TabMeta: Table Metadata Generation with LLM-Curated Dataset and LLM-Judges

Anonymous EMNLP submission

Abstract

Recent advances in LLMs have found use in several tabular related tasks including Text2SQL, data wrangling, imputation, Q&A, and other table-related tasks. Crucially however, researchers have often overlooked the fact that the downstream data consumers are often decoupled from the data producers. Downstream data users therefore, neither precisely know which tables to request access for and make use of, nor can easily understand complex cryptic terminology (in column names, etc) employed by the data producers. Specifically, the lack of descriptive metadata for tables has emerged as a significant obstacle to effective data governance and utilization. To tackle this, our work introduces TABMETA, a new natural language task aimed at automatically generating comprehensive metadata for arbitrarily complex tables, enabling non-expert users to discover, understand and use relevant data more effectively. First, we curate a unique benchmark dataset for the TABMETA task, consisting of table descriptions and column descriptions for 302 tables spanning 30 industry domains. Second, we propose two novel tabular metadata evaluation strategies (a) a *robust and consistent* LLM-Judge based framework which aligns with human judgement and employs confidence scores suited for tabular metadata and (b) ML based metrics to capture quality of the generated metadata such as *conciseness, coherence and information gain*. Finally, we also show that our metadata enhancement framework substantially improves the performance of tabular data discovery and search by a factor of 3-4x.

1 Introduction

The last couple of years have seen a positively disruptive influence of Large Language Models (LLMs) (Zhao et al., 2023) and Foundational Models (FMs) (Bommasani et al., 2021) for enterprise scale tabular data and databases (Orr et al., 2022; Arora et al., 2023; Narayan et al., 2022). Primarily, they have found utility in a variety of tasks such as Text2SQL translation (Li et al., 2024a; Zhang et al., 2024; Sun et al., 2023), Tabular Q&A and reasoning, data wrangling, imputation and various other tasks on tables (Kong et al., 2024; Sui et al., 2024; Lei et al., 2023; Li et al., 2023b).

However, these use-cases assume that data consumers can conveniently query, retrieve, and comprehend tables for appropriate use. In reality, this assumption is often unsatisfied due to complex data governance policies and access restrictions (Khatri and Brown, 2010; Rosenbaum, 2010; Abraham et al., 2019) within organizations. Data producers, owners, and consumers belong to different verticals, and users have to request access *via* search. Unfortunately searching for tabular data, without open access to confidential information is challenging due to inconsistent terminology used by producers and consumers, such as cryptic column and table names in the column, table names (Zhang et al., 2023a), making tabular search and subsequent user understanding difficult (Figure 1).

Prior literature (Brickley et al., 2019; Li et al., 2021; Christophides et al., 2019), recommends meta data enrichment as a mechanism to alleviate the above concerns – making data assets more amenable to search and discovery. In similar vein, we propose TABMETA, a natural language task, where the goal is to enrich tabular metadata, making it easier to search and more understandable for downstream users, without exposing any confidential data present in the tables.

To enrich tabular metadata, we use LLMs to add descriptive summaries (see Figure 1) for the entire table (akin to tabular data summarization (Zhang et al., 2020a)) as well as its constituent columns which facilitates conversational search (Zamani et al., 2023) in additional to traditional keyword/semantic search. While we use LLMs to enrich the metadata, the work is broadly applicable to generative text models, such as diffusion models for text (Austin et al., 2021; Gong et al., 2022).

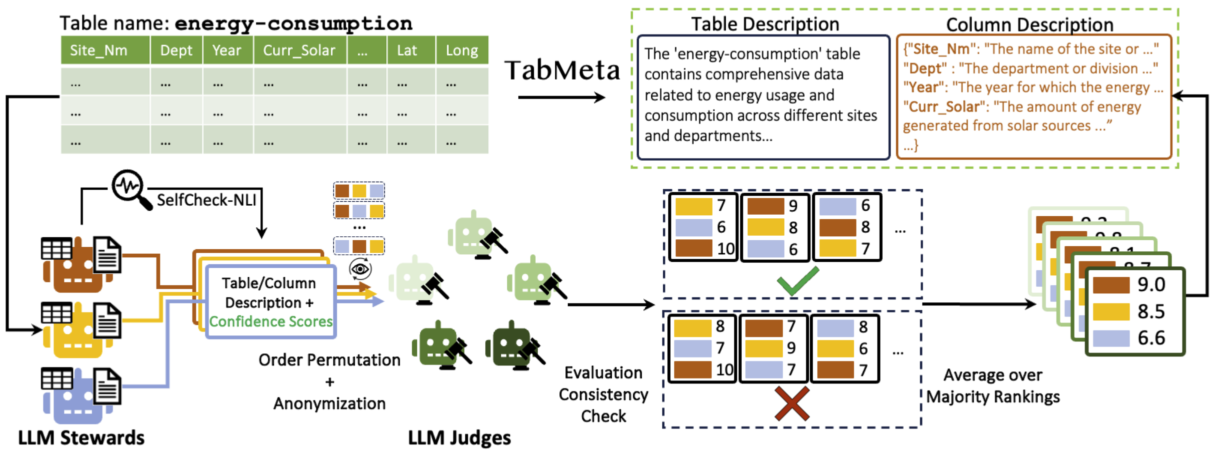

Figure 1: Schematic of the TabMeta benchmark creation pipeline.

However, the number of publicly available LLMs have grown multi-fold and LLMs are known to hallucinate (Azamfirei et al., 2023; Ye et al., 2023), producing unreliable content, specifically for cryptic/complex technical content not found in the training data (Zhang et al., 2023a). *Selecting the appropriate LLMs, generating high quality metadata and evaluating the efficacy of* TabMeta is therefore of utmost importance. Specifically, this requires a carefully curated benchmark which spans multiple industrial domains to ensure wide applicability – and to the best of our knowledge no such benchmarks currently exist. To tackle this, ***first***, we present a benchmark of 302 tables which span 30 different domains. To control the associated scale, time and costs of the evaluation process without sacrificing on quality, we carefully downsample data from multiple tabular data-sources including Kaggle, GitTables (Hulsebos et al., 2023), and BIRD-SQL (Li et al., 2023a), ensuring the selection of the most representative tables while eliminating redundancy.

Second, we present two different but complimentary evaluation mechanisms which together can help select the appropriate LLM for the table, detect hallucinations without sacrificing on informativeness, conciseness, coherence, etc of the content generated. The first of these, adapts the LLM-Judges framework (Zheng et al., 2023), where secondary LLMs act as judges that compare and evaluate the generated metadata candidates from multiple LLMs. This framework however suffers from multiple biases such as lack of consistency, self-enhancement biases and position biases. To overcome this, we craft a mechanism which leverages confidence scores specifically designed for tabular data to significantly enhance consistency and mitigate these biases. Secondly, we design and adapt multiple ML metrics, gauging Q&A-based and semantic-based precision & recall, as well as capturing/approximating various other criteria such as coherence, cohesion, information gain, conciseness. ***Third***, we show that employing our tabular metadata enrichment framework can aid BM25-based retrieval by a factor of 3-4x for keyword based search (Robertson et al., 2009)

Our key *contributions* can be summarized as:

- We introduced TabMeta, a task for table metadata generation, with a goal to aid table search, data governance in general.
- We curated a benchmark dataset for the TabMeta task, utilizing multiple LLMs in an iterative feedback driven fashion.
- We developed an LLM-based judging method leveraging confidence scores to enhance judge consistency, ensuring a more reliable and robust assessment.
- We established a set of machine learning-based metrics for performance evaluation in TabMeta task which captures diverse properties such as informativeness, conciseness, etc.

