AutoDDG: Automated Dataset Description Generation using Large Language Models

HAOXIANG ZHANG*, New York University, USA
YURONG LIU*, New York University, USA
AÉCIO SANTOS, New York University, USA
WEI-LUN (ALLEN) HUNG, New York University, USA
JULIANA FREIRE, New York University, USA

The proliferation of datasets across open data portals and enterprise data lakes presents an opportunity for deriving data-driven insights. Widely-used dataset search systems rely on keyword search over dataset metadata to support discovery. Therefore, when metadata is incomplete, missing, or inconsistent with dataset contents, findability is severely compromised. To address this limitation, we introduce AutoDDG, a framework that automatically generates textual descriptions of tabular data. By adopting a data-driven approach to summarize dataset contents and leveraging large language models (LLMs) to enrich summaries with semantic information and produce human-readable text, AutoDDG derives descriptions that are comprehensive, accurate, and concise. A critical challenge in this problem is evaluating the effectiveness of description generation methods and assessing the quality of the generated descriptions. We propose a comprehensive evaluation methodology that combines retrieval, reference-based, and reference-free assessment, with human validation. Our experimental results using new benchmarks demonstrate that AutoDDG generates high-quality, accurate descriptions at scale, significantly improving dataset retrieval performance across diverse use cases. AutoDDG is available at https://github.com/VIDA-NYU/AutoDDG.

CCS Concepts: • **Information systems** → **Data management systems**; **Enterprise information systems**; **Data encoding and canonicalization**; **Document filtering**; **Summarization**; **Relevance assessment**; **Mediators and data integration**.

Additional Key Words and Phrases: dataset discovery, dataset search, table to text, foundational models, LLMs, FAIR data

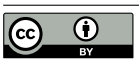
ACM Reference Format:
Haoxiang Zhang, Yurong Liu, Aécio Santos, Wei-Lun (Allen) Hung, and Juliana Freire. 2026. AutoDDG: Automated Dataset Description Generation using Large Language Models. *Proc. ACM Manag. Data* 4, 1 (SIGMOD), Article 12 (February 2026), 27 pages. https://doi.org/10.1145/3786626

1 Introduction

We have witnessed a proliferation of data portals and data lakes [25, 29, 33, 34, 43, 63, 66, 93] as more structured data is generated and made available. It is estimated that there are tens of millions of datasets on the web [65]. *However, a significant fraction of these datasets remains effectively invisible due to inadequate or missing descriptions* [48]. While these datasets present significant opportunities

*Equal contribution.

Authors' Contact Information: Haoxiang Zhang, haoxiang.zhang@nyu.edu, New York University, New York, NY, USA; Yurong Liu, yurong.liu@nyu.edu, New York University, New York, NY, USA; Aécio Santos, aecio.santos@nyu.edu, New York University, New York, NY, USA; Wei-Lun (Allen) Hung, allen.hung@nyu.edu, New York University, New York, NY, USA; Juliana Freire, juliana.freire@nyu.edu, New York University, New York, NY, USA.



ACM 2836-6573/2026/2-ART12
https://doi.org/10.1145/3786626

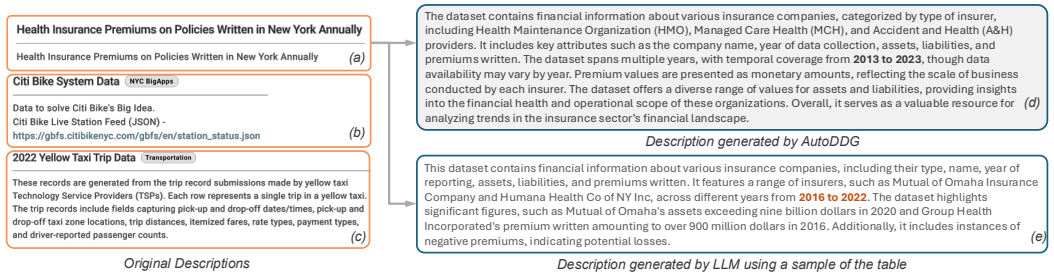


Fig. 1. Dataset descriptions from NYC Open Data: (a) and (b) lack sufficient details, while (c) contains inconsistent information–the *2022 Yellow Taxi Trip Data* dataset includes taxi trips that span multiple years, not just 2022. (d) shows a description automatically generated by AutoDDG for the dataset (a), while (e) was generated by an LLM using a data sample.

for data-driven insights, they must be easily discoverable and interpretable to maximize their utility [90].

Dataset search systems that power data portals [9, 22, 77] follow the web search paradigm: they support keyword queries over metadata, such as dataset names and descriptions. Consequently, discoverability depends on metadata quality–how effectively it conveys the dataset contents and aligns with users' information needs. The critical importance of descriptions is exemplified by Google Dataset Search, which excludes from its index datasets lacking descriptions [8]. Platforms such as CKAN do not mandate descriptions but encourage their use to improve search precision and recall, facilitate dataset understanding, and help users determine relevance [23]. Yet many datasets are published with inadequate or missing descriptions. In the index of the Auctus dataset search engine [12], 3,121 out of 23,520 datasets (13.2%) have no descriptions, and 2,346 datasets (approximately 10%) have descriptions with 10 words or fewer. Figure 1 (a) and (b) show dataset descriptions from NYC Open Data [66] that are terse and fail to convey key characteristics of the dataset; (c) is actually *inconsistent* with the actual contents–it says that the dataset includes taxi trips from 2022, when the actual data contains records that span multiple years.

Datasets without detailed and accurate descriptions may be technically available, but they are effectively invisible. Koesten et al. [48] compare the problem of finding data today to the early days of the web, when users needed to know the URL of web pages or relied on manually created directories such as DMOZ to access content. Beyond discoverability, descriptions are crucial for relevance assessment. Just as web search users rely on snippets to decide which results to explore, dataset search users depend on descriptions to filter relevant datasets. This issue is particularly pronounced for datasets, where downloading and inspecting large files is far more costly than simply opening a webpage.

LLMs for Automatic Description Generation. The ability of language models to produce fluent, readable text, combined with their broad world knowledge [10, 84], enables them to generate coherent descriptions enriched with inferred semantic and contextual information. Prior work has demonstrated the effectiveness of LLMs in semantic inference tasks for tabular data, including the ability to determine the semantic types of attributes and the table class [28, 45]. By leveraging

[1]The *Health Insurance* dataset: https://data.ny.gov/Economic-Development/Health-Insurance-Premiums-on-Policies-Written-in-N/xek8-zfrt/about_data
[2]The *Citi Bike* dataset: https://data.cityofnewyork.us/NYC-BigApps/Citi-Bike-System-Data/vsnr-94wk/about_data
[3]The *Yellow Taxi* dataset: https://data.cityofnewyork.us/Transportation/2022-Yellow-Taxi-Trip-Data/qp3b-zxtp/about_data

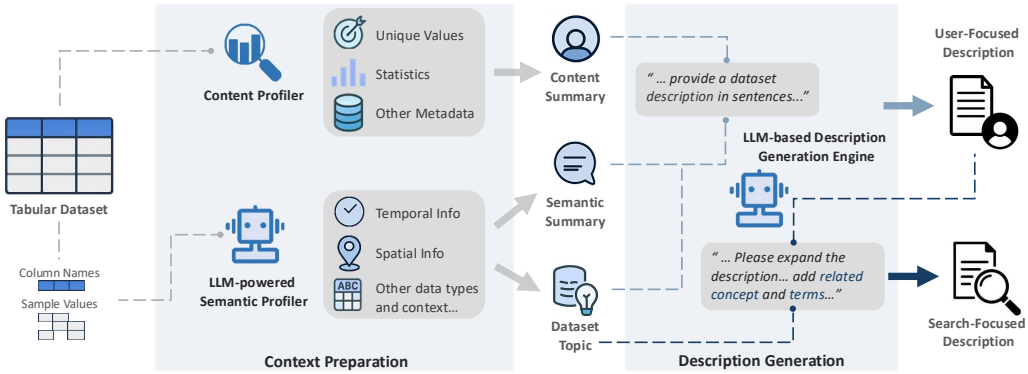


Fig. 2. *AutoDDG: a multi-stage framework for tabular dataset description generation.* In the *Context Preparation* stage, a content and a semantic profiler derive summaries of the dataset. The LLM-powered *Description Generation Engine* uses the summaries to produce dataset descriptions. The system can generate descriptions tailored to specific needs.

external knowledge not explicitly encoded in the data, LLMs can enhance dataset descriptions in ways that manual approaches might overlook.

However, using LLMs for this task presents several challenges. LLMs are designed for processing and generating textual data, whereas datasets are structured in tabular form. This mismatch raises questions about how to best represent tables in prompts, a challenge that remains an active area of research in table representation learning [82]. Moreover, fixed context windows limit LLMs' ability to process large datasets in their entirety [28, 45, 53]. Relying on data samples risks yielding incomplete or misleading descriptions that fail to capture crucial global properties such as the spatial and temporal coverage of the dataset. This is illustrated in Figure 1(e): the description derived by an LLM using a sample states that the data covers the years 2016-2022, while it actually includes records from 2013-2023.

Furthermore, LLMs can generate hallucinated or inaccurate content [36, 89], making it critical to implement safeguards to ensure that descriptions remain grounded in the dataset's contents.

Limitations of Existing Approaches. The main limitation of existing methods is their inability to synthesize comprehensive and accurate descriptions of heterogeneous datasets that are common in data lakes and open data repositories. Pre-trained table-to-text models, such as MVP [83] and ReasTAP [99], were designed for domain-specific narrative tasks, including sports summaries and biographical sentences. When applied to diverse datasets, these models exhibit poor generalization and generate descriptions that do not align with users' needs for discoverability. Simpler heuristics, such as concatenating column headers with sample values (Header+Sample), and LLM-only approaches, rely solely on data samples for context generation. This reliance on samples risks producing incomplete or misleading descriptions that fail to capture crucial global dataset properties, such as the full temporal extent or statistical ranges across all rows. Furthermore, as illustrated above, the absence of grounding in global statistics can lead LLM-only methods to hallucinate, resulting in descriptions with factual errors. Pneuma [5] leverages LLMs for schema narration and row samples to generate table summaries for data discovery. Since it uses row samples, it misses the global properties of columns. Additionally, Pneuma produces dense representations designed primarily for retrieval rather than human readability, limiting their utility for users assessing dataset relevance.

Our Approach: AutoDDG. To address these limitations, we introduce AutoDDG, a systematic LLM-powered framework for the automatic generation of tabular dataset descriptions. Instead of directly feeding data samples to LLMs, which risks missing global properties and generating hallucinations, AutoDDG employs a two-stage architecture (Figure 2):
Context Preparation: It profiles datasets using two complementary approaches (1) *Content profiling* uses algorithmic techniques to capture global statistical properties [12, 64], ensuring descriptions are grounded in actual data characteristics and fit within LLM context windows; (2) *Semantic profiling* leverages LLMs to enrich summaries with contextual information not explicitly present in the data, such as the dataset topic and types of analyses the dataset can be used for, improving discoverability through meaningful keywords and topical cues.
Description Generation: An LLM-based engine uses these summaries to generate two types of descriptions that balance conciseness and comprehensiveness: User-Focused Descriptions (UFD) optimized for readability, providing succinct yet informative overviews; and Search-Focused Descriptions (SFD), which incorporate dataset overviews, themes, and keyword-rich snippets for improved retrieval (Section 2.2). These descriptions can be indexed by standard search engines (e.g., Lucene [60], Elasticsearch [27], and Solr [4]) as well as used to create embeddings that support the efficient evaluation of textual queries, thus improving discoverability without requiring major infrastructure changes. Unlike existing approaches, AutoDDG does not require training and generalizes to different domains. It derives descriptions that are grounded in dataset statistics, enriched with contextual information, and human-readable. As we demonstrate in Section 4, the AutoDDG descriptions are 1) highly effective for retrieval; 2) accurate–as judged by expert evaluators; and 3) readable–attaining high readability scores from both LLM judges and expert evaluators.

