 workbooks and 566,018 worksheets, serving as a substantial resource for NLP and document intelligence research.

3.2 Preprocessing Pipeline

A comprehensive preprocessing pipeline standardizes data, enhances quality, and eliminates redundancies through format conversion, cleaning, language filtering, and deduplication.

Format Standardization. All `.xls` files are converted to the modern `.xlsx` format using `pyexcel`, as the newer XML-based format provides better data integrity and compatibility with contemporary tools. The `openpyxl` library then extracts cell contents, formulas, and metadata, serializing them into JSON with fields for filename, sheetname, UsedRange, Cells, and others.

Data Cleaning. First, language filtering uses `lingua` to identify dominant languages. For each worksheet, we concatenate non-empty, non-numeric, and non-formula cells (e.g., excluding cells starting with =) row-wise to form detection contexts. Worksheets with fewer than 20 valid cells are excluded as linguistically unverifiable. Language detection applies a confidence threshold of 0.8, with mixed-language detection and a lower threshold (0.5) retry for ambiguous cases to reduce "unknown" classifications. Across 295,672 valid worksheets, English dominates (77.40%, 228,843 worksheets), followed by "unknown" (13.34%) and minor languages like Latin (1.76%).

Second, formulas are filtered to retain only those that are syntactically valid and functionally complex. We exclude formulas with cross-sheet references (e.g., Sheet2!A1) and those consisting solely of a single text-manipulation function (e.g., CONCAT, TEXTJOIN, LEFT) without range references. Retained formulas must contain at least one valid Excel function (e.g., SUM, VLOOKUP, IF) from a predefined set of standard official functions and reference non-empty cells within the same worksheet. Syntactic correctness is verified by tokenizing formulas and validating cell ranges (e.g., A1, A1:B10).

Spreadsheet Deduplication. To eliminate redundant worksheets in the corpus, we employ a deduplication pipeline based on MinHash [19] and Locality-Sensitive Hashing (LSH) [20], leveraging Jaccard similarity as the core metric. Spreadsheets are naturally suited for set-based similarity analysis: each worksheet can be represented as a set of non-empty, non-numeric cell values (excluding formulas), capturing its textual content. Jaccard similarity, defined as the size of the intersection divided by the union of two sets, effectively measures content overlap between worksheets, making it ideal for identifying near-duplicates in this context.

Initially, we extracted non-empty, non-numeric cells from each worksheet, excluding worksheets with fewer than 20 valid cells to guarantee sufficient content for reliable comparisons. For the remaining worksheets, we computed MinHash signatures (length = 1000) to efficiently approximate Jaccard similarities. These signatures were clustered using LSH with a Jaccard similarity threshold of 0.8, effectively grouping worksheets exhibiting significant content overlap. The chosen LSH configuration employed parameters of r=100 and b=10, carefully balancing precision and recall. Specifically, the high value of r ensured precision by demanding strong within-band similarity, thus minimizing false positives (dissimilar worksheets incorrectly clustered). Concurrently, a moderate value of b maintained acceptable recall, effectively capturing most near-duplicates without incurring excessive computational costs. The selected configuration of r=100, b=10 significantly reduced the corpus size from 566,018 to 290,509 worksheets (a 48.7% reduction).

[1] Numbers reflect workbooks/worksheets that were successfully parsed and accessible; some files were excluded due to corruption or format obsolescence.

[2] `https://github.com/pyexcel/pyexcel`

[3] `https://openpyxl.readthedocs.io/en/stable/`

[4] `https://github.com/pemistahl/lingua-py`.

An alternative configuration (r=10, b=100) yielded excessive clustering and missed genuine duplicates. Our selected configuration (r=100, b=10), validated via manual inspection, ensured high precision and sufficient recall. The Union-Find algorithm finalized clustering in approximately 9 minutes, producing 290,509 unique worksheets: Enron (62,612 → 28,032), Fuse (182,784 → 105,097), and ExcelForum (320,489 → 157,380).

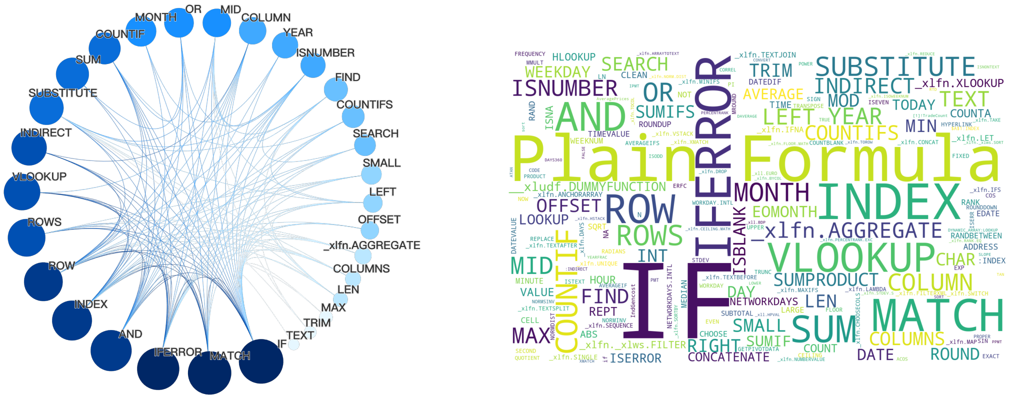
4 Dataset Statistics

This section analyzes the final deduplicated spreadsheet corpus of 290,509 worksheets to characterize its scale, diversity, and suitability for NLP tasks. The analysis examines formula pattern distributions, corpus characteristics at workbook and worksheet levels, and language distribution. Through statistical summaries, visualizations, and language detection, we highlight the corpus's prevalent formula patterns, skewed distributions and English dominance, providing insights into its structure and utility.

4.1 Formula Pattern Distribution

We analyzed the distribution of individual functions and their co-occurrence patterns in formulas. Non-arithmetic formula patterns are predominantly led by `IF`, `MATCH`, `IFERROR`, and `AND` (Figure1a). The co-occurrence network (Figure1a) reveals that functions such as `IF`, `SUM`, `VLOOKUP`, and `COUNTIF` occupy important positions with larger nodes and numerous connections, indicating their high usage frequency and frequent combination with other functions. This aligns with common knowledge, as these functions are foundational in Excel, serving essential purposes like conditional logic (`IF`), aggregation (`SUM`), and data lookup (`VLOOKUP`). Notably, `IFERROR` exhibits strong connections with `IF` and `VLOOKUP`, reflecting its common use in handling potential errors from these functions, such as wrapping `VLOOKUP` to avoid error values—a practical pattern observed in real-world applications.

The word cloud (Figure 1b) further highlights the prominence of `IF`, `SUM`, and `VLOOKUP`, consistent with their dominance in the co-occurrence network. Additionally, the presence of terms like `ERROR` and `ISERROR` in the word cloud underscores error handling as a critical aspect of formula design, corroborating the frequent use of `IFERROR` observed in the network.



(a) Co-occurrence network of the top 30 functions.

