Large Language Models for Data Discovery and Integration: Challenges and Opportunities

Juliana Freire {juliana.freire@nyu.edu}, Grace Fan, Benjamin Feuer,
Christos Koutras, Yurong Liu, Eduardo Pena, Aécio Santos, Cláudio Silva, Eden Wu
New York University, New York, USA

Abstract

The exponential growth of data across diverse sources—from the web and scientific repositories to enterprise systems—has amplified the need for effective data discovery and integration methods. While data integration has been extensively studied for decades, traditional approaches face limitations in generalization and scalability, often requiring a time-consuming and error-prone process that demands manual effort. Dataset discovery, though a more recent research direction, encounters similar challenges. A fundamental issue across both domains is semantic heterogeneity and ambiguity, as existing methods frequently struggle to understand the underlying meaning of data, limiting their effectiveness across diverse applications.

Large Language Models (LLMs) have emerged as a promising solution to enhance both data integration and discovery tasks. These models encode vast amounts of knowledge from extensive training data, enabling them to contextualize datasets—both their schema and contents—within real-world semantics. Notably, LLMs have demonstrated remarkable generalization capabilities, successfully performing various data tasks even without task-specific fine-tuning. This paper examines how LLMs are being applied across various integration and discovery tasks, synthesizing recent developments in this rapidly evolving field. We identify key limitations in current methods and suggest potential directions for future research, offering insights into both the potential and challenges of LLM-driven solutions.

1 Introduction

The increasing availability of datasets presents unprecedented opportunities for innovation across government, industry, and scientific research. By integrating data from multiple sources, it is possible to address complex and important questions that would otherwise be difficult or impossible to answer. For example, integrating genomic, transcriptomic, proteomic, and clinical data from multiple studies allows scientists to carry out pan-cancer analyses and identify common patterns, shared molecular mechanisms, and unique characteristics across cancer types [74]. Similarly, in climate science, combining satellite imagery with ground sensor data improves the accuracy of climate models and predictions [140]. Data integration unlocks new insights, driving advancements across many domains.

Despite efforts to make data FAIR–findable, accessible, interoperable and reusable [133]–discovering and integrating datasets remains a significant challenge [12, 49, 62, 96]. While datasets published on the Web are easily *accessible*, finding specific datasets can be difficult. They are often spread across different repositories that typically rely on simple keyword-based search interfaces, which are insufficient for users to express important information needs [10, 62, 105]. This challenge is further compounded by data heterogeneity: different datasets may use varying terminology for the same attribute or, conversely, the same term for distinct concepts. Such heterogeneity hinders the accuracy and recall of discovery queries, creating barriers to *interoperability* and *re-usability* and making it difficult to combine data from diverse sources [27].

Dataset Discovery and Integration. Dataset discovery involves systematically identifying and retrieving relevant datasets from data repositories or data lakes [8, 10, 36, 38, 42, 109]. Datasets can be discovered through text-based search, where information needs are expressed in keyword-based queries that are matched against metadata and contents [124, 126, 151], as well as dataset-oriented queries, where users provide a query table and search for relevant, related datasets [3, 25, 29, 30, 43, 51, 60, 90, 105, 160]. These search paradigms largely rely on schema similarity or content-based relevance, but incomplete metadata and semantic heterogeneity remain key challenges.

Data integration refers to the process of combining data from disparate sources into a unified, coherent dataset [27]. It encompasses several key tasks such as schema matching [102], which identifies semantically similar schema elements (e.g., attributes) across datasets, and entity resolution [19], which determines when different records refer to the same real-world entity. Data integration has been a long-standing research area, with a rich body of literature addressing its challenges [69]. Related concepts have emerged in specific domains. For instance, *data harmonization* is a common concept in digital healthcare [88] and social sciences [33] that refers to the process of combining datasets with the explicit goal of maximizing dataset compatibility and comparability [16]. Ultimately, data integration aims to create datasets with standardized formats and structures that enable meaningful comparisons and analysis while addressing challenges such as semantic heterogeneity, data quality, and structural differences across sources. Despite many advances in the area, data integration remains a time-consuming and error-prone task.

Dataset discovery and data integration share several fundamental challenges related to understanding tabular datasets, their schemas and attributes. For instance, to integrate two datasets, one must identify correspondences between attributes across the datasets. Similarly, in dataset-oriented discovery, these correspondences must be determined to enable effective retrieval of related datasets, e.g., that can be joined or concatenated with a query dataset [38, 42, 105].

LLMs for Data Discovery and Integration. State-of-the-art methods for both dataset discovery and integration struggle with semantic heterogeneity and ambiguity, as they often fail to *understand* the semantics of the underlying data, limiting their effectiveness. Recent advances in language models, particularly *Large Language Models* (LLMs), offer promising opportunities to address these challenges [44]. LLMs encode vast amounts of knowledge from extensive training data, allowing them to contextualize datasets—both their schema and contents—within real-world semantics. Moreover, these models have demonstrated the ability to generalize beyond their training data and perform data integration tasks [45, 59, 89] without task-specific fine-tuning. Unsurprisingly, the data management community has shown growing interest in leveraging LLMs for these tasks. In this paper, we survey recent advances, challenges, and opportunities in applying LLMs to dataset discovery and integration.

Contributions. While previous surveys and studies have examined the role of language models in data management [150, 155], they have largely focused on smaller models like BERT or overlooked the specific challenges of using LLMs for dataset discovery. In contrast, this paper explores the emerging role of LLMs, such as GPT and LLaMA, which primarily operate through prompt-based mechanisms, and how these models are reshaping dataset discovery and integration practices. We begin with an overview of language models in Section 2. We then review established techniques for dataset discovery and integration and examine how LLMs have been used to address key challenges in Sections 3 and 4. Finally, in Section 5, we discuss the limitations of existing LLM-based approaches, particularly challenges related to scalability and consistency, and outline potential research directions in both dataset discovery and integration, including opportunities to leverage their synergy to improve each of these tasks.

2 Large Language Models (LLMs): An Overview

A *Language Model* (LM) is trained to model the likelihood of future (autoregressive) or missing (masked or denoising) tokens in a sequence, conditional on all previous tokens in that sequence (the input, often called the context) [111]. These tokens are presumed to be drawn from some fixed vocabulary, typically consisting of common subwords and characters in particular languages. In this work, for simplicity's sake, we refer to all discriminative language models whose final output is an embedding as *Pretrained Language Models* (PLMs), and all generative language models whose final output is a string as *Large Language Models* (LLMs). In fact, LLMs lack a formal definition, as the threshold for what qualifies as large is subjective, and has evolved over time. The phrase was not, for instance, employed by [26] when they introduced their 400 Mn parameter language model, although the phrase "large pre-trained model" is. The authors in [5] use the phrase several times in reference to GPT-3, a 175 Bn parameter language model, and are likely responsible for the term's popularization.

One notable and often discussed property of LLMs that distinguishes them from PLMs is their ability to learn to solve novel problems "in-context" – either through few-shot learning, where they are provided examples with or without instructions, or zero-shot learning, where only instructions are provided [5, 101]. It has been observed that such abilities, which extend well beyond traditional language modeling to encompass, in principle, any task which can be expressed in language, tend to emerge as data and parameter count scale up [22, 104].

The current standard practice with LLMs is to train them in two stages. In the pretraining stage, raw strings are curated and filtered from a range of sources, tokenized, and served for training. This stage, which generally accounts for over 90% of training compute, is where the model learns world knowledge and internal algorithms it will rely on, such as copying [92]. In the post-training stage, the LLM learns to follow instructions and adopt human preferences [22, 94, 130]. It is fine-tuned on a variety of tasks presented as instructions [120], and typically aligned with human preferences using some form of human preference feedback [94]. Post-training has been observed to work better with larger models [22, 130], which in turn has increased the emphasis on scale. Another, more recent area of intensive research has been the scaling of test-time compute, exemplified by OpenAI's O1 and O3 models [157]. Prompting strategies such as chain-of-thought (CoT), which encourage the model to generate more tokens before answering, give the autoregressive model more "time to think" and can improve performance on complex arithmetic, commonsense, and symbolic reasoning tasks [131].

As LLMs can be applied to any task which can be expressed in natural language, attempting to use them to solve long-standing problems in processing tabular data is both logical and appealing. Recent approaches have used LLMs for data cleaning and integration [71, 89, 118, 138, 146], data profiling [45, 52], transforming tables [53, 91], reasoning and question answering over tables [142, 156], and more. However, using LLMs for table related tasks entails several challenges [28, 57, 82, 117]. Naive serialization of large tables as context rapidly explodes the number of tokens per query, increasing the cost and decreasing the efficacy of autoregressive models [73, 86]. Test-time scaling methods exacerbate this challenge, as they rely on the availability of long context, which they consume with their internal chains of thought [157]. Although many methods for subsampling long contexts for tabular data have been proposed, some of which we will discuss later, they tend to vary in efficacy depending on the particulars of the task and the data. The inherent heterogeneity of tabular data poses another kind of challenge: a table can contain anything, ranging from dense numerical attributes to sparse or high-cardinality categorical features, to natural language strings and embeddings [41]. Last but not least, metadata for tables in the wild (e.g., from open-data repositories) is often incomplete; the most reliable source are column headers, but even this can be absent in web-scraped tables [7]. This fact makes it more challenging to deploy solutions which rely on table metadata.

In the following sections, we discuss recent work on using LLMs on tabular data, specifically data discovery and integration tasks, and highlight limitations and open research challenges.

3 Large Language Models for Dataset Discovery

As more tabular datasets become available from academic institutions, private companies, and governments, there is greater opportunity for innovation and advancements across technology, society, and the economy [37]. However, dataset collections, made available in open repositories or closed data lakes, often contain a large number of datasets with varying sizes and complexity, making manual exploration and retrieval practically infeasible. As a result, there has been growing interest in the data management community to develop discovery systems that enable users to efficiently explore and retrieve datasets from large collections.

Query-driven dataset (table) discovery systems address this challenge by allowing users to query for relevant tables from a data lake. These systems support different types of queries, including (1) keyword-based queries (e.g., users specify keywords such as "*new york*"); (2) natural language queries (e.g., users can ask, "*What is the expected wait time of taxi cabs in NYC?*"); and (3) query tables (e.g., users have a table about NYC Taxis and would like to find other *related* datasets). Methods that support query tables consider different notions of table relatedness. For instance, some systems find tables that can join with a query table on shared attributes [43, 159, 160], while others find tables that can union with a query table to extend it with additional tuples [25, 60, 90]. Additionally, some methods support task-specific table discovery, such as finding tables that are joinable and correlated with the query table to augment it with additional features to improve the performance of machine learning models [105].