By using summaries, AutoDDG produces descriptions with consistent structure, enhancing the usability of dataset search tools and addressing a common user frustration with encountering metadata that varies unpredictably across search results [79]. Additionally, AutoDDG can complement end-to-end data discovery systems such as Pneuma [5] by providing high-quality dataset descriptions that improve both retrieval effectiveness and human readability.

We introduce a comprehensive evaluation methodology to assess the quality of generated descriptions that measures: 1) the improvement in dataset retrieval when automatically generated descriptions are used, 2) the similarity between the generated descriptions and existing ones (when available) to evaluate the ability of AutoDDG to produce descriptions that resemble human-generated descriptions, (3) intrinsic text quality metrics, using LLMs as a judge [31, 58, 76, 100] (and validated by human experts) to evaluate language-level aspects such as readability and conciseness (Section 3.1), (4) the cost and scalability of the approach.

Summary of Contributions. This paper presents the first systematic LLM-powered framework for automated dataset description generation to improve dataset discoverability and relevance assessment in keyword-based search engines. Our main contributions can be summarized as follows:

- *The AutoDDG Framework:* A dual-profiling architecture that addresses fundamental systems challenges—processing large datasets within LLM context limits while maintaining data faithfulness and enriching the descriptions with semantic information. AutoDDG generates descriptions optimized for different goals–readability and retrieval.
- *Benchmarks and Evaluation Methodology:* We propose a multi-pronged approach to evaluate different aspects of descriptions, combining reference-based, reference-free, and expert assessment. We also construct two benchmarks, ECIR-DDG and NTCIR-DDG, tailored to evaluate the quality of dataset descriptions for keyword-based search and retrieval tasks.
- *Experimental Evaluation:* We report the results of an extensive experimental evaluation that validates our design decisions and demonstrates AutoDDG's effectiveness at generating descriptions that improve dataset retrieval up to 30% compared to the original descriptions, with robust

performance across commercial and open-weight LLMs. We also propose two optimized versions of AutoDDG that improve efficiency and assess its cost and scalability relative to other dataset description generation methods.

2 AutoDDG: LLM-Powered Automated Dataset Description Generation

Problem Definition. Given a tabular dataset D with columns C and rows R, our task is to *automatically* generate a descriptive textual summary S_D that effectively captures the key characteristics of D, such as column names, data types, statistical properties, and semantically enriched information, to support dataset discoverability and relevance assessment. Formally, our goal to design a function $\mathcal{G}(\cdot)$ that takes as input the dataset D and generates a textual description S_D:

$$S_D = \mathcal{G}(D)$$

In this paper, we focus on the subclass of functions $\mathcal{G}$ that relies on LLMs to generate descriptions. Implementing such a function is non-trivial due to the large design space for both the input context provided to the LLM and the instructions necessary to shape the desired output. In practice, this entails designing a set of prompt templates P, a dataset context $X(D)$, as well as algorithms to orchestrate the generation process. More specifically, our problem involves designing effective methods for:

Description Generation: Crafting prompts P that guide the LLM to generate accurate and rich descriptions tailored to specific use cases, such as enhancing human readability or discoverability.

Context Preparation: Extracting information and representing the dataset context $X(D)$ in a way that captures the heterogeneous characteristics of tabular data, fits within the input limitations of LLMs, and is aligned with users' information needs. In this paper, we limit AutoDDG to use only the table content and the information automatically derived from it (e.g., using LLMs and data profilers) to create the context $X(D)$. We deliberately exclude existing metadata to assess the framework's ability to generate descriptions from data alone, addressing the common practice of datasets published with inadequate or no metadata. However, when additional metadata is available (e.g., data provenance or existing categorizations), it can be incorporated as context. This is a direction that we intend to pursue in future work.

Desiderata for Dataset Descriptions. We ground our design decisions on several studies on information-seeking behavior and data discovery. These studies have revealed a gap between the datasets available and those that users can effectively locate [14]. They have also highlighted shortcomings in existing dataset metadata that hinder discoverability and the assessment of relevance [48, 68, 79]. Building on these findings, which we summarize in Section 5, we propose a set of desiderata for dataset descriptions:

- *Faithfulness to Data:* Dataset descriptions must be accurate representations of the dataset contents.
- *Uniformity:* While variability is expected in dataset collections, it is important to maintain a certain uniformity in the descriptions to facilitate comparison and relevance assessment.
- *Conciseness and Readability:* Dataset descriptions should be concise to enable quick relevance assessment and yet detailed enough to convey meaningful insights.
- *Comprehensiveness and Alignment with Users' Information Needs:* Widely-used search systems [9, 22, 77] often do not account for dataset contents, limiting users' ability to locate relevant data. Thus, descriptions should provide details about the contents, ensuring that they address users' needs.

Discussion. There is an inherent trade-off between conciseness and comprehensiveness: overly brief descriptions may fail to capture key aspects of the dataset, whereas excessively detailed

Content summary for “Health Insurance Dataset”.

```
Number of Rows: 790
Number of Columns: 6
Columns:
  - Name: Year
    - Data Types: Text, DateTime
    - Coverage: 2013 to 2022
    - Unique Values: 10
  - Name: Liabilities
    - Data Types: Integer
    - Coverage: 0 to 2682301090.0
    - Unique Values: 757
  ...
  [other attributes are omitted due to space limitations]
```

Fig. 3. Example summary created by the Content Profiler.

descriptions can be difficult and time-consuming for users to process. Furthermore, while readability enhances users' ability to assess relevance, it is less critical for indexing and automated discoverability.

To achieve a balance between these conflicting goals, we design our framework to generate multiple outputs tailored to different objectives. In particular, our framework currently generates descriptions to support (1) users' understanding of dataset contents and (2) building indexes for dataset search engines. These descriptions prioritize, respectively, readability and comprehensiveness.

2.1 Context Preparation

The context preparation step takes as input a table and profiles it, generating *dataset summaries* as outputs. We currently implement two profilers: the *Content Profiler* (Section 2.1.1) and the Semantic Profiler (Section 2.1.2). The output of these profilers is combined to create the *dataset context*, which is used to generate the description.

2.1.1 Content Profiler. The summary generated by the *Content Profiler* is derived directly from the table's contents. It captures structural and statistical properties that are commonly extracted by traditional data profilers [1, 64, 70]. In our implementation, we use the Datamart Profiler [70]. Each table is profiled by analyzing all rows of each column to extract information such as attribute data types, value distributions, and uniqueness. Figure 3 shows an example of such a content summary. These elements provide a global view of a dataset, summarizing its structure and content in a concise manner that can be effectively fed into LLMs. Beyond statistical summaries, content profiling could be extended to domain-specific data (e.g., from the biomedical domain) to extract metadata that aligns with specific application needs.

2.1.2 Semantic Profiler. The *semantic summary* goes beyond the dataset contents to capture contextual information. AutoDDG uses LLMs to enrich the dataset context with external knowledge that complements the content summary and caters to users' information needs. This includes the dataset topic and semantic information about all columns in the table and their potential uses, which improves interpretability and relevance [47].

Algorithm 1 describes the structured prompting approach we use to guide the LLM. For each column, it generates a prompt that includes the column name, sample values, and data type. The LLM responds with a JSON-formatted classification of the column based on predefined semantic categories. Examples of the output of the structure-defined prompt are shown in Figure 4. The structure-defined template and the complete prompt for semantic enrichment analysis are available in the extended version of this paper [95]. Note that AutoDDG supports different execution modes,

Algorithm 1 SemanticProfiler

Input: Tabular Dataset D, LLM model M, Number of Samples $sample\_size$
Output: Semantically enriched information for each column in D
Initialize list $semantic\_summary$ as empty
for each column C_i in D **do**
    $sample\_values$ = GetSample(C_i, $sample\_size$)
    $semantic\_info$ = Prompt(M, C_i, $sample\_values$)
    Create a human-readable summary $column\_summary$ based on the semantic information $semantic\_info$
    Append $column\_summary$ to $semantic\_summary$
end for
return $semantic\_summary$

Example 1: *Year* column with values like 2018, 2020, 2023.

```
Temporal:
 - isTemporal: True
 - resolution: Year
Spatial:
 - isSpatial: False
 - resolution:
Entity Type: Temporal Entity
Domain-Specific Types: General
Function/Usage Context:
  Aggregation Key
```

Example 2: *Liabilities* column with values like 137790801, 43992755, 599895.

```
Temporal:
 - isTemporal: False
 - resolution:
Spatial:
 - isSpatial: False
 - resolution:
Entity Type: Monetary Value
Domain-Specific Types: Financial
Function/Usage Context:
  Measurement
```

Fig. 4. Examples of semantic information derived from datasets by the Semantic Profiler.

including a high-concurrency mode for speed and a group-prompting mode for token efficiency; we discuss these in Section 4.

The structured output is then serialized into descriptive sentences, and the summaries of all columns are concatenated to form the final output of the semantic profiler module. For example, the serialized semantic summary for Example 1 (from Figure 4) is:

```
**Year**: Represents temporal entity. Contains temporal data (resolution: Year).
Domain-specific type: general. Function/ Usage Context: Aggregation Key.
```

These enriched outputs provide a description of each column's semantic properties, enabling the generation of high-quality, contextually relevant dataset descriptions.

To enhance dataset discoverability, we also incorporate dataset topics into the semantic summary. The process leverages LLMs to analyze dataset metadata and samples to extract meaningful topics that provide a high-level overview by capturing the primary theme of a dataset in 2-3 words. This entails two key steps: (1) *prompt design*, where a tailored prompt is dynamically constructed using the dataset's title, original description (if available), and sample data; (2) *topic generation*, where the LLM processes the prompt and generates a brief topic. Figure 5 shows the prompt used by the dataset topic generator.

Discussion. The design of our prompts was guided by the findings of studies on information-seeking requirements for dataset search (summarized in Section 5). Our goal is to obtain information that is aligned with how users formulate search queries. For example, the semantic profiler includes information about temporal and spatial attributes, as well as their resolution, as this was a common search pattern observed by Koesten et al. [48]. The semantic profiler (and corresponding prompts)

Dataset Topic Generation Prompt

```
Using the dataset information provided, generate a concise topic in 2-3 words that best describes the
dataset's primary theme:
- Title: {title}
- Original Description: {original_description} (optional)
- Dataset Sample: {dataset_sample}
- Topic (2-3 words):
```

Fig. 5. Prompt for generating concise dataset topics based on the dataset title, description, and sample data.

User-Focused Description Prompt

```
Answer the question using the following information. First, consider the dataset sample: {D_sample}.
Additionally, the dataset profile is as follows: {D_profile}. Based on this profile, please add sentence(s)
to enrich the dataset description. Furthermore, the semantic profile of the dataset columns is as follows:
{D_semantic}. Based on this information, please add sentence(s) discussing the semantic profile in the
description. Moreover, the dataset topic is: {D_topic}. Based on this topic, please add sentence(s) describing
what this dataset can be used for. Based on the information above and the requirements, provide a dataset
description in sentences. Use only natural, readable sentences without special formatting.
```

Example Output:

```
This dataset contains wind speed and direction measurements from a specific time period in 2003. The data
includes 4433 unique time stamps, with a temporal coverage of May 13 to June 12, 2003, at a resolution
of minutes. The average wind speed ranges from 0 to 27.0, with a standard deviation ranging from 0 to
2.78. The average wind direction ranges from 0 to 338.0, with 16 unique values. The dataset provides a
comprehensive view of wind patterns during this time period, making it suitable for environmental studies
and research.
```

Fig. 6. Prompt and example of a UFD.

can be adapted to support other information needs and domains. It can also be extended to include other useful information, such as other semantic types of interest [28, 45].