(b) Word cloud visualizing formula structure patterns.

Figure 1: Visualization of formula pattern distribution

4.2 Spreadsheet Corpus Characteristics

Table 1 summarizes the spreadsheet corpus at workbook and worksheet levels, covering cell, row, column, and worksheet counts. The data reveals a highly skewed distribution, with most spreadsheets being small but a few exhibiting extreme sizes. At the workbook level, cell counts range from 20 to 60,614,911 (mean: 21,290.72, median: 270), indicating that most workbooks are compact, with 75% having 1,218 or fewer cells (Q3). Row (median: 67) and column (median: 10) counts show similar skewness, and worksheet counts (mean: 1.37, median: 1) confirm that single-sheet workbooks dominate. At the worksheet level, cell counts (mean: 15,487.37, median: 300) and row/column counts (medians: 48 and 10) reflect compact worksheets, with 75% having 1,161 or fewer cells.

Table 1: Statistical summary of the spreadsheet corpus at workbook and worksheet levels.

Level	Metric	Min	Max	Mean	Median	Q1	Q3	Mode
Workbook	Cell Count	20	60,614,911	21,290.72	270	201	1,218	201
	Row Count	1	3,145,590	931.58	67	23	87	67
	Column Count	1	98,304	49.07	10	3	20	3
	Worksheet Count	1	123	1.37	1	1	1	1
Worksheet	Cell Count	20	60,614,872	15,487.37	300	184	1,161	201
	Row Count	1	1,048,576	677.65	48	21	75	67
	Column Count	1	16,384	35.69	10	4	17	3

4.3 Language Distribution

The language distribution of 247,909 worksheets, identified using the `lingua` library after filtering, is highly skewed. English dominates, comprising 78.85% of the dataset (195,479 worksheets), followed by an “Unknown” category at 11.99% (29,726 worksheets). The significant Unknown portion likely reflects worksheets with minimal text, mixed languages, or formats challenging for automated detection. Among the 70 other identified languages, Latin (1.78%, 4,423 worksheets) and Yoruba (1.73%, 4,281 worksheets) are the most prevalent minor languages, followed by 68 others (e.g., German: 0.44%, Spanish: 0.43%) collectively contributing 9.16%. This distribution highlights the predominance of English-language materials, with moderate linguistic diversity across a long tail of minor languages.

5 Spreadsheet Corpus for Downstream Tasks

This section presents a spreadsheet corpus designed to enhance LLM performance on spreadsheet-related tasks through fine-tuning. We focus on two tasks: Natural Language to Semantic Range (NL2SR) and Natural Language to Formula (NL2Formula). To address the challenge of acquiring high-quality training and test data, we employ distinct data generation strategies: rejection sampling for the training set and a combination of LLM generation with human review for the test set. Fine-tuned models demonstrate significant improvements, achieving up to 97.50% accuracy for NL2SR and 71.67% for NL2Formula, validated through comprehensive experiments.

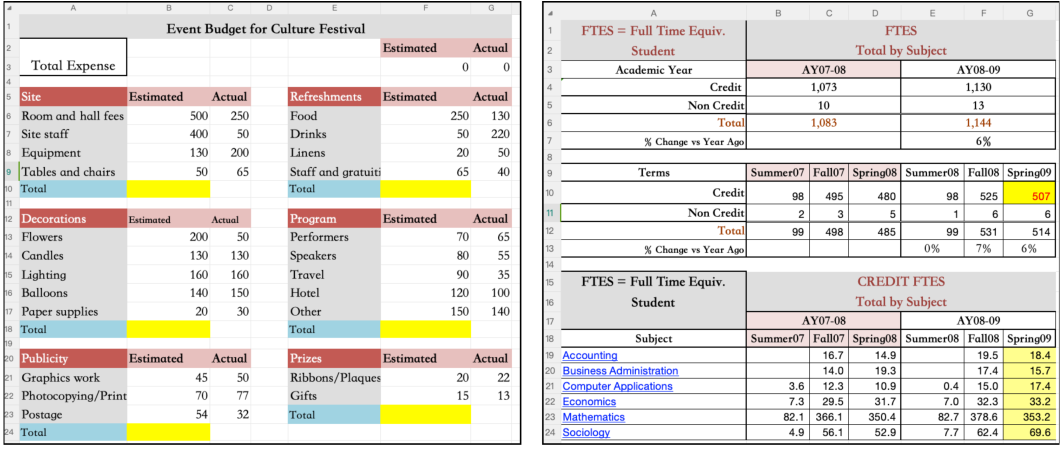
5.1 Task Definitions

This work focuses on two spreadsheet-specific tasks: Natural Language to Semantic Range (NL2SR) and Natural Language to Formula (NL2Formula). These tasks are designed to evaluate a model’s ability to interpret and manipulate spreadsheets, capturing their structural and semantic complexity. NL2SR involves mapping natural language queries to precise cell ranges in a spreadsheet, evaluated by the accuracy of the identified range. NL2Formula requires generating syntactically correct and semantically accurate formulas from natural language queries, assessed by the correctness of the formula’s output in context.

These tasks are fundamental because spreadsheets combine structured data (e.g., cell grids, references) with semantic intent (e.g., calculations, aggregations). NL2SR tests a model’s understanding of spatial relationships and data organization within the table structure, critical for tasks like data selection or filtering. NL2Formula evaluates the ability to translate semantic intent into executable formulas, requiring both syntactic precision and contextual reasoning. High-quality training and test data are essential to address these challenges, as they enable models to learn the intricate interplay of structure and semantics, overcoming the primary bottleneck in spreadsheet intelligence. Examples of these tasks are illustrated in Figure 2.

5.2 Dataset Construction

We construct a spreadsheet corpus comprising training, validation, and test sets, using tailored strategies to ensure data quality for NL2SR and NL2Formula tasks.



Query: Detailed Budget Situation of Cultural Festival Expenses

	A	B	C	D	E	F	G
1	Event Budget for Culture Festival						
2	Total Expense					Estimated	Actual
3						0	0
4							
5	Site	Estimated	Actual		Refreshments	Estimated	Actual
6	Room and hall fees	500	250		Food	250	130
7	Site staff	400	50		Drinks	50	220
8	Equipment	130	200		Linens	20	50
9	Tables and chairs	50	65		Staff and gratuiti	65	40
10	Total				Total		
11							
12	Decorations	Estimated	Actual		Program	Estimated	Actual
13	Flowers	200	50		Performers	70	65
14	Candles	130	130		Speakers	80	55
15	Lighting	160	160		Travel	90	35
16	Balloons	140	150		Hotel	120	100
17	Paper supplies	20	30		Other	150	140
18	Total				Total		
19							
20	Publicity	Estimated	Actual		Prizes	Estimated	Actual
21	Graphics work	45	50		Ribbons/Plaques	20	22
22	Photocopying/Print	70	77		Gifts	15	13
23	Postage	54	32		Total		
24	Total						

Label: B12, F12, B20, F20, B26, F25
(Yellow Regions)