In this section, we provide an overview of approaches to table discovery, and survey recent approaches that leverage LLMs for different discovery tasks. Table 1 summarizes LLM-based approaches for dataset discovery.

3.1 Table Search

Table search (or table retrieval) is a data discovery method similar to traditional web search– it aims to find tables that satisfy the information needs described in a textual query. Text-based table retrieval systems initially focused on matching keywords in user queries against the dataset metadata [4, 23, 115] or the content of tables [151], similar to conventional search engines. However, as language models evolved and became more sophisticated, the task advanced to address more complex challenges, such as identifying tables that can answer questions posed in natural language [124, 126]. Regardless of the content of the queries, these methods usually aim to generate a ranked list of tables, denoted as $(T_1, ..., T_k)$, selected from a collection of tables C, in response to a textual query q. This task is often referred to as *ad-hoc table retrieval* since the relevance of each table T_i is determined independently of the other tables T_j (where $i \neq j$). Consequently, the ranking assigns scores to each table, which are then arranged in descending order based on these scores.

Overview of Related Work. Zhang and Balog [151] were one of the first to formalize the table search problem in recent literature and propose deep-learning methods to match keyword queries to table content. However, the problem had previously appeared in earlier work in the context of web tables [7, 8] (we refer the reader to [152] for a longer list of related work). Since then, more recent work has followed and proposed improvements, including algorithms based on PLMs. For example, Chen et al. [14] proposed to leverage a pre-trained BERT model to encode the table content. To workaround BERT's input size limit, they proposed and evaluated different ways to select content from the table that improves the overall ranking quality. Inspired by TaBERT [143], a pre-trained LM that jointly learns representations for natural language sentences and semi-structured tables, Trabelsi et al. [121]

introduced StruBERT. This new model, which was designed for table search and matching, combines textual and structural information of a table to produce context-aware representations for both textual and tabular content. They expand on the concept of vertical self-attention from TaBERT and introduce horizontal self-attention, allowing for equal treatment of both dimensions of a table. Graph-based models have also been proposed that capture table layout, including tables with nested structure [124].

Table Search using LLMs. A common approach to leverage LLMs in table search is to use them as generators of training data to build smaller and more efficient models. For instance, Fujita et al. [46] explored different strategies for generating labels (relevant/irrelevant) for a given table and query. Silva and Barbosa [113] introduced a method for generating synthetic queries based on dataset descriptions. The resulting pairs of queries and descriptions are regarded as soft matches when training fine-tuned dense retrieval models for re-ranking. Wang and Fernandez [126] used an LLM to generate synthetic training data to train a lightweight encoder model that generates embeddings that are used to efficiently retrieve tables that answer natural language questions. Specifically, their pipeline includes a fine-tuned T5 model to translate SQL queries into natural language questions. Another possibility is to use LLMs for query understanding. For instance, Chen et al. [13] used GPT-3.5 Turbo to decompose a natural language query into multiple sub-queries that can potentially be mapped to different tables and columns. They aimed to solve the problem of answering questions that require retrieving multiple tables and joining them through a join plan that cannot be easily discerned from the user query.

Dataset search systems and infrastructure that power data portals [4, 23, 115] treat datasets as documents and rely on metadata (i.e., dataset names and descriptions) to build an inverted index for keyword-based queries. Findability is thus dependent on the quality of dataset descriptions. For data in the wild, descriptions are often incomplete and sometimes inconsistent with the data contents. Zhang et al. [147] proposed a data-driven approach that uses LLMs to automatically generate dataset descriptions and showed that the derived descriptions lead to improved accuracy and recall for table retrieval.

3.2 Query-by-Tables

Semantic Joinable Table Search. Joinable table search aims to find tables that can be joined with a query table to augment it with additional attributes. This type of search is useful for data scientists who want to find new features to improve machine learning models, enrich data for analysis and support decision-making. There are different types of joinable table search, including equi-join, which finds exact matches between joinable columns; fuzzy join, which finds approximate column matches; and semantic join, which matches tables based on semantic relationships between columns. Traditional techniques for joinable table search have often relied on syntactic similarity measures, including Jaccard similarity and set overlap, to find potential joinable tables [43, 68, 159, 160]. More recently, there has been a shift towards methods that capture semantic relationships between columns by using embeddings and pre-trained language models [24, 29, 30, 61] to improve precision and recall.

To find semantic joinable tables, PEXESO [29] encodes columns into high-dimensional vectors using word embeddings such as fastText and GloVE. Joinable tables are then retrieved by comparing vector representations using similarity predicates. Similarly, DeepJoin [30] encodes columns as vectors, and uses column vector similarity to find joinable tables. Unlike PEXESO, DeepJoin uses a pre-trained language model (DistilBERT or MPNet) as the column encoder, which is trained in a self-supervised manner. This way, DeepJoin is able to consider table semantics. To support both equi-joins and semantic joins, the model is fine-tuned on labeled data specific to each joinability task. WarpGate [24] also performs semantic join discovery by leveraging pre-trained language models. WarpGate uses pre-trained web table embeddings [50] to capture the semantic relationships between tables. TabSketchFM [61] was introduced as a sketch-based tabular pre-training model that can be fine-tuned for different search tasks. TabSketchFM leverages data sketches to represent tabular data and combines embeddings of these sketches with column and token embeddings to create an input embedding for a BERT encoder model. While these approaches for semantic joinable search focus on table retrieval, DTT [91] addresses the challenge of joining values in semantically joinable columns. DTT leverages ByT5 and fine-tunes it to learn transformation rules to align and transform values for joins.

Table Union Search. In table union search, the objective is to discover tables that can be unioned (or concatenated) with a query table to extend it with additional tuples. This type of search is particularly valuable for data scientists who want to compile training or test data for machine learning models or expand the scope of their query tables to cover different geographical regions or time periods, among other use cases. Early work defined unionable tables as entity-complements that share the same subject column and similar schema [109]. More recently, Nargesian et al. [90] relaxed this assumption that unionable tables share the same schema as the query table. They formally defined table unionability based on attribute unionability, such that tables are considered unionable if they have attributes that originate from the same domain as the query table. Bogatu et al. [3] adopted this definition and used five similarity metrics to find unionable and joinable columns. This definition was further refined by Khatiwada et al. [60], who considered relationships between columns in addition to individual column unionability when finding unionable tables that share similar semantics as the query table.

Recent approaches for table union search [25, 38, 51] leverage pre-trained language models to capture column semantics more effectively. Starmie [38] finds unionable tables via self-supervised learning, namely contrastive learning, leveraging a pre-trained language model (RoBERTa) to capture table context when encoding column embeddings. To determine unionability, Starmie computes cosine similarity between column vectors and explores various column aggregation techniques to produce table unionability scores. Similarly, Pylon [25] employs self-supervised contrastive learning for table union search. Pylon explores different encoder models, including fastText, web table embeddings [50], and BERT, to generate column representations. In contrast, AutoTUS [51] shifts the focus to encoding the relationships between column pairs, rather than the columns themselves. By leveraging BERT, AutoTUS produces column relational representations that capture table contexts.

Query-by-Table Approaches using LLMs. Recent methods for joinable and unionable table search have largely been embedding based. While the use of large language models (LLMs) for these tasks has yet to be explored, it comes with its own set of challenges, which we discuss in Section 3.4. There are also many opportunities to take advantage of the power of LLMs to help perform these tasks, we discuss these in Section 5.

3.3 Other Goal-Oriented Dataset Discovery Tasks

Beyond the tasks described above, some discovery methods aim to satisfy more specific information needs. For instance, find tables that (1) satisfy specific distributional characteristics, such as percentile predicates [1], (2) are joinable and contain attributes correlated to columns in the input query table [35, 105, 106, 108, 122], (3) improve the performance of machine learning models [17, 55, 56, 78, 79], (4) uncover causal links among attributes [80, 144], (5) provide explanations for salient features in data [11, 18].

Overview of Related Work: PLMs and LLMs. Since the majority of work on goal-oriented tasks involves numerical data, the approaches used are typically based on traditional algorithmic techniques such as sketches and indexes. However, recent studies have shown that PLMs and LLMs can potentially be used in these tasks. Trummer [122] empirically demonstrates that PLMs can effectively predict correlations between table attributes using only their schemas in many cases. This shows that schemas are important for data profiling and allow language models to extract insights about the data such as correlations between columns. For the problem of causal dataset discovery, where the goal is to discover datasets containing columns with causal relationships to those in a query table, Liu et al. [80] leverage LLMs to infer causal relationships. They use LLMs to determine whether there are causal links between the correlated tables and to infer the direction of these links. They experiment with GPT-3.5 and GPT-4 and different prompting techniques such as Chain-of-Thought [131] as well as fine-tuning, and show that GPT-4 performs particularly well in identifying causal relationships and their directions.

3.4 Limitations and Research Gaps

Table Discovery methods face similar challenges when leveraging PLMs and LLMs. First, these models rely primarily on textual values in tables and often struggle to capture the semantics of numerical values. This poses a significant limitation for tasks like joinable table search, when join columns are numerical, or unionable table search, that often requires identifying matching numerical columns. Additionally, LLMs may struggle with the complexity of joins, especially transitive joins that involve many tables. Scalability is another major concern, particularly when searching over a large number of tables. Last but not least, LLMs' susceptibility to hallucination raises concerns about their reliability [137].

The effectiveness of techniques that rely on pre-trained models are heavily dependent on the size of the training data. At the same time, the limited context windows of PLMs and LLMs make it challenging to process large amounts of tabular data. Moreover, PLM techniques often lack generalizability. While methods like TabSketchFM [61] demonstrate task generalizability–for example, a model fine-tuned for joinable table search can also be applied to table union search–they often struggle to generalize to unseen data or entirely new domains. Finally, embedding-based approaches lack interpretability, making it difficult to explain retrieval results. Despite these challenges, embedding-based methods highlight the importance of encoding table semantics into column representations when discovering related tables in a large data lake. Building on this idea, we discuss potential opportunities to utilize LLMs for data discovery in Section 5.

4 Large Language Models for Data Integration

Data integration pipelines depend on identifying and connecting relevant elements across disparate datasets. Schema matching and entity resolution represent foundational techniques that identify semantically related elements and enable the creation of unified views from heterogeneous data sources.

