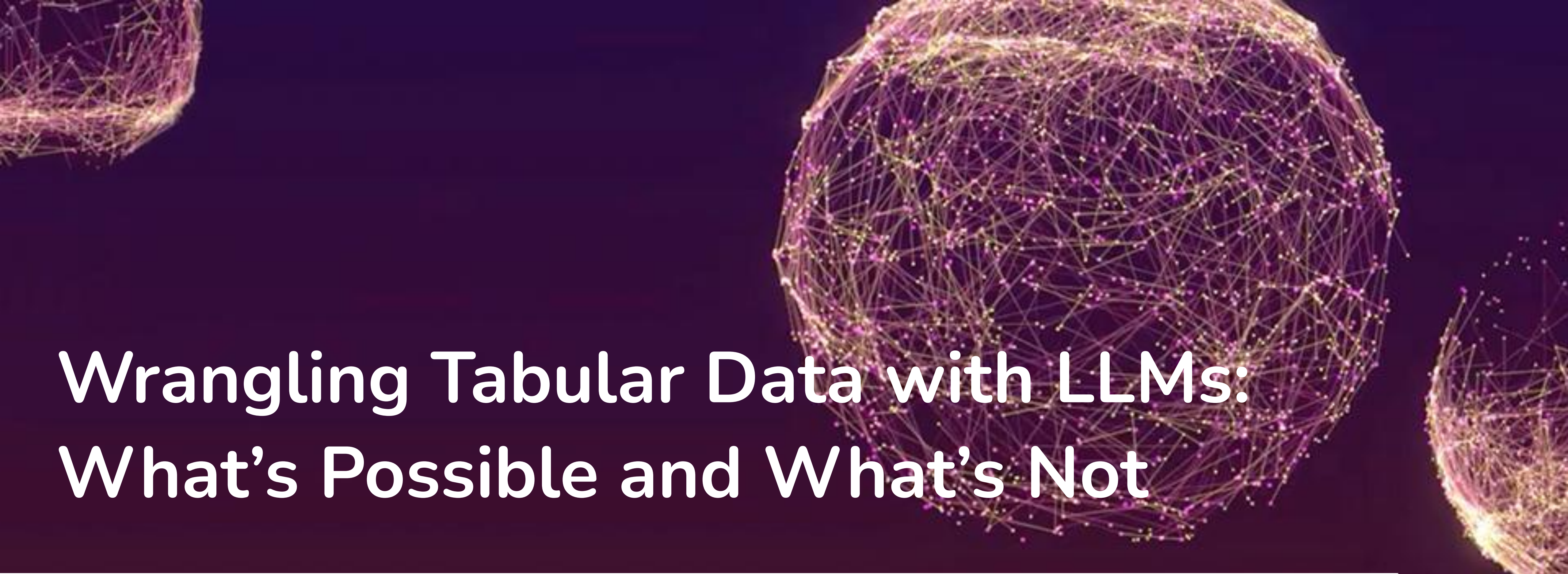


Wrangling Tabular Data with LLMs: What's Possible and What's Not

CSG Data Management

Jan-Micha Bodensohn and Liane Vogel

NHR4CES
NHR for Computational Engineering Science

NHR
Hessisches Ministerium für Wissenschaft und Kunst
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen
Gefördert vom Bundesministerium für Bildung und Forschung
Technische Universität Darmstadt
RWTH Aachen University

Collaborators

Jan-Micha Bodensohn

Ulf Brackmann

Liane Vogel

Matthias Urban

Anupam Sanghi

Carsten Binnig

Agenda

(Brief) Introduction to LLMs

LLMs for Data Engineering

- What works?
- What doesn’t work (yet)?
- What’s to come?

(Brief) Introduction to LLMs

Introduction to LLMs

trained on vast amounts of data

WikiDBs: A Large-Scale Corpus of Relational Databases from Wikidata

Liane Vogel[1] Jan-Micha Bodensohn[2,1] Carsten Binnig[1,2]
[1]Technical University of Darmstadt, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Darmstadt, Germany

Abstract

Deep learning on tabular data, and particularly tabular representation learning, has recently gained growing interest. However, representation learning for relational databases with *multiple* tables is still an underexplored area, which may be attributed to the lack of openly available resources. To support the development of foundation models for tabular data and relational databases, we introduce **WikiDBs, a novel open-source corpus of 100,000 relational databases**. Each database consists of multiple tables connected by foreign keys. The corpus is based on Wikidata and aims to follow certain characteristics of real-world databases. In this paper, we describe the dataset and our method for creating it. By making our code publicly available, we enable others to create tailored versions of the dataset, for example, by creating databases in different languages. Finally, we conduct a set of initial experiments to showcase how WikiDBs can be used to train for data engineering tasks, such as missing value imputation and column type annotation.

1 Introduction

The importance of representation learning on relational data. Foundation models for text, images, and videos support users in many every-day tasks, reducing the manual effort required for end tasks such as machine translation, image generation, and automated software development [13].

While research on learning such foundation models is currently dominated by text and images, considerable progress has recently been made on other modalities such as tabular data [14, 6]. This is important since a non-negligible amount of data is expressed in tabular form, in particular enterprise data [11]. Pre-trained foundation models that are specialized to handle relational data could reduce the manual effort required for many data-heavy tasks such as data integration and data cleaning [47].

Over the last years, several deep learning-based approaches have been developed to solve data engineering tasks like entity matching [38] and missing value imputation [35] on individual tables. pported by various large-scale table corpora, such as GitTables [23] and vide the training data required to train these deep learning models.

relational databases. Considerable amounts of tabular data, especially not stored in individual self-contained tables, but rather in relational ltiple tables that are closely connected by foreign key relationships. For atabases, however, there is a lack of large-scale openly available training network architectures that can process multiple related tables. Collecting ational data is not trivial. Due to the sensitivity of the data stored in world enterprise databases are typically highly confidential and therefore ific community, resulting in a lack of openly available corpora.

formation Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

LLMs

→

Understand
and
generate
language

Introduction to LLMs

trained on vast amounts of data

WikiDBs: A Large-Scale Corpus of Relational Databases from Wikidata

Liane Vogel[1] Jan-Micha Bodensohn[2,1] Carsten Binnig[1,2]
[1]Technical University of Darmstadt, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Darmstadt, Germany

Abstract

Deep learning on tabular data, and particularly tabular representation learning, has recently gained growing interest. However, representation learning for relational databases with *multiple* tables is still an underexplored area, which may be attributed to the lack of openly available resources. To support the development of foundation models for tabular data and relational databases, we introduce **WikiDBs, a novel open-source corpus of 100,000 relational databases**. Each database consists of multiple tables connected by foreign keys. The corpus is based on Wikidata and aims to follow certain characteristics of real-world databases. In this paper, we describe the dataset and our method for creating it. By making our code publicly available, we enable others to create tailored versions of the dataset, for example, by creating databases in different languages. Finally, we conduct a set of initial experiments to showcase how WikiDBs can be used to train for data engineering tasks, such as missing value imputation and column type annotation.

1 Introduction

The importance of representation learning on relational data. Foundation models for text, images, and videos support users in many every-day tasks, reducing the manual effort required for end tasks such as machine translation, image generation, and automated software development [13].