Query: Calculate the Credit enrollment for Spring 2009

	A	B	C	D	E	F	G
1	FTES = Full Time Equiv.	FTES					
2	Student	Total by Subject					
3	Academic Year	AY07-08			AY08-09		
4	Credit	1,073			1,130		
5	Non Credit	10			13		
6	Total	1,083			1,144		
7	% Change vs Year Ago				6%		
8							
9	Terms	Summer07	Fall07	Spring08	Summer08	Fall08	Spring09
10	Credit	98	495	480	98	525	507
11	Non Credit	2	3	5	1	6	6
12	Total	99	498	485	99	531	514
13	% Change vs Year Ago				0%	7%	6%
14							
15	FTES = Full Time Equiv.	CREDIT FTES					
16	Student	Total by Subject					
17		AY07-08			AY08-09		
18	Subject	Summer07	Fall07	Spring08	Summer08	Fall08	Spring09
19	Accounting		16.7	14.9		19.5	18.4
20	Business Administration		14.0	19.3		17.4	15.7
21	Computer Applications	3.6	12.3	10.9	0.4	15.0	17.4
22	Economics	7.3	29.5	31.7	7.0	32.3	33.2
23	Mathematics	82.1	366.1	350.4	82.7	378.6	353.2
24	Sociology	4.9	56.1	52.9	7.7	62.4	69.6

Label: =SUM(G19:G25)
(Yellow Regions)

Figure 2: Examples of NL2SR (left) and NL2Formula (right) tasks. The left side illustrates mapping a natural language query to a semantic cell range, while the right side shows generating a formula from a natural language query.

Training Set Generation The training set is constructed through an iterative rejection sampling framework (Algorithm 1). For each spreadsheet target (formula or range), we generate a context prompt and sample $k = 5$ candidate queries using Gemini-Flash-2.0 [21](generation temperature $\tau_g = 0.7$), balancing diversity and coherence. Each candidate is evaluated by Claude-3.7-Sonnet [22] following the LLM-as-a-Judge protocol, scoring four dimensions: clarity (30%), accuracy (30%), conciseness (20%), and completeness (20%). Candidates achieving a composite quality score of at least $\gamma = 0.7$ are retained for dataset augmentation.

Test Set Generation The test set, consisting of 240 samples, is generated using Claude-3.7-Sonnet for both query generation and scoring, following a process similar to the training set but with an additional human review step. Queries are generated with the same parameters ($k = 5$, $\tau_g = 0.7$) and scored using the same four-dimensional criteria. Human experts review and refine the queries to ensure correctness and relevance, enhancing the test set's reliability for benchmarking.

Algorithm 1 Iterative Query Generation Algorithm

```
Require: Original spreadsheet dataset $\mathcal{D}$, generation model $M_g$, scoring model $M_s$
Ensure: Augmented dataset $\mathcal{D}_{aug}$
for each spreadsheet $T \in \mathcal{D}$ do
    for each target $o \in T.targets$ do
        /* Target: formula (Formula) or range (Range) */
        Construct context prompt $p \leftarrow$ generate_prompt$(T, o)$
        Generate candidate queries $\{q_i\}_{i=1}^{k} \leftarrow M_g(p)$
        Compute quality scores $s_i \leftarrow M_s(q_i, o, T)$
        Filter high-quality samples $\mathcal{Q} \leftarrow \{q_i \mid s_i \geq \gamma\}$
        Add training pairs $\mathcal{D}_{aug} \leftarrow \mathcal{D}_{aug} \cup \{(q, o, T) \mid q \in \mathcal{Q}\}$
    end for
end for
```

Dataset Statistics The training and validation sets are split in a 9:1 ratio, with statistics summarized in Table 2. The test set is curated separately to ensure diversity and quality.

Table 2: Dataset Statistics for NL2Formula and NL2SemanticRange Tasks (Token Lengths)

Task	**Input Length**				**Output Length**			
	Mean	**Min**	**Max**	**Med**	**Mean**	**Min**	**Max**	**Med**
NL2Formula								
Train ($n = 1,957$)	4,400	474	16,382	2,904	28	13	335	21
Valid ($n = 210$)	4,405	502	16,194	3,192	25	13	90	19
NL2SemanticRange								
Train ($n = 2,204$)	5,076	1,220	16,364	3,758	16	15	19	17
Valid ($n = 239$)	5,103	1,277	16,376	3,782	16	15	19	17

5.3 Experimental Setup

This subsection outlines the fine-tuning strategies, baseline models, and evaluation metrics used to assess model performance on the NL2SR and NL2Formula tasks.

Fine-Tuning Strategies Fine-tuning is performed on an NVIDIA 8xA800 GPU setup using two approaches. The first, LoRA[23], employs rank 16, targeting all parameters, with the AdamW optimizer (learning rate 2×10^{-5}, cosine schedule, 0.1 warmup ratio), batch size 1 per device, 8 gradient accumulation steps, 5 epochs, and bf16 precision. The second, full-parameter fine-tuning, uses the same hyperparameters with DeepSpeed optimizations for efficiency. We evaluate single-task fine-tuning (NL2SR or NL2Formula) and mixed-data fine-tuning (both tasks) to analyze multi-task performance trade-offs.

Baseline Models The evaluation includes five baseline models selected for their diversity in scale, architecture, and accessibility. LLaMA-3.1-8B-Instruct [24] and Qwen2.5-7B-Instruct [25] are efficient open-source models, valued for academic and industrial applications, with Qwen2.5-7B offering strong multilingual capabilities. LLaMA-3.3-70B and Qwen2.5-72B, larger open-source models, provide insights into scaling effects. GPT-4o[26], a powerful closed-source model, serves as a high-performance benchmark. This selection balances computational feasibility, open-source availability, and cutting-edge performance, enabling a comprehensive comparison across model sizes and training paradigms.

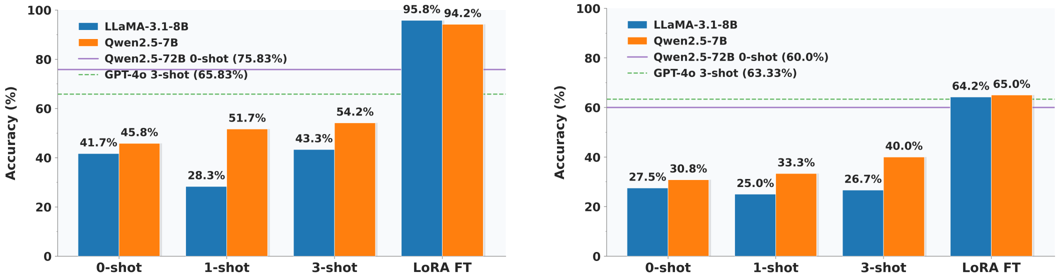
Evaluation Metrics Performance is measured using accuracy, defined as: (1) for NL2SR, the proportion of correct range mappings; (2) for NL2Formula, the proportion of formulas that exactly match the ground truth (exact match required). Both tasks are evaluated on a 240-sample test set.