2 Preliminaries

2.1 Notation

We consider a countable set of tables across different domains (e.g. finance, automobile, pharmaceuticals, etc) $\mathcal{T} = \{t_1, t_2, ..., t_n\}$ where each table $t_i \in \mathcal{T}$ where $t_i \equiv (n_i, m_i, \phi_i)$ has n_i a set of columns in the table, m_i sampled rows (which we don't have access to i.e. $|m_i| = 0$) as well as optional existing metadata ϕ_i (such as table names, attribute/column names and data types, as well as additional metadata).

Our goal here is to generate high quality human understandable descriptions of the table, as a whole as well as human understandable descriptions of every constituent column, i.e. yield a function $f : t_i \mapsto (d_{(t_i,t)}$: table description of t_i , $d_{(t_i,c)}$:

column descriptions for $t_i) \in \mathcal{D}$, where we use $\mathcal{D}$ to denote the space of all possible generated descriptions, and $d_{t_i} \equiv (d_{(t_i,t)}, d_{(t_i,c)}) \in \mathcal{D}$ for description corresponding to table t_i.

Our desiderata is an inductive function f which works across different industrial domains – and the generated text is accurate, informative, concise and coherent. Towards this, we seek to employ LLMs, in order to take advantage of the exogenous information they can add to enrich the metadata.

2.2 LLM-Judges

LLM-as-a-judge (Zheng et al., 2023; Zhu et al., 2023; Wang et al., 2023c) offers a proxy solution to human evaluations on judging generated textual data from multiple sources (e.g. LLMs) – specifically when it is hard to acquire human experts (a.k.a gold standard) across a wide spectrum of domains. Our goal is to comparatively evaluate text generations for a given table from multiple LLMs i.e. for a table t_i, and multiple LLMs $\mathrm{LLM}^1, \ldots, \mathrm{LLM}^k$, output an ordering amongst $\{d_{t_i}^{LLM^1}, d_{t_i}^{LLM^2}, \ldots, d_{t_i}^{LLM^k}\}$, therefore serving as a proxy to identify the best LLM-generated candidate(s) for a given table t_i.

3 Benchmark Creation

We employed a multi-step procedure to curate a diverse dataset from GitTables (Hulsebos et al., 2023), BIRD-SQL (Li et al., 2023a), and Kaggle. As a part of our final dataset, we have included all tables from BIRD-SQL (incl. training and dev splits) and selected 500 tables manually from Kaggle. Since, the original GitTables dataset contained around 1M tables, we first filtered out tables with less than 10 columns to ensure a minimum level of table complexity and also removed tables with licensing issues. However, with such a large number of tables, costs associated with LLMs (e.g. GPT-4) can be exceedingly high - even without including evaluations based on LLM-as-a-judge.

To tackle this, we employ an aggressive but robust down-sampling procedure to ensure that our dataset and evaluation framework forms a reliable and cost-effective testbed for future works. Specifically, for a given schema, we independently obtain BERT embeddings for the column names. Since there is no explicit ordering of the columns in a table, we aggregate the column name embedding via mean-pooling to get embeddings for the entire schema (a permutation invariant strategy). Alternatively, more complex permutation invariant strategies (Zaheer et al., 2017; Murphy et al., 2018), can be employed if the underlying topic of the schema can only be captured jointly across all column name embeddings. Subsequently, we use k-means clustering on the table schema embeddings to identify those closest to the distinct cluster centroids as the representative examples.

Finally, we ensure we have an broad distribution of tables from different industrial domains, we use an LLM to infer the domain of each table based on its schema akin to topic modeling (Wang et al., 2023b, 2024). Our overall down-sampling framework yields a final dataset comprising 302 tables with a comprehensive coverage of 30 distinct industry domains, each table averaging 14.5 columns. Our down-sampling procedure is robust to say, that uniformly duplicating tables or removing tables from the original dataset, does not cause significant alterations to the representative samples obtained.

3.1 Metadata Enrichment by LLM Stewards

As a part of our framework, LLMs serve as data annotators and stewards, enriching table metadata utilizing only the schema details (table name and column titles) and any available metadata, while preserving security by operating without access to the table's content or any sampled data. This methodology closely resembles how humans comprehend tabular data – i.e. initiating the process by using LLMs to provide descriptive annotations for individual columns. Subsequently, these column descriptions, in conjunction with the schema and the metadata, are used towards constructing a comprehensive table description. This enriched metadata includes a high-level summary of the table's contents and identifies potential end-users and use cases, see an example in Table 5 (Appendix). Details about the prompts used are provided in Figure 8 and 9 (Appendix).

3.2 Metadata Quality Control by LLM Judges

Analogous to other human expert based evaluation tasks (Ouyang et al., 2022; Bai et al., 2022; Taori et al., 2023; Diao et al., 2023), the TabMeta task is also labor-intensive and complex nature. The complexity arises from the need to jointly understand the underlying data structure and its contextual relevance. Human evaluation of tabular metadata can be prone to inconsistency, subjectivity, and a high time investment, particularly for large and complex databases. Indeed, recent studies (Hosking et al., 2023; Huang et al., 2023; Gilardi et al., 2023) have shown that LLMs can effectively replace human evaluators in many tasks.

These factors underscore the necessity for automated evaluation solution for the TabMeta task using LLMs as judges. To ensure the quality of the table metadata generated from LLM stewards, we leverage five powerful LLM judges including GPT-4-Turbo, GPT-3.5-Turbo, Claude-v2, Claude-v1 and LLaMA-2-70B to conduct independent evaluations for the candidate annotations from LLM stewards. Another notable challenge during the evaluation is how to detect and penalize potential hallucination and non-factual statements in the table metadata generated by LLM stewards. This could be a common issue due to ambiguous or cryptic column names, or lack of informative table context. Therefore, we introduce the sentence-level confidence scores for each candidate response during LLM judging. The confidence scores were derived from: $f_{\text{conf}}(d_{t_i}) = 1 - \mathcal{S}_{\text{NLI}}(d_{t_i}) = 1 - \frac{1}{N}\sum_{n=1}^{N} P(\text{contradict}|x_{t_i}, d_{t_i}^{n}); \quad x_{t_i} \in d_{t_i}$, where x_{t_i} denotes the sentence being assessed from the description d_{t_i}, and $\mathcal{S}_{\text{NLI}}(i)$ is the sentence-level hallucination probability estimated using SelfCheck-NLI (Manakul et al., 2023). It is computed as a contradiction probability averaged over N stochastic responses from the same LLM data steward against the main response, using a DeBERTa-based textual entailment classifier (He et al., 2023) fine-tuned on MNLI (Williams et al., 2018) (we set $N = 3$). The confidence scores served as a proxy for the likelihood that the generated content was free from hallucinations. It is important to note that the stochastic responses were not provided by the judge LLMs but were instead generated concurrently with the candidates. We demonstrate through experiments that they serve as a powerful guardrail that significantly mitigates commonly observed biases in LLM judge settings (Section 3.2.1). The prompt template used for LLM judging is shown in Figure 10 (Appendix).