2.2 Description Generation

AutoDDG uses LLMs to generate two types of dataset descriptions: User-Focused Descriptions (UFD) and Search-Focused Descriptions (SFD). For this task, we carefully design prompts to guide the LLM toward producing descriptions optimized for dataset search engine users to assess relevance and to improve dataset discoverability.

2.2.1 User-Focused Description (UFD). The UFD is designed to provide a clear, concise, and accurate overview of the dataset, prioritizing human readability and presentation. This type of description works best for scenarios where the dataset needs to be communicated to users in a way that is easily understood, such as in reports, dashboards, or data catalogs aimed at human readers. Although its primary goal is readability, UFD can also be effective for search purposes, as it offers a well-structured overview containing key terms and concepts related to the dataset. The UFD is generated by prompting the LLM to describe the dataset based on dataset samples and the content summary (Section 2.1). The UFD prompt and a sample output are shown in Figure 6. The description includes information about the temporal extent and resolution, number of records, as well as a summary of the contents (e.g., the range of wind speed values) and an overview statement about the dataset and what it can be used for.

2.2.2 Search-Focused Description (SFD). The SFD is optimized to enhance the discoverability of datasets in search systems. The process begins with a tabular dataset, which is processed by an LLM to generate an initial description and identify the dataset topic. By including a specific topic

Search-Focused Description Prompt

```
You are given a dataset about the topic {D_topic}, with the following initial description: {D_initial_description}.
Please expand the description by including the exact topic. Additionally, add as many related concepts, synonyms, and relevant terms as possible based on the initial description and the topic. Unlike the initial description, which is focused on presentation and readability, the expanded description is intended to be indexed at backend of a dataset search engine to improve searchability. Therefore, focus less on readability and more on including all relevant terms related to the topic. Make sure to include any variations of the key terms and concepts that could help improve retrieval in search results. Please follow the structure of following example template: {Template}.
```

Example Abridged Output:

```
Dataset Overview: This dataset contains wind speed and direction measurements [... an overview based on the UFD.]
Related Topics: - Climate/Weather Patterns - Renewable Energy - Wind Energy - Meteorology ...
Concepts and Synonyms: - Wind Speed/Velocity - Wind Direction - Average Wind Speed ...
Applications and Use Cases: - Analysis of wind patterns for renewable energy projects - Understanding climate trends and predicting wind behavior ...
Additional Context: - This dataset can be used to address questions such as "What are the typical wind patterns in a given region?" or "How does climate change affect wind behavior?" - It can be integrated with other datasets, such as climate models or energy consumption data, to provide a more comprehensive understanding of the relationship between wind patterns and energy generation. [additional information omitted]
```

Fig. 7. Prompt and example of an abridged SFD.

or area related to the dataset, the LLM can focus on expanding the description with relevant terms, concepts, synonyms, and keyword variations. This helps improve search engine indexing and retrieval performance. For example, suppose a user wishes to publish a dataset on "climate data" that contains detailed measurements across various regions. The system identifies "climate data" as the topic and enhances the description by including climate-related keywords such as "temperature trends," "precipitation," "regional climate analysis," and "weather patterns." This process enables the SFD to incorporate terms related to the topic, increasing the likelihood that users searching for climate-related datasets will find the dataset in question.

Figure 7 shows the prompt template used for generating an SFD, which includes the dataset topic *D_topic* and an initial description *D_initial_description* as inputs; the output structure is defined by the SFD template. The result is a description densely packed with terms relevant to the topic, which significantly improves the dataset's ranking in search results and makes it easier for users to discover the dataset when searching for related topics. A complete SFD example output is given in the extended version of this paper [95].

2.2.3 Discussion: UFDs and SFDs. UFD and SFD are optimized for different purposes. The UFD is crafted with the end user in mind, making the description easy to understand and presenting a well-rounded summary of the dataset. In contrast, the SFD is more technical and keyword-driven, focusing on enhancing search engine performance rather than prioritizing human readability. The generation of both types of descriptions allows AutoDDG to effectively serve both user-friendly and search-optimized needs.

There are different ways in which these descriptions can be generated. Here, we describe our initial approach that treats the SFD as an extension of the UFD. In the SFD, semantic information about the dataset's topic is reinforced to generate expanded keywords, concepts, and use cases, as illustrated in Figure 7. Indexing SFD descriptions in search engines improves dataset findability by incorporating richer, more relevant terms. In future work, we plan to explore different strategies to combine the data-driven and LLM-derived information.

3 Evaluation and New Benchmarks

We propose the use of multiple strategies to assess both the *intrinsic* quality of descriptions and their impact on dataset retrieval. Next, we describe the evaluation strategies and metrics we use as well as new benchmarks that we have derived from existing resources to support the evaluation of dataset descriptions.

3.1 Evaluating Description Quality

We use three complementary strategies to evaluate description quality: (1) *dataset retrieval evaluation*, which assesses the impact of descriptions on search effectiveness, (2) *reference-based evaluation*, which compares generated descriptions to existing ones using natural language generation metrics, and (3) *reference-free evaluation*, which leverages LLMs to assess descriptions without reference texts. The following subsections detail each approach.

Dataset Retrieval Evaluation. Evaluating the impact of generated descriptions on dataset retrieval performance is critical, as keyword-based search remains the primary application for dataset search engines. This evaluation assesses explicitly the ability of description generation models to enhance dataset discoverability by improving search relevance and ranking–arguably the most critical metric for practical use cases. To measure retrieval performance, we use the Normalized Discounted Cumulative Gain (NDCG@k) metric [41], a widely-used standard for ranking evaluation. NDCG@k measures the effectiveness of a ranking by comparing the ideal ranking of relevant items to the system's actual ranking up to position k. It accounts for both the relevance of the datasets and their positions in the result list, assigning higher scores to datasets that appear earlier.

In our evaluation, we integrate the generated descriptions into a keyword-based search engine and perform search queries representative of typical user behavior. By calculating NDCG@k scores for search results with and without the generated descriptions, we can quantify the improvement in dataset discoverability attributable to our method. This retrieval-based evaluation provides a direct measure of description effectiveness in supporting dataset search engines. Unlike intrinsic evaluations, which focus on the quality of the descriptions themselves, this extrinsic evaluation measures their practical impact on improving search outcomes.

Reference-Based Evaluation. For datasets that already have existing descriptions, we can compare generated descriptions with the original ones. This allows us to assess whether our automated approach produces descriptions that have similar quality to those written by humans. This is particularly important for data portals seeking to enhance or update their metadata, ensuring that automated descriptions meet user expectations.

METEOR: The METEOR score [6] combines precision, recall, synonymy, and stemming for a more flexible comparison between the generated and reference descriptions.

ROUGE: The ROUGE score [54] focuses on recall by measuring the overlap of n-grams, longest common subsequences, and skip-bigrams between the generated and reference texts.

BERTScore: BERTScore [98] uses pre-trained language models, such as BERT, to measure the similarity of embeddings between the generated text and the reference text. Unlike ROUGE and METEOR, it evaluates semantic similarity by comparing contextualized word representations, providing an assessment of whether the generated description conveys the same information as the reference. In this work, we report the F1 variant of BERTScore.

These metrics provide a structured, quantitative evaluation of how well the generated description matches the reference in terms of wording, content, and meaning.

Reference-Free Evaluation. For datasets without reference descriptions, reference-free evaluation is essential. In this scenario, we leverage large language models (LLMs) as judges to evaluate the quality of the generated descriptions. Instead of relying on direct comparison with reference texts, LLM-based evaluation assesses attributes such as coherence, relevance, clarity, and coverage of key

dataset features [31, 58]. We also conduct expert evaluation on a sample of descriptions to validate their faithfulness to the dataset and the reliability of LLM evaluation results.

LLM-Based Evaluations. We first adopt *Pointwise* evaluation by following the prompt design from G-Eval [58], incorporating task introduction, evaluation criteria, and evaluation steps. Additionally, we include example evaluations to guide the LLM in rating dataset descriptions. In this setting, we evaluate each candidate item individually based on the following criteria: *completeness*, *conciseness*, and *readability*. For instance, compared with search-focused descriptions (SFD), user-focused descriptions (UFD) are expected to score higher on conciseness and readability but lower on completeness. In addition, we use *Pairwise* evaluation in which an LLM is presented with two candidate descriptions and must select the superior one [11, 59]. For each dataset, we uniformly sample ten unordered pairs of descriptions generated by distinct methods. Each pair is judged twice—once for each presentation order—to mitigate positional bias. Similarly, the LLM evaluates the candidates along three dimensions: *completeness*, *conciseness*, and *readability*. We report the *win rate*, defined as the number of victories divided by the total number of pairwise comparisons in which the method participates.

However, it is important to acknowledge that LLM-based evaluations may introduce biases inherent in the models' training data. The models might favor certain styles or content, especially when evaluating outputs generated by the same model [67]. To mitigate this issue, we utilize cross-evaluation—for example, we use Llama to evaluate descriptions generated by GPT (and vice versa). By employing different models for generation and evaluation, we reduce the likelihood of shared biases influencing the assessment.

Expert Evaluation. To validate the LLM-based metrics, three annotators evaluated a stratified sample of 48 datasets (12 from ECIR-DDG, 36 from NTCIR-DDG) across four description types: AutoDDG variants (SFD-GPT and UFD-GPT), original descriptions, and LLM-GPT (our strongest baseline). Annotators assessed four dimensions on a 1–10 scale: *completeness*, *conciseness*, *readability* (matching automated metrics), and *faithfulness*—the accuracy with which descriptions represent actual dataset content without hallucinations or factual errors, including the dataset's variables, structure, and statistics, etc. The evaluation guidelines are given in the extended version of this paper [95]. We assess inter-annotator agreement (IAA) and LLM judge-human alignment by comparing GPT and Llama evaluations against normalized human consensus using Pearson's r on z-score normalized ratings.

Reference-free evaluation complements traditional metrics by incorporating semantic and contextual assessments, which are crucial for datasets with no or low-quality descriptions.

3.2 New Dataset Retrieval Benchmarks

Since our primary goal is to generate descriptions that improve dataset retrieval, we need benchmark collections that include not only tabular datasets but also queries and human-labeled results. Moreover, to capture realistic and challenging real-world scenarios, we must include tables that contain a large number of rows and columns. Therefore, we consider the following criteria to select table search benchmarks: (1) support for keyword-based queries; (2) inclusion of CSV datasets collected from Open Data Portals; and (3) availability of query relevance data, typically represented as triples of (CSV dataset, keyword query, relevance score). While there are several table search benchmarks [15, 20, 37, 42, 51, 55, 96, 97], only two meet these criteria: ECIR [19] and NTCIR [44]. Below, we describe how we adapted them to evaluate the effectiveness of description generation for keyword-based dataset retrieval. The new benchmarks are summarized in Table 1.

ECIR-DDG Benchmark. ECIR contains 2,417 tabular datasets published by the U.S. Federal Government in data.gov [24] and consists of six tasks, covering various domains such as public health, economic data, and environmental statistics, each of which describes an information need

Table 1. Statistics of the ECIR-DDG and NTCIR-DDG benchmarks: number of queries, the average, minimum, and maximum number of relevant tables per query (Rel. Tabs/Query), total tables per query (Tabs/Query), total tables per benchmark (Tabs/Bench), rows per table (Rows/Tab) and columns per table (Cols/Tab). These statistics highlight the diversity and scale of the datasets in the benchmarks.