5.4 Experimental Results

Few-Shot Performance Table 3 compares the accuracy of baselines in zero-shot, one-shot, and three-shot settings across NL2SR and NL2Formula tasks, based on 120 test samples per task. Here, Qwen2.5-72B achieves the highest NL2SR accuracy (73.33–75.83%), surpassing GPT-4o (65.83–69.17%) across all few-shot settings. For NL2Formula, GPT-4o leads (63.33–64.17%), with Qwen2.5-72B close behind (59.17–61.67%). Smaller models, LLaMA-3.1-8B and Qwen2.5-7B, show lower performance (25.00–54.17%), indicating limitations in zero-shot and few-shot generalization.

Table 3: Accuracy Comparison of Models in Few-Shot Settings (%)

Model	**NL2SR**			**NL2Formula**		
	0-shot	**1-shot**	**3-shot**	**0-shot**	**1-shot**	**3-shot**
GPT-4o	66.67	69.17	65.83	**63.33**	**64.17**	**63.33**
LLaMA-3.1-8B	41.67	28.33	43.33	27.50	25.00	26.67
Qwen2.5-7B	45.83	51.67	54.17	30.83	33.33	40.00
LLaMA-3.3-70B	66.67	67.50	70.00	47.50	59.17	57.50
Qwen2.5-72B	**75.83**	**74.17**	**73.33**	60.00	59.17	61.67



(a) Accuracy comparison on NL2SR task.

(b) Accuracy comparison on NL2Formula task.

Figure 3: Performance of LoRA fine-tuning versus few-shot learning. For both NL2SR and NL2Formula tasks, LoRA fine-tuning provides a substantial accuracy boost, outperforming all few-shot settings and surpassing strong baselines from larger models.

Fine-Tuning Results Figures 3a and 3b compare LLaMA-3.1-8B and Qwen2.5-7B across zero-shot, one-shot, three-shot, and LoRA fine-tuning settings. For NL2SR, Qwen2.5-7B outperforms LLaMA-3.1-8B in few-shot settings, with both reaching near 95% accuracy after LoRA fine-tuning. For NL2Formula, Qwen2.5-7B maintains a slight edge, both surpass the performance of GPT-4o.

To further explore LLaMA-3.1-8B's fine-tuning strategies, Figure 4 visualizes the test accuracy of LoRA and Full-Param approaches across NL2SR, NL2Formula, and Mixed data types for both tasks, with darker colors indicating higher performance. LoRA fine-tuning on NL2SR data achieves the highest NL2SR accuracy (97.50%) but significantly degrades NL2Formula performance (5.00%), as shown in Figure 4. Conversely, Full-Param fine-tuning on NL2Formula data yields the best NL2Formula accuracy (71.67%) while maintaining competitive NL2SR performance (37.50%). Mixed-data strategies, particularly Full-Param, balance both tasks effectively (96.67% NL2SR, 70.00% NL2Formula), highlighting the dataset's quality and the trade-offs between LoRA's efficiency and Full-Param's multi-task robustness.

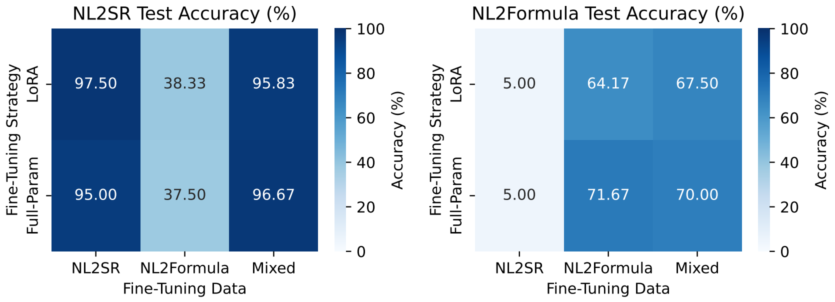


Figure 4: Heatmaps comparing fine-tuning strategies (LoRA and Full-Param) on LLaMA-3.1-8B across NL2SR, NL2Formula, and Mixed data types for NL2SR and NL2Formula tasks.

6 Conclusion

We introduced Sheetpedia, a large-scale, diverse, and formula-rich spreadsheet corpus comprising over 290,000 unique worksheets, significantly surpassing existing datasets in scale, diversity, and semantic richness. Sheetpedia serves as a valuable resource for spreadsheet intelligence, enabling the development and benchmarking of advanced NLP models through two newly introduced tasks: Natural Language to Semantic Range (NL2SR) and Natural Language to Formula (NL2Formula). The rejection sampling-based fine-tuning strategy demonstrated strong performance gains, achieving 97.5% accuracy on NL2SR and 71.7% on NL2Formula. Sheetpedia, along with its benchmarks and methods, sets a new foundation for future research in spreadsheet understanding, fostering improvements in both theoretical modeling and practical application.

7 Limitations

Despite these advancements, our work also faces certain limitations. The corpus predominantly contains English-language spreadsheets, potentially limiting applicability to multilingual contexts. Additionally, our deduplication and formula filtering methods might omit smaller spreadsheets or complex cross-sheet formulas, which may impact dataset diversity and representativeness. Furthermore, while our benchmarks and tasks are comprehensive, they may not fully capture all intricacies and challenges encountered in real-world spreadsheet usage. Future research could extend Sheetpedia by incorporating multilingual data, refining deduplication and formula inclusion methods, and broadening task coverage to enhance the dataset's generalizability and practical relevance.

Acknowledgments and Disclosure of Funding

The authors wish to thank the anonymous reviewers for their constructive feedback. This work was supported by the Beijing Natural Science Foundation (No. L253020) and the National Natural Science Foundation of China (No. 62036001). This research was also supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative (AISG Award No. AISG-NMLP-2024-002), and by the Ministry of Education, Singapore, under its AcRF Tier 2 Funding (Proposal ID: T2EP20123-0052). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation or the Ministry of Education, Singapore.

References

[1] M. Fisher and G. Rothermel. The euses spreadsheet corpus: A shared resource for supporting experimentation with spreadsheet dependability mechanisms. *ACM SIGSOFT Software Engineering Notes*, 30(4):1–5, 2005.

[2] Felienne Hermans and Emerson Murphy-Hill. Enron's spreadsheets and related emails: A dataset and analysis. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, volume 2, pages 7–16, 2015.

[3] Titus Barik, Kevin Lubick, Justin Smith, John Slankas, and Emerson Murphy-Hill. Fuse: a reproducible, extendable, internet-scale corpus of spreadsheets. In *2015 IEEE/ACM 12th Working Conference on Mining Software Repositories*, pages 486–489. IEEE, 2015.

[4] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. Turl: Table understanding through representation learning. *Proceedings of the VLDB Endowment*, 13(11):1985–1994, 2020.

[5] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, 2018. Association for Computational Linguistics.