While research on learning such foundation models is currently dominated by text and images, considerable progress has recently been made on other modalities such as tabular data [14, 6]. This is important since a non-negligible amount of data is expressed in tabular form, in particular enterprise data [11]. Pre-trained foundation models that are specialized to handle relational data could reduce the manual effort required for many data-heavy tasks such as data integration and data cleaning [47].

Over the last years, several deep learning-based approaches have been developed to solve data engineering tasks like entity matching [38] and missing value imputation [35] on individual tables.
pported by various large-scale table corpora, such as GitTables [23] and
vide the training data required to train these deep learning models.

relational databases. Considerable amounts of tabular data, especially
not stored in individual self-contained tables, but rather in relational
ltiple tables that are closely connected by foreign key relationships. For
atabases, however, there is a lack of large-scale openly available training
network architectures that can process multiple related tables. Collecting
ational data is not trivial. Due to the sensitivity of the data stored in
world enterprise databases are typically highly confidential and therefore
ific community, resulting in a lack of openly available corpora.

formation Processing Systems (NeurIPS 2024) Track on Datasets and Benchmarks.

→

LLMs

→

to solve tasks via prompting

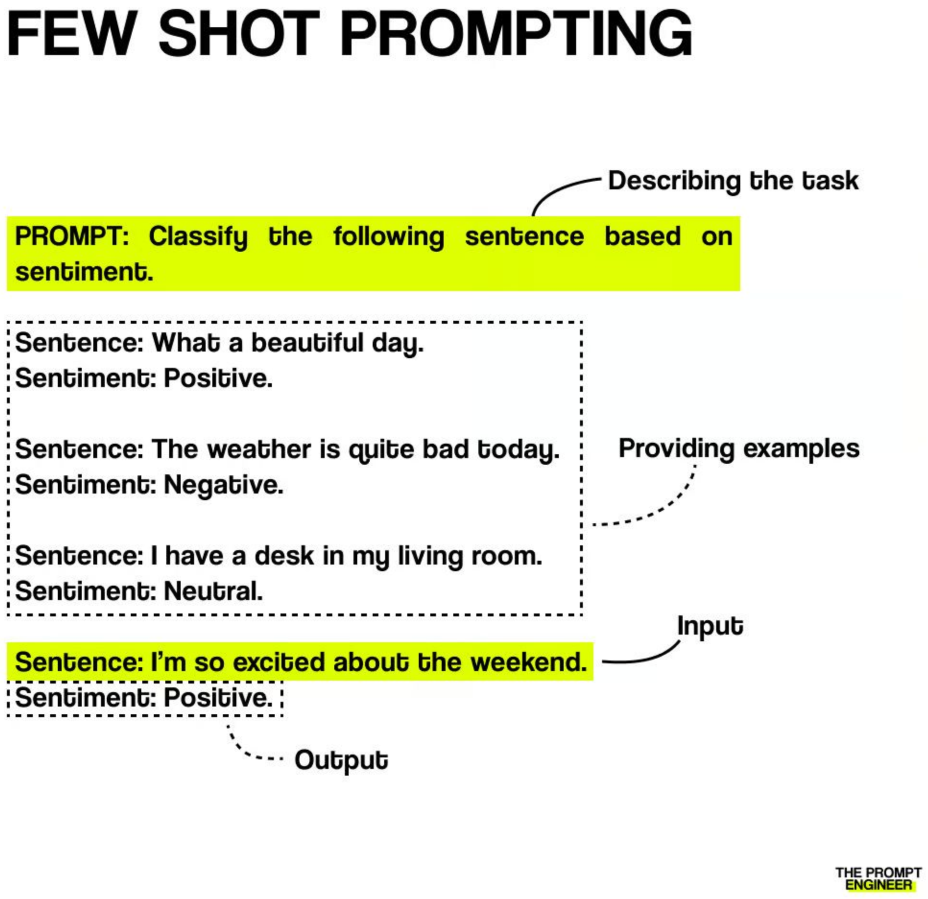
```
Translate the
following e-mail
into German.
```

Solve Tasks via Prompting

Zero Shot Prompting
= provide only the instructions

Few Shot Prompting
= give input & output examples

Chain-of-Thought Prompting
& **Reasoning**
= LLM generates reasoning chain before answering



FEW SHOT PROMPTING

Describing the task

PROMPT: Classify the following sentence based on sentiment.

Sentence: What a beautiful day.
Sentiment: Positive.

Sentence: The weather is quite bad today.
Sentiment: Negative.

Sentence: I have a desk in my living room.
Sentiment: Neutral.

Providing examples

Input

Sentence: I'm so excited about the weekend.
Sentiment: Positive.

Output

THE PROMPT ENGINEER

Image from https://the-prompt-engineer.beehiiv.com/p/3-fewshot-prompting

LLMs - Providers & Models

LLMs

OpenAI

ChatGPT

o1

GPT-4o

Google

Gemini

Meta

Llama

ANTHROP\C

Claude

Mistral

Alibaba

Qwen

LLMs - Beyond text

Text

WikiDBs: A Large-Scale Corpus of Relational Databases from Wikidata

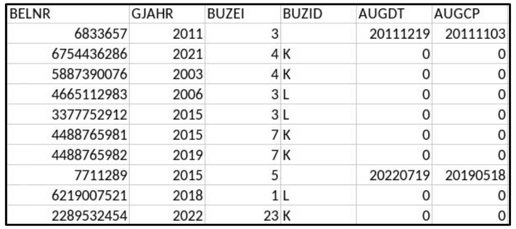
Liane Vogel[1] Jan-Micha Bodensohn[2,1] Carsten Binnig[1,2]
[1]Technical University of Darmstadt, Germany
[2]German Research Center for Artificial Intelligence (DFKI), Darmstadt, Germany

Abstract

Deep learning on tabular data, and particularly tabular representation learning, has recently gained growing interest. However, representation learning for relational databases with *multiple* tables is still an underexplored area, which may be attributed to the lack of openly available resources. To support the development of foundation models for tabular data and relational databases, we introduce **WikiDBs, a novel open-source corpus of 100,000 relational databases**. Each database consists of multiple tables connected by foreign keys. The corpus is based on Wikidata and aims to follow certain characteristics of real-world databases. In this paper, we describe the dataset and our method for creating it. By making our code publicly available, we enable others to create tailored versions of the dataset, for example, by creating databases in different languages. Finally, we conduct a set of initial experiments to showcase how WikiDBs can be used to train for data engineering tasks, such as missing value imputation and column type annotation.