Benchmark	Query	Rel. Tabs/Query			Tabs/Query			Tabs/Bench	Rows/Tab			Cols/Tab		
		Avg	Min	Max	Avg	Min	Max		Avg	Min	Max	Avg	Min	Max
ECIR-DDG	120	12	8	16	169	76	549	1015	17977	4	183539	22	4	337
NTCIR-DDG	32	10	5	22	19	10	42	615	78630	3	2717008	14	2	74

and the associated relevant datasets along with relevance scores. For each task, 20 distinct keyword queries were created using Mechanical Turk, and relevance judgments for *task-dataset pairs* were obtained through crowdsourcing. To validate the benchmark's *query-dataset pairs*, for each query q, we randomly sampled datasets D_{rel} labeled as relevant and datasets D_{irrel} labeled as irrelevant for q.

We manually reviewed the dataset snippets—including the title, description, column names, and a sample of rows—to determine how many datasets in D_{rel} were truly relevant to the query q (true positives) and how many in D_{irrel} were truly irrelevant to q (true negatives). Manual validation revealed a true-positive rate of 9.87%, with 23 out of 233 dataset-query pairs (D_{rel}, q) identified as valid. The true negative rate was 98.5%, with 65 out of 66 dataset-query pairs (D_{irrel}, q) confirmed as irrelevant.

Given the low true-positive rate, we used Google Dataset Search (GDS) to construct a new collection of relevant datasets for the queries. We issued the keyword queries to GDS and manually verified the retrieved datasets for relevance. The collected datasets, along with their original titles and descriptions, are considered relevant to the benchmark. We retain the original irrelevant candidates without modification.

NTCIR-DDG Benchmark. NTCIR was built as a community effort carried out by six research groups. They collected data from Data.gov (U.S. Government's open data) [24] and e-Stat (Japan's Government open data) [40], comprising documents and datasets across domains such as climate, population, economy, and transportation. Queries were constructed from real user questions found in community Q&A forums and refined through crowdsourcing. They include geographical, temporal, and numerical terms, reflecting common information needs in dataset search. We focus on the English component of NTCIR, which encompasses datasets in different formats (e.g., CSV, PDF, JSON, XML, and RDF). Given that we aim to evaluate descriptions of tabular data, we use only CSV datasets. To ensure sufficient data for meaningful evaluation, we selected queries with at least five relevant datasets and at least 10 total datasets (relevant and irrelevant combined). This filtering process resulted in a benchmark with 32 queries and a total of 615 datasets (Table 1). We manually validated a sample of the queries and found the relevance assessment to be accurate.

4 Experimental Evaluation

4.1 Experiment Overview

To evaluate our dataset description generation framework, we conduct experiments following the evaluation strategy introduced in Section 3.1 using multiple baselines and description models.

We consider the following models for constructing descriptions:

(1) **Original**: the dataset descriptions provided with the datasets.

(2) **Header+Sample**: a simple method that concatenates column headers with sample values from each column.

(3) **MVP** [83]: the Multi-task Supervised Pre-training for Natural Language Generation (MVP) model leverages multi-task training for improved text generation, including data-to-text generation.

(4) **ReasTAP** [99]: a table pre-training model that incorporates reasoning skills to enhance table comprehension, including table-to-text generation.
(5) **Pneuma** [5]: an end-to-end retrieval-augmented generation (RAG) system designed to support tabular data discovery using natural language queries; we use its `Table Summarizer`, which generates schema summaries via LLM and row summaries by mechanically pairing randomly sampled values with column headers.
(6) **LLM-GPT/Llama**: an LLM-based baseline (GPT or Llama) that generates descriptions from 5 randomly sampled rows, using the same sampling strategy as AutoDDG.

We use the HuggingFace models available for MVP [62] and ReasTAP [73]. Given the limitations of these models with respect to window size, we provide as input samples of rows. For Pneuma, we use its authors' implementation [69].

Finally, we evaluate our proposed AutoDDG framework, which uses large language models (LLMs) to generate dataset descriptions automatically. We selected two cost-effective LLMs for our experiments: `GPT-4o-mini`[1] and `LLaMA-3.1-8B-Instruct`[2], which we refer to as "GPT" and "Llama", respectively, in method names. We instantiate AutoDDG with both models:
(7) **AutoDDG-UFD-GPT/Llama**: generates user-focused descriptions (UFDs) optimized for readability and clarity while maintaining relevance for search.
(8) **AutoDDG-SFD-GPT/Llama**: generates search-focused descriptions (SFDs) designed to improve dataset retrieval in keyword-based search engines.

4.2 Retrieval Performance

To evaluate the effectiveness of dataset descriptions in search tasks, we measure retrieval performance with **BM25** [74] and **SPLADE** [30]. BM25 focuses on lexical matching, ranking documents based on statistics derived from term frequency and inverse document frequency. In contrast, SPLADE retrieves documents using vector search over sparse text representations, which are derived by expanding query terms with semantically related words. For example, it may expand 'lunar' to include 'moon', thereby capturing broader context.

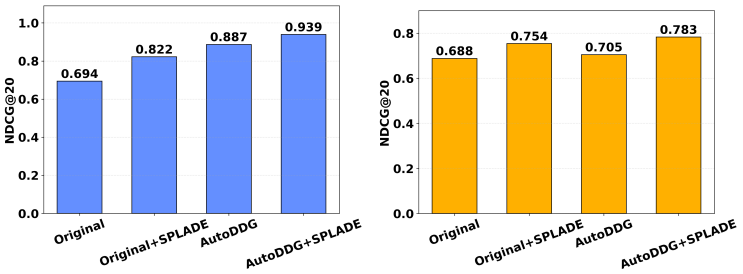
We experimented with both the ECIR-DDG and NTCIR-DDG benchmarks. We created indices for both BM25 and SPLADE using the dataset descriptions generated by the models. For BM25, we created an inverted index, in which terms from each dataset description were tokenized and stored along with their term frequencies and document frequencies.[3] During evaluation, keyword-based queries were issued against this index, and BM25 ranked the dataset descriptions by calculating their relevance scores. For SPLADE, we created a sparse vector representation of each dataset description.[4] During evaluation, keyword-based queries were processed, and the dataset descriptions were ranked by computing cosine similarity between the query's sparse representation and each document's sparse embedding.

BM25 Results. The results in Table 2 show that AutoDDG-SFD methods consistently outperform UFD across both ECIR-DDG and NTCIR-DDG benchmarks (for a given LLM), confirming the effectiveness of SFDs for retrieval. By tailoring descriptions to maximize keyword matching with query topics, SFDs improve search relevance. The GPT-4o-mini model leads to higher NDCG values than Llama on the ECIR-DDG benchmark, and Llama shows better performance on the NTCIR-DDG benchmark. This suggests that different LLMs exhibit varying strengths depending on the characteristics of the dataset and query topics.

[1] https://platform.openai.com/docs/models
[2] https://deepinfra.com/meta-llama/Meta-Llama-3.1-8B-Instruct
[3] We used Okapi BM25 from this library: https://github.com/dorianbrown/rank_bm25
[4] We used SPLADE with FastEmbed: https://qdrant.github.io/fastembed/examples/SPLADE_with_FastEmbed/

Table 2. BM25. Comparison of NDCG scores across different description generation models on the ECIR-DDG and NTCIR-DDG benchmarks. Higher NDCG scores indicate better alignment between the generated descriptions and the relevance of retrieved results for keyword-based search. Bold values represent the highest scores for each metric, while underlined values indicate the second highest.

	ECIR-DDG				NTCIR-DDG			
Model	**NDCG@5**	**NDCG@10**	**NDCG@15**	**NDCG@20**	**NDCG@5**	**NDCG@10**	**NDCG@15**	**NDCG@20**
Original	0.666	0.658	0.668	0.694	0.439	0.575	0.662	0.688
Header+Sample	0.589	0.573	0.575	0.593	0.323	0.451	0.553	0.590
MVP	0.557	0.538	0.533	0.543	0.356	0.472	0.575	0.617
ReasTAP	0.638	0.598	0.599	0.609	0.406	0.485	0.583	0.627
Pneuma	0.652	0.695	0.725	0.757	0.423	0.561	0.651	0.682
LLM-GPT	0.794	0.800	0.803	0.828	0.415	0.536	0.615	0.651
LLM-Llama	0.774	0.783	0.784	0.802	0.422	0.559	0.638	0.674
AutoDDG-UFD-GPT	0.800	0.804	0.815	0.840	0.415	0.553	0.645	0.682
AutoDDG-UFD-Llama	0.760	0.762	0.777	0.801	0.425	0.565	0.652	0.679
AutoDDG-SFD-GPT	**0.849**	**0.856**	**0.867**	**0.887**	**0.458**	0.585	**0.672**	**0.705**
AutoDDG-SFD-Llama	0.766	0.785	0.806	0.828	0.456	**0.594**	0.668	0.696



(a) ECIR-DDG Benchmark (b) NTCIR-DDG Benchmark

Fig. 8. Comparison of NDCG@20 scores for Original, Original+SPLADE, AutoDDG, and AutoDDG+SPLADE on ECIR-DDG and NTCIR-DDG benchmarks shows improvements from combining AutoDDG and SPLADE.

The performance of UFD is comparable to that of SFD. However, as expected, the focus on user readability limits its effectiveness in purely search-oriented metrics. Traditional baselines such as MVP and ReasTAP perform significantly worse than both UFD and SFD, highlighting their limitations in generating effective descriptions for retrieval tasks. Interestingly, original descriptions sometimes outperform these baselines, underscoring the importance of domain-specific context and the limitations of these data-to-text approaches in keyword-based retrieval.

SPLADE Results. Figure 8 compares retrieval performance using the original and AutoDDG's SFD descriptions with BM25 and SPLADE. The results show that SFD descriptions consistently outperform the original descriptions when using BM25's lexical matching. Similarly, we observe that SPLADE's semantic term expansion in the vector space yields significant gains over BM25 on the original description. This is expected, since both AutoDDG and SPLADE use different approaches to expand the information contained in the original descriptions. However, the best results are achieved by combining SPLADE with AutoDDG descriptions. These results suggest that the approaches are complementary and that AutoDDG's SFD descriptions provide additional helpful information not present in the original description. A complete table of NDCG scores with SPLADE is available in the extended version of this paper [95].

4.3 Evaluation of Dataset Description Quality

We follow the methodology introduced in Section 3.1 to evaluate the quality of the generated dataset descriptions using both reference-based and reference-free methods. We: 1) assess alignment with

Table 3. *Pointwise* evaluation of dataset description quality on reference-free metrics and *Pairwise* evaluation across all models for both the ECIR-DDG and NTCIR-DDG benchmarks. Scores are reported as GPT/Llama pairs. Completeness (Comp), Conciseness (Conc), and Readability (Read) are scaled from 0 to 10, while Pairwise is scaled from 0 to 1. Higher values indicate better performance. Bold values denote the highest scores for each metric, and underlined values denote the second-highest.