3.2.1 Handling Limitations of LLM Judges

Self-Enhancement Bias. When serving as judges, LLMs tend to favor candidates generated by themselves, known as self-enhancement bias (Zheng et al., 2023). During the evaluation of TabMeta task, this bias is present even we anonymize the model ID of the LLM stewards as seen in left panel of Figure 2, the corresponding judges including Claude-v1, Claude-v2, and GPT-3.5-turbo consistently exhibit a preference towards the table/column descriptions generated by themselves as data stewards. In contrast, including the sentence-level

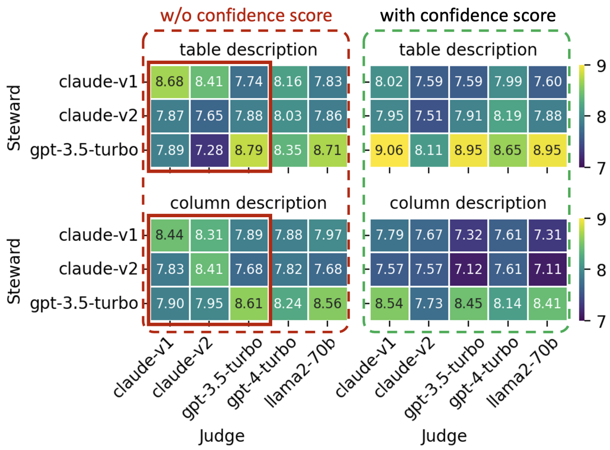

Figure 2: Overall LLM judge ratings for table and column descriptions generated by LLM stewards. Scoring scale is from 0 to 10.

confidence scores significantly alleviates this bias: the judges reflect a consistent preference towards annotations from GPT-3.5-turbo (right panel of Figure 2). This consistent preference is also corroborated in Section 5.1 from the overall enhancement in search $P@k$ and MRR by including enriched metadata from different LLM stewards (Table 2).

Position Bias. Position bias refers to the preference to the answers or candidate responses located in a certain position of the task description / prompt, when making the judgement. Top tier LLMs including GPT-4 and Claude are not immune to position bias potentially due to the architecture of autoregressive transformers and the pre-training data, and this bias is also common in human decision-making (Zheng et al., 2023; Li et al., 2024b; Wang et al., 2023a; Zhang et al., 2023b; Zeng et al., 2023). To mitigate this issue, we use all six permutations of the annotations from three LLM stewards. For each order, the average scores from all judges are used to determine the rankings for each candidate responses, whereas the majority ranking results were subsequently used to select the ground truth table/column descriptions. In addition to permuting the order of candidate responses during evaluation, the consistency among each judge can be important. We posited that presenting the confidence scores as additional information would enhance the reliability and consistency of the evaluations from LLM judges. To support this presumption, we tried LLM evaluations under three more scenarios:

- **No confidence scores used**: Evaluations were done without presenting any confidence scores.
- **Perturbed confidence score**: Each original confidence score was modified by adding noise from a uniform distribution $\mathcal{U}(-0.5, 0.5)$, with the fi-

Table 1: Intra-judge ranking consistency for different LLM judges under different scenarios, defined by the existence of a majority ranking (more than half) in each judge's rating across the six possible order permutations of the candidate results.

Cat.	Judge	Full Conf.	Pert. Conf.	No Conf.	Rand. Conf.
$d_{(t_i,t)}$	gpt-4-turbo	**14.3**	7.6	10.3	7.1
	gpt-3.5-turbo	**24.5**	8.9	2.0	2.4
	claude-v2	**20.9**	10.1	8.9	9.4
	claude-v1	**15.3**	2.1	8.3	5.9
	llama2-70b	**17.6**	10.1	2.3	3.3
	aggregated	**69.5**	**49.4**	**44.5**	**40.0**
$d_{(t_i,c)}$	gpt-4-turbo	**23.0**	15.2	10.4	8.2
	gpt-3.5-turbo	**6.3**	2.5	1.0	4.7
	claude-v2	**17.2**	2.5	3.0	2.4
	claude-v1	**20.1**	11.4	7.9	3.5
	llama2-70b	**7.2**	2.1	0.9	3.3
	aggregated	**65.9**	**48.1**	**36.9**	**31.8**

nal score capped between 0 and 1.

- **Random confidence score**: Each confidence score was replaced with a random value generated from a uniform distribution $\mathcal{U}(0, 1)$.

The results in Table 1 clearly demonstrate the impact of the above scenarios on evaluation consistency. Overall, the judge consistency rate, defined as the percentage of table results with a majority ranking from the judge among all order permutations, increases drastically by combining all the LLM judges as opposed to using results from a single judge. Specifically, using full confidence scores resulted in the highest aggregated intra-judge consistency, reaching 69.5% for table descriptions and 65.9% for column descriptions. When using perturbed confidence scores, the consistency levels dropped below the full confidence but was still above no confidence scenarios, indicating the benefit of even partially accurate confidence scores. The lowest consistency were observed when random confidence scores were used. We also present evidence of alignment between human evaluations and LLM judges, showing consistent preferences for GPT-3.5-Turbo generated metadata in Table 4 (Appendix). These findings underscore the importance of accurate confidence information in enhancing the reliability and consistency of evaluations by LLM judges.

3.2.2 Selecting Ground Truth for Supervised Evaluation

Although evaluating the metadata generation in TabMeta task is highly subjective and open-ended, for each table we still provide a sample description for the entire table and each column as the ground truth, which enables computing the supervised ML metrics introduced in Section 4. For each table, the ground truth was determined by selecting the top result based on the majority ranking derived from averaging across all LLM judge scores. This approach was applied to tables with consistent rankings in over half of the permutations. For the small percentage of tables lacking a majority ranking, the ground truth was chosen as the top result averaged across all permutations.

4 Quantitative and Deterministic Evaluation Methods

Evaluation of generative models for text is still an ambiguous problem (Theis et al., 2015; Betzalel et al., 2022). Our goal here is to measure the quality of tabular metadata generation with respect to accuracy, coverage, conciseness, etc. To this end, we introduce a set of deterministic supervised and unsupervised metrics for TabMeta, to capture the subtleties and complexities associated with such evaluation. Subsequently, we also analyze the key characteristics of the evaluation metrics that align with LLM judges in TabMeta evaluation.

4.1 Conciseness

Approximation of Kolmogorov Complexity: The Kolmogorov complexity (Li et al., 2008) $K(d_{t_i})$ of a description d_{t_i} is the length of the shortest possible representation of d_{t_i} in some fixed universal description language, which is utilized as a measure of the computational resources needed to specify a string. As the true Kolmogorov complexity is usually non-computable, it is approximated via the use compression algorithms: the length of the compressed version of a string is a proxy for its Kolmogorov complexity. In our case, we leverage a heuristic to approximate the Kolmogorov complexity using BERT embeddings and `gzip` compression. Given multiple options of generated text(with the same semantic content), the size (in bytes) of the compressed embeddings is used as the approximation, wherein lower values indicates more concise generations.