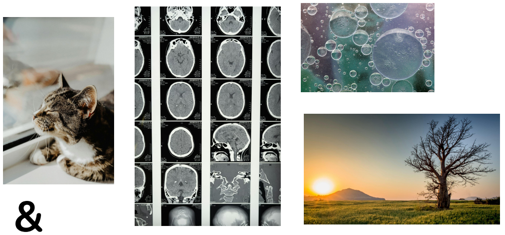
1 Introduction

Code



BELNR	GJAHR	BUZEI	BUZID	AUGDT	AUGCP
6833657	2011	3		20111219	20111103
6754436286	2021	4	K	0	0
5887390076	2003	4	K	0	0
4665112983	2006	3	L	0	0
3377752912	2015	3	L	0	0
4488765981	2015	7	K	0	0
4488765982	2019	7	K	0	0
7711289	2015	5		20220719	20190518
6219007521	2018	1	L	0	0
2289532454	2022	23	K	0	0

Tables & Databases



Images & Videos

LLMs

Translate e-mails

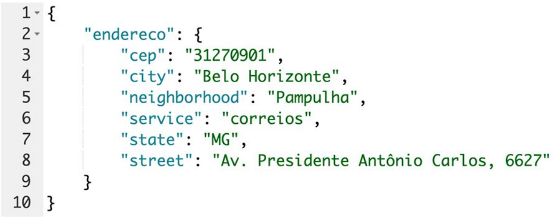
Polish paper abstracts

Describe images

Write code to process data

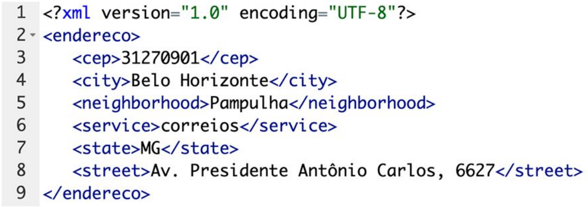
LLMs & Structured data

{json}



```
{
    "endereco": {
        "cep": "31270901",
        "city": "Belo Horizonte",
        "neighborhood": "Pampulha",
        "service": "correios",
        "state": "MG",
        "street": "Av. Presidente Antônio Carlos, 6627"
    }
}
```

XML



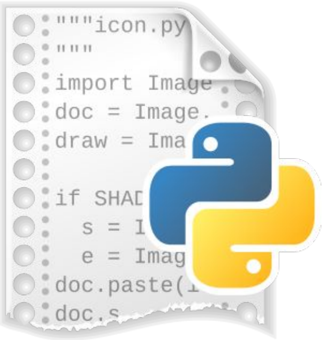
```
<?xml version="1.0" encoding="UTF-8"?>
<endereco>
   <cep>31270901</cep>
   <city>Belo Horizonte</city>
   <neighborhood>Pampulha</neighborhood>
   <service>correios</service>
   <state>MG</state>
   <street>Av. Presidente Antônio Carlos, 6627</street>
</endereco>
```

LLMs

CSV

```
BSEG
MANDT,BUKRS,BELNR,GJAHR,BUZEI,BUZID,AUGDT,AUGCP,AUGBL,...
1,D054,5930568205,2013,5,H,20140503,20140501,9836283674,...
1,D054,5829473293,2021,7,H,20221123,20221119,3485949047,...
1,D037,3168347239,2012,43,L,20120913,20120831,7554950694,...
```

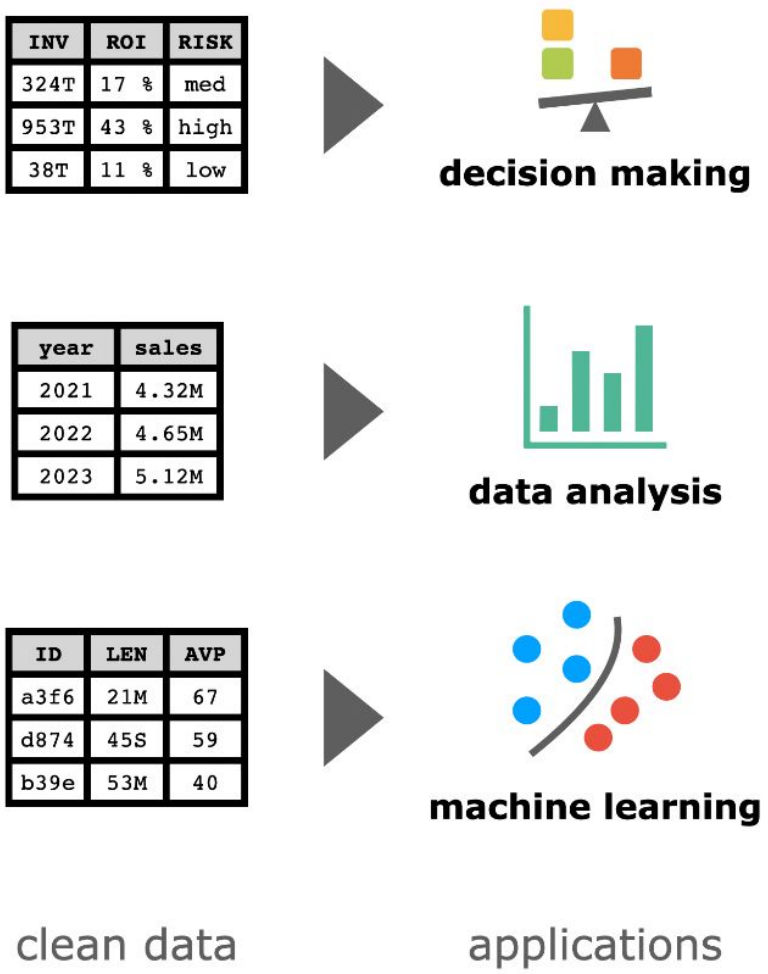
Python



pandas

LLMs for Data Engineering

Applications Need Clean Data



INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

decision making

year	sales
2021	4.32M
2022	4.65M
2023	5.12M

data analysis

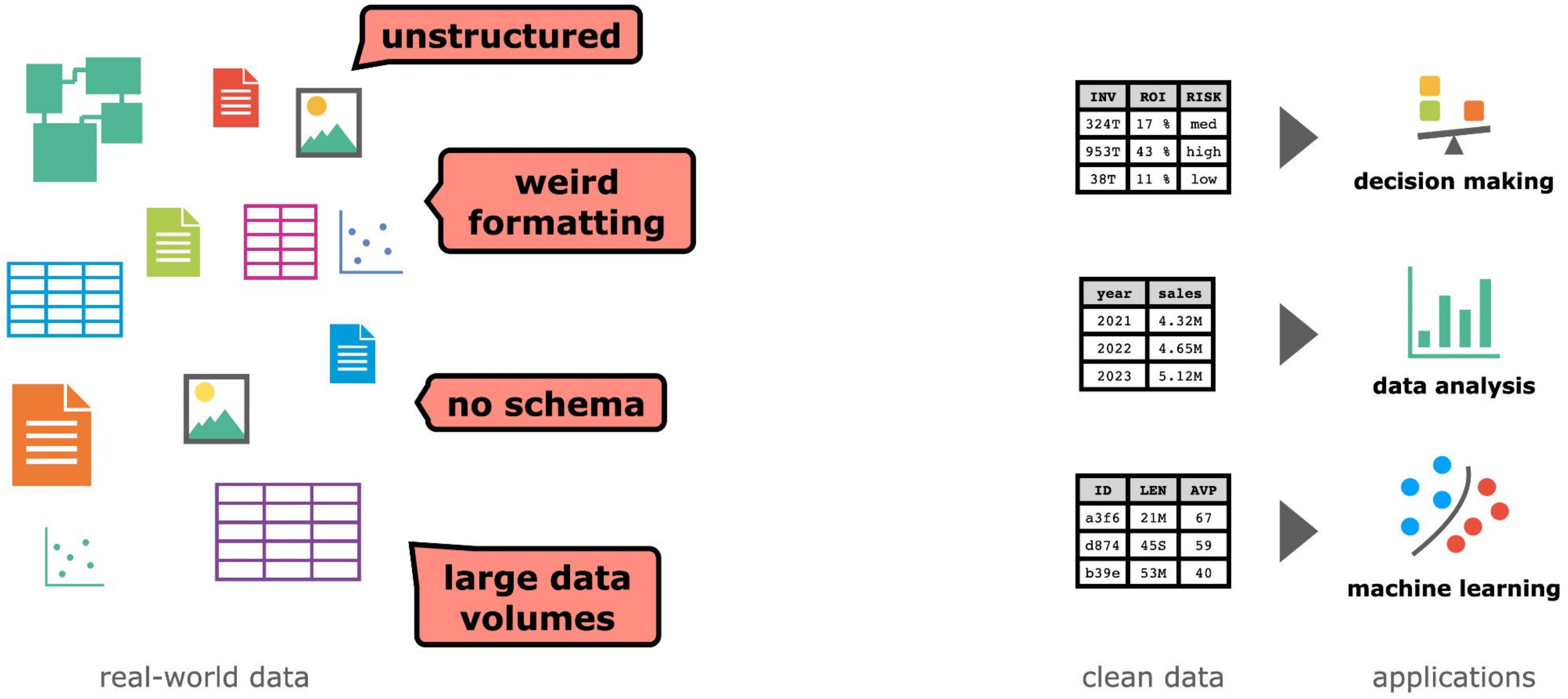
ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