[6] Wei Zhao, Zhitao Hou, Siyuan Wu, Yan Gao, Haoyu Dong, Yao Wan, Hongyu Zhang, Yulei Sui, and Haidong Zhang. Nl2formula: Generating spreadsheet formulas from natural language queries. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 2377–2388, St. Julian's, Malta, 2024. Association for Computational Linguistics.

[7] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[8] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex

Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021.

[9] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[10] Yutao Zhang, Yilin Wang, and Yilin Zhang. Flame: A small language model for spreadsheet formulas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[11] Oscar Romero, Elvis Koci, Maik Thiele, and Wolfgang Lehner. Deco: A dataset of annotated spreadsheets for layout and table recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 139–144, Sydney, Australia, September 2019. IEEE. Presents the DECO dataset with 1,165 annotated spreadsheets for layout and table recognition.

[12] Julian Eberius, Christoper Werner, Maik Thiele, Katrin Braunschweig, Lars Dannecker, and Wolfgang Lehner. Deexcelerator: A framework for extracting relational data from partially structured documents. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 2477–2480, San Francisco, CA, USA, October 2013. ACM. Introduces the DeExcelerator framework for extracting relational data from spreadsheets and HTML tables.

[13] Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online, 2020. Association for Computational Linguistics.

[14] Sumit Gulwani. Automating string processing in spreadsheets using input-output examples. In *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 317–326, 2011.

[15] Xinyun Chen, Petros Maniatis, Rishabh Singh, Charles Sutton, Hanjun Dai, Max Lin, and Denny Zhou. Spreadsheetcoder: Formula prediction from semi-structured context. In *Proceedings of the 38th International Conference on Machine Learning*, pages 1661–1672, 2021.

[16] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *ArXiv*, abs/1709.00103, 2017.

[17] Derek Flood, Kevin Mc Daid, and Fergal Mc Caffery. Nlp-sir: A natural language approach for spreadsheet information retrieval. *arXiv preprint arXiv:0908.1193*, 2009.

[18] Zhoujun Cheng, Haoyu Dong, Ran Jia, Pengfei Wu, Shi Han, Fan Cheng, and Dongmei Zhang. FORTAP: Using formulas for numerical-reasoning-aware table pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1093–1106, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[19] Andrei Z. Broder. On the resemblance and containment of documents. *Proceedings of the Compression and Complexity of Sequences*, pages 21–29, 1997. Introduced Minhash for estimating Jaccard similarity between sets.

[20] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing (STOC '98)*, pages 604–613. ACM, 1998. Foundational work on Locality-Sensitive Hashing (LSH) for approximate nearest neighbor search.

[21] DeepMind. Gemini-2.0-flash, 2025. Accessed: March 31, 2025.

[22] Anthropic. Claude-3.7-sonnet research, 2025. Accessed: March 31, 2025.

[23] Edward J Hu, Yelong Shen, Phil Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[24] Meta AI. Llama 3: The next generation of open, large language models, 2024. Accessed: 2025-02-28.

[25] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report, 2024.

[26] Open AI. Chatgpt, 2024. Accessed: 2025-02-28.

8 Technical Appendices and Supplementary Material

This appendix provides supplementary materials to support the main findings and methodologies presented in the Sheetpedia corpus study. It includes detailed visualizations, statistical tables, and prompt templates used for data generation and evaluation. These materials offer deeper insights into the corpus's characteristics, formula distributions, language diversity, and the construction of the NL2SR and NL2Formula tasks. Each figure and table is accompanied by a brief description to guide readers through the content, ensuring accessibility and clarity for researchers aiming to replicate or extend our work.

Data Sanitization and PII Masking To ensure data privacy and ethical compliance, we implemented a comprehensive pipeline to identify and mask Personally Identifiable Information (PII) within the corpus. The process was executed using a Python script leveraging multi-core processing via the `concurrent.futures` library to efficiently handle the large volume of XLSX files. For each spreadsheet, the script systematically iterated through every cell and applied a series of masking rules.

The core of this process involved a multi-faceted approach to PII detection and replacement:

- **Person Names:** We utilized the `spaCy` natural language processing library (specifically, the `en_core_web_sm` model) for Named Entity Recognition (NER). Detected person names were replaced with a consistent, file-specific placeholder. For instance, the first unique name found in a file would be replaced with `[Person_1]`, the second with `[Person_2]`, and so on. This method ensures that all occurrences of the same name within a single document are mapped to the same placeholder, preserving contextual integrity while guaranteeing anonymity.
- **Email Addresses:** A regular expression was used to identify email addresses. To balance privacy with the potential utility of domain information, we masked the local-part (username) of the email with a generic `[USER]` token while preserving the domain (e.g., `example@edu.com` becomes `[USER]@edu.com`).
- **Phone Numbers:** Phone numbers were also detected using regular expressions. The script masked the initial digits with a `[PHONE_PREFIX]` token but retained the final four digits. This approach removes the identifiable portion of the number while leaving a non-identifiable suffix that could potentially be used for record linkage without compromising privacy.

Overview of Language Distribution Figure 5 visualizes the language distribution of the deduplicated Sheetpedia corpus, highlighting the predominance of English and the presence of minor languages. This chart complements the language statistics in Section 4.3, providing a clear visual representation of linguistic diversity.

Formula Pattern Insights Figure 6 illustrates the distribution of the top 20 Excel formula patterns in the deduplicated corpus, using a logarithmic scale to emphasize the prevalence of common functions like `IF` and `SUM`. This visualization supports the analysis in Section 4.1, revealing key patterns in formula usage.

Language Distribution Details The table below presents the detailed language distribution of the deduplicated Sheetpedia corpus, listing the top 15 languages by count and percentage. This complements Figure 5 and Section 4.3, offering precise statistics for researchers studying linguistic diversity in spreadsheet data.

Formula Usage Before Deduplication This table lists the top 30 Excel formula patterns in the raw corpus before deduplication, showcasing the frequency of key functions like `IF` and `SUM`. It provides a baseline for understanding formula prevalence, as discussed in Section 4.1.

Formula Usage After Deduplication This table details the top 30 Excel formula patterns in the deduplicated corpus, reflecting changes in frequency after removing redundant worksheets. It supports the analysis in Section 4.1 and highlights the impact of deduplication on formula distribution.