This section first provides definitions of these core data integration problems, followed by an overview of traditional state-of-the-art methods. We then examine recent innovations using Large Language Models (LLMs) for data integration tasks, analyzing their potential advantages and limitations. Our discussion primarily focuses on tabular datasets, which are ubiquitous across enterprises, the scientific community, and the web. Table 2 summarizes the LLM-based approaches for data integration discussed in this paper, highlighting their distinctive characteristics.

4.1 Schema Matching

Schema matching refers to the process of identifying semantic relationships between elements in different schemas. For tabular datasets, this process involves finding column pairs from different tables that are semantically similar. The typical input for a schema matching method consists of two or more tables, while the output comprises of either pairwise column correspondences or clusters of semantically similar columns. The effectiveness of schema matching approaches depends on two key factors: the input information considered and the similarity metrics employed to identify related columns. In what

Table 1: Characteristics of LLM-based methods in the literature for Table Discovery.

Papers	Query Type		Task				Models Type			Inference	
	Textual	Table	Table Search	Joinable	Unionable	Others	Open	Closed	Fine-tuned	Zero-shot	Few-shot
[46]	✓		✓					✓		✓	
[113]	✓		✓				✓		✓		
[126]	✓		✓				✓		✓		
[13]	✓		✓			✓		✓			
[80]		✓				✓		✓	✓	✓	
[91]		✓		✓					✓		✓
[147]	✓		✓				✓	✓		✓	

Table 2: Characteristics of LLM-based methods in the literature for Schema Matching (SM) and Entity Resolution (ER).

Papers	Task		Input			Models Type			Inference	
	SM	ER	Metadata	Values	External Knowledge	Open	Closed	Fine-tuned	Zero-shot	Few-shot
[59]	(equi-joins)		✓	✓			✓		✓	
[53]	(PK-FK)		✓				✓		✓	
[97]	✓		✓				✓		✓	
[112]	✓		✓				✓		✓	
[110]	✓		✓		✓		✓		✓	
[83]	✓		✓		✓	✓	✓			✓
[81]	✓		✓	✓			✓			✓
[89]	✓	✓	✓	(ER only)			✓		✓	✓
[136]	✓	✓	✓	(ER only)	✓	✓	✓			✓
[71]	✓	✓	✓	✓			✓	✓	✓	✓
[146]	✓	✓	✓	(ER only)		✓	✓	✓	✓	✓
[138]	✓	✓	✓	(ER only)		✓		✓		✓
[99]		✓		✓		✓	✓	✓	✓	✓
[128]		✓		✓			✓		✓	
[54]		✓	✓	✓		✓				✓
[39]		✓	✓	✓		✓	✓			✓
[116]		✓		✓		✓	✓	✓	✓	
[31]		✓	✓	✓		✓		✓	✓	
[15]		✓	✓	✓			✓		✓	✓

follows, we review approaches proposed in the literature that demonstrate considerable diversity in both the similarity metrics utilized and the types of information leveraged.

Overview of Related Work. Early schema matching methods relied primarily on syntactic similarity measures between columns [102] and their value distribution [149], drawing mainly on information captured in column names and their corresponding column values. Some approaches expanded beyond syntax by incorporating external knowledge sources such as dictionaries and domain-specific thesauri [84] to capture semantic relationships.

The emergence of word and character embedding models, such as GloVe [100] and fastText [85], enabled the creation of semantically rich column representations that could be compared for similarity assessment [9, 110]. However, these embedding-based methods have shown inconsistent behavior and struggled with noisy data [65]. Specifically, traditional word embedding models do not properly handle formatting discrepancies, such as typos, and they attach the same meaning to each syntactically unique word, without accounting for context and polysemy (words with multiple meanings).

More recently, PLM-based methods, which produce contextualized representations of text, have been proposed for schema matching. Specifically, several methods have been proposed to fine-tune PLMs and perform schema matching [32, 123, 154]. The effectiveness of these methods is comparable to or even better than schema matching techniques that rely on syntactic measures or word embeddings. However, these PLM-based methods require large amounts of labeled column pairs. This labeled set is not always available, like in the case of tables in the wild, and thus requires methods to generate training examples with similar data distributions as the test data, such as contrastive learning. In addition, other methods have been proposed that leverage other (self-)supervised deep learning models (e.g., GNNs) to produce column representations for schema matching [66, 67, 148].

Schema Matching with LLMs. Several methods use LLMs with varied prompting approaches and information. Narayan et al. [89] employed zero-shot and few-shot LLM prompts to address data cleaning and integration tasks. Their prompts consist of serialized attribute/data values and, optionally, task demonstrations (selected randomly or manually) from a pool of labeled data. When using only column names in few-shot LLM prompts, the authors showed improvement over the deep-learning state-of-the-art schema matching method in [148]. Kayali et al. [59] designed prompts specifically for identifying join columns between two distinct *pandas DataFrames*, providing instructions with data samples and answer templates to guide the LLM in completing a *pandas.merge* operation. Parciak et al. [97] explored various prompting strategies for matching source attributes to a target schema. Huang et al. [53] developed a method that utilizes LLM prompts containing table descriptions (which are generated using LLMs) and schema information to capture PK-FK relationships (a special type of column match) between column pairs. In an effort to improve prompts for schema matching, Xu et al. [136] incorporated manual rules and additional insights extracted from either the input datasets or external knowledge bases.

Beyond using off-the-shelf LLMs, several researchers have explored fine-tuning LLMs specifically for data wrangling and integration tasks. Li et al. [71] proposed Table-GPT, which fine-tunes LLMs to address various table-related tasks, including schema matching. It incorporates fine-tuning for both complex reasoning tasks and simpler table manipulation operations to enhance the model's overall table processing capabilities. They evaluated various prompt and table serialization templates, demonstrating the importance of diverse augmentation techniques during fine-tuning. However, their experimental results for schema matching are inconclusive, as both out-of-the-box and fine-tuned LLMs demonstrate "perfect" effectiveness. This study leaves several questions unanswered, including how the number of tuples in table serialization affects performance and how the approach scales to large tables (in terms of both columns and rows).

Similarly, Zhang et al. [146] developed Jellyfish, which also employs LLM fine-tuning for data wrangling and integration tasks. Unlike Table-GPT, Jellyfish places special emphasis on prompt content and intended output, sometimes incorporating injected knowledge about input data and requiring answers that include reasoning information. Their evaluation shows that Jellyfish's fine-tuned models slightly outperform a previous schema matching technique [148] on a benchmark using only attribute names and descriptions (similar to the evaluation in [89]). Yan et al. [138] proposed a Mixture of Experts (MoE) approach based on LLMs for various data preprocessing tasks, including schema matching. Their results demonstrate improved effectiveness compared to both open-source and proprietary off-the-shelf models, while also showing better efficiency than Jellyfish's fine-tuned model [146], suggesting that MoE represents a promising approach.

It is worth noting that several studies [89, 97, 136, 138, 146] evaluate schema matching using datasets that only include table/attribute names and descriptions. However, there has not been a comprehensive evaluation to compare all these approaches.

Instead of using LLMs on input data to directly capture matches, a number of methods incorporate them as part of more complex schema matching pipelines. ReMatch [112] introduces a method that refines candidates based on embeddings before prompting the LLM. Specifically, the authors focus on the problem of matching a set of source tables to a set of target tables when table and attribute descriptions and names are given. Their method first transforms each table to a document that includes table/column names and descriptions; using GPT-4, each such document is transformed into an embedding. Then, for each source attribute they leverage embeddings to find the most relevant documents, i.e., tables, from which they retrieve the final matching target attributes based on the LLM response. Matchmaker [110] targets the same variant of schema matching problem and proposes a multi-stage method with several LLM-calls to suggest and refine candidate matches between source and target schemata, while also assigning confidence scores. In contrast to ReMatch, Matchmaker uses both LLM-calls and PLM embeddings to filter out candidate target attributes, while it also employs LLMs in a different way to refine and finally match them to the query source attribute: matching is formulated as a multiple choice question, where the task is to find the most relevant target column with respect to a given source column. Both methods show improvements over a state-of-the-art deep-learning schema matching method [148], while Matchmaker shows effectiveness gains over ReMatch and Jellyfish [146]. Instead of relying only on the information in the input datasets, Ma et al. [83] proposed to leverage knowledge graphs. Their method retrieves relevant knowledge graph triplets and uses them to augment LLM prompts for answering whether a source attribute corresponds to a target one, on top of providing their names, descriptions and demonstration examples; retrieval can be either LLM-based or employ vector search over PLM-based embeddings.

Magneto [81] introduces a new approach for schema matching that combines small-PLMs (SLMs) and LLMs in a novel way. Like ReMatch, it works in two steps. First, it leverages a pre-trained or fine-tuned SLM to produce embeddings of columns and given a source column, outputs, a ranked list of similar target attributes. Note that instead of relying on manually-labeled data, Magneto uses LLMs to generate training data for fine-tuning the SLM. For the second step, the ranked list of matches is given to an LLM for re-ranking. To deal with the context-window limitations of both PLMs and LLMs, they explored different methods for sampling and serializing tables (including values). Experimental results show that using LLMs for re-ranking can be effective, regardless of the SLM used to derive the initial ranking, and that Magneto outperforms or performs comparably to state-of-the-art methods, including [123] and [32].

Limitations and Research Gaps. Despite the progress in applying LLMs to schema matching, several critical limitations remain unaddressed. First, existing work largely ignore the challenges of processing large input sequences when tables contain a large number columns and rows. The tendency to disregard data instances in favor of relying solely on column names and descriptions restricts these methods' applicability in many real-world scenarios.

Interestingly, current LLM-based approaches predominantly focus on matching source data to standardized target schemas, with matching datasets to OMOP CDM (Observational Medical Outcomes Partnership Common Data Model) [93] emerging as the most common evaluation scenario. Since these benchmarks contain table/attribute names and descriptions, research has gravitated toward metadata-based schema matching, resulting in a scarcity of LLM-based methods that effectively utilize actual data instances.

Few studies include comparisons against other LM-based methods, impeding meaningful progress in the field. Instead, most methods report improvements against pre-PLM schema matching techniques or out-of-the-box models. Notably absent are detailed analyses of specific column match cases where LLM-based methods significantly outperform previous state-of-the-art approaches. The absence of granular performance analysis makes it difficult to identify genuine advancements and understand the specific strengths of LLM-based approaches. Furthermore, evaluation results from multi-tasking models like TableGPT [71] and Jellyfish [146] successfully demonstrate general applicability but fail to provide task-specific insights crucial for understanding schema matching performance.