machine learning

clean data

applications

Real-World Data Is Messy



unstructured

weird formatting

no schema

large data volumes

real-world data

INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

year	sales
2021	4.32M
2022	4.65M
2023	5.12M

ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

clean data

decision making

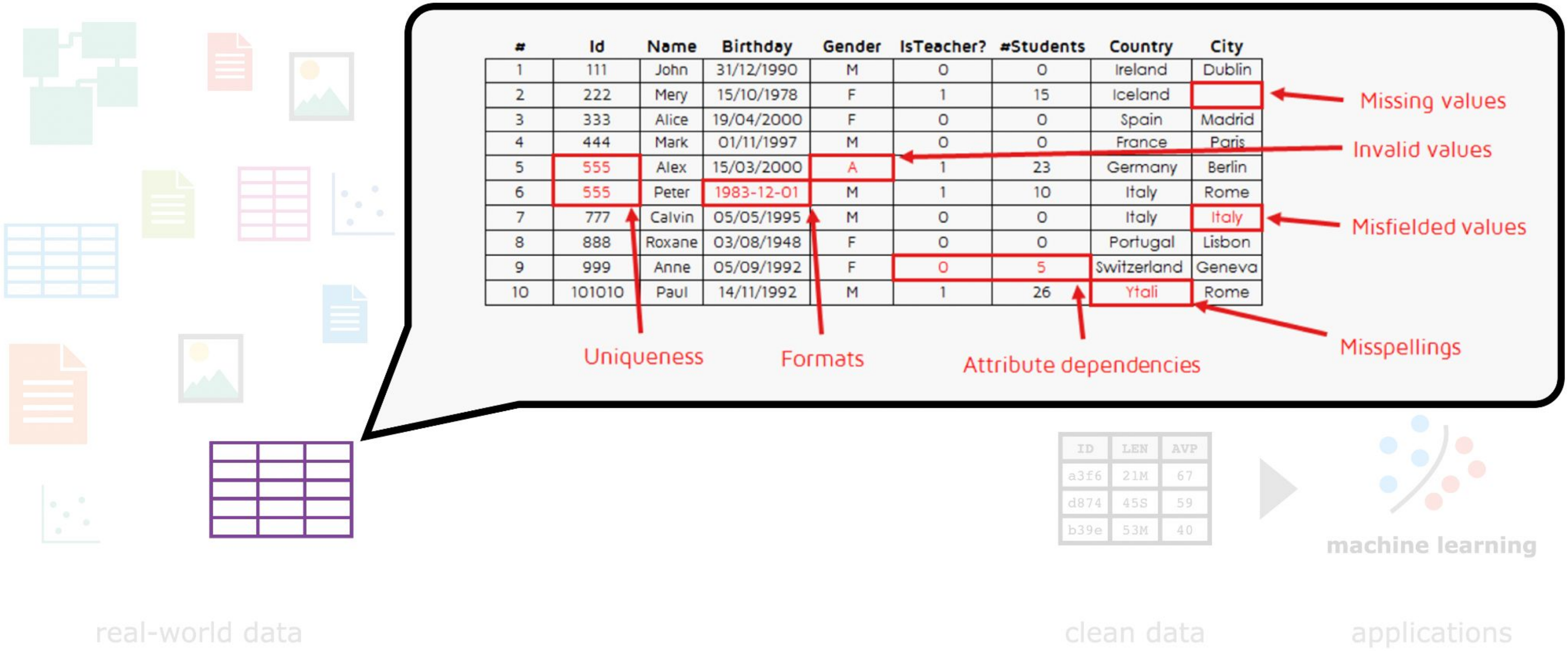
data analysis

machine learning

applications

Even Tables Have Problems

NHR4CES
NHR for Computational Engineering Science



#	Id	Name	Birthday	Gender	IsTeacher?	#Students	Country	City
1	111	John	31/12/1990	M	0	0	Ireland	Dublin
2	222	Mery	15/10/1978	F	1	15	Iceland	
3	333	Alice	19/04/2000	F	0	0	Spain	Madrid
4	444	Mark	01/11/1997	M	0	0	France	Paris
5	555	Alex	15/03/2000	A	1	23	Germany	Berlin
6	555	Peter	1983-12-01	M	1	10	Italy	Rome
7	777	Calvin	05/05/1995	M	0	0	Italy	Italy
8	888	Roxane	03/08/1948	F	0	0	Portugal	Lisbon
9	999	Anne	05/09/1992	F	0	5	Switzerland	Geneva
10	101010	Paul	14/11/1992	M	1	26	Ytali	Rome