Approximation of Minimum Description Length via Embedding Variance: Minimum Description Length (MDL) (Grünwald, 2007) is a principle that relates to the best compression of a set of data. If we regard a piece of text as "data", MDL can be interpreted as the smallest length (in terms of some encoding) at which this data can be represented without loss of information. Since, MDL on text is hard to compute directly, we measure the variance of the embeddings for words within the generated descriptions. Intuitively, if a piece of generated description is concise and information-dense, the word embeddings of that would have higher variance (spreading across various topics or semantics). In contrast, repetitive or verbose descriptions would have embeddings that are clustered more closely together, leading to lower variance.

4.2 Informativeness

Semantic Entropy: Here we focus on the diversity of information contained within text generated by a language model. Towards computing the semantic entropy for a generated description $\mathcal{D}$, we first tokenize the text and obtain embeddings. These embeddings are then clustered based on similarity, with a defined threshold (we use 0.9) to ensure meaningful grouping. Subsequently, we calculate the entropy as $-\sum_i p(d_{t_i}) \log_2 p(d_{t_i})$, where $p(d_{t_i})$ represents the probability of each cluster. Intuitively, a higher semantic entropy suggests more informative and diverse content, accounting for synonymous terms and reducing the impact of repetitive but differently phrased information.

KL Divergence. We use KL Divergence to compute the difference of the information content between the original schema $s_{t_i} \in \mathcal{S}$ and the generated metadata $d_{t_i} \in \mathcal{D}$, as a proxy for information gain. For generated text (distribution P) and the reference text (distribution Q), the texts are first tokenized to generate BERT embeddings. K-means clustering is then applied to these embeddings to create a summarized representation of the text in terms of key "semantic" clusters. A probability distribution is then constructed based on cluster frequencies, i.e. the probability of sentences within each piece of text that fall within the clusters and then the value is computed as : $KL = -\Sigma_i p(d_{t_i}) \log_2 \frac{p(d_{t_i})}{q(s_{t_i})}$.

4.3 Reliability and Coverage

Semantic Overlap F1. To estimate the semantic overlaps between the reference and prediction, we use instruct-xl embedder (Su et al., 2023) to generate sentence-level embeddings. The generated embeddings are used to compute pairwise similarity scores between each sentence in the candidate paragraph and each sentence in the reference paragraph. Unlike existing sentence-level metrics for evaluation like BertScore (Zhang et al., 2020b) and BartScore (Yuan et al., 2021), which puts more emphasis on token-wise embedding similarity, we computed similarities on the sentence-level embeddings, therefore the semantic overlaps between the long summary candidates can be better captured. This is especially important for the table-level and column-level descriptions in TABMETA, since these summaries typically contain long and narrative sentences. With the reference sentences $x = x_1, ..., x_k$ (embeddings $\mathbf{x} = \mathbf{x}_1, ..., \mathbf{x}_k$) and the candidate sentences $\hat{x} = \hat{x}_1, ..., \hat{x}_k$ (embeddings $\hat{\mathbf{x}} = \hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_k$), We compute the F1 score of semantic overlap by: $F_{\text{SemOv}} = 2 \times (P_{\text{SemOv}} \times R_{\text{SemOv}})/(P_{\text{SemOv}} + R_{\text{SemOv}})$, where the precision and recall are calculated by: $P_{\text{SemOv}} = \frac{1}{|x|} \sum_{x_i \in x} \max_{\hat{x}_j \in \hat{x}} \mathbf{x}_i^T \hat{\mathbf{x}}_j$, and $R_{\text{SemOv}} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \mathbf{x}_i^T \hat{\mathbf{x}}_j$.

QA Overlap F1. Intuitively, a high-quality summary should encompass key concepts accurately, mirroring the essential elements found in the ground truth or reference. Inspired by FEQA (Durmus et al., 2020), an automated faithfulness metric based on question answering, we leverage a LLM (specifically GPT-4-turbo) to execute the following (see Figure 3): (i) QG-QA for reference: identify and extract k entities that could form answer spans from the reference and formulate questions pertaining to each of the answers. For our evaluation, we set $k = 5$. (ii) QA for candidate: utilizes candidate description as input for the LLM to extract answers for those questions generated in prior steps. (iii) Compute average BertScores (precision, recall, F1) between the answers generated by the LLM for the same set of questions but with the reference and candidate descriptions as contextual inputs. As such, the QA Overlap F1 is aimed at effectively assessing the reliability of table summaries by measuring their alignment with established ground truths.

4.4 Coherence and Cohesion

Coherence *via* Embeddings. We compute the cosine similarity scores of embeddings from instruct-xl for each individual sentence in the generated metadata. Then, the embedding coherence is com-

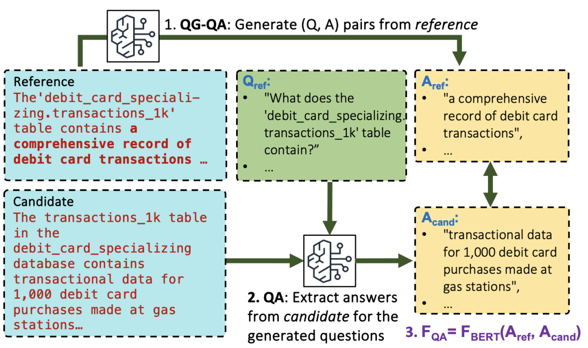

Figure 3: Illustration of computing QA overlap F1 given a reference.

puted by averaging cosine similarity between consecutive sentences throughout the paragraph, where higher values imply more coherent description. Note this metric only applies to table description.

Lexical Cohesion. This is a metric reliant on identifying the recurrence of lexical items, such as using pronouns to refer back to nouns, or the repetition of certain words and phrases which helps in linking different parts of a text. In this case, the lexical cohesion score is simply computed by the ratio of repeated words to the total number of words.

Perplexity. This metric is derived from the perplexity scores of a pretrained autoregressive model. It assesses the congruence between the model's predicted word probabilities and the actual distribution in the pre-training corpus. Lower perplexity often correlates with more human-like text generation.

5 Experiments

5.1 Enhancing Keyword Search by LLM-Enriched Table Metadata

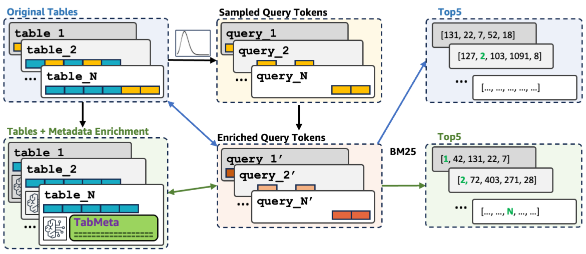

Figure 4: Customized keyword search workflow.