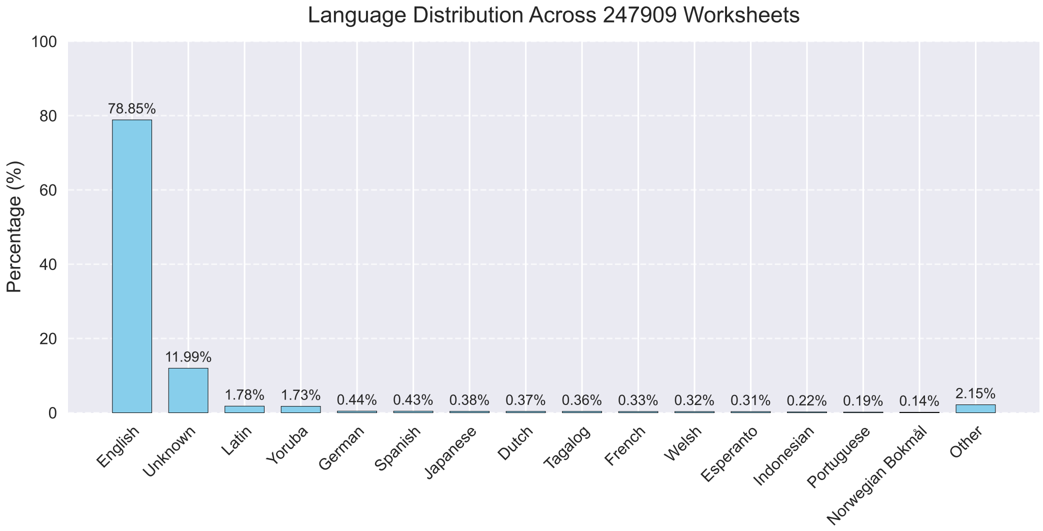


Figure 5: Language distribution of deduplicated corpus (percentage)

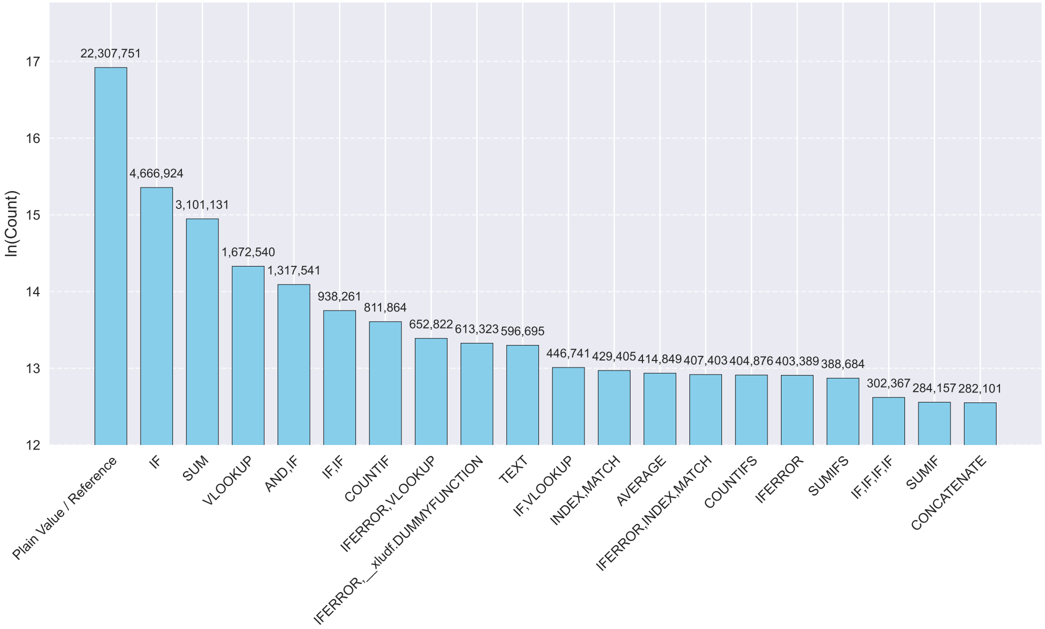


Figure 6: Distribution of formula patterns in deduplicated corpus

Table 4: Language distribution in the corpus (Top 15)

Language Code	Count	Percentage (%)
en	195,479	78.85
unknown	29,726	11.99
la	4,423	1.78
yo	4,281	1.73
de	1,093	0.44
es	1,061	0.43
ja	939	0.38
nl	915	0.37
tl	901	0.36
fr	818	0.33
cy	793	0.32
eo	768	0.31
id	554	0.22
pt	462	0.19
nb	357	0.14

Distribution of Merged Cells To understand the structural complexity of the spreadsheets within the corpus, we analyzed the prevalence of merged cells—a common feature used for formatting and presentation. Our analysis of all 290,509 successfully processed worksheets revealed that 56,494 of them contained at least one merged cell. This corresponds to approximately 19.45% of the total worksheets. The significant presence of merged cells highlights a key challenge in automated spreadsheet processing, as such structures can complicate data extraction and table recognition. This finding underscores the necessity for robust parsing techniques that can accurately interpret visually formatted layouts, a critical consideration for developing effective spreadsheet intelligence models.

Prompt for NL2SR Query Generation The prompt shown in Figure 7 outlines the instructions provided to the language model for generating natural language queries for the NL2SR task. It specifies the input format, guidelines for query precision, and output structure, as described in Section 5.2.

Prompt for NL2Formula Query Generation The prompt in Figure 8 details the instructions for generating natural language queries for the NL2Formula task, specifying the input format and requirements for capturing formula semantics. This supports the dataset construction process outlined in Section 5.2.

Prompt for NL2SR Query Scoring The prompt in Figure 9 specifies the criteria and scoring rubric for evaluating the quality of NL2SR queries, as used in the dataset construction pipeline (Section 5.2). It ensures queries meet standards for clarity, accuracy, conciseness, and completeness.

Prompt for NL2Formula Query Scoring The prompt in Figure 10 defines the evaluation criteria for NL2Formula queries, ensuring they accurately reflect formula logic, as described in Section 5.2.

Prompt for NL2SR Testing The prompt in Figure 11 provides the instructions for testing models on the NL2SR task, specifying how to map user queries to cell ranges, as used in Section 5.3.

Prompt for NL2Formula Testing The prompt in Figure 12 outlines the instructions for testing models on the NL2Formula task, detailing how to generate formulas from user queries, as described in Section 5.3.

Chain-of-Thought (CoT) Prompting Analysis To assess the applicability of Chain-of-Thought (CoT) prompting for spreadsheet-related tasks, we conducted an exploratory analysis on NL2SR and NL2Formula using baseline LLMs in a zero-shot setting. CoT was induced by appending the phrase "Please think step by step." to the standard prompts. The results, summarized in Table 7, show that CoT does not consistently improve performance and, in many cases, degrades accuracy compared

Table 5: Top 30 Most Used Excel Formulas: Before Deduplication

Formula	Count
Plain Formula	30,798,987
IF	8,121,168
SUM	5,723,627
AND,IF	3,155,740
VLOOKUP	2,547,198
IF,IF	1,562,982
IFERROR,VLOOKUP	1,163,168
COUNTIF	1,162,861
TEXT	939,481
AVERAGE	824,825
INDEX,MATCH	814,055
IFERROR,INDEX,MATCH	784,835
IFERROR,_xludf.DUMMYFUNCTION	781,777
IFERROR	656,478
IF,VLOOKUP	639,355
IF,IF,IF,IF	567,283
SUMIF	540,015
CHAR,CHAR,CLEAN,SUBSTITUTE,SUBSTITUTE,TRIM	539,178
SUMIFS	534,398
COUNTIFS	519,826
CONCATENATE	510,803
IF,SUM	488,113
COLUMNS,MID,REPT,SUBSTITUTE,TRIM	459,785
WEEKDAY	429,244
IF,IF,IF,IF,IF,IF,IF,IF	369,177
MONTH	368,708
LEFT	363,611
_xlfn.XLOOKUP	343,869
IF,ISBLANK	336,105
IF,IF,IF	335,063

to standard zero-shot evaluation. We hypothesize that this outcome is because our tasks primarily require direct semantic mapping and precise formula generation, rather than the multi-step logical or numerical reasoning where CoT typically excels.