Missing values

Invalid values

Misfielded values

Uniqueness

Formats

Attribute dependencies

Misspellings

ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

machine learning

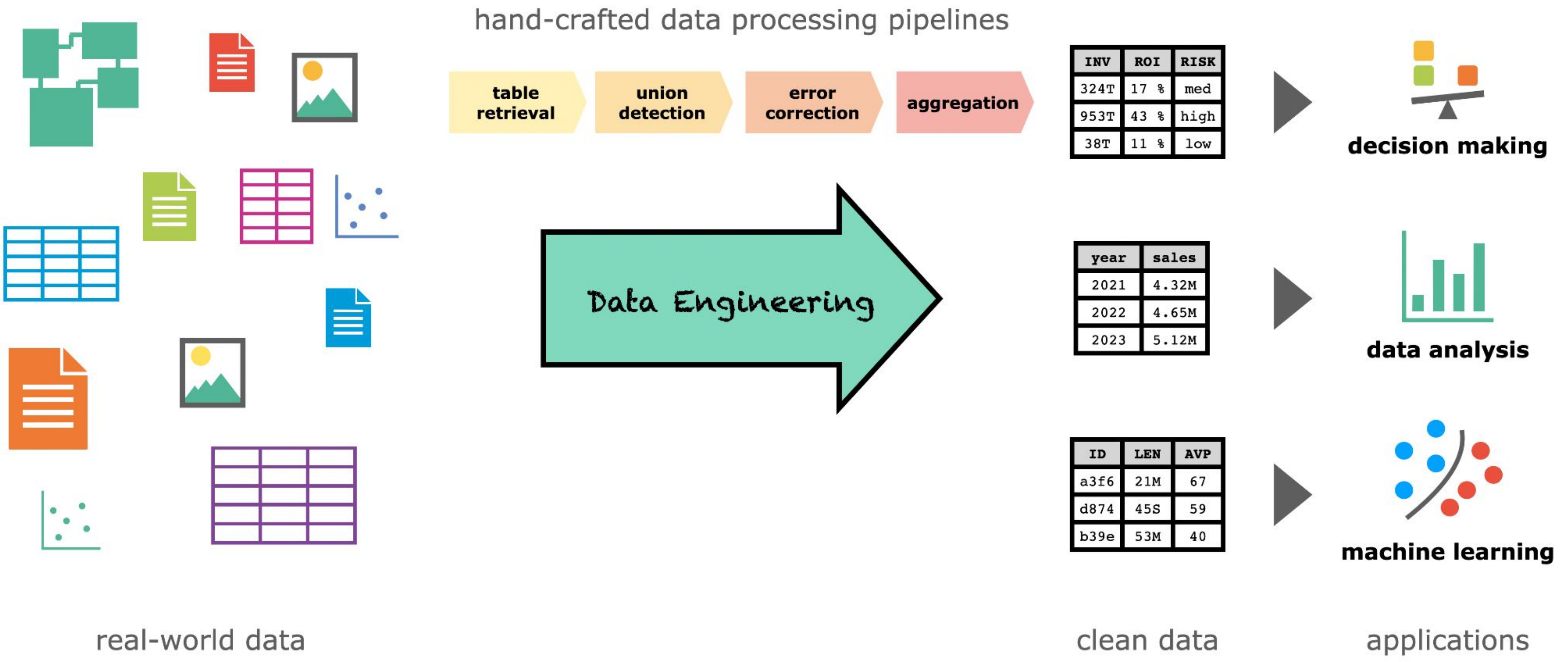
real-world data

clean data

applications

Illustration adapted from https://medium.com/@analystsam007/prediction-model-1bc5ea113231.