In this experiment, we investigate whether table metadata generated by LLM stewards can improve keyword-based table search effectiveness. The search queries for this study were generated by sampling tokens from table schemas. The number of tokens sampled per query ranged from 1 to 5, determined by sampling from a Poisson distribution (λ=3). Without prior knowledge of the database and its specific formatting, user query keywords often do not exactly match the schema; they are more likely to be in alternate forms, including synonyms, abbreviations, or expansions. Therefore, to simulate a more realistic search experience, we enrich the sampled queries using an LLM (refer to Figure 4). It is important to note that the assumption is based on the search keywords originating solely from raw data; no keywords or variants derived from exogenous information were employed. Using the enriched queries and metadata (column descriptions and table descriptions) generated from different LLM stewards, we conduct the retrieval

Table 2: Precision at k and mean reciprocal rank (MRR) for enriched query search over BIRD-SQL dataset without metadata enrichment (schema-only) and with table/column descriptions.

Method	P@1	P@5	P@10	MRR
s_{t_i} Only	8.8	12.6	19.0	12.6
$s_{t_i}, d_{(t_i,c)}$ (claude-v1)	25.6	34.7	55.8	35.7
$s_{t_i}, d_{(t_i,c)}$ (claude-v2)	26.6	37.7	56.1	36.8
$s_{t_i}, d_{(t_i,c)}$ (gpt-3.5-turbo)	27.6	39.4	58.3	38.4
$s_{t_i}, d_{(t_i,c)}, d_{(t_i,t)}$ (claude-v1)	30.8	42.0	60.5	41.2
$s_{t_i}, d_{(t_i,c)}, d_{(t_i,t)}$ (claude-v2)	32.2	43.4	62.3	42.6
$s_{t_i}, d_{(t_i,c)}, d_{(t_i,t)}$ (gpt-3.5-turbo)	**33.5**	**45.1**	**62.8**	**43.7**

using BM25 and measure the search performance by precision at k retrieved results, as well as mean reciprocal rank (MRR).

As shown in Table 2, including solely the column descriptions already significantly enhances the search performance compared to using schema-only information. The Precision at 1 (P@1) metric notably improved from 8.8% with the schema-only approach to 25.6%, 26.6%, and 27.6% when enriched with column descriptions from Claude-v1, Claude-v2, and GPT-3.5-Turbo, respectively. This pattern of improvement is consistent across other precision metrics (P@5 and P@10), indicating that LLM-enriched metadata provides more relevant search results at various result depths. Furthermore, the integration of both column and table descriptions ($d_{(t_i,t)}$ and $d_{(t_i,c)}$) led to an even more pronounced improvement. For example, the P@1 for these combinations showed an increase to 33.5% using GPT-3.5-Turbo, demonstrating that the addition of table descriptions further refines the retrieval relevance. This trend is similarly observed in the MRR, where the inclusion of both column and table descriptions resulted in the highest scores across all models. These results underscore the significance of TabMeta in enhancing keyword-based table retrieval, even in scenarios where the user's query does not directly align with the underlying schema.

5.2 Metric Analysis

In our evaluation, we assessed the table descriptions generated by three LLM stewards using the automatic metrics outlined in Section 4. The supervised metrics were computed against the ground truth of TabMeta benchmark. The results, presented in Table 3, indicate that over half of these metrics are consistent with the preferences of LLM judges. This consistency is evident both in the rankings derived from metric scores and the correlation between these scores and the LLM judges' evaluations, with a notable preference for results generated from GPT-3.5-turbo, see also from scatter

Table 3: Average metric scores computed for table and column descriptions from different LLM stewards, and the correlation coefficients between the metric scores and the average judge scores. Superscripts u and s denote unsupervised and supervised metrics, respectively. Metrics with the highest scores are highlighted in **blue bold** for comparisons across LLM stewards, and **red bold** signifies the strongest correlation with judges' scores.

	Average Metric Scores (LLM steward)						**Correlation with LLM Judge Scores**			
	Claude-v1		Claude-v2		GPT-3.5-turbo		Pearson		Spearman	
Metric Name	$d_{(t_i,t)}$	$d_{(t_i,c)}$	$d_{(t_i,t)}$	$d_{(t_i,c)}$	$d_{(t_i,t)}$	$d_{(t_i,c)}$	$d_{(t_i,t)}$	$d_{(t_i,c)}$	$d_{(t_i,t)}$	$d_{(t_i,c)}$
Approx. Kolmogorov Complexity ^{u} ↓	8.35E5	**4.94E5**	**7.19E5**	5.11E5	1.18E6	5.61E5	0.318	0.086	0.341	0.113
Embedding Variance ^{u} ↑	0.213	0.177	0.212	0.177	**0.223**	**0.178**	0.251	0.124	0.249	0.110
Semantic Entropy ^{u} ↑	**6.638**	3.186	6.343	3.289	6.592	**3.392**	0.165	0.092	0.169	0.087
KL Divergence ^{u} ↑	**4.930**	4.582	4.538	5.040	4.394	**5.105**	-0.036	-0.030	-0.009	-0.041
Semantic Overlap F1 ^{s} ↑	0.875	0.929	0.893	0.923	**0.950**	**0.952**	**0.756**	**0.692**	**0.742**	**0.721**
QA Overlap F1 ^{s} ↑	0.787	0.891	0.800	0.889	**0.909**	**0.928**	0.552	0.604	0.659	0.685
Coherence ^{u} ↑	0.687	-	0.681	-	**0.729**	-	0.310	-	0.362	-
Lexical Cohesion ^{u} ↑	0.155	0.105	**0.169**	0.120	0.167	**0.123**	0.062	0.112	0.046	0.109
Perplexity ^{u} ↓	29.933	178.693	30.382	172.477	**13.665**	**141.298**	-0.236	-0.117	-0.347	-0.163

plots in Figure 5 and Figure 6 (Appendix). For instance, F1 scores for semantic overlap (supervised), exhibited the highest Pearson correlation scores, reaching 0.756 and 0.692 for table and column descriptions, respectively. However, certain metrics including semantic entropy, KL divergence, and lexical cohesion showed very low correlation, suggesting these aspects were less valued by the LLM judges. Interestingly, despite being a measure of conciseness, the approximated Kolmogorov complexity demonstrated a positive correlation with LLM judge scores, indicating a preference for completeness over conciseness in their assessments.

6 Related Works

Prior works on meta data enrichment for tabular data. have primarily taken three different directions (i) Column Semantic Type Annotation (CSTA) (ii) Table Summarization (iii) Semantic matching to help with better search/ understanding of the underlying tabular data.

Column Semantic Type Annotation: CSTA associates every column name in the table to a pre-defined glossary to enhance search and understanding. Prior deep learning methods like Sherlock (Hulsebos et al., 2019) and SATO(Zhang et al., 2019), use column statistics and character distributions as features to their models. CSTA often is limited to a pre-defined glossary and also requires human-annotated training data, which can be difficult to obtain in real-life - and also do not add a table-wide unique tag understandable by downstream users different from the data producers.

Table Summarization: Prior works on tabular data summarization (Lo et al., 2000; Zhang et al., 2020a; Kumar et al., 2022; Ienco et al., 2013) have largely leveraged rules and constraints to summarize the contents of a table or its schema – with outputs also limited to a certain pre-defined and small vocabulary. In addition to making the implicit assumption that the consumer is often familiar with terminology used by the producer, these mechanisms were not designed to work on arbitrarily complex tables from different industrial domains.