While few-shot inference with LLMs can achieve state-of-the-art effectiveness, this approach requires carefully selected demonstration examples from labeled data pools. This presents a significant challenge, as ground truth data for schema matching tasks is notoriously limited. Even when labeled data exists, it may not accurately reflect the intrinsic characteristics of test data, potentially leading to suboptimal performance in real-world applications.

4.2 Entity Resolution

Entity Resolution involves identifying the same entities across different datasets. In tabular data, the objective is to determine which tuples refer to the same entity. Entity resolution pipelines typically consist of several distinct tasks [19]. Initially, input tables may undergo an optional pre-processing step that includes integration and curation tasks such as schema matching (as discussed in Section 4.1), data cleaning [21], and other preparatory operations. Following this, the blocking phase examines all tuples across datasets and produces a set of candidate pairs that might be semantically similar; essentially, blocking serves as the filtering step within an entity resolution pipeline. Finally, the matching phase evaluates each candidate pair and determines whether it constitutes a valid match. In the following sections, we discuss proposed methods that focus on either the blocking phase exclusively [95], the matching phase exclusively [76], or address both phases [20]. We then examine approaches that leverage LLMs for entity resolution tasks.

Overview of Related Work. The majority of entity resolution methods target the matching step. Specifically, several approaches rely on human experts to devise rules [40, 114], guide the entity matching process through crowd-sourcing [48, 125], or provide labeled data for training machine learning models [63]. Moreover, automated methods using supervised learning with deep learning models [34, 87] exhibit considerable effectiveness gains.

More recently, various methods have leveraged the representational capabilities of PLMs to build fine-tuned models that capture entity matches in a supervised manner [6, 75, 77, 98, 123]. While these approaches show performance improvements, they continue to require a substantial amount of labeled data. To address this limitation, alternative research directions have explored self-supervised [47, 127] and unsupervised [134, 145] frameworks. These approaches have shown comparable or superior effectiveness relative to supervised models, helping overcome the labeled data bottleneck that often constrains learning-based entity resolution methods.

To address the blocking phase of entity resolution, methods have been proposed to employ various filtering techniques to identify potential matches across datasets, including rule-based filtering, hash-based comparisons, sorted key comparisons, similarity functions, and ensemble methods combining multiple approaches [95]. Deep learning solutions have also emerged in the blocking space [34, 119], which project tuples into embedding spaces and leverage these representations to construct clusters of potentially matching entities.

Moving beyond the traditional separation of blocking and matching, some researchers have proposed integrated approaches [70, 135]. Such methods demonstrate the benefits of addressing both tasks simultaneously, allowing signals from each phase to inform and improve the other's performance. This integrated perspective represents a promising direction for enhancing overall entity resolution effectiveness.

Entity Resolution with LLMs. Recent research has explored using LLMs out-of-the-box with dedicated prompts for direct entity matching decisions. Narayan et al. [89] demonstrated that an LLM prompt that serializes column names and values for tuple pairs alongside demonstration examples can sometimes outperform state-of-the-art solutions like Ditto [75, 77]. Peeters et al. [99] conducted a comprehensive experimental study on LLM-based tuple-pair matching. Their investigation encompassed multiple LLMs using both zero-shot and few-shot prompting approaches, while also exploring fine-tuning with labeled data from existing annotated dataset pairs. Their findings revealed that no single model or prompting technique consistently outperforms others, with traditional non-LLM supervised methods sometimes achieving superior results. Wang et al. [128] introduced a novel approach that considers matching of a single tuple against a set of tuples, rather than focusing on tuple-pair matching. This problem formulation enabled their method, ComEM, to explore diverse input-output prompting strategies and effectively combine them for enhanced performance. When compared against state-of-the-art approaches—including supervised methods (e.g., Ditto [77]), self-supervised techniques (e.g., Sudowoodo [127]), and out-of-the-box LLM prompts [99], ComEM demonstrated comparable or occasionally superior effectiveness. Dou et al. [31] explored the advantages of jointly training blocking and matching components through supervised learning. Specifically, they introduced an innovative architecture supports fine-tuning either traditional PLMs or instruction-tuned open-source LLMs for the matching phase. Their experimental results reveal that while instruction-tuned LLMs demonstrate superior effectiveness compared to PLMs, they still fall short of the performance achieved by proprietary models like GPT-4 when used out-of-the-box. Xu et al. [136] presented a method that claims applicability to entity matching tasks through the use of manual instructions and external knowledge integration. However, their evaluation focuses exclusively on schema matching, leaving the method's effectiveness for entity resolution tasks unverified.

Recent work has explored multiple strategies to optimize LLM-based entity matching, ranging from model architecture modifications to prompt engineering and computational efficiency improvements. Table-GPT [71] demonstrates that LLMs specifically tuned on tabular data outperform their untuned counterparts in both zero-shot and few-shot settings for tuple pair matching. Building on this foundation, Jellyfish [146] achieves superior entity matching results compared to both Table-GPT and traditional non-LLM methods. However, Jellyfish's performance advantage is most pronounced when the test data distribution is represented in the fine-tuning dataset, and comparisons against state-of-the-art LLMs like GPT-4 have yielded inconclusive results. A systematic study by Steiner et al. [116] further explores fine-tuning approaches across both open-source and proprietary LLMs, analyzing various data representation formats and example selection strategies. They also evaluated model robustness against domain shifts between training and test data. While their fine-tuned models demonstrate enhanced performance for in-domain generalization, an interesting finding reveals that zero-shot approaches actually achieve superior results when handling test data that comes from a domain different from the training data. This highlights an important trade-off between fine-tuning benefits and domain adaptability.

Alternative architectural approaches have also shown promise. The Mixture-of-Experts (MoE) LLM-based model proposed in [138] brings improvements in entity matching over standard open/closed-source and fine-tuned models. It demonstrates superior performance compared to both conventional and fine-tuned LLM models, highlighting the advantages of specialized task experts within a unified framework. Taking a different approach, Seed [15] introduces a flexible system that leverages LLMs in two ways: either for code and training data generation, or for direct entity matching tasks. A key innovation of Seed is its optimization framework, which generates and selects execution plans based on both computational efficiency and effectiveness metrics. While Seed does not match the accuracy levels of state-of-the-art supervised methods like Ditto [77] or pure LLM approaches [99], it achieves a balance between effectiveness and LLM-cost efficiency.

Huh et al. [54] investigated demonstration example selection by leveraging pre-trained language models (PLMs) to embed tuple pairs and identify similar examples in the representation space. Interestingly, their findings reveal that this sophisticated approach does not consistently outperform simpler random or manual example selection methods. Addressing computational efficiency, BatchER [39] demonstrates that batch processing of tuple pairs within single prompts can achieve both cost savings and effectiveness gains. BatchER achieves comparable results to state-of-the-art PLM-based entity matching methods that require extensive labeled datasets, while maintaining lower computational overhead.

Limitations and Research Gaps. While LLMs show considerable promise for entity matching tasks, there are also important challenges. First, prompt engineering significantly impacts matching accuracy, with different model architectures exhibiting varying levels of prompt sensitivity. Peeters et al. [99] demonstrated that while GPT-4 maintains consistent performance across different prompt formulations, other models like GPT-mini, Llama2, Llama3.1, and Mixtral show higher sensitivity to prompt variations. This finding highlights the critical importance of careful prompt design, particularly when using more cost-effective models.

The evaluation landscape presents a mixed picture. Unlike schema matching, entity matching research benefits from common benchmark datasets, enabling direct comparisons between LLM-based approaches and traditional methods. However, inconsistent result presentation and reporting practices often lead to inconclusive comparisons. This challenge is particularly evident when trying to determine the precise advantages of LLM-based methods over existing approaches. A significant gap in current research is the lack of detailed analysis of specific use cases or scenarios where LLM-based methods demonstrate clear advantages over traditional approaches. While studies often report aggregate performance metrics, they frequently fail to identify and analyze the particular types of entity matching problems where LLMs excel or struggle compared to conventional methods. This limitation makes it difficult for practitioners to make informed decisions about when to adopt LLM-based solutions.

A significant challenge in deploying LLMs for entity matching is managing computational costs, particularly with hosted commercial models. The cost structure, based on token count for both input prompts and model outputs, becomes especially problematic when handling tuples with extensive textual attributes that require longer prompts. While advanced models like GPT-4 provide superior performance, their higher per-token costs create a substantial economic barrier compared to smaller models like GPT-mini. The cost-quality trade-off becomes more complex when considering performance optimization strategies. In-context learning can significantly improve matching accuracy through demonstration examples, but each additional example or rule increases the token count and consequently the operational costs [39]. Similarly, while fine-tuning offers a promising path to improve performance of smaller open-source models, its benefits are often limited to scenarios where test data closely aligns with the training domain [99, 116]. Furthermore, the fine-tuning process itself presents two significant barriers: the high computational overhead and the requirement for substantial labeled training data—a limitation shared with traditional PLM-based methods [77].

5 Research Challenges and Future Directions

Despite the rapid advancement in applying LLMs to data discovery and integration tasks, significant research challenges and opportunities remain unexplored. Building on our analysis of existing methods (Sections 3 and 4), we identify and discuss both task-specific challenges and broader research directions that warrant further investigation. Our discussion encompasses not only technical limitations identified in current approaches but also emerging opportunities to enhance the effectiveness and practical applicability of LLM-based solutions in this domain.

5.1 Dataset Discovery

Although the direct application of LLMs to large-scale tasks like semantic joinable table search and table union search faces *scalability* challenges, LLMs' sophisticated semantic understanding capabilities present promising opportunities for advancing dataset discovery tasks.

Improved Semantic Understanding and Metadata Enrichment. While existing PLM-based methods [24, 25, 30, 38, 51, 61] have demonstrated the value of semantic approaches in finding related tables, LLMs could potentially enable more precise semantic matching and relationship identification. LLMs have shown success in fundamental table understanding tasks, including column-type annotation [45, 59, 64, 71, 132, 146, 153], table-class detection [52, 59, 64], and column-relation extraction [59, 153]. These capabilities can be leveraged to enhance tables with rich semantic information, thereby improving joinable and unionable table search accuracy.

Cross-Task Integration. Advances in LLM-based entity matching (Section 4.2) could be extended to multi-table entity matching and attribute discovery. Similarly, schema matching techniques (Section 4.1) could be adapted to identify semantically similar columns for unionable table search. This cross-pollination of techniques offers promising paths for improvement. Research is needed on how to effectively combine different LLM-based tasks (e.g., type annotation, relation extraction, and matching) in a unified discovery pipeline.