	ECIR-DDG				NTCIR-DDG			
Model	**Comp (G/L)**	**Conc (G/L)**	**Read (G/L)**	**Pairwise (G/L)**	**Comp (G/L)**	**Conc (G/L)**	**Read (G/L)**	**Pairwise (G/L)**
Original	4.07/5.02	6.83/7.35	6.04/7.14	0.35/0.31	4.46/4.81	7.04/6.14	6.17/6.37	0.34/0.39
H+S	3.50/3.56	5.39/4.00	4.36/4.15	0.18/0.26	3.24/3.57	5.01/3.92	3.98/4.57	0.24/0.30
MVP	1.52/1.57	3.24/3.25	2.41/2.84	0.05/0.09	1.29/1.51	2.57/2.90	1.96/2.43	0.03/0.11
ReasTAP	1.96/1.91	4.38/5.81	3.33/4.85	0.03/0.07	1.69/1.83	3.63/4.81	2.54/3.98	0.04/0.06
Pneuma	5.25/7.09	5.65/4.78	5.19/5.81	0.28/0.46	5.26/6.64	5.89/5.27	5.47/6.32	0.28/0.48
LLM-GPT	6.40/6.75	8.50/**8.00**	8.17/8.08	0.50/0.45	6.40/6.54	8.30/7.87	8.15/8.03	0.48/0.46
LLM-Llama	6.17/6.50	8.26/7.92	8.04/8.03	0.52/0.45	6.03/6.17	8.23/7.82	8.00/7.91	0.49/0.42
UFD-GPT	8.19/8.57	**8.97/8.00**	**8.99/8.99**	0.73/0.72	8.17/8.58	**8.49/8.00**	**8.89/8.99**	0.77/0.68
UFD-Llama	7.72/8.43	8.51/7.86	8.63/8.50	0.69/0.68	7.59/8.06	8.33/7.88	8.42/8.65	0.60/0.60
SFD-GPT	**8.96/8.98**	7.35/5.97	7.90/7.00	**0.86**/0.71	**9.00/9.00**	8.00/6.31	9.00/8.16	0.87/0.70
SFD-Llama	8.95/8.89	6.24/5.56	7.19/6.79	0.77/**0.78**	8.99/9.00	7.96/6.01	8.94/8.04	**0.79/0.70**

ground-truth descriptions using different methods—METEOR, ROUGE, and BERTScore, which capture phrase similarity, lexical overlap, and semantic consistency; 2) evaluate the descriptions' intrinsic quality using LLM-based scoring for completeness, conciseness, readability–these metrics assess how well the descriptions convey relevant information while maintaining clarity and accuracy; 3) carry out a user evaluation in which experts assess descriptions to validate both their faithfulness to source datasets and the reliability of LLM-based evaluation.

This multi-pronged approach to evaluation offers insights into how well the descriptions meet human readability standards and perform in automated search and retrieval tasks. This is useful for users seeking to publish new datasets, as these evaluations can predict the performance of the generated descriptions for both front-end user interaction and back-end dataset retrieval.

Reference-Based Evaluation. The trends observed in the reference-based evaluation (Table 4) are consistent with those from other evaluations. The AutoDDG-UFD models perform exceptionally well on *METEOR*, indicating improved semantic alignment with reference descriptions. This suggests that UFDs, designed with human-centered presentation in mind, better capture the intent and meaning of the original data description. In contrast, AutoDDG-SFD achieves superior scores in *ROUGE*, signaling a higher overlap with the reference text's exact terms and phrases. This aligns with the typical goal of search-focused descriptions: ensuring high precision in matching key data points. The AutoDDG methods achieve slightly lower BERTScores than LLM-only baselines, likely because BERTScore favors embedding similarity to reference phrasing. In contrast, our approach prioritizes factual grounding, generating semantically accurate but linguistically diverse descriptions, as evidenced by competitive METEOR and ROUGE scores.

Reference-Free Evaluation. The results of this evaluation, summarized in Table 3, show that AutoDDG descriptions consistently outperform other methods. AutoDDG-UFD-GPT excels in *Conciseness* and *Readability*, confirming the expectation that user-focused descriptions (UFDs) prioritize clarity and readability. This reinforces our design decision of creating user-centered descriptions. Conversely, AutoDDG-SFD-GPT ranks highest in *Completeness*, as expected from search-focused descriptions (SFDs) that prioritize comprehensive data coverage, even at the expense of readability or conciseness. The LLM-only methods achieve competitive *Conciseness* scores by generating shorter descriptions without profiling, but their significantly lower *Completeness* underscores their incomplete knowledge of the data. Furthermore, the *Pairwise* comparison included in Table 3 also demonstrates that AutoDDG outperforms other methods. Despite numerical differences, GPT and Llama evaluations show consistent trends across all reference-free metrics.

Table 4. Evaluation of dataset description quality on reference-based metrics for both the ECIR-DDG and NTCIR-DDG benchmarks. Higher values indicate better performance. Bold values represent the highest scores for each metric, while underlined values indicate the second highest.

	ECIR-DDG			NTCIR-DDG		
Model	**METEOR**	**ROUGE**	**BERTScore**	**METEOR**	**ROUGE**	**BERTScore**
Original	-	-	-	-	-	-
Header+Sample	2.28	3.61	75.17	2.41	4.25	75.46
MVP	1.55	1.83	78.85	1.58	2.60	78.85
ReasTAP	6.48	8.82	79.99	4.94	6.71	79.93
Pneuma	7.58	26.97	77.68	10.24	28.29	78.48
LLM-GPT	11.93	20.14	<u>82.45</u>	12.60	22.28	<u>82.89</u>
LLM-Llama	12.04	19.03	**82.85**	13.27	22.72	**83.11**
AutoDDG-UFD-GPT	<u>15.48</u>	29.78	82.24	**15.90**	<u>30.56</u>	82.47
AutoDDG-UFD-Llama	**16.46**	30.08	82.26	<u>15.64</u>	27.80	82.60
AutoDDG-SFD-GPT	13.00	**34.50**	80.07	14.49	**35.45**	80.24
AutoDDG-SFD-Llama	12.27	<u>33.85</u>	79.17	13.80	32.48	79.40

Table 5. Human evaluation of description quality across methods. Scores represent mean ± standard deviation on a 1–10 scale, computed from z-score normalized ratings across three annotators (n = 192 per method). Inter-annotator agreement (IAA) is measured by Pearson's r on normalized scores.

Model	**Comp.**	**Conc.**	**Read.**	**Faith.**	**Average**
Original	4.33 ± 1.68	6.45 ± 1.75	6.35 ± 1.81	5.47 ± 1.65	5.65 ± 0.85
LLM-GPT	6.95 ± 0.76	<u>8.73 ± 0.55</u>	<u>8.83 ± 0.58</u>	7.59 ± 1.19	8.03 ± 0.79
UFD-GPT	<u>8.67 ± 0.50</u>	**8.75 ± 0.70**	**8.99 ± 0.61**	<u>9.00 ± 0.89</u>	**8.85 ± 0.14**
SFD-GPT	**9.37 ± 0.45**	7.40 ± 0.55	7.61 ± 0.58	**9.09 ± 0.82**	<u>8.37 ± 0.87</u>
IAA (avg r)	0.792	0.477	0.494	0.593	0.719

User Evaluation. To validate LLM-based evaluation, we conduct a user evaluation on a stratified sample of 48 datasets (192 descriptions total). Three users assessed the faithfulness of the generated descriptions by comparing them with the original datasets. We normalize user scores to remove individual annotator bias prior to analysis. Table 5 shows that AutoDDG descriptions have substantially higher faithfulness than the ones derived by LLM-GPT. This indicates that the integration of data profilers and topic generators effectively reduces hallucinations and factual errors—a critical advantage over pure LLM generation. Original descriptions achieve only 5.47 ± 1.65 for faithfulness, showing high variance and indicating inconsistent quality. Both AutoDDG methods also excel in completeness and overall quality, which aligns with the LLM-as-judge results. Annotator comments reveal that original descriptions provide only metadata and external links rather than content descriptions, achieving *"conciseness at the cost of completeness"*. LLM-only descriptions frequently contain *factual errors including wrong temporal ranges, incorrect statistics, and unsupported inferences*, which AutoDDG methods largely eliminate. Users identified occasional errors in the AutoDDG descriptions, which we traced to bugs in the algorithmic profiler—for example, misidentifications of temporal coverage or units of measure. Nonetheless, the descriptions were consistent with the information output by the profiler.

Figure 9 shows that there is a high correlation between human annotators and the scores derived by the LLMs, confirming the reliability of the automated LLM evaluation. Panel (a) shows substantial inter-annotator agreement (r=0.72) on overall assessment, computed by averaging the four scores

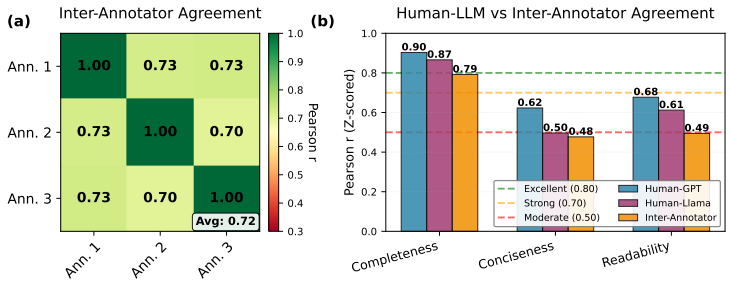


Fig. 9. **(a)** Inter-annotator agreement on overall assessment (avg $r = 0.72$). **(b)** Human-LLM vs inter-annotator agreement across three metrics. All correlations use Pearson's r on z-score normalized ratings ($n = 192$, $p < 0.001$).

for each dataset before measuring agreement between annotators. Panel (b) reveals strong LLM-human agreement on objective completeness metrics. Inter-annotator agreement on conciseness and readability is lower, reflecting the subjective nature of these dimensions. By analyzing the comments from the annotators, we can see that they considered different stylistic criteria: *one prioritized logical flow and appropriate detail levels for dataset complexity, while another focused on redundancy across sections and representative examples.* This variation is expected for subjective writing assessments where individual preferences naturally differ. Notably, LLM evaluators achieve better agreement with mean human ratings (GPT: $r = 0.62$, Llama: $r = 0.5$), indicating that automated evaluation effectively aggregates diverse human judgments on these subjective dimensions. Overall, LLM-as-judge achieves human-level reliability for assessing description quality.

Summary. Our experimental evaluation confirms our hypothesis: UFDs score higher in conciseness, readability, and semantic coherence (as seen in *BERTScore*), while SFDs perform better in completeness and exact overlap with reference descriptions. These insights suggest that, although UFDs are more suitable for user assessment and semantic clarity, SFDs are better suited to search-related tasks that benefit from more detailed descriptions. Therefore, using both descriptions in a dataset search engine improves both the user experience and the quality of search results.

4.4 Cost Analysis

We evaluate the LLM-related computational cost and scalability of the AutoDDG pipeline across a sample of 283 datasets from the NTCIR and ECIR benchmarks. We compare our approach against all baselines except the heuristic-based H+S, whose runtime is negligible. To ensure a fair comparison, all LLM-based methods (AutoDDG, Pneuma, and LLM-GPT) utilize `gpt-4o-mini` as the underlying backbone. For conciseness, we report results for the AutoDDG UFD-GPT configuration; for SFD, the cost is incremental, requiring one additional LLM call using the UFD output.

To mitigate the cost and runtime overheads inherent in making multiple LLM API calls for the Semantic Profiling stage (Algorithm 1), we implemented two optimization strategies:
–*AutoDDG-MT* (multi-threaded) applies Algorithm 1 concurrently by issuing distinct API calls for each column, utilizing a pool of 64 worker threads to improve throughput.
–*AutoDDG-GP* (group prompting) utilizes a batching strategy where multiple columns are grouped into a single prompt. This enables the sharing of system instructions across columns within a single context window, thereby reducing the total number of input tokens.
We refer to the sequential strategy as *AutoDDG-Seq*.