Data Engineering Bridges This Gap



hand-crafted data processing pipelines

table retrieval → **union detection** → **error correction** → **aggregation**

Data Engineering

real-world data

clean data

INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

year	sales
2021	4.32M
2022	4.65M
2023	5.12M

ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

applications

decision making

data analysis

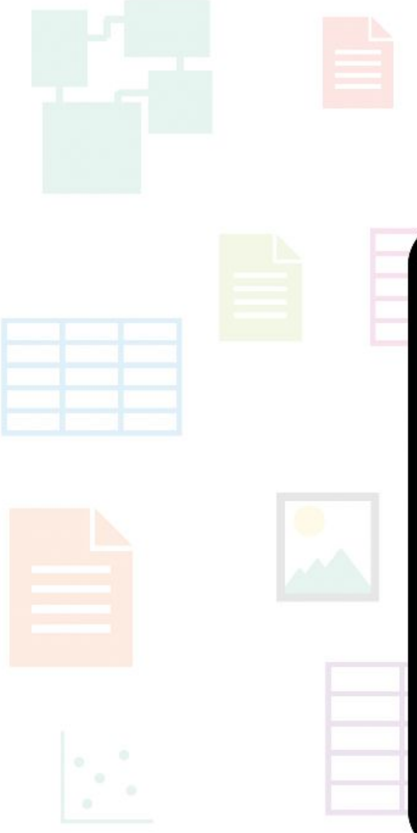
machine learning

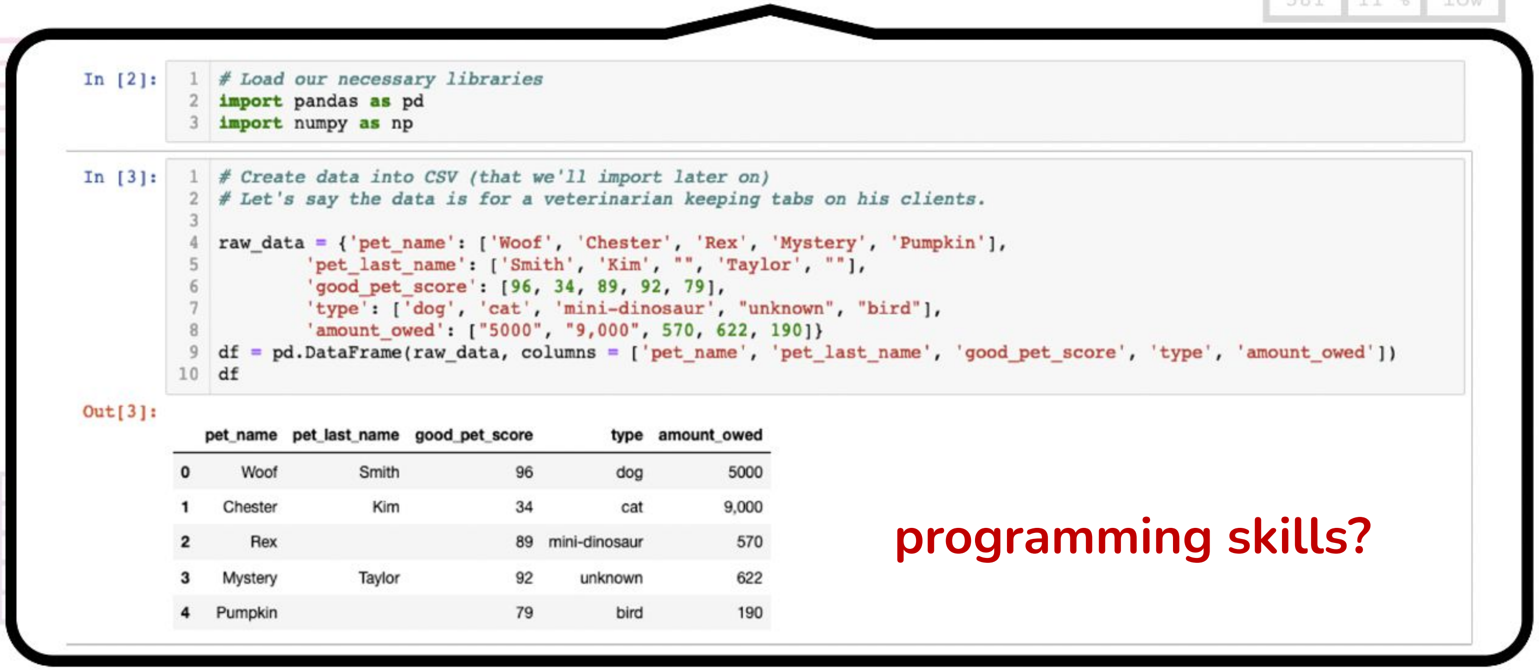
Data Engineering Means Python Code

NHR4CES NHR for Computational Engineering Science

hand-crafted data processing pipelines

table retrieval → **union detection** → **error correction** → **aggregation**





```
In [2]:  # Load our necessary libraries
         import pandas as pd
         import numpy as np
```

```
In [3]:  # Create data into CSV (that we'll import later on)
         # Let's say the data is for a veterinarian keeping tabs on his clients.

         raw_data = {'pet_name': ['Woof', 'Chester', 'Rex', 'Mystery', 'Pumpkin'],
                 'pet_last_name': ['Smith', 'Kim', "", 'Taylor', ""],