Hybrid Approaches. Another direction of research is to further explore hybrid approaches that combine LLMs with traditional techniques. Such approaches could leverage LLMs' semantic understanding while maintaining the efficiency of established methods. Particularly promising is the integration of LLMs with existing join discovery techniques like sketches and indexes [30, 43, 106, 160], which could enhance emerging join-aware textual query methods that retrieve multiple tables [2, 13].

Explainable Discovery. Lastly, LLMs' ability to generate natural language explanations for semantic matches and relationships between columns and tables represents an underexplored opportunity. Such explanations could significantly improve user trust and understanding of discovery results, facilitating more effective use in downstream tasks. Research is needed on how to generate explanations that are both technically accurate and accessible to users with varying levels of expertise.

5.2 Schema Matching

Handling Instance Data. Current LLM-based schema matching methods predominantly focus on scenarios where only metadata is available for matching source datasets to target schemas (Section 4.1). However, schema matching must also handle more challenging scenarios where column names may be opaque or missing, making instance data the primary source of matching information. This disconnect presents several key challenges to the development of LLM-based schema matching approaches. A fundamental challenge is managing large input sequences when tables contain numerous columns and rows. While recent LLM architectures offer expanded context windows exceeding 100,000 tokens, two critical issues emerge: First, studies indicate that LLMs face accuracy degradation with longer contexts [72] and efficiency challenges [139]. Second, even with larger context windows, the computational costs and performance trade-offs must be carefully balanced. This points to promising research directions. Rather than using LLMs for direct schema matching, approaches like [81] demonstrate the potential of leveraging LLMs as sophisticated reasoning engines in post-processing steps. Future research should explore approaches to efficiently sample and summarize instance data to create informative yet compact LLM inputs, develop adaptive strategies that selectively invoke LLM processing based on data characteristics, design cost-effective architectures that maintain accuracy while managing computational overhead.

Profiling for Schema Matching. LLMs present significant opportunities for enhancing schema matching through automated data profiling and metadata enrichment, particularly in scenarios where metadata is missing or unreliable. Recent works demonstrate promising applications: generating detailed table and column descriptions [52, 147], and inferring semantic types for columns [45]. The information obtained through profiling can be used to identify matching columns. In addition to improving the effectiveness of schema matching methods, the generated metadata an help prune the search space of potential column matches, contributing to lower computation cost and execution time.

5.3 Entity Resolution

Handling Long Entities. While LLM-based entity matching solutions benefit from significantly longer context windows compared to PLM-based methods, incorporating all attribute values into entity representations may not always improve matching accuracy. The impact of long attribute values on LLM reasoning can vary significantly - for instance, extensive user comments might either obscure or enhance product matching depending on their content. This complexity necessitates sophisticated pre-processing strategies for optimizing entity representations in LLM prompts. Several promising research directions emerge for effective pre-processing, including: the development of profiling techniques to identify and prioritize informative attributes; integration of schema matching techniques to focus on comparable attributes, following successful approaches from traditional entity matching methods [20]; and techniques for condensing long attribute values while preserving matching-relevant information [77]

Blocking. While LLMs have shown promise in entity matching (Section 4.2), their application to blocking—a crucial step in real-world entity resolution [95]—remains largely unexplored. This gap stems from the inherent complexity of blocking tasks: unlike pair-wise entity matching, blocking must efficiently process the entire search space of possible tuple pairs across datasets to identify candidate matches. The computational demands and complexity of this task make direct application of LLMs impractical, even with their expanding context windows and enhanced reasoning capabilities. However, LLMs can contribute to blocking effectiveness through indirect approaches. For example, through metadata enhancement – enriching tuples with additional semantic information, generating standardized representations of attribute values. Recent work [129] has demonstrated the potential of LLM-based tuple enrichment for improving blocking effectiveness

5.4 General Directions

Prompt Engineering. The effectiveness of LLMs in data integration and discovery tasks heavily depends on prompt engineering, but there are significant challenges in developing robust prompting strategies. Experimental evaluations of methods discussed in Sections 3 and 4 point out that there are no universally effective prompting templates and strategies. Even minor changes in table serialization formats or task descriptions can significantly impact performance [71, 99]. Moreover, the rapid evolution of prompting strategies [103] makes it difficult to establish best practices.

This complexity points to several critical research directions in automated prompt optimization and the integration of context and knowledge. Inspired by recent advances in automatic prompt engineering efforts [158], automated prompt optimization methods could be tailored to each given input and table-related tasks. Moreover, choosing suitable in-context examples [54] and effective retrievers for incorporating external knowledge to prompts [58] is equally important. Finally, recent advanced prompting techniques that enable tool calling abilities [141] are an exciting direction that allows the development of agentic systems that integrate existing efficient algorithms with LLM-based reasoning and interactive user interfaces [107].

Cost Considerations. Evaluating LLMs' feasibility in practical scenarios demands rigorous analysis of cost and runtime factors, which present significant deployment barriers. The current landscape presents a complex trade-off between hosted and open-source solutions, each with distinct challenges. Hosted models pose several significant challenges for practical deployment. Leading providers' high API costs per token [1][2][3] can accumulate rapidly, becoming prohibitive for large-scale deployments – involving large datasets or a large number of datasets. Response latency issues become particularly acute when processing large datasets, potentially impacting real-time applications.

Open-source models present a different set of challenges. Models like LLaMA2 require a significant upfront investment in specialized hardware infrastructure, particularly high-performance GPUs. Beyond the initial investment, organizations must consider ongoing operational and maintenance costs, as well as the technical expertise required for effective deployment and optimization. These factors can make the total cost of ownership substantial, even without per-token API fees.

This dichotomy highlights a critical research gap: the absence of standardized frameworks for comparing hosted and open-source models. Such frameworks would need to address multiple dimensions including effectiveness-cost ratios, scalability characteristics, total cost of ownership, and operational complexity. The development of comprehensive comparison methodologies would enable organizations to make more informed deployment decisions based on their specific needs and constraints.

Comprehensive Task-Specific Evaluations. Recent LLM-based approaches that target multiple tabular data curation and integration tasks [71, 138, 146] have established their effectiveness through extensive experimental evaluations that typically assess each task using dedicated benchmarks and compare against state-of-the-art methods using standard metrics like F1-score. Additionally, they often include task-agnostic ablation studies examining the impact of different prompting strategies, demonstration examples, and LLM architectures.

While these evaluations demonstrate the general applicability and advantages of LLM-based methods, they often lack granular, task-specific insights that could better illuminate their unique strengths. Current evaluation approaches rarely analyze specific challenging instances where LLMs demonstrate particular advantages. For example, cases involving columns or entities with significant syntactic differences but semantic similarities could provide valuable insights into LLMs' semantic understanding capabilities. Such fine-grained analysis could reveal where LLMs excel compared to traditional approaches. Future evaluations should therefore incorporate detailed analysis of specific challenging cases that highlight LLMs' distinctive capabilities. This could include examining performance on instances requiring complex semantic reasoning, handling of ambiguous or context-dependent matches, and cases where traditional methods typically struggle. By providing concrete examples of where LLMs bring meaningful improvements, such task-specific evaluations would offer stronger justification for their adoption and better guide their application in practice.

Acknowledgments. This work was supported by NSF awards IIS-2106888 and OAC-2411221, and the

[1] https://openai.com/api/pricing/

[2] https://www.anthropic.com/pricing#anthropic-api

[3] https://ai.google.dev/pricing#1_5flash

DARPA Automating Scientific Knowledge Extraction and Modeling (ASKEM) program, Agreement No. HR0011262087. 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, the U.S. Government, or NSF.

References

[1] Lennart Behme, Sainyam Galhotra, Kaustubh Beedkar, and Volker Markl. Fainder: A fast and accurate index for distribution-aware dataset search. *Proceedings of the VLDB Endowment*, 17 (11):3269–3282, 2024.

[2] Jan-Micha Bodensohn and Carsten Binnig. Rethinking table retrieval from data lakes. In *Proceedings of the Seventh International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pages 1–5, 2024.

[3] Alex Bogatu, Alvaro A. A. Fernandes, Norman W. Paton, and Nikolaos Konstantinou. Dataset discovery in data lakes. In *ICDE*, pages 709–720, 2020.

[4] Dan Brickley, Matthew Burgess, and Natasha Noy. Google dataset search: Building a search engine for datasets in an open web ecosystem. In *The World Wide Web Conference*, pages 1365–1375, 2019.

[5] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

[6] Ursin Brunner and Kurt Stockinger. Entity matching with transformer architectures-a step forward in data integration. In *23rd International Conference on Extending Database Technology, Copenhagen, 30 March-2 April 2020*, pages 463–473. OpenProceedings, 2020.

[7] Michael J Cafarella, Alon Halevy, Daisy Zhe Wang, Eugene Wu, and Yang Zhang. Webtables: exploring the power of tables on the web. *Proceedings of the VLDB Endowment*, 1(1):538–549, 2008.

[8] Michael J Cafarella, Alon Halevy, and Nodira Khoussainova. Data integration for the relational web. *Proceedings of the VLDB Endowment*, 2(1):1090–1101, 2009.

[9] Riccardo Cappuzzo, Paolo Papotti, and Saravanan Thirumuruganathan. Creating embeddings of heterogeneous relational datasets for data integration tasks. In *SIGMOD*, pages 1335–1349, 2020.

[10] Sonia Castelo, Rémi Rampin, Aécio Santos, Aline Bessa, Fernando Chirigati, and Juliana Freire. Auctus: A dataset search engine for data discovery and augmentation. *Proceedings of the VLDB Endowment*, 14(12):2791–2794, 2021.

[11] Yeuk-Yin Chan, Fernando Chirigati, Harish Doraiswamy, Cláudio T. Silva, and Juliana Freire. Querying and exploring polygamous relationships in urban spatio-temporal data sets. In *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, pages 1643–1646. ACM, 2017. doi: 10.1145/3035918.3058741. URL `https://doi.org/10.1145/3035918.3058741`.

[12] Adriane Chapman, Elena Simperl, Laura Koesten, George Konstantinidis, Luis-Daniel Ibáñez, Emilia Kacprzak, and Paul Groth. Dataset search: a survey. *The VLDB Journal*, 29(1):251–272, 2020.

[13] Peter Baile Chen, Yi Zhang, and Dan Roth. Is table retrieval a solved problem? exploring join-aware multi-table retrieval. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2687–2699, 2024.