Scalability: Runtime and Token Usage. We assessed the scalability of the different approaches with respect to dataset width–varying number of columns. For AutoDDG, the total runtime includes

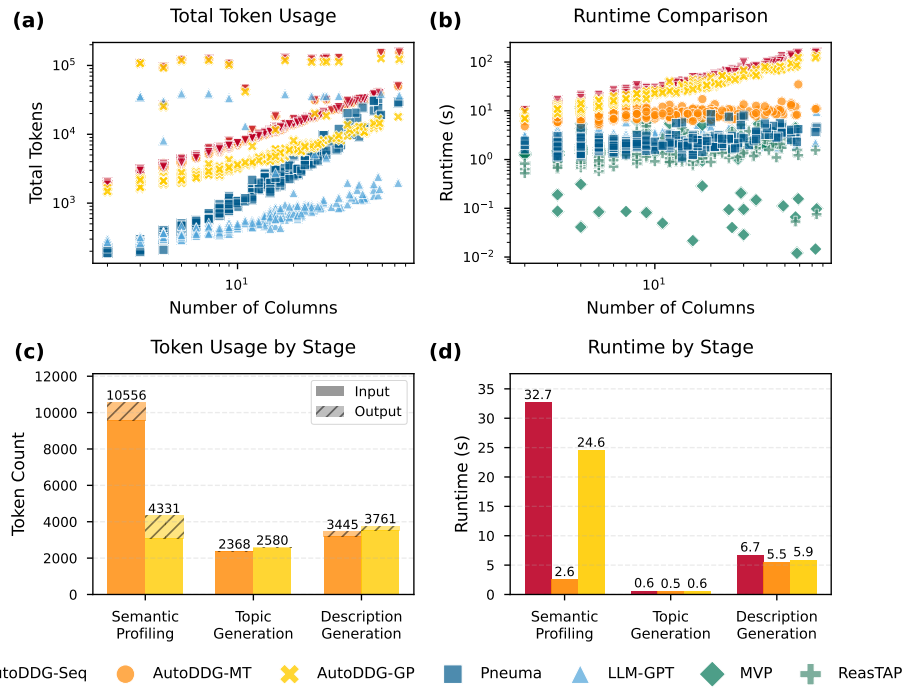


Fig. 10. Cost and scalability analysis across 283 datasets. (a-b) runtime and LLM token usage scaling relative to dataset width (log-log scale) (token usage is shown for LLM-based methods). AutoDDG maintains predictable scalability despite higher overhead compared to baselines. (c-d) Stage-wise cost breakdown for AutoDDG. The comparison between multi-threading (orange) and group prompting (yellow) reveals a key trade-off: group prompting reduces input tokens but increases latency due to serialization bottlenecks–its runtime is similar to that of the AutoDDG sequential variant (red).

semantic profiling, topic generation, and description generation. To ensure a fair comparison, we also ran the Pneuma baseline using multiple threads. The results are shown in Figure 10.

The concurrent execution of API calls results in a substantial performance improvement: for the maximum number of columns, *AutoDDG-MT* is about 10 times faster than *AutoDDG-Seq*. *AutoDDG-MT* has predictable scaling with dataset width. While its runtimes are longer than those of the baselines, they are of the same order of magnitude. On average, *AutoDDG-MT* takes 8.63 sec per dataset, compared to 2.40 sec for Pneuma, 2.76 sec for LLM-GPT, 1.93 sec for MVP, and 2.01 sec for ReasTAP. To test enterprise-level scalability, we ran AutoDDG-MT over a table with 337 columns–it completed the task in 17.16 seconds.

The token usage for the different LLM-based methods is shown in Figure 10(b). Since the token usage is identical for *AutoDDG-Seq* and *AutoDDG-MT*, we present the results for the former. As expected, the number of tokens used by AutoDDG grows with dataset width. The new group prompting strategy (*AutoDDG-GP*) is effective at reducing token usage: it reduces the average number of tokens per dataset from 16,370 to 10,674 and significantly dampens the growth rate for wide tables. Note that AutoDDG uses more tokens than the LLM-only baseline (which issues a single prompt per dataset) and Pneuma–while Pneuma, like AutoDDG, issues one call per column, it does not execute the token-heavy description-generation step.

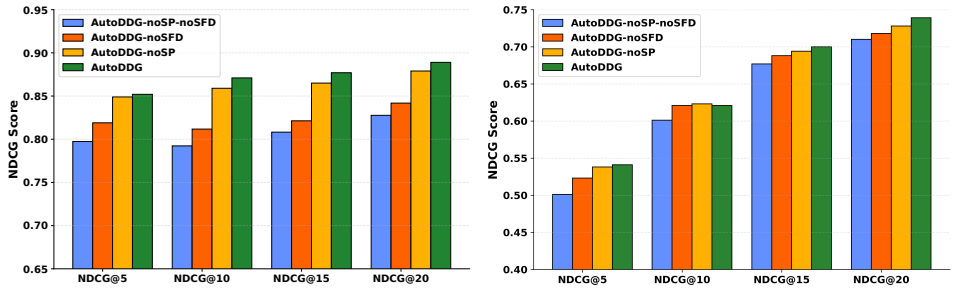


Fig. 11. Comparison of NDCG scores across different settings of AutoDDG (noSP-noSFD, noSFD, noSP and full configuration) on BM25 for the ECIR-DDG (left) and NTCIR-DDG (right) benchmarks, showing the impact of enabling semantic profile (SP) and search-focused description (SFD) on retrieval performance.

In Figure 10(b), we also observe coincident token-count outliers for AutoDDG and the LLM-GPT baseline, which are absent for Pneuma. While all three approaches use an LLM, their input strategies differ: AutoDDG and LLM-GPT use sampled row values to construct prompts, resulting in token counts that increase substantially when datasets contain very long values in their columns. Since Pneuma does not include value samples during LLM-powered schema narration, it does not incur token costs for those samples. Importantly, Figure 10(a) confirms that these input-token surges do not translate into runtime outliers, demonstrating that *AutoDDG-MT* remains efficient even when processing long textual values.

To further assess the scalability of the different methods, we experimented with a very wide table containing 337 columns. For this dataset, Pneuma consumed 1,202,309 tokens, whereas AutoDDG-MT required 218,760 tokens. This difference arises from their respective architectural designs: Pneuma incorporates the entire schema as context for each column description to maintain global coherence, whereas AutoDDG processes columns independently during semantic profiling. This shows that engineering systems to use LLMs efficiently is challenging, as many trade-offs must be considered. It also suggests that AutoDDG could serve as a complementary tool for Pneuma's table-summarization stage, particularly for handling datasets with many columns.

We find that the modest overhead in runtime and token consumption for AutoDDG is justified by the significant performance gains in both retrieval effectiveness (Table 2) and generated description quality (Tables 3, 4, and 5).

Optimizing LLM Usage. We further analyze the trade-offs between the two optimization strategies. As Figure 10(d) shows, group prompting (yellow) is highly effective at reducing cost. Sharing system instructions reduces the semantic-profiling input token count by approximately 60%. However, Figure 10(c) reveals a critical trade-off between token and runtime efficiency: multi-threaded single-column prompting (orange) achieves far superior performance (~2.6s) compared to group prompting (~24.6s). This performance gap stems from the auto-regressive nature of LLM decoding: even when columns are grouped, the semantic profiling is performed sequentially for each column within a single response window (intra-batch serialization). *AutoDDG-MT* eliminates this serialization bottleneck. In practice, we can use multi-threading for latency-sensitive applications, while group prompting remains a viable cost-saving alternative for resource-constrained environments where high concurrency is not feasible.

AutoDDG: Total Cost. The description generation stage dominates the remaining computational costs, averaging 3,445 tokens and 5.5 seconds per dataset. The total cost per dataset is approximately $0.003 (using `gpt-4o-mini`), with Group Prompting further reducing this to ≈$0.0023, confirming AutoDDG's applicability for large dataset collections.

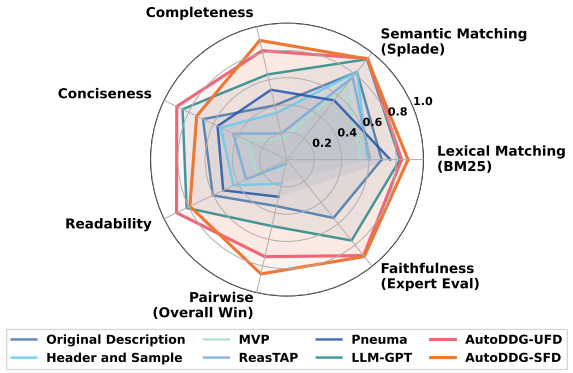


Fig. 12. Radar chart comparing the performance of various dataset description methods across evaluation metrics.

4.5 Ablation Study

To assess the contributions of each module in AutoDDG, we conduct an ablation study by selectively disabling key components and analyzing their impact on retrieval performance. This allows us to isolate the effects of the Semantic Profiler (SP) and the Search-Focused Description (SFD). We compare the following variations: (1) **AutoDDG-noSP-noSFD**: Generates descriptions without SP and SFD, relying only on basic dataset samples and statistics. (2) **AutoDDG-noSFD**: Includes SP but omits SFD, enhancing descriptions with semantic insights while not optimizing for search retrieval. (3) **AutoDDG-noSP**: Includes SFD but omits SP, test the configuration optimizing for search retrieval without SP. (4) **AutoDDG**: Includes both SP and SFD, leveraging semantic insights while optimizing descriptions for search retrieval.

Impact of Different Modules in AutoDDG. Figure 11 summarizes the results, measured by NDCG scores. The inclusion of the semantic profile (SP) module (AutoDDG-noSFD) improves retrieval compared to the baseline (AutoDDG-noSP-noSFD). Adding the search-focused description (SFD) module alone (AutoDDG-noSP) provides further gains. The AutoDDG with all components has the best performance, reinforcing its role in enhancing search relevance.

4.6 Overview of Experimental Results

The radar chart in Figure 12 provides a comparative overview, illustrating how different methods perform across these dimensions. The results show that UFD excels in readability and conciseness, making it more suitable for user-facing applications, while SFD improves completeness and retrieval relevance, increasing dataset findability. By combining these strengths, AutoDDG achieves a balanced and superior performance compared to baseline methods (Original, Header+Sample, MVP, and ReasTAP), demonstrating its effectiveness in generating high-quality descriptions for both human interpretation and machine-based search.

5 Related Work

Dataset Search. Studies on information-seeking behavior reveal significant challenges in dataset discovery. Koesten et al. [48] found that users often cannot find needed data and identified common search strategies and key dataset attributes in queries: category/topic, geographic region, data granularity (spatial and temporal resolution), and relevance indicators like summary statistics and data semantics. Papenmeier et al. [68] reported that “current systems fail to sufficiently support scientists in their data-seeking process” and because the metadata associated with datasets does

not cover information that is sought by scientists. This underscores the disconnect between users' information needs and the datasets they can actually find, and the importance of data descriptions that accurately characterize the data [14]. Sostek et al. [79] confirmed these metadata shortcomings in Google Dataset Search and identified an additional challenge: inconsistency in dataset descriptions increases mental load during relevance assessment. We use the findings from these studies to guide the design of the data-driven and semantic summaries in AutoDDG. By using the dataset's actual contents to generate summaries and augmenting them with semantic information, we can derive high-quality descriptions. While it may not be possible to derive completely uniform descriptions, given that datasets contain different information, an automated process can be applied to all datasets to enforce a common structure. For example, it is not possible to summarize the spatial extent of a dataset that lacks spatial data; however, for all spatial datasets, this information can be provided and represented in a similar manner.

Discovery queries that go beyond keyword search have been proposed in which a dataset D is the query, and the results include datasets that are related to D, e.g., can be joined or concatenated, or are correlated [13, 46, 75, 101]. These queries are orthogonal to and can be combined with keyword search.

Table Understanding and Representation. Many models have been developed to enhance table understanding and facilitate various downstream tasks by semantically representing table content [26, 28, 35, 38, 39, 45, 49, 56, 81, 85, 88, 92, 94]. These areas include, but are not limited to, table question answering, column-type analysis, and table-type classification. Recently, large language models (LLMs) have begun to play a significant role in various data-related tasks, including column type annotation (CTA) [28, 45, 49] and table-class detection [45]. While these approaches advance semantic understanding of tables, they do not directly address the problem of generating dataset descriptions. Nonetheless, the information they derive can be used to enrich dataset descriptions. For example, AutoDDG uses the strategy proposed in [45] to obtain the dataset topic. In future work, we will explore incorporating additional semantic information, such as column types [28].