                 'good_pet_score': [96, 34, 89, 92, 79],
                 'type': ['dog', 'cat', 'mini-dinosaur', "unknown", "bird"],
                 'amount_owed': ["5000", "9,000", 570, 622, 190]}
         df = pd.DataFrame(raw_data, columns = ['pet_name', 'pet_last_name', 'good_pet_score', 'type', 'amount_owed'])
         df
```

Out[3]:

	pet_name	pet_last_name	good_pet_score	type	amount_owed
0	Woof	Smith	96	dog	5000
1	Chester	Kim	34	cat	9,000
2	Rex		89	mini-dinosaur	570
3	Mystery	Taylor	92	unknown	622
4	Pumpkin		79	bird	190

programming skills?

INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

decision making

data analysis

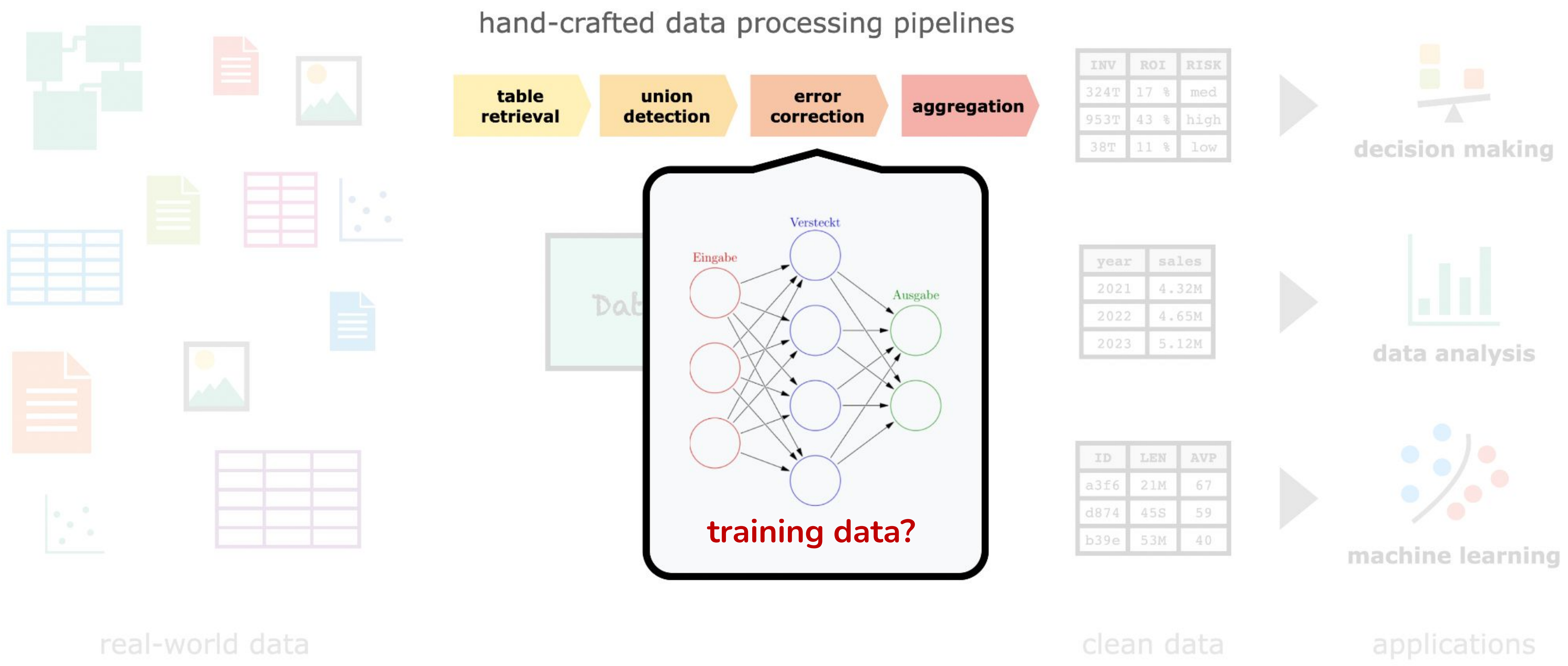
machine learning

real-world data

clean data

applications

Data Engineering Means ML Models



hand-crafted data processing pipelines

table retrieval → **union detection** → **error correction** → **aggregation**

INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

year	sales
2021	4.32M
2022	4.65M
2023	5.12M

ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

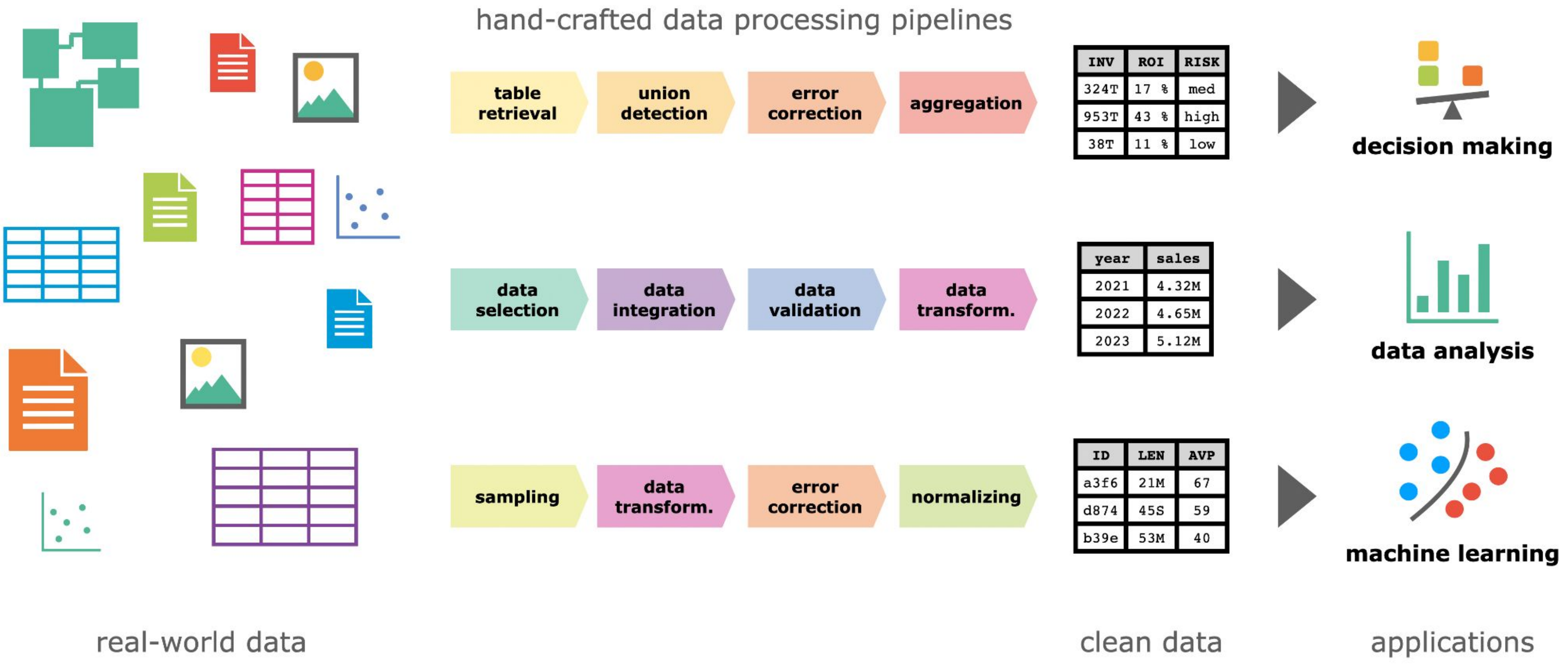
training data?