[14] Zhiyu Chen, Mohamed Trabelsi, Jeff Heflin, Yinan Xu, and Brian D Davison. Table search using a deep contextualized language model. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*, pages 589–598, 2020.

[15] Zui Chen, Lei Cao, Sam Madden, Tim Kraska, Zeyuan Shang, Ju Fan, Nan Tang, Zihui Gu, Chunwei Liu, and Michael Cafarella. Seed: Domain-specific data curation with large language models. *arXiv e-prints*, pages arXiv–2310, 2023.

[16] Cindy Cheng, Luca Messerschmidt, Isaac Bravo, Marco Waldbauer, Rohan Bhavikatti, Caress Schenk, Vanja Grujic, Tim Model, Robert Kubinec, and Joan Barceló. A general primer for data harmonization. *Scientific data*, 11(1):152, 2024.

[17] Nadiia Chepurko, Ryan Marcus, Emanuel Zgraggen, Raul Castro Fernandez, Tim Kraska, and David R. Karger. ARDA: automatic relational data augmentation for machine learning. *Proc. VLDB Endow.*, 13(9):1373–1387, 2020.

[18] Fernando Chirigati, Harish Doraiswamy, Theodoros Damoulas, and Juliana Freire. Data polygamy: the many-many relationships among urban spatio-temporal data sets. In *ACM SIGMOD*, pages 1011–1025, 2016.

[19] Peter Christen. *Data Matching - Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection.* Data-Centric Systems and Applications. Springer, 2012. ISBN 978-3-642-31163-5. doi: 10.1007/978-3-642-31164-2. URL `https://doi.org/10.1007/978-3-642-31164-2`.

[20] Vassilis Christophides, Vasilis Efthymiou, Themis Palpanas, George Papadakis, and Kostas Stefanidis. An Overview of End-to-End Entity Resolution for Big Data. *ACM Comput. Surv.*, 53 (6):127:1–127:42, December 2020. ISSN 0360-0300. doi: 10.1145/3418896.

[21] Xu Chu, Ihab F Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 international conference on management of data*, pages 2201–2206, 2016.

[22] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022.

[23] CKAN. `https://ckan.org`, 2024.

[24] Tianji Cong, James Gale, Jason Frantz, H. V. Jagadish, and Çagatay Demiralp. Warpgate: A semantic join discovery system for cloud data warehouses. In *CIDR*, 2023.

[25] Tianji Cong, Fatemeh Nargesian, and HV Jagadish. Pylon: Semantic table union search in data lakes. *arXiv preprint arXiv:2301.04901*, 2023.

[26] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1423. URL `https://doi.org/10.18653/v1/n19-1423`.

[27] AnHai Doan, Alon Halevy, and Zachary Ives. *Principles of Data Integration.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition, 2012. ISBN 0124160441.

[28] Haoyu Dong and Zhiruo Wang. Large language models for tabular data: Progresses and future directions. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2997–3000, 2024.

[29] Yuyang Dong, Kunihiro Takeoka, Chuan Xiao, and Masafumi Oyamada. Efficient joinable table discovery in data lakes: A high-dimensional similarity-based approach. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 456–467. IEEE, 2021.

[30] Yuyang Dong, Chuan Xiao, Takuma Nozawa, Masafumi Enomoto, and Masafumi Oyamada. Deepjoin: Joinable table discovery with pre-trained language models. *Proceedings of the VLDB Endowment*, 16(10):2458–2470, 2023.

[31] Wenzhou Dou, Derong Shen, Xiangmin Zhou, Hui Bai, Yue Kou, Tiezheng Nie, Hang Cui, and Ge Yu. Enhancing deep entity resolution with integrated blocker-matcher training: Balancing consensus and discrepancy. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 508–518, 2024.

[32] Xingyu Du, Gongsheng Yuan, Sai Wu, Gang Chen, and Peng Lu. In situ neural relational schema matcher. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 138–150. IEEE, 2024.

[33] Joshua Kjerulf Dubrow and Irina Tomescu-Dubrow. The rise of cross-national survey data harmonization in the social sciences: emergence of an interdisciplinary methodological field. *Quality & Quantity*, 50:1449–1467, 2016.

[34] Muhammad Ebraheem, Saravanan Thirumuruganathan, Shafiq R. Joty, Mourad Ouzzani, and Nan Tang. Distributed representations of tuples for entity resolution. *Proc. VLDB Endow.*, 11 (11):1454–1467, 2018. doi: 10.14778/3236187.3236198. URL `http://www.vldb.org/pvldb/vol11/p1454-ebraheem.pdf`.

[35] Mahdi Esmailoghli, Jorge-Arnulfo Quiané-Ruiz, and Ziawasch Abedjan. *Cocoa: Correlation coefficient-aware data augmentation.* Konstanz, Germany: OpenProceedings. org, University of Konstanz, University . . . , 2021.

[36] Mahdi Esmailoghli, Christoph Schnell, Renée J Miller, and Ziawasch Abedjan. Blend: A unified data discovery system. *arXiv preprint arXiv:2310.02656*, 2023.

[37] Grace Fan, Jin Wang, Yuliang Li, and Renée J. Miller. Table discovery in data lakes: State-of-the-art and future directions. In *SIGMOD Conference Companion*, pages 69–75. ACM, 2023.

[38] Grace Fan, Jin Wang, Yuliang Li, Dan Zhang, and Renée J. Miller. Semantics-aware dataset discovery from data lakes with contextualized column-based representation learning. *Proc. VLDB Endow.*, 16(7):1726–1739, 2023.

[39] Meihao Fan, Xiaoyue Han, Ju Fan, Chengliang Chai, Nan Tang, Guoliang Li, and Xiaoyong Du. Cost-effective in-context learning for entity resolution: A design space exploration. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 3696–3709. IEEE, 2024.

[40] Wenfei Fan, Xibei Jia, Jianzhong Li, and Shuai Ma. Reasoning about record matching rules. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):407–418, 2009.

[41] Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. Large language models(llms) on tabular data: Prediction, generation, and understanding – a survey, 2024. URL https://arxiv.org/abs/2402.17944.

[42] Raul Castro Fernandez, Ziawasch Abedjan, Famien Koko, Gina Yuan, Samuel Madden, and Michael Stonebraker. Aurum: A data discovery system. In *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pages 1001–1012. IEEE, 2018.

[43] Raul Castro Fernandez, Jisoo Min, Demitri Nava, and Samuel Madden. Lazo: A cardinality-based method for coupled estimation of jaccard similarity and containment. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1190–1201. IEEE, 2019.

[44] Raul Castro Fernandez, Aaron J. Elmore, Michael J. Franklin, Sanjay Krishnan, and Chenhao Tan. How large language models will disrupt data management. *Proc. VLDB Endow.*, 16(11): 3302–3309, 2023. doi: 10.14778/3611479.3611527. URL https://www.vldb.org/pvldb/vol16/p3302-fernandez.pdf.

[45] Benjamin Feuer, Yurong Liu, Chinmay Hegde, and Juliana Freire. Archetype: A novel framework for open-source column type annotation using large language models. *Proceedings of the VLDB Endowment*, 17(9):2279–2292, May 2024. ISSN 2150-8097. doi: 10.14778/3665844.3665857. URL http://dx.doi.org/10.14778/3665844.3665857.

[46] Yukihisa Fujita, Teruaki Hayashi, and Masahiro Kuwahara. Inferring relationships between tabular data and topics using llm for a dataset search task. In *2024 IEEE International Conference on Big Data (BigData)*, pages 6564–6573. IEEE, 2024.

[47] Congcong Ge, Pengfei Wang, Lu Chen, Xiaoze Liu, Baihua Zheng, and Yunjun Gao. Collaborem: A self-supervised entity matching framework using multi-features collaboration. *IEEE Transactions on Knowledge and Data Engineering*, 35(12):12139–12152, 2021.

[48] Chaitanya Gokhale, Sanjib Das, AnHai Doan, Jeffrey F Naughton, Narasimhan Rampalli, Jude Shavlik, and Xiaojin Zhu. Corleone: Hands-off crowdsourcing for entity matching. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*, pages 601–612, 2014.

[49] Kathleen Gregory and Laura Koesten. *Human-centered data discovery*. Springer, 2022.

[50] Michael Günther, Maik Thiele, Julius Gonsior, and Wolfgang Lehner. Pre-trained web table embeddings for table discovery. In *Proceedings of the Fourth International Workshop on Exploiting Artificial Intelligence Techniques for Data Management*, pages 24–31, 2021.

[51] Xuming Hu, Shen Wang, Xiao Qin, Chuan Lei, Zhengyuan Shen, Christos Faloutsos, Asterios Katsifodimos, George Karypis, Lijie Wen, and Philip S. Yu. Automatic table union search with tabular representation learning. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3786–3800. Association for Computational Linguistics, 2023.

[52] Zezhou Huang and Eugene Wu. Cocoon: Semantic table profiling using large language models. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, pages 1–7, 2024.

[53] Zezhou Huang, Jia Guo, and Eugene Wu. Transform table to database using large language models. *Proceedings of the VLDB Endowment. ISSN*, 2024.

[54] Joon Suk Huh, Changho Shin, and Elina Choi. Pool-search-demonstrate: Improving data-wrangling llms via better in-context examples. In *NeurIPS 2023 Second Table Representation Learning Workshop*, 2023.

[55] Andra Ionescu, Rihan Hai, Marios Fragkoulis, and Asterios Katsifodimos. Join path-based data augmentation for decision trees. In *2022 IEEE 38th International Conference on Data Engineering Workshops (ICDEW)*, pages 84–88. IEEE, 2022.

[56] Andra Ionescu, Kiril Vasilev, Florena Buse, Rihan Hai, and Asterios Katsifodimos. Autofeat: Transitive feature discovery over join paths. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1861–1873. IEEE, 2024.

[57] Deyi Ji, Lanyun Zhu, Siqi Gao, Peng Xu, Hongtao Lu, Jieping Ye, and Feng Zhao. Tree-of-table: Unleashing the power of llms for enhanced large-scale table understanding. *arXiv preprint arXiv:2411.08516*, 2024.

[58] Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. Target: Benchmarking table retrieval for generative tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

[59] Moe Kayali, Anton Lykov, Ilias Fountalis, Nikolaos Vasiloglou, Dan Olteanu, and Dan Suciu. Chorus: Foundation models for unified data discovery and exploration. *Proceedings of the VLDB Endowment*, 17(8):2104–2114, 2024.

[60] Aamod Khatiwada, Grace Fan, Roee Shraga, Zixuan Chen, Wolfgang Gatterbauer, Renée J. Miller, and Mirek Riedewald. Santos: Relationship-based semantic table union search. In *SIGMOD*, 2023.