8.1 Accessing Sheetpedia

The Sheetpedia corpus and associated benchmarks are available for research purposes. Researchers interested in accessing the dataset or exploring the NL2SR and NL2Formula tasks can contact the authors or visit the project repository at https://huggingface.co/datasets/tianzl66/Sheetpedia and https://github.com/TTtianTT/Sheetpedia. We hope these supplementary materials facilitate further advancements in spreadsheet intelligence and NLP research.

Table 6: Top 30 Most Used Excel Formulas: After Deduplication

Formula	Count
Plain Formula	22,307,751
IF	4,666,924
SUM	3,101,131
VLOOKUP	1,672,540
AND,IF	1,317,541
IF,IF	938,261
COUNTIF	811,864
IFERROR,VLOOKUP	652,822
IFERROR,__xludf.DUMMYFUNCTION	613,323
TEXT	596,695
IF,VLOOKUP	446,741
INDEX,MATCH	429,405
AVERAGE	414,849
IFERROR,INDEX,MATCH	407,403
COUNTIFS	404,876
IFERROR	403,389
SUMIFS	388,684
IF,IF,IF,IF	302,367
SUMIF	284,157
CONCATENATE	282,101
CHAR,CHAR,CLEAN,SUBSTITUTE,SUBSTITUTE,TRIM	259,789
COUNTIF,IF	256,165
IF,SUM	243,636
IF,ISBLANK	242,460
LEFT	241,113
IF,IF,IF,IF,IF,IF,IF,IF,IF	234,509
COLUMNS,MID,REPT,SUBSTITUTE,TRIM	224,934
_xlfn.XLOOKUP	222,150
IF,IFERROR	216,615
MONTH	214,274

Table 7: Performance comparison of Chain-of-Thought (CoT) prompting versus standard zero-shot prompting on NL2SR and NL2Formula tasks.

Model	NL2SR (CoT)	NL2SR (0-shot)	NL2Formula (CoT)	NL2Formula (0-shot)
GPT-4o	67.56	66.67	50.00	63.33
LLaMA-3.1-8B	12.50	41.67	18.33	27.50
LLaMA-3.3-70B	54.17	66.67	28.57	47.50
Qwen2.5-7B	31.67	45.83	26.63	30.83
Qwen2.5-72B	73.57	75.83	56.03	60.00

NL2SR Query Generation Prompt

```
System: Given the content of the spreadsheet and the selected cell
ranges, your task is to generate an accurate query that precisely
reflects the data within these ranges. The query should reveal the
semantics of the selected cell ranges. When the query is given, we
should be able to pinpoint a unique cell range in the sheet.
Importantly, this unique range should be identical to the selected
cell range. The generated query must solely reflect the data within
the range and should not include any other cells. While generating
an accurate query, you should aim to keep the query as concise as
possible. Avoid referring to the address and content of the
selected cell. Avoid lengthy queries. The generated query should
be in line with my needs as a user, as if it's a succinct request
made during a conversation with the model.

The sheet data will be provided to you in a format as follows: Each data
cell in the spreadsheet is represented by a pair consisting of the cell
address and cell content, separated by a comma, such as 'A1,Year'. This
means that 'A1' is the cell's address, and 'Year' is its content. Cells
are separated by a vertical bar ('|'), like 'A1,Year|A2,Profit'. The cell
content can be empty, resulting in cell data like 'A1,|A2,Profit'. Cells
are organized in row-major order, with different rows in the spreadsheet
separated by line breaks. If there are merged cells in the sheet, they
are split into multiple cells and only the first cell will be filled with
content, other cells will be left as blank. You can visualize the sheet
data as a matrix of cells. Following the matrix, all the merged cells are
provided in the format '<upper-left address>:<lower-right address>,' like
'A3:C3', with each line representing one merged cell.

Example User:

{Example i:}
Original Sheet Content:
{Example Sheet}

The selected cell ranges: {Example Range Info}

Example Assistant:

Corresponding Query: {Example Query}

User:
Now your turn,

My Sheet Content:
{Sheet String}

The selected cell ranges: {Range Info}

Output Format: Please generate a query based on the selected cell ranges
. The output should be provided in a JSON format, with a key of 'query'
and the generated query as the corresponding value.

Tell me the content in the selected cell ranges first and then generate
the corresponding query, keep the query precise and concise.
```

Figure 7: NL2SR Query Generation Prompt

NL2Formula Query Generation Prompt

```
System: You will receive sheet data along with an Excel formula. Your
task is to generate a query that can reflect the semantics of the
corresponding formula.

The sheet data will be provided to you in a format as follows: Each data
cell in the spreadsheet is represented by a pair consisting of the cell
address and cell content, separated by a comma, such as ’A1,Year’. This
means that ’A1’ is the cell’s address, and ’Year’ is its content. Cells
are separated by a vertical bar (’|’), like ’A1,Year|A2,Profit’. The
cell content can be empty, resulting in cell data like ’A1,|A2,Profit’.
Cells are organized in row-major order, with different rows in the
spreadsheet separated by line breaks. If there are merged cells in the
sheet, they are split into multiple cells and only the first cell will
be filled with content, other cells will be left as blank. You can
visualize the sheet data as a matrix of cells. Following the matrix,
all the merged cells are provided in the format ’<upper-left address>:
<lower-right address>,’ like ’A3:C3’, with each line representing one
merged cell.

{Output Format}

{Guidelines}

Example User:

Original Sheet Content:

{Example Sheet}

{
    "formula": Example Formula,
    "address": Example Address
}

Example Assistant:

Corresponding Query: {Example Query}

User:
Now your turn,

{
    "sheetString": {Sheet String},

    "Formula Address": {

        "formula": {Formula},
        "address": {Address}

    }

}
```

Figure 8: NL2Formula dataset query generation prompt

NL2SR Query Scoring Prompt

```
[Task]
Evaluate the quality of this natural language query for describing an
Excel cell range. Consider:

1. Clarity (0-3): Is the purpose/use of the cell range unambiguous?
2. Accuracy (0-3): Does the query correctly reflect the cell range’s
application (e.g., data processing, references)?
3. Conciseness (0-2): Is it free of unnecessary details?
4. Completeness (0-2): Are all the meanings of the cells included in the
query?

[Scoring Rubric]
• 9-10: Perfectly describes the cell range’s scope and purpose
• 7-8: Minor inaccuracies or omissions
• 5-6: Partial accuracy with vague references
• <5: Fails to characterize the cell range

[Input]
Cell Range: {cell_range}
Context: {context}
Query: {query}

[Output Format]
Strict JSON:
{
    "score": total_score,
    "breakdown": {
        "clarity": score,
        "accuracy": score,
        "conciseness": score,
        "completeness": score
    },
    "rationale": "Brief explanation"
}

IMPORTANT:
1. Do NOT include any additional text before or after the JSON object
2. Ensure the JSON is valid and properly formatted
```