Text Generation from Tables. Table-to-text generation models have attained significant attention due to their ability to transform structured data into coherent natural language text [17, 18, 32, 57, 71, 72, 76, 80, 83, 87, 99]. Among recent advancements, ReasTAP [99] introduces a novel table pre-training approach that enhances models' reasoning capabilities through synthetic question-answering examples and demonstrates notable performance gains in producing logically faithful text across various downstream tasks. Additionally, the Multi-task Supervised Pre-training for Natural Language Generation (MVP) model [83] leverages multi-task supervised pre-training to excel in a broad range of natural language generation tasks, including knowledge-graph-to-text and data-to-text generation. However, these models are trained on datasets such as Rotowire [91], WikiBio [50], and LogicNLG [16], which are designed to generate narrative text tailored to specific tasks, such as sports game summaries and biographical sentences. While effective for generating text from structured data, these models lack the ability to: (1) create descriptions for keyword-based search and high-level overviews; (2) handle large and heterogeneous datasets with complex structures. Moreover, they require training for a specific objective. In contrast, we aim to generate summaries that are sufficiently general to accommodate a wide range of datasets and address diverse information needs, without requiring specific training for each dataset.

Pneuma [5] is an end-to-end retrieval-augmented generation (RAG) system designed to support tabular data discovery using natural language queries. While Pneuma and AutoDDG aim to improve dataset discoverability, they have important technical differences. Pneuma leverages LLMs to narrate table schemas to extract attribute semantics and relies on row samples to represent the dataset's contents. In contrast, AutoDDG applies both a data-driven and an LLM-powered profiler to obtain both semantic information and derive a global summary of the dataset contents. In Section 4, we

show that AutoDDG has better retrieval performance than Pneuma, suggesting that Pneuma could benefit from integrating AutoDDG's approach for description generation.

LLM-based Metadata Extraction from Documents. Automating metadata extraction from scholarly documents using LLMs is an emerging research area, though challenges remain regarding robustness and accuracy [2, 89]. Frameworks like MOLE [3] employ LLMs to extract and validate over 30 distinct attributes that describe datasets (e.g., license, volume, tasks) from full-text papers. Data Gatherer [61] focuses on extracting structured dataset references (identifiers and repository names) from open-access scientific publications available in databases such as PubMed Central. In domain-specific applications, LLMs were utilized to complete missing metadata fields in physical sample records (e.g., catalog records or specimen records) [78]. Effective strategies for this task include record summarization before classification, few-shot prompting, and integrating hierarchical domain taxonomy via bottom-up traversal. While these works share AutoDDG's goal of improving resource findability, they differ in the types of resources and in the approaches to analyzing data and generating LLM context.

LLMs as Judges. LLMs have recently been explored as scalable, consistent evaluators of natural language generation (NLG) outputs, offering an alternative to costly and variable human assessments. These automated evaluations typically follow two paradigms: *pointwise* scoring and *pairwise* comparison [52]. The pointwise approach assigns numerical ratings to outputs based on dimensions such as fluency, relevance, and faithfulness. For example, Chiang and Lee [21] demonstrated that GPT-4's Likert-scale ratings for story generation and adversarial text examples align closely with expert human judgments, and Wang et al. [86] reported that ChatGPT's pointwise evaluations on summarization, story generation, and data-to-text tasks achieve state-of-the-art correlations with human references. In contrast, pairwise evaluation requires the model to choose the better of two candidates. This method provides more discriminative judgments [59] and has been confirmed to closely approximate human rankings in large-scale studies [7]. Our evaluation design is inspired by these prior works.

6 Conclusions and Future Work

Effective dataset descriptions are essential for improving findability and helping users assess dataset relevance. However, many datasets lack informative descriptions, limiting their discoverability and usability in search systems. In this paper, we introduce AutoDDG, an end-to-end framework for automated dataset description generation that systematically addresses this challenge. AutoDDG combines data-driven profiling with LLM-powered semantic augmentation to generate high-quality descriptions that balance comprehensiveness, faithfulness, conciseness, and readability. The derived descriptions can be indexed by standard search engines and used to create embeddings that support the efficient evaluation of textual queries–improving discoverability without requiring infrastructure changes. They also enable RAG systems to incorporate datasets alongside traditional document sources.

We proposed a multi-pronged evaluation strategy for dataset descriptions that measures improvements in dataset retrieval, assesses alignment with existing descriptions, and employs LLM-based scoring for intrinsic quality evaluation. Recognizing the limitations of existing benchmarks, we introduced two new dataset search benchmarks, ECIR-DDG and NTCIR-DDG, to enable rigorous assessment.

Our experimental results demonstrate that AutoDDG significantly improves dataset retrieval performance, produces high-quality, accurate descriptions, and helps users better understand and assess datasets. Beyond its immediate impact on dataset search engines, our framework provides a scalable and systematic approach to metadata enhancement, benefiting data repositories, open

data portals, and enterprise data lakes. Furthermore, it open new opportunities for LLMs to makes datasets understandable for LLMs.

Limitations and Future Directions. While AutoDDG demonstrates strong performance for tabular datasets, an important avenue for future work is to extend its applicability to a broader range of dataset types and content structures. In particular, will explore approaches to incorporate additional data modalities, such as images, to generate richer descriptions.

Our data-driven and semantic profilers were designed to extract information identified as necessary for dataset search [48, 68, 79]. Expanding their capabilities by incorporating functional dependency analysis and domain-specific customization during table sampling could enhance their effectiveness by better capturing inter-column relationships and serving diverse research domains and use cases. While our implementation relies on API-based lightweight models, transitioning to local GPU deployment would lead to lower cost and latency. It would also enable optimizations such as batch inference, as used in [5].

A critical challenge in LLM-generated descriptions is hallucination, where the model may introduce incorrect information [36]. Developing robust detection and mitigation strategies to ensure descriptions remain faithful to the dataset's content and context is an important research direction. Finally, while automatic description generation significantly improves dataset findability, the generated descriptions are necessarily incomplete. Exploring human-in-the-loop techniques, in which users can enrich (e.g., by adding provenance and information on data-collection methodology) or validate descriptions, offers an opportunity to further enhance accuracy, completeness, and trustworthiness, particularly in high-stakes domains.

Acknowledgments

This work was supported by NSF awards IIS-2106888 and OAC-2411221, the DARPA ASKEM program Agreement No. HR0011262087, and the ARPA-H BDF program. The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the DARPA, ARPA-H, the U.S. Government, or NSF.

References

[1] Ziawasch Abedjan, Lukasz Golab, and Felix Naumann. 2015. Profiling relational data: a survey. *The VLDB Journal* 24 (2015), 557–581.

[2] Zaid Alyafeai, Maged S Al-Shaibani, and Bernard Ghanem. 2025. MeXtract: Light-Weight Metadata Extraction from Scientific Papers. *arXiv preprint arXiv:2510.06889* (2025).

[3] Zaid Alyafeai, Maged S Al-Shaibani, and Bernard Ghanem. 2025. MOLE: Metadata Extraction and Validation in Scientific Papers Using LLMs. *arXiv preprint arXiv:2505.19800* (2025).

[4] apachesolr [n.d.]. Apache SOLR. https://github.com/apache/solr.

[5] Muhammad Imam Luthfi Balaka, David Alexander, Qiming Wang, Yue Gong, Adila Krisnadhi, and Raul Castro Fernandez. 2025. Pneuma: Leveraging LLMs for Tabular Data Representation and Retrieval in an End-to-End System. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, Vol. 3. Article 200, 28 pages. https://doi.org/10.1145/3725337

[6] Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*. 65–72.

[7] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of Human Judges? A Large Scale Empirical Study across 20 NLP Evaluation Tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 238–255. https://doi.org/10.18653/v1/2025.acl-short.20

[8] Omar Benjelloun, Shiyu Chen, and Natasha Noy. 2020. Google dataset search by the numbers. In *International Semantic Web Conference*. Springer, 667–682.

[9] 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*. 1365–1375.
[10] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33 (2020), 1877–1901.
[11] Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. 2024. Compassjudger-1: All-in-one judge model helps model evaluation and evolution. *arXiv preprint arXiv:2410.16256* (2024).
[12] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. 2021. Auctus: a dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment (PVLDB)* 14, 12 (July 2021), 2791–2794. https://doi.org/10.14778/3476311.3476346
[13] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. 2019. Lazo: A Cardinality-Based Method for Coupled Estimation of Jaccard Similarity and Containment. In *IEEE ICDE*. 1190–1201. https://doi.org/10.1109/ICDE.2019.00109
[14] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. 2020. Dataset search: a survey. *The VLDB Journal* 29, 1 (2020), 251–272.
[15] Qiaosheng Chen, Weiqing Luo, Zixian Huang, Tengteng Lin, Xiaxia Wang, Ahmet Soylu, Basil Ell, Baifan Zhou, Evgeny Kharlamov, and Gong Cheng. 2024. ACORDAR 2.0: A Test Collection for Ad Hoc Dataset Retrieval with Densely Pooled Datasets and Question-Style Queries. In *ACM SIGIR*. 303–312.
[16] Wenhu Chen, Jianshu Chen, Yu Su, Zhiyu Chen, and William Yang Wang. 2020. Logical Natural Language Generation from Open-Domain Tables. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 7929–7942. https://doi.org/10.18653/v1/2020.acl-main.708
[17] Wenhu Chen, Yu Su, Xifeng Yan, and William Yang Wang. 2020. KGPT: Knowledge-Grounded Pre-Training for Data-to-Text Generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 8635–8648.
[18] Zhiyu Chen, Harini Eavani, Wenhu Chen, Yinyin Liu, and William Yang Wang. 2020. Few-Shot NLG with Pre-Trained Language Model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault (Eds.). Association for Computational Linguistics, Online, 183–190. https://doi.org/10.18653/v1/2020.acl-main.18
[19] Zhiyu Chen, Haiyan Jia, Jeff Heflin, and Brian D Davison. 2020. Leveraging schema labels to enhance dataset search. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*. Springer, 267–280.
[20] Zhiyu Chen, Shuo Zhang, and Brian D Davison. 2021. WTR: A test collection for web table retrieval. In *ACM SIGIR*. 2514–2520.
[21] Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937* (2023).
[22] CKAN. 2024. https://ckan.org/.
[23] CKAN. 2024. CKAN User guide. https://docs.ckan.org/en/2.10/user-guide.html.
[24] datagov [n.d.]. U.S. Government's Open Data. https://data.gov. Accessed in October 2025.
[25] dataverse [n.d.]. Dataverse Project. https://dataverse.org.
[26] Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record* 51, 1 (2022), 33–40.
[27] elasticsearh [n.d.]. Elasticsearch. https://github.com/elastic/elasticsearch.
[28] Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. 2024. ArcheType: A Novel Framework for Open-Source Column Type Annotation Using Large Language Models. *Proceedings of the VLDB Endowment (PVLDB)* 17, 9 (2024), 2279–2292.
[29] figshare [n.d.]. Figshare. https://info.figshare.com.
[30] Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. 2021. SPLADE: Sparse lexical and expansion model for first stage ranking. In *ACM SIGIR*. 2288–2292.
[31] Mingqi Gao, Xinyu Hu, Xunjian Yin, Jie Ruan, Xiao Pu, and Xiaojun Wan. 2025. LLM-based NLG Evaluation: Current Status and Challenges. *Computational Linguistics* 51 (June 2025), 661–687. https://doi.org/10.1162/coli_a_00561
[32] Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*. 1978–1988.
[33] Kathleen Gregory and Laura Koesten. 2022. *Human-centered data discovery*. Springer.
[34] James Hendler, Jeanne Holm, Chris Musialek, and George Thomas. 2012. US government linked open data: semantic. data. gov. *IEEE Intelligent Systems* 27, 03 (2012), 25–31.