Semantic Matching: Semantic matching methods (Li et al., 2021) broadly comprise of techniques such as schema matching, entity matching and linking. In the case of schema matching, it identifies columns which are similar/ identical across tables which can help with joins/ unions, etc. While these methods can help search and discover related tables, they still do not make discovery or understanding of any given table easier for a data consumer without knowledge of the data producer's terminology. Entity matching and linking methods on the other hand are useful when rows in different tables are different attributes of the same entity (orthogonal to our work, as we don't work with table content).

7 Conclusion

Our work introduced TABMETA, a natural language task that generates comprehensive metadata for arbitrarily complex tables, enabling non-expert users to discover, understand and use relevant data more effectively. As a part of our contributions, we curated a unique benchmark dataset for the TABMETA task, comprising table descriptions and column descriptions for 302 tables spanning 30 industry domains. We also put forward two tabular metadata evaluation strategies (a) a *robust and consistent* LLM-Judge based framework which employed confidence scores suited for tabular metadata and (b) ML based metrics to capture quality of the generated metadata such as *conciseness, coherence, information gain*, etc. Finally, we also showed that our metadata enhancement framework substantially improves the performance of tabular data discovery and search by a factor of 3-4x.

8 Limitations

While our work introduces an innovative approach to generating metadata for complex tables, several areas for further enhancement exist. Although we conducted a preliminary human evaluation showing alignment with LLM judges, a more extensive human evaluation would further validate our findings. Our dataset, with 302 tables across 30 domains, provides a strong foundation but may not encompass all real-world diversity, and scaling to larger datasets involves higher costs. Despite using LLM judges and confidence scores to reduce biases and inaccuracies, the reliance on large language models can still pose challenges. While we acknowledge the potential of advanced prompt engineering strategies to improve the quality of generated metadata, it is not the primary focus of this work. Lastly, our metrics are only proxies, as the true evaluation is intractable to compute, suggesting that further refinement of these metrics could enhance future research.

References

Rene Abraham, Johannes Schneider, and Jan Vom Brocke. 2019. Data governance: A conceptual framework, structured review, and research agenda. *International journal of information management*, 49:424–438.

Simran Arora, Brandon Yang, Sabri Eyuboglu, Avanika Narayan, Andrew Hojel, Immanuel Trummer, and Christopher Ré. 2023. Language models enable simple systems for generating structured views of heterogeneous data lakes. *arXiv preprint arXiv:2304.09433*.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021. Structured denoising diffusion models in discrete state-spaces. *Advances in Neural Information Processing Systems*, 34:17981–17993.

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):120.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Eyal Betzalel, Coby Penso, Aviv Navon, and Ethan Fetaya. 2022. A study on the evaluation of generative models. *arXiv preprint arXiv:2206.10935*.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Dan Brickley, Matthew Burgess, and Natasha Noy. 2019. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The world wide web conference*, pages 1365–1375.

Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. 2019. End-to-end entity resolution for big data: A survey. *arXiv preprint arXiv:1905.06397*.

Shizhe Diao, Rui Pan, Hanze Dong, Ka Shun Shum, Jipeng Zhang, Wei Xiong, and Tong Zhang. 2023. Lmflow: An extensible toolkit for finetuning and inference of large foundation models.

Esin Durmus, He He, and Mona Diab. 2020. FEQA: A question answering evaluation framework for faithfulness assessment in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5055–5070, Online. Association for Computational Linguistics.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30).

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

Peter D Grünwald. 2007. *The minimum description length principle*. MIT press.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Tom Hosking, Phil Blunsom, and Max Bartolo. 2023. Human feedback is not gold standard. *arXiv preprint arXiv:2309.16349*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23. ACM.

Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. Gittables: A large-scale corpus of relational tables. *Proceedings of the ACM on Management of Data*, 1(1):1–17.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1500–1508.

Dino Ienco, Yoann Pitarch, Pascal Poncelet, and Maguelonne Teisseire. 2013. Knowledge-free table summarization. In *Data Warehousing and Knowledge Discovery: 15th International Conference, DaWaK 2013, Prague, Czech Republic, August 26-29, 2013. Proceedings 15*, pages 122–133. Springer.

Vijay Khatri and Carol V Brown. 2010. Designing data governance. *Communications of the ACM*, 53(1):148–152.

Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. Opentab: Advancing large language models as open-domain table reasoners. *arXiv preprint arXiv:2402.14361*.

Dibyakanti Kumar, Vivek Gupta, Soumya Sharma, and Shuo Zhang. 2022. Realistic data augmentation framework for enhancing tabular reasoning. *arXiv preprint arXiv:2210.12795*.

Fangyu Lei, Tongxu Luo, Pengqi Yang, Weihao Liu, Hanwen Liu, Jiahe Lei, Yiming Huang, Yifan Wei, Shizhu He, Jun Zhao, et al. 2023. Tableqakit: A comprehensive and practical toolkit for table-based question answering. *arXiv preprint arXiv:2310.15075*.

Han Li, Yash Govind, Sidharth Mudgal, Theodoros Rekatsinas, and AnHai Doan. 2021. Deep learning for semantic matching: A survey. *Journal of Computer Science and Cybernetics*, 37(4):365–402.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Rongyu Cao, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C. C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2023a. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2024a. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36.

Ming Li, Paul Vitányi, et al. 2008. *An introduction to Kolmogorov complexity and its applications*, volume 3. Springer.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023b. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Tianxiang Li, Shuai Wang, Cuiyun Gao, and Yang Liu. 2024b. Split and merge: Aligning position biases in large language model based evaluators.

Ming-Ling Lo, Kun-Lung Wu, and Philip S Yu. 2000. Tabsum: A flexible and dynamic table summarization approach. In *Proceedings 20th IEEE International Conference on Distributed Computing Systems*, pages 628–635. IEEE.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. 2023. Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models.

Ryan L Murphy, Balasubramaniam Srinivasan, Vinayak Rao, and Bruno Ribeiro. 2018. Janossy pooling: Learning deep permutation-invariant functions for variable-size inputs. *arXiv preprint arXiv:1811.01900*.

Avanika Narayan, Ines Chami, Laurel Orr, Simran Arora, and Christopher Ré. 2022. Can foundation models wrangle your data? *arXiv preprint arXiv:2205.09911*.

Laurel J Orr, Karan Goel, and Christopher Ré. 2022. Data management opportunities for foundation models. In *CIDR*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Sara Rosenbaum. 2010. Data governance and stewardship: designing data stewardship entities and advancing data access. *Health services research*, 45(5p2):1442–1455.

Hongjin Su, Weijia Shi, Jungo Kasai, Yizhong Wang, Yushi Hu, Mari Ostendorf, Wen tau Yih, Noah A. Smith, Luke Zettlemoyer, and Tao Yu. 2023. One embedder, any task: Instruction-finetuned text embeddings.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Ruoxi Sun, Sercan O Arik, Hootan Nakhost, Hanjun Dai, Rajarishi Sinha, Pengcheng Yin, and Tomas Pfister. 2023. Sql-palm: Improved large language modeladaptation for text-to-sql. *arXiv preprint arXiv:2306.00739*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Lucas Theis, Aäron van den Oord, and Matthias Bethge. 2015. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023a. Large language models are not fair evaluators.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2023b. Large language models are implicitly topic models: Explaining and finding good demonstrations for in-context learning. In *Workshop on Efficient Systems for Foundation Models@ ICML2023*.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. 2024. Large language models are latent variable models: Explaining and finding good demonstrations for in-context learning. *Advances in Neural Information Processing Systems*, 36.

Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, Wei Ye, Shikun Zhang, and Yue Zhang. 2023c. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Hongbin Ye, Tong Liu, Aijia Zhang, Wei Hua, and Weiqiang Jia. 2023. Cognitive mirage: A review of hallucinations in large language models. *arXiv preprint arXiv:2309.06794*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. 2017. Deep sets. *Advances in neural information processing systems*, 30.

Hamed Zamani, Johanne R Trippas, Jeff Dalton, Filip Radlinski, et al. 2023. Conversational information seeking. *Foundations and Trends® in Information Retrieval*, 17(3-4):244–456.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following.

Bin Zhang, Yuxiao Ye, Guoqing Du, Xiaoru Hu, Zhishuai Li, Sun Yang, Chi Harold Liu, Rui Zhao, Ziyue Li, and Hangyu Mao. 2024. Benchmarking the text-to-sql capability of large language models: A comprehensive evaluation. *arXiv preprint arXiv:2403.02951*.

Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2019. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311*.

Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Shen Wang, Huzefa Rangwala, and George Karypis. 2023a. Nameguess: Column name expansion for tabular data. *arXiv preprint arXiv:2310.13196*.

Shuo Zhang, Zhuyun Dai, Krisztian Balog, and Jamie Callan. 2020a. Summarizing and exploring tabular data in conversational search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1537–1540.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020b. Bertscore: Evaluating text generation with bert.

Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023b. Wider and deeper llm networks are fairer llm evaluators.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges.

9 Appendix

9.1 Analysis of Automated Evaluation Methods

We compare the overall LLM judge scores for each table's metadata with each individual automated evaluation metric proposed in Section 4 in Figure 5 (table description) and Figure 6. The different generations from candidate LMs including GPT-3.5-Turbo, Claude-v1, and Claude-v2 were highlighted in different colors. Note that the descriptions for each individual table column are non-consecutive, therefore the Coherence metric were not computed for column descriptions.

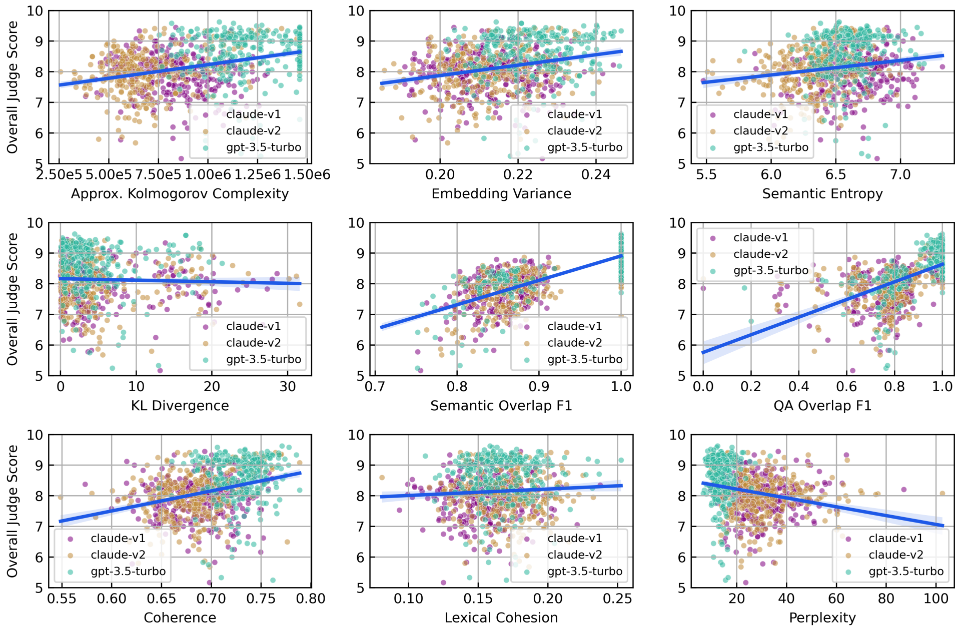

Figure 5: Scatter plots for supervised and unsupervised evaluation metrics for table descriptions from LLM stewards versus the overall ratings (out of 10) from LLM judges.

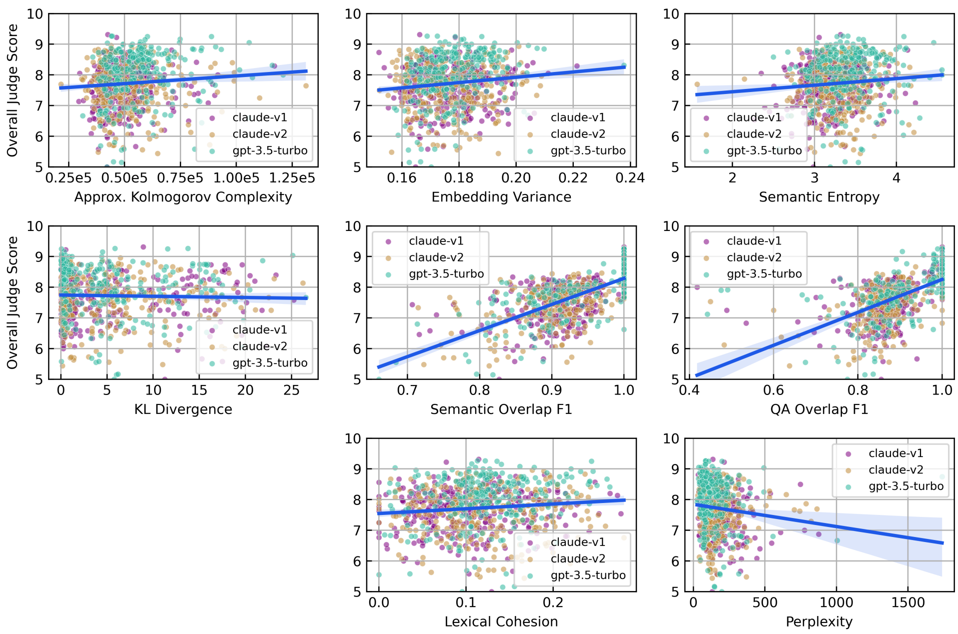

Figure 6: Scatter plots for supervised and unsupervised evaluation metrics for column descriptions from LLM stewards versus the overall ratings (out of 10) from LLM judges.