[61] Aamod Khatiwada, Harsha Kokel, Ibrahim Abdelaziz, Subhajit Chaudhury, Julian Dolby, Oktie Hassanzadeh, Zhenhan Huang, Tejaswini Pedapati, Horst Samulowitz, and Kavitha Srinivas. Tabsketchfm: Sketch-based tabular representation learning for data discovery over data lakes. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024.

[62] Laura M Koesten, Emilia Kacprzak, Jenifer FA Tennison, and Elena Simperl. 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*, pages 1277–1289, 2017.

[63] Pradap Venkatramanan Konda. *Magellan: Toward building entity matching management systems*. The University of Wisconsin-Madison, 2018.

[64] Keti Korini and Christian Bizer. Column type annotation using chatgpt. In *CEUR Workshop Proceedings*, volume 3462, pages 1–12, 2023.

[65] Christos Koutras, George Siachamis, Andra Ionescu, Kyriakos Psarakis, Jerry Brons, Marios Fragkoulis, Christoph Lofi, Angela Bonifati, and Asterios Katsifodimos. Valentine: Evaluating matching techniques for dataset discovery. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 468–479. IEEE, 2021.

[66] Christos Koutras, Rihan Hai, Kyriakos Psarakis, Marios Fragkoulis, and Asterios Katsifodimos. Sima: Effective and efficient data silo federation using graph neural networks. *arXiv preprint arXiv:2206.12733*, 2022.

[67] Christos Koutras, Jiani Zhang, Xiao Qin, Chuan Lei, Vasileios Ioannidis, Christos Faloutsos, George Karypis, and Asterios Katsifodimos. Omnimatch: Effective self-supervised any-join discovery in tabular data repositories. *arXiv preprint arXiv:2403.07653*, 2024.

[68] Oliver Lehmberg, Dominique Ritze, Petar Ristoski, Robert Meusel, Heiko Paulheim, and Christian Bizer. The mannheim search join engine. *J. Web Semant.*, 35:159–166, 2015.

[69] Maurizio Lenzerini. Data integration: a theoretical perspective. In *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, page 233–246, New York, NY, USA, 2002. Association for Computing Machinery. ISBN 1581135076. doi: 10.1145/543613.543644. URL `https://doi-org.proxy.library.nyu.edu/10.1145/543613.543644`.

[70] Bing Li, Yukai Miao, Yaoshu Wang, Yifang Sun, and Wei Wang. Improving the efficiency and effectiveness for bert-based entity resolution. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15):13226–13233, 2021.

[71] Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. Table-gpt: Table fine-tuned gpt for diverse table tasks. *Proceedings of the ACM on Management of Data*, 2(3):1–28, 2024.

[72] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*, 2024.

[73] Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. Long-context llms struggle with long in-context learning, 2024. URL `https://arxiv.org/abs/2404.02060`.

[74] Yize Li, Yongchao Dou, Felipe Da Veiga Leprevost, Yifat Geffen, Anna P Calinawan, François Aguet, Yo Akiyama, Shankara Anand, Chet Birger, Song Cao, et al. Proteogenomic data and resources for pan-cancer analysis. *Cancer cell*, 41(8):1397–1406, 2023. PMCID: PMC10506762.

[75] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, AnHai Doan, and Wang-Chiew Tan. Deep entity matching with pre-trained language models. *Proceedings of the VLDB Endowment*, 14(1):50–60, 2020.

[76] Yuliang Li, Jinfeng Li, Yoshihiko Suhara, Jin Wang, Wataru Hirota, and Wang-Chiew Tan. Deep Entity Matching: Challenges and Opportunities. *J. Data and Information Quality*, 13(1):1:1–1:17, January 2021. ISSN 1936-1955. doi: 10.1145/3431816.

[77] Yuliang Li, Jinfeng Li, Yoshi Suhara, AnHai Doan, and Wang-Chiew Tan. Effective entity matching with transformers. *The VLDB Journal*, 32(6):1215–1235, 2023.

[78] Jiaming Liang, Chuan Lei, Xiao Qin, Jiani Zhang, Asterios Katsifodimos, Christos Faloutsos, and Huzefa Rangwala. Featnavigator: Automatic feature augmentation on tabular data. *arXiv preprint arXiv:2406.09534*, 2024.

[79] Jiabin Liu, Chengliang Chai, Yuyu Luo, Yin Lou, Jianhua Feng, and Nan Tang. Feature augmentation with reinforcement learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 3360–3372. IEEE, 2022.

[80] Junfei Liu, Shaotong Sun, and Fatemeh Nargesian. Causal dataset discovery with large language models. In *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, pages 1–8, 2024.

[81] Yurong Liu, Eduardo Pena, Aecio Santos, Eden Wu, and Juliana Freire. Magneto: Combining small and large language models for schema matching. *arXiv preprint arXiv:2412.08194*, 2024.

[82] Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. Large language model for table processing: A survey. *Frontiers of Computer Science*, 19(2):192350, 2025.

[83] Chuangtao Ma, Sriom Chakrabarti, Arijit Khan, and Bálint Molnár. Knowledge graph-based retrieval-augmented generation for schema matching. *arXiv preprint arXiv:2501.08686*, 2025.

[84] Jayant Madhavan, Philip A Bernstein, and Erhard Rahm. Generic schema matching with cupid. In *vldb*, volume 1, pages 49–58, 2001.

[85] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[86] Ali Modarressi, Hanieh Deilamsalehy, Franck Dernoncourt, Trung Bui, Ryan A. Rossi, Seunghyun Yoon, and Hinrich Schütze. Nolima: Long-context evaluation beyond literal matching, 2025. URL `https://arxiv.org/abs/2502.05167`.

[87] Sidharth Mudgal, Han Li, Theodoros Rekatsinas, AnHai Doan, Youngchoon Park, Ganesh Krishnan, Rohit Deep, Esteban Arcaute, and Vijay Raghavendra. Deep learning for entity matching: A design space exploration. In *Proceedings of the 2018 international conference on management of data*, pages 19–34, 2018.

[88] Yang Nan, Javier Del Ser, Simon Walsh, Carola Schönlieb, Michael Roberts, Ian Selby, Kit Howard, John Owen, Jon Neville, Julien Guiot, Benoit Ernst, Ana Pastor, Angel Alberich-Bayarri, Marion I. Menzel, Sean Walsh, Wim Vos, Nina Flerin, Jean-Paul Charbonnier, Eva M. van Rikxoort, Avishek Chatterjee, Henry C. Woodruff, Philippe Lambin, Leonor Cerdá Alberich, Luis Martí-Bonmatí, Francisco Herrera, and Guang Yang. Data harmonisation for information fusion in digital healthcare: A state-of-the-art systematic review, meta-analysis and future research directions. *CoRR*, abs/2201.06505, 2022. URL `https://arxiv.org/abs/2201.06505`.

[89] Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. Can foundation models wrangle your data? *Proceedings of the VLDB Endowment*, 16(4):738–746, 2022.

[90] Fatemeh Nargesian, Erkang Zhu, Ken Q. Pu, and Renée J. Miller. Table union search on open data. *Proc. VLDB Endow.*, 11(7):813–825, 2018.

[91] Arash Dargahi Nobari and Davood Rafiei. DTT: An example-driven tabular transformer for joinability by leveraging large language models. *Proceedings of the ACM on Management of Data*, 2(1):1–24, 2024.

[92] Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, Michal Guerquin, Hamish Ivison, Pang Wei Koh, Jiacheng Liu, Saumya Malik, William Merrill, Lester James V. Miranda, Jacob Morrison, Tyler Murray, Crystal Nam, Valentina Pyatkin, Aman Rangapur, Michael Schmitz, Sam Skjonsberg, David Wadden, Christopher Wilhelm, Michael Wilson, Luke Zettlemoyer, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. 2 olmo 2 furious, 2024. URL https://arxiv.org/abs/2501.00656.

[93] omop. Observational medical outcomes partnership (omop) common data model (cdm). https://www.ohdsi.org/data-standardization, 2024.

[94] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html.

[95] George Papadakis, Dimitrios Skoutas, Emmanouil Thanos, and Themis Palpanas. Blocking and Filtering Techniques for Entity Resolution: A Survey. *ACM Comput. Surv.*, 53(2):31:1–31:42, March 2020. ISSN 0360-0300. doi: 10.1145/3377455.

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

[97] Marcel Parciak, Brecht Vandevoort, Frank Neven, Liesbet M Peeters, and Stijn Vansummeren. Schema matching with large language models: an experimental study. *Proceedings of the VLDB Endowment. ISSN*, 2150:8097, 2024.

[98] Ralph Peeters and Christian Bizer. Dual-objective fine-tuning of bert for entity matching. *Proceedings of the VLDB Endowment*, 14:1913–1921, 2021.

[99] Ralph Peeters, Aaron Steiner, and Christian Bizer. Entity matching using large language models. *Proceedings 28th International Conference on Extending Database Technology, EDBT 2025, Barcelona, Spain, March 25-28, 2025*, pages 529–541, 2025. doi: 10.48786/EDBT.2025.42. URL https://doi.org/10.48786/edbt.2025.42.

[100] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[101] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023.

[102] Erhard Rahm and Philip A Bernstein. A survey of approaches to automatic schema matching. *the VLDB Journal*, 10:334–350, 2001.

[103] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[104] Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal V. Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Févry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. Multitask prompted training enables zero-shot task generalization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL `https://openreview.net/forum?id=9Vrb9D0WI4`.

[105] Aécio Santos, Aline Bessa, Fernando Chirigati, Christopher Musco, and Juliana Freire. Correlation sketches for approximate join-correlation queries. In *Proceedings of the 2021 International Conference on Management of Data*, pages 1531–1544, 2021.

[106] Aécio Santos, Aline Bessa, Christopher Musco, and Juliana Freire. A sketch-based index for correlated dataset search. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 2928–2941. IEEE, 2022.

[107] Aécio Santos, Eduardo HM Pena, Roque Lopez, and Juliana Freire. Interactive data harmonization with llm agents. *arXiv preprint arXiv:2502.07132*, 2025.

[108] Aécio Santos, Flip Korn, and Juliana Freire. Efficiently estimating mutual information between attributes across tables. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 193–206, 2024. doi: 10.1109/ICDE60146.2024.00022.

[109] Anish Das Sarma, Lujun Fang, Nitin Gupta, Alon Y Halevy, Hongrae Lee, Fei Wu, Reynold Xin, and Cong Yu. Finding related tables. In *SIGMOD Conference*, volume 10, pages 2213836–2213962, 2012.