[35] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Mueller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly Supervised Table Parsing via Pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 4320–4333.

[36] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. 2025. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* 43, 2 (2025), 1–55.

[37] Madelon Hulsebos, Çagatay Demiralp, and Paul Groth. 2023. GitTables: A Large-Scale Corpus of Relational Tables. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, Vol. 1. Article 30, 17 pages. https://doi.org/10.1145/3588710

[38] 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*. 1500–1508.

[39] Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. TABBIE: Pretrained Representations of Tabular Data. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, Online, 3446–3456. https://doi.org/10.18653/v1/2021.naacl-main.270

[40] japangov [n.d.]. Japan Government's Open Data. https://www.e-stat.go.jp. Accessed in October 2025.

[41] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.

[42] Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. 2024. TARGET: Benchmarking Table Retrieval for Generative Tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

[43] Maxat Kassen. 2013. A promising phenomenon of open data: A case study of the Chicago open data project. *Government information quarterly* 30, 4 (2013), 508–513.

[44] Makoto P Kato, Hiroaki Ohshima, Ying-Hsang Liu, and Hsin-Liang Chen. 2021. A test collection for ad-hoc dataset retrieval. In *ACM SIGIR*. 2450–2456.

[45] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. 2024. Chorus: Foundation Models for Unified Data Discovery and Exploration. *Proceedings of the VLDB Endowment (PVLDB)* 17, 8 (2024), 2104–2114.

[46] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. 2023. SANTOS: Relationship-based Semantic Table Union Search. *Proceedings of the VLDB Endowment (PVLDB)* 1, 1, Article 9 (2023), 25 pages.

[47] Laura Koesten, Elena Simperl, Tom Blount, Emilia Kacprzak, and Jeni Tennison. 2020. Everything you always wanted to know about a dataset: Studies in data summarisation. *International journal of human-computer studies* 135 (2020), 102367.

[48] Laura M Koesten, Emilia Kacprzak, Jenifer FA Tennison, and Elena Simperl. 2017. The Trials and Tribulations of Working with Structured Data: -a Study on Information Seeking Behaviour. In *Proceedings of the 2017 CHI conference on human factors in computing systems*. 1277–1289.

[49] Keti Korini and Christian Bizer. 2023. Column type annotation using chatgpt. *arXiv preprint arXiv:2306.00745* (2023).

[50] Rémi Lebret, David Grangier, and Michael Auli. 2016. Neural Text Generation from Structured Data with Application to the Biography Domain. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Jian Su, Kevin Duh, and Xavier Carreras (Eds.). Association for Computational Linguistics, Austin, Texas, 1203–1213. https://doi.org/10.18653/v1/D16-1128

[51] Aristotelis Leventidis, Martin Pekár Christensen, Matteo Lissandrini, Laura Di Rocco, Katja Hose, and Renée J Miller. 2024. A Large Scale Test Corpus for Semantic Table Search. In *ACM SIGIR*. 1142–1151.

[52] Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579* (2024).

[53] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024. Long-context LLMs Struggle with Long In-context Learning. arXiv:2404.02060 [cs.CL]

[54] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[55] Tengteng Lin, Qiaosheng Chen, Gong Cheng, Ahmet Soylu, Basil Ell, Ruoqi Zhao, Qing Shi, Xiaxia Wang, Yu Gu, and Evgeny Kharlamov. 2022. ACORDAR: a test collection for ad hoc content-based (RDF) dataset retrieval. In *ACM SIGIR*. 2981–2991.

[56] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2021. TAPEX: Table Pre-training via Learning a Neural SQL Executor. In *International Conference on Learning Representations*.

[57] Tianyu Liu, Xin Zheng, Baobao Chang, and Zhifang Sui. 2021. Towards faithfulness in open domain table-to-text generation from an entity-centric view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13415–13423.

[58] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

[59] Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*. 139–151.

[60] lucene [n.d.]. Apache Lucene. https://lucene.apache.org.

[61] Pietro Marini, Aécio Santos, Nicole Contaxis, and Juliana Freire. 2025. Data Gatherer: LLM-Powered Dataset Reference Extraction from Scientific Literature. In *Proceedings of the Fifth Workshop on Scholarly Document Processing (SDP 2025)*. 114–123.

[62] MVP [n.d.]. MVP implementation. https://huggingface.co/docs/transformers/en/model_doc/mvp.

[63] Fatemeh Nargesian, Erkang Zhu, Renée J Miller, Ken Q Pu, and Patricia C Arocena. 2019. Data lake management: challenges and opportunities. *Proceedings of the VLDB Endowment (PVLDB)* 12, 12 (2019), 1986–1989.

[64] Felix Naumann. 2014. Data profiling revisited. *ACM SIGMOD Record* 42, 4 (2014), 40–49.

[65] Natasha Noy and Omar Benjelloun. 2020. An Analysis of Online Datasets Using Dataset Search. https://research.google/blog/an-analysis-of-online-datasets-using-dataset-search-published-in-part-as-a-dataset.

[66] NYC OpenData. 2024. https://opendata.cityofnewyork.us/.

[67] Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076* (2024).

[68] Andrea Papenmeier, Thomas Krämer, Tanja Friedrich, Daniel Hienert, and Dagmar Kern. 2021. Genuine information needs of social scientists looking for data. *Proceedings of the Association for Information Science and Technology* 58, 1 (2021), 292–302.

[69] pneuma [n.d.]. Pneuma implementation. https://github.com/TheDataStation/Pneuma.

[70] profiler [n.d.]. datamart-profiler. https://pypi.org/project/datamart-profiler. Accessed on Jan 2025.

[71] Ratish Puduppully, Li Dong, and Mirella Lapata. 2019. Data-to-text generation with content selection and planning. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 6908–6915.

[72] Ratish Puduppully and Mirella Lapata. 2021. Data-to-text generation with macro planning. *Transactions of the Association for Computational Linguistics* 9 (2021), 510–527.

[73] reastap [n.d.]. ReasTAP implementation. https://huggingface.co/docs/transformers/en/model_doc/mvp.

[74] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[75] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. 2021. Correlation Sketches for Approximate Join-Correlation Queries. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 1531–1544. https://doi.org/10.1145/3448016.3458456

[76] Kwangwook Seo, Jinyoung Yeo, and Dongha Lee. 2024. Unveiling Implicit Table Knowledge with Question-Then-Pinpoint Reasoner for Insightful Table Summarization. *arXiv preprint arXiv:2406.12269* (2024).

[77] socrata [n.d.]. Socrata. https://open-source.socrata.com.

[78] Hyunju Song, Steven Bethard, and Andrea Thomer. 2024. Metadata Enhancement Using Large Language Models. In *Proceedings of the Fourth Workshop on Scholarly Document Processing (SDP 2024)*, Tirthankar Ghosal, Amanpreet Singh, Anita Waard, Philipp Mayr, Aakanksha Naik, Orion Weller, Yoonjoo Lee, Shannon Shen, and Yanxia Qin (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 145–154. https://aclanthology.org/2024.sdp-1.14/

[79] Katrina Sostek, Daniel M. Russell, Nitesh Goyal, Tarfah Alrashed, Stella Dugall, and Natasha Noy. 2024. Discovering Datasets on the Web Scale: Challenges and Recommendations for Google Dataset Search. *Harvard Data Science Review* Special Issue 4 (apr 2 2024). https://hdsr.mitpress.mit.edu/pub/psnc8zsr.

[80] Yixuan Su, Zaiqiao Meng, Simon Baker, and Nigel Collier. 2021. Few-shot table-to-text generation with prototype memory. *arXiv preprint arXiv:2108.12516* (2021).

[81] Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of the International Conference on Management of Data (SIGMOD)*. 1493–1503.

[82] 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*. 645–654.

[83] Tianyi Tang, Junyi Li, Wayne Xin Zhao, and Ji-Rong Wen. 2023. MVP: Multi-task Supervised Pre-training for Natural Language Generation. In *The 61st Annual Meeting Of The Association For Computational Linguistics*.

[84] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971* (2023).

[85] Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. 2021. Tcn: Table convolutional network for web table interpretation. In *Proceedings of the Web Conference 2021*. 4020–4032.

[86] Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023. Is ChatGPT a Good NLG Evaluator? A Preliminary Study. In *Proceedings of the 4th New Frontiers in Summarization Workshop*, Yue Dong, Wen Xiao, Lu Wang, Fei Liu, and Giuseppe Carenini (Eds.). Association for Computational Linguistics, Singapore, 1–11. https://doi.org/10.18653/v1/2023.newsum-1.1

[87] Peng Wang, Junyang Lin, An Yang, Chang Zhou, Yichang Zhang, Jingren Zhou, and Hongxia Yang. 2021. Sketch and Refine: Towards Faithful and Informative Table-to-Text Generation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, Online, 4831–4843. https://doi.org/10.18653/v1/2021.findings-acl.427

[88] Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1780–1790.

[89] Yu Watanabe, Koichiro Ito, and Shigeki Matsubara. 2024. Capabilities and Challenges of LLMs in Metadata Extraction from Scholarly Papers. In *International Conference on Asian Digital Libraries*. Springer, 280–287.

[90] Mark D Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E Bourne, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific data* 3, 1 (2016), 1–9.

[91] Sam Wiseman, Stuart Shieber, and Alexander Rush. 2017. Challenges in Data-to-Document Generation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Martha Palmer, Rebecca Hwa, and Sebastian Riedel (Eds.). Association for Computational Linguistics, Copenhagen, Denmark, 2253–2263. https://doi.org/10.18653/v1/D17-1239

[92] Pengcheng Yin, Graham Neubig, Wen tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for Joint Understanding of Textual and Tabular Data. In *Annual Conference of the Association for Computational Linguistics (ACL)*.

[93] zenodo [n.d.]. Zenodo. https://zenodo.org.

[94] Dan Zhang, Madelon Hulsebos, Yoshihiko Suhara, Çağatay Demiralp, Jinfeng Li, and Wang-Chiew Tan. 2020. Sato: contextual semantic type detection in tables. *Proceedings of the VLDB Endowment (PVLDB)* 13, 12 (2020), 1835–1848.

[95] Haoxiang Zhang, Yurong Liu, Aécio Santos, Wei-Lun Hung, and Juliana Freire. 2025. AutoDDG: Automated Dataset Description Generation using Large Language Models. (2025). https://arxiv.org/abs/2502.01050

[96] Shuo Zhang and Krisztian Balog. 2018. Ad hoc table retrieval using semantic similarity. In *Proceedings of the 2018 world wide web conference*. 1553–1562.

[97] Shuo Zhang and Krisztian Balog. 2021. Semantic table retrieval using keyword and table queries. *ACM Transactions on the Web (TWEB)* 15, 3 (2021), 1–33.

[98] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. BERTScore: Evaluating Text Generation with BERT. In *International Conference on Learning Representations*.

[99] Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022. ReasTAP: Injecting Table Reasoning Skills During Pre-training via Synthetic Reasoning Examples. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 9006–9018.

[100] Yilun Zhao, Haowei Zhang, Shengyun Si, Linyong Nan, Xiangru Tang, and Arman Cohan. 2023. Investigating Table-to-Text Generation Capabilities of Large Language Models in Real-World Information Seeking Scenarios. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*. Association for Computational Linguistics, 160–175. https://doi.org/10.18653/v1/2023.emnlp-industry.17

[101] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. 2019. JOSIE: Overlap Set Similarity Search for Finding Joinable Tables in Data Lakes. In *In Proceedings of the ACM International Conference on Management of Data (SIGMOD)*. 847–864. https://doi.org/10.1145/3299869.3300065

Received July 2025; revised October 2025; accepted November 2025