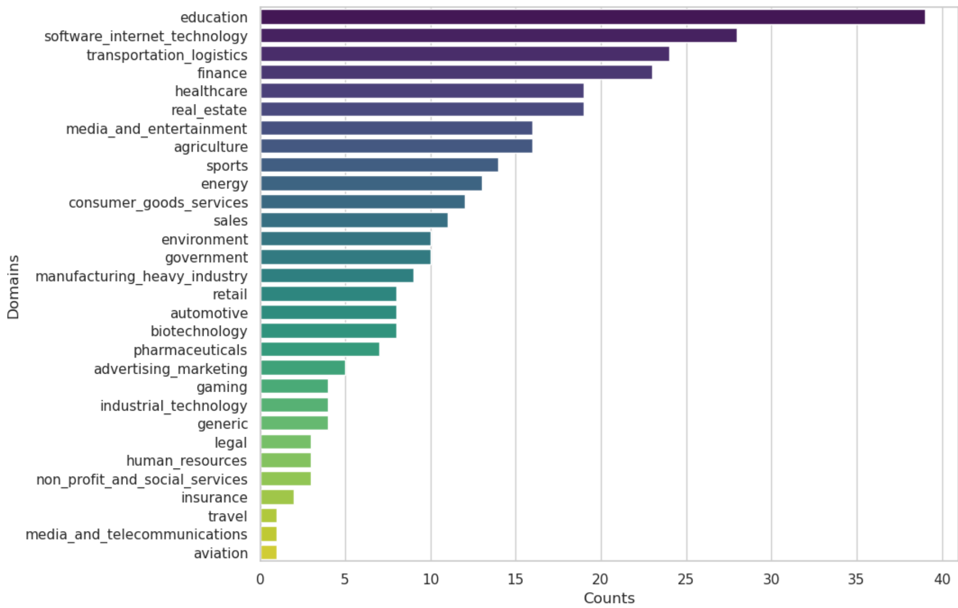

Figure 7: Distribution of domains for tables included in the benchmark.

9.2 Additional Details about Dataset Curation

We conducted a human evaluation study by randomly sampling the LLM-generated metadata for 20 tables, and asked a group of three data scientists and analysts to assess the quality of the generated metadata using the exact instruction/rubric for LLM judges. The scores from the three human evaluators were averaged and compared with the LLM judge scores averaged from different order permutations (with using confidence scores). As shown in Table 4, the averaged human scores reflect the same preference to the metadata generated by GPT-3.5-Turbo model, consistent with the LLM-evaluation approach.

Table 4: Overall LLM judge scores and average human evaluation scores for the 20 sampled table metadata.

	Judges					
Steward	claude-v1	claude-v2	gpt-3.5-turbo	gpt-4-turbo	llama2-70b	human
	Table Description					
claude-v1	8.00	7.50	7.85	7.85	7.80	7.58
claude-v2	8.00	7.25	8.05	8.20	8.07	7.28
gpt-3.5-turbo	**9.17**	**8.00**	**9.00**	**8.35**	**9.00**	**7.73**
	Column Description					
claude-v1	7.89	7.70	7.35	7.60	7.20	6.50
claude-v2	7.58	7.20	6.90	7.60	6.87	6.62
gpt-3.5-turbo	**8.68**	**7.85**	**8.40**	**7.85**	**8.33**	**7.33**

9.3 Prompts Used for Metadata Generation and LLM Evaluation

```
For the table named {table_name}, with schema '{schema_list}' ({len(schema_list)} attributes), provide
detailed descriptions for each column. Use the following format for each column on separate lines:
'[Column Name] | [Description]'. Ensure that the descriptions are clear, informative, and precise. Do
not generate any additional text at the beginning or end of the response.
```

Figure 8: Prompt template for generating column-level descriptions.

```
Given the table name {table_name}, schema '{schema_list}', along with the detailed column descriptions:
'{column_description_dict}', generate a comprehensive and reliable global description for the table.
The description should provide a broad understanding of the data contained within the table, its
relevance, the relationships among different columns, and any potential implications or insights it
might offer. While crafting the description, seamlessly incorporate the column descriptions into the
narrative to provide a cohesive understanding of the table's structure and content. Do not generate
any additional text at the beginning or end of the response.
```

Figure 9: Prompt template for generating table-level descriptions.

```
You are an expert database catalog creator who is evaluating metadata for a table drafted by different models, based only on the
table schema. For each sentence, there is a corresponding confidence score for your reference.
Candidate metadata for this table in JSON:
    {
      "table_name": "debit_card_specializing.transactions_1k",
      "llm_results": {
        "model1": {
          "table description": {
            "description": [
              "The debit_card_specializing.transactions_1k table contains records of transactions made using debit cards at gas
stations.",
              "Each record includes a unique TransactionID to identify the transaction.",
              ...
            ],
            "confidence": [
              0.9908298118971288,
              0.9989090043818578,
              ...
            ]
          },
          "attribute description": {
            "attribute name": [
              "TransactionID",
              "Date",
              ...
            ],
            "description": [
              "Unique identifier for each transaction record",
              "Date the transaction occurred",
              ...
            ],
            "confidence": [
              0.9939870447851717,
              0.9790911888703704,
              ...
            ]
          }
        },
      "model2": {
        ...
      },
      "model3": {
        ...
      }
    }
 Please provide an overall score from 1 to 10 for each table description and each set of column descriptions, considering their
accuracy, clarity, consistency, completeness, context awareness, handling of ambiguity, and informativeness. A score of 1 represents
extremely poor performance across these aspects, while a score of 10 indicates exceptional performance in all areas.  Avoid any
potential biases.
Before giving the score, provide a detailed reasoning of your evaluation, and the order of the candidate responses should not affect
your judgement. The response should follow the reasonings and contain the example JSON code snippet.
  {
    "column": {
      "model1": # score between 1 to 10, worst to best,
      "model2": # score between 1 to 10, worst to best,
      "model3": # score between 1 to 10, worst to best
    },
    "table": {
      "model1": # score between 1 to 10, worst to best,
      "model2": # score between 1 to 10, worst to best,
      "model3": # score between 1 to 10, worst to best
    }
  }
Response:
```

Figure 10: Prompt template for LLM judge.

Table 5: Example from TABMETA Benchmark from `affordable-housing-by-town-2011-2022` Table

Table Description	
The 'affordable-housing-by-town-2011-2022' table provides a comprehensive overview of affordable housing units in various towns from 2011 to 2022. The table contains information on the number of affordable housing units, including government-assisted units, tenant rental assistance, single-family CHFA/USDA mortgages, and deed-restricted units. The 'Year' column indicates the specific year for which the data is recorded, allowing for temporal analysis of affordable housing trends over time. The 'Town Code' and 'Town' columns provide the unique code and name of each town, enabling the identification and comparison of affordable housing statistics across different locations. The '2010 Census Units' column offers a baseline for understanding the total housing units in each town, providing context for the proportion of affordable housing within the overall housing stock. The 'Total Assisted Units' column aggregates the various types of assisted housing units, offering a consolidated view of the overall impact of government assistance and rental programs on affordable housing availability. The 'Percent Affordable' column calculates the percentage of affordable housing units relative to the total housing units, providing a key metric for assessing the level of affordability within each town.	
Attribute Name	**Description**
Year	The year in which the data was recorded.
Town Code	A unique code assigned to each town for identification purposes.
Town	The name of the town for which the data is being reported.
2010 Census Units	The number of housing units recorded in the 2010 census for the respective town.
Government Assisted	The number of housing units that received government assistance for affordability.
Tenant Rental Assistance	The number of housing units that received rental assistance for tenants.
Single Family CHFA/ USDA Mortgages	The number of single-family housing units that received mortgages from the Connecticut Housing Finance Authority (CHFA) or the United States Department of Agriculture (USDA).
Deed Restricted Units	The number of housing units with deed restrictions to maintain affordability.
Total Assisted Units	The total number of housing units that received any form of assistance for affordability.
Percent Affordable	The percentage of housing units in the town that are considered affordable based on the provided assistance.