[110] Nabeel Seedat and Mihaela van der Schaar. Matchmaker: Self-improving large language model programs for schema matching. In *GenAI for Health: Potential, Trust and Policy Compliance*, 2024.

[111] Sofia Serrano, Zander Brumbaugh, and Noah A. Smith. Language models: A guide for the perplexed, 2023. URL `https://arxiv.org/abs/2311.17301`.

[112] Eitam Sheetrit, Menachem Brief, Moshik Mishaeli, and Oren Elisha. Rematch: Retrieval enhanced schema matching with llms. *arXiv preprint arXiv:2403.01567*, 2024.

[113] Levy Silva and Luciano Barbosa. Improving dense retrieval models with llm augmented data for dataset search. *Knowledge-Based Systems*, 294:111740, 2024.

[114] Rohit Singh, Vamsi Meduri, Ahmed Elmagarmid, Samuel Madden, Paolo Papotti, Jorge-Arnulfo Quiané-Ruiz, Armando Solar-Lezama, and Nan Tang. Generating concise entity matching rules. In *Proceedings of the 2017 ACM International Conference on Management of Data*, pages 1635–1638, 2017.

[115] Socrata. `https://open-source.socrata.com`, 2024.

[116] Aaron Steiner, Ralph Peeters, and Christian Bizer. Fine-tuning large language models for entity matching. *arXiv preprint arXiv:2409.08185*, 2024.

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

[118] Nan Tang, Chenyu Yang, Ju Fan, Lei Cao, Yuyu Luo, and Alon Halevy. Verifai: verified generative ai. *Conference on Innovative Data Systems Research (CIDR)*, 2024.

[119] Saravanan Thirumuruganathan, Han Li, Nan Tang, Mourad Ouzzani, Yash Govind, Derek Paulsen, Glenn Fung, and AnHai Doan. Deep learning for blocking in entity matching: a design space exploration. *Proceedings of the VLDB Endowment*, 14(11):2459–2472, 2021.

[120] Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, YaGuang Li, Hongrae Lee, Huaixiu Steven Zheng, Amin Ghafouri, Marcelo Menegali, Yanping Huang, Maxim Krikun, Dmitry Lepikhin, James Qin, Dehao Chen, Yuanzhong Xu, Zhifeng Chen, Adam Roberts, Maarten Bosma, Vincent Zhao, Yanqi Zhou, Chung-Ching Chang, Igor Krivokon, Will Rusch, Marc Pickett, Pranesh Srinivasan, Laichee Man, Kathleen Meier-Hellstern, Meredith Ringel Morris, Tulsee Doshi, Renelito Delos Santos, Toju Duke, Johnny Soraker, Ben Zevenbergen, Vinodkumar Prabhakaran, Mark Diaz, Ben Hutchinson, Kristen Olson, Alejandra Molina, Erin Hoffman-John, Josh Lee, Lora Aroyo, Ravi Rajakumar, Alena Butryna, Matthew Lamm, Viktoriya Kuzmina, Joe Fenton, Aaron Cohen, Rachel Bernstein, Ray Kurzweil, Blaise Aguera-Arcas, Claire Cui, Marian Croak, Ed Chi, and Quoc Le. Lamda: Language models for dialog applications, 2022.

[121] Mohamed Trabelsi, Zhiyu Chen, Shuo Zhang, Brian D Davison, and Jeff Heflin. Strubert: Structure-aware bert for table search and matching. In *Proceedings of the ACM Web Conference 2022*, pages 442–451, 2022.

[122] Immanuel Trummer. Can large language models predict data correlations from column names? *Proc. VLDB Endow.*, 16(13):4310–4323, September 2023. ISSN 2150-8097. doi: 10.14778/3625054.3625066. URL `https://doi.org/10.14778/3625054.3625066`.

[123] Jianhong Tu, Ju Fan, Nan Tang, Peng Wang, Guoliang Li, Xiaoyong Du, Xiaofeng Jia, and Song Gao. Unicorn: A unified multi-tasking model for supporting matching tasks in data integration. *Proceedings of the ACM on Management of Data*, 1(1):1–26, 2023.

[124] Fei Wang, Kexuan Sun, Muhao Chen, Jay Pujara, and Pedro Szekely. Retrieving complex tables with multi-granular graph representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1472–1482, 2021.

[125] Jiannan Wang, Tim Kraska, Michael J Franklin, and Jianhua Feng. Crowder: Crowdsourcing entity resolution. *Proceedings of the VLDB Endowment*, 5(11), 2012.

[126] Qiming Wang and Raul Castro Fernandez. Solo: Data discovery using natural language questions via a self-supervised approach. *Proceedings of the ACM on Management of Data*, 1(4):1–27, 2023.

[127] Runhui Wang, Yuliang Li, and Jin Wang. Sudowoodo: Contrastive self-supervised learning for multi-purpose data integration and preparation. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1502–1515. IEEE, 2023.

[128] Tianshu Wang, Xiaoyang Chen, Hongyu Lin, Xuanang Chen, Xianpei Han, Hao Wang, Zhenyu Zeng, and Le Sun. Match, compare, or select? an investigation of large language models for entity matching. *arXiv preprint arXiv:2405.16884*, 2024.

[129] Yaoshu Wang and Mengyi Yan. Unsupervised domain adaptation for entity blocking leveraging large language models. In *2024 IEEE International Conference on Big Data (BigData)*, pages 159–164. IEEE, 2024.

[130] Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022.

[131] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL `http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html`.

[132] Lindsey Linxi Wei, Guorui Xiao, and Magdalena Balazinska. Racoon: An llm-based framework for retrieval-augmented column type annotation with a knowledge graph. *NeurIPS Third Table Representation Learning Workshop*, 2024.

[133] 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. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3(1):1–9, 2016.

[134] Renzhi Wu, Sanya Chaba, Saurabh Sawlani, Xu Chu, and Saravanan Thirumuruganathan. Zeroer: Entity resolution using zero labeled examples. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pages 1149–1164, 2020.

[135] Shiwen Wu, Qiyu Wu, Honghua Dong, Wen Hua, and Xiaofang Zhou. Blocker and matcher can mutually benefit: A co-learning framework for low-resource entity resolution. *Proceedings of the VLDB Endowment*, 17(3):292–304, 2023.

[136] Yongqin Xu, Huan Li, Ke Chen, and Lidan Shou. Kcmf: A knowledge-compliant framework for schema and entity matching with fine-tuning-free llms. *arXiv preprint arXiv:2410.12480*, 2024.

[137] Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024.

[138] Mengyi Yan, Yaoshu Wang, Kehan Pang, Min Xie, and Jianxin Li. Efficient Mixture of Experts based on Large Language Models for Low-Resource Data Preprocessing. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '24, pages

3690–3701, New York, NY, USA, August 2024. Association for Computing Machinery. doi: 10.1145/3637528.3671873.

[139] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Shaochen Zhong, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data*, 18(6):1–32, 2024.

[140] Jun Yang, Peng Gong, Rong Fu, Minghua Zhang, Jingming Chen, Shunlin Liang, Bing Xu, Jiancheng Shi, and Robert Dickinson. The role of satellite remote sensing in climate change studies. *Nature climate change*, 3(10):875–883, 2013.

[141] Shunyu Yao, Jeffrey Zhao, Dian Yu, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. URL https://openreview.net/forum?id=tvI4u1ylcqs.

[142] Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 174–184, 2023.

[143] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745.

[144] Brit Youngmann, Michael Cafarella, Babak Salimi, and Anna Zeng. Causal data integration. *arXiv preprint arXiv:2305.08741*, 2023.

[145] Xiaocan Zeng, Pengfei Wang, Yuren Mao, Lu Chen, Xiaoze Liu, and Yunjun Gao. Multiem: Efficient and effective unsupervised multi-table entity matching. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 3421–3434. IEEE, 2024.

[146] Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. Jellyfish: Instruction-tuning local large language models for data preprocessing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8754–8782, 2024.

[147] Haoxiang Zhang, Yurong Liu, Aécio Santos, Juliana Freire, et al. Autoddg: Automated dataset description generation using large language models. *arXiv preprint arXiv:2502.01050*, 2025.

[148] Jing Zhang, Bonggun Shin, Jinho D Choi, and Joyce C Ho. Smat: An attention-based deep learning solution to the automation of schema matching. In *Advances in Databases and Information Systems: 25th European Conference, ADBIS 2021, Tartu, Estonia, August 24–26, 2021, Proceedings 25*, pages 260–274. Springer, 2021.

[149] Meihui Zhang, Marios Hadjieleftheriou, Beng Chin Ooi, Cecilia M Procopiuc, and Divesh Srivastava. Automatic discovery of attributes in relational databases. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of data*, pages 109–120, 2011.

[150] Meihui Zhang, Zhaoxuan Ji, Zhaojing Luo, Yuncheng Wu, and Chengliang Chai. Applications and challenges for large language models: From data management perspective. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 5530–5541. IEEE, 2024.

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

[152] Shuo Zhang and Krisztian Balog. Web table extraction, retrieval, and augmentation: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(2):1–35, 2020.

[153] Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. Tablellama: Towards open large generalist models for tables, 2023.

[154] Yunjia Zhang, Avrilia Floratou, Joyce Cahoon, Subru Krishnan, Andreas C Müller, Dalitso Banda, Fotis Psallidas, and Jignesh M Patel. Schema matching using pre-trained language models. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, pages 1558–1571. IEEE, 2023.

[155] Yunyi Zhang, Ming Zhong, Siru Ouyang, Yizhu Jiao, Sizhe Zhou, Linyi Ding, and Jiawei Han. Automated mining of structured knowledge from text in the era of large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6644–6654, 2024.

[156] Yilun Zhao, Lyuhao Chen, Arman Cohan, and Chen Zhao. Tapera: enhancing faithfulness and interpretability in long-form table qa by content planning and execution-based reasoning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12824–12840, 2024.

[157] Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, et al. Evaluation of openai o1: Opportunities and challenges of agi. *arXiv preprint arXiv:2409.18486*, 2024.

[158] Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[159] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. LSH ensemble: Internet-scale domain search. *Proc. VLDB Endow.*, 9(12):1185–1196, 2016.

[160] Erkang Zhu, Dong Deng, Fatemeh Nargesian, and Renée J. Miller. JOSIE: overlap set similarity search for finding joinable tables in data lakes. In *SIGMOD*, pages 847–864, 2019.