real-world data

clean data

applications

New Task or Data: Start Over



hand-crafted data processing pipelines

table retrieval → **union detection** → **error correction** → **aggregation**

INV	ROI	RISK
324T	17 %	med
953T	43 %	high
38T	11 %	low

decision making

data selection → **data integration** → **data validation** → **data transform.**

year	sales
2021	4.32M
2022	4.65M
2023	5.12M

data analysis

sampling → **data transform.** → **error correction** → **normalizing**

ID	LEN	AVP
a3f6	21M	67
d874	45S	59
b39e	53M	40

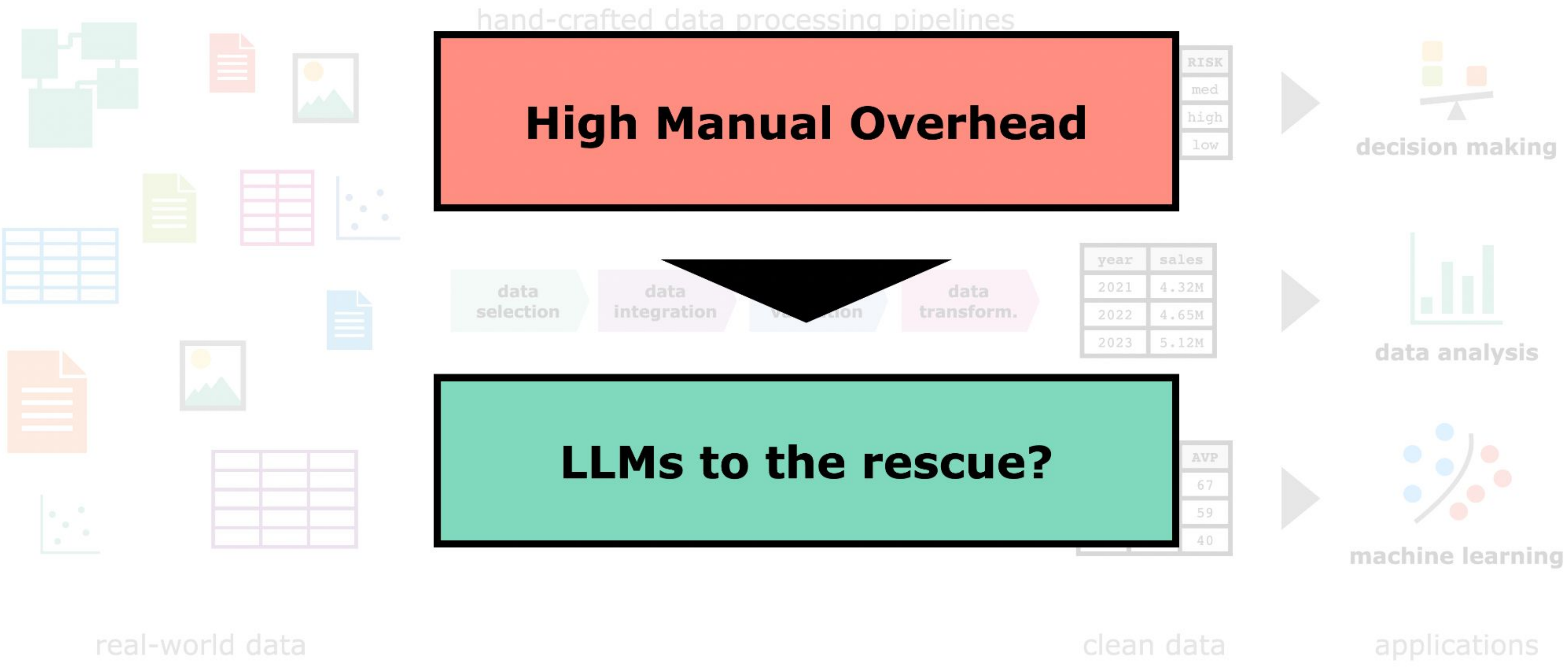
machine learning

real-world data

clean data

applications

Data Engineering Has High Overheads

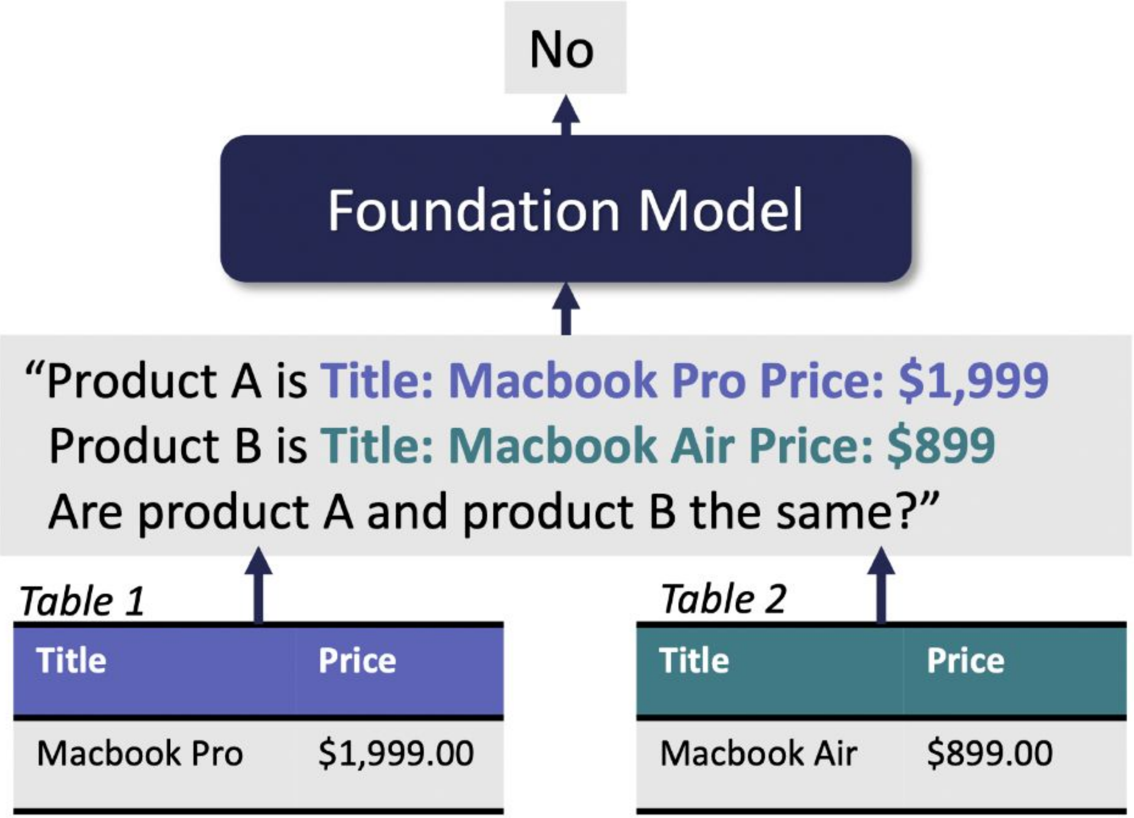


High Manual Overhead

LLMs to the rescue?

LLMs Can **Automate Many Tasks**

- entity matching
- error detection
- value imputation
- schema matching
- ...



No

Foundation Model

"Product A is **Title: Macbook Pro Price: $1,999**
Product B is **Title: Macbook Air Price: $899**
Are product A and product B the same?"

Table 1

Title	Price
Macbook Pro	$1,999.00

Table 2

Title	Price
Macbook Air	$899.00

Illustration adapted from Narayan et al. (2022).

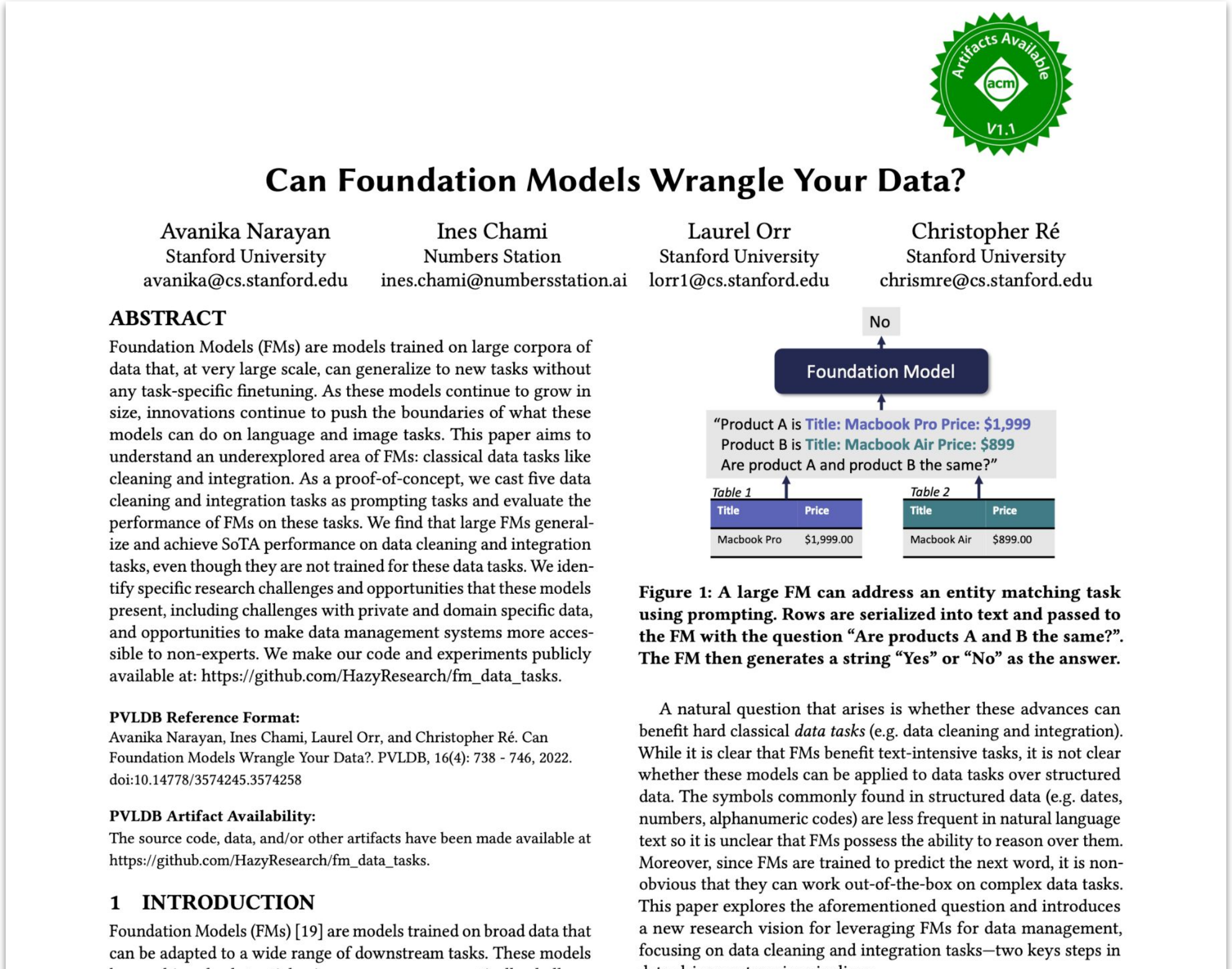
no model training required

anyone can do it

easily adapted to new tasks & data

Narayan et al. (2022) Can Foundation Models Wrangle Your Data?

Seminal Work: Narayan et al. (2022)



Can Foundation Models Wrangle Your Data?

Avanika Narayan
Stanford University
avanika@cs.stanford.edu

Ines Chami
Numbers Station
ines.chami@numbersstation.ai

Laurel Orr
Stanford University
lorr1@cs.stanford.edu

Christopher Ré
Stanford University
chrismre@cs.stanford.edu

ABSTRACT

Foundation Models (FMs) are models trained on large corpora of data that, at very large scale, can generalize to new tasks without any task-specific finetuning. As these models continue to grow in size, innovations continue to push the boundaries of what these models can do on language and image tasks. This paper aims to understand an underexplored area of FMs: classical data tasks like cleaning and integration. As a proof-of-concept, we cast five data cleaning and integration tasks as prompting tasks and evaluate the performance of FMs on these tasks. We find that large FMs generalize and achieve SoTA performance on data cleaning and integration tasks, even though they are not trained for these data tasks. We identify specific research challenges and opportunities that these models present, including challenges with private and domain specific data, and opportunities to make data management systems more accessible to non-experts. We make our code and experiments publicly available at: https://github.com/HazyResearch/fm_data_tasks.

PVLDB Reference Format:
Avanika Narayan, Ines Chami, Laurel Orr, and Christopher Ré. Can Foundation Models Wrangle Your Data?. PVLDB, 16(4): 738 - 746, 2022.
doi:10.14778/3574245.3574258

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/HazyResearch/fm_data_tasks.