Figure 9: NL2SR Query Scoring Prompt

NL2Formula Query Scoring Prompt

```
[Task]
Evaluate the quality of this natural language query for describing an
Excel formula. Consider:

1. Clarity (0-3): Is the calculation purpose unambiguous?
2. Accuracy (0-3): Does it match the formula's logic?
3. Conciseness (0-2): Is it free of redundant information?
4. Completeness (0-2): Are cell ranges/specifics included?

[Scoring Rubric]
• 9-10: Perfectly captures all formula aspects
• 7-8: Minor omissions but generally accurate
• 5-6: Partial accuracy with some ambiguities
• <5: Significant discrepancies

[Input]
Formula: {formula}
Context: {context}
Query: {query}

[Output Format]
Strict JSON:
{
    "score": total_score,
    "breakdown": {
        "clarity": score,
        "accuracy": score,
        "conciseness": score,
        "completeness": score
    },
    "rationale": "Brief explanation"
}

IMPORTANT:
1. Do NOT include any additional text before or after the JSON object
2. Ensure the JSON is valid and properly formatted
```

Figure 10: NL2Formula Query Scoring Prompt

NL2SR Test Prompt

[Task]
As a data scientist, you are presented with a spreadsheet and a user query. Your task is to interpret the spreadsheet and identify the specific cell range that corresponds to the given user query. Ensure the identified range should be as precise as possible, and should not include any other irrelevant cells.

The sheet data will be provided to you in a format as follows: Each data cell in the spreadsheet is represented by a pair consisting of the cell address and cell content, separated by a comma, such as ’A1,Year’. This means that ’A1’ is the cell’s address, and ’Year’ is its content. Cells are separated by a vertical bar (’|’), like ’A1,Year|A2,Profit’. The cell content can be empty, resulting in cell data like ’A1,|A2,Profit’. Cells are organized in row-major order, with different rows in the spreadsheet separated by line breaks. If there are merged cells in the sheet, they are split into multiple cells and only the first cell will be filled with content, other cells will be left as blank. You can visualize the sheet data as a matrix of cells. Following the matrix, all the merged cells are provided in the format ’<upper-left address>:<lower-right address>,’ like ’A3:C3’, with each line representing one merged cell.

Output Format: Please generate the cell range based on the user query. The output should be provided in a JSON format, enclosed in ‘‘‘Json and ‘‘‘ markdown code blocks, with a key of ’cell range’ and the generated cell range as the corresponding value.

Figure 11: NL2SR Test Prompt

NL2Formula Test Prompt

[Task]
As a data scientist, your task is to generate a formula based on a specific cell in a given spreadsheet. The query of the formula is provided. Use the spreadsheet and the query to generate the correct formula.

The sheet data will be provided to you in a format as follows: Each data cell in the spreadsheet is represented by a pair consisting of the cell address and cell content, separated by a comma, such as ’A1,Year’. This means that ’A1’ is the cell’s address, and ’Year’ is its content. Cells are separated by a vertical bar (’|’), like ’A1,Year|A2,Profit’. The cell content can be empty, resulting in cell data like ’A1,|A2,Profit’. Cells are organized in row-major order, with different rows in the spreadsheet separated by line breaks. If there are merged cells in the sheet, they are split into multiple cells and only the first cell will be filled with content, other cells will be left as blank. You can visualize the sheet data as a matrix of cells. Following the matrix, all the merged cells are provided in the format ’<upper-left address>:<lower-right address>,’ like ’A3:C3’, with each line representing one merged cell.

Output Format: Please generate the formula based on the user query. The output should be provided in a JSON format, enclosed in ‘‘‘Json and ‘‘‘ markdown code blocks, with a key of ’formula’ and the generated formula as the corresponding value.

Figure 12: NL2Formula Test Prompt

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly outline the creation of Sheetpedia, a 290,509-worksheet corpus, the introduction of NL2SR and NL2Formula tasks, and the fine-tuning results (97.5% NL2SR, 71.7% NL2Formula), which are supported by detailed methodology and results in Sections 3–5.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Section 7 (Limitations) explicitly discusses the English-centric language distribution, deduplication constraints, formula filtering limitations, and the limited test set size, addressing key assumptions and scope.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: The paper does not present theoretical results or proofs, focusing instead on empirical contributions (dataset creation, task definitions, and model fine-tuning).

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: Section 5.3 (Experimental Setup) details the fine-tuning strategies (LoRA, full-parameter), baseline models, hyperparameters, and evaluation metrics. Dataset construction is described in Section 5.2, including the rejection sampling algorithm (Algorithm 1).

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We will submit the data url alongwith the paper.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5.3 (Experimental Setup) specifies data splits (9:1 train/validation, 240 test samples), hyperparameters (learning rate, optimizer, batch size), and hardware (NVIDIA 8xA800 GPU). Table 5 provides dataset statistics.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper reports accuracy metrics (Section 5.4, Figures 6–7) but does not include error bars, confidence intervals, or statistical significance tests, likely due to the deterministic nature of the tasks and the focus on exact match accuracy.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: Section 5.3 (Experimental Setup) specifies the use of an NVIDIA 8xA800 GPU setup for fine-tuning, though memory and exact execution times are not detailed. This is sufficient for the main claims but could be expanded.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The paper adheres to ethical data collection (Section 3.1, respecting robots.txt for ExcelForum crawling) and uses publicly available datasets (Enron, Fuse). No human subjects or unethical practices are involved.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: The paper focuses on technical contributions and limitations (Section 6) but does not explicitly discuss societal impacts, such as potential misuse of spreadsheet automation or accessibility benefits, likely due to its foundational research nature.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.

- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The dataset (Sheetpedia) and fine-tuned models do not pose high risks for misuse (e.g., disinformation or surveillance), as they are designed for spreadsheet tasks. No specific safeguards are needed or described.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper credits the Enron and Fuse datasets (Section 3.1) and cites libraries like pyexcel and openpyxl (Section 3.2). ExcelForum data collection respects terms of service (robots.txt). Licenses are not explicitly stated but implied to be public domain or open.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

Justification: Sheetpedia is a new asset (Section 3), and its creation is well-documented (data sources, preprocessing in Sections 3.1–3.2). Data url will be provided alongwith the paper.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: The paper does not involve crowdsourcing or human subjects, except for minimal human review in test set generation (Section 5.2), which does not require participant instructions or compensation details.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: The paper does not involve human subjects research beyond minimal expert review (Section 5.2), which does not require IRB approval or risk disclosure.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: The paper describes the use of LLMs (Gemini-Flash-2.0, Claude-3.7-Sonnet) for training and test set generation via rejection sampling (Section 5.2, Algorithm 1), which is integral to the methodology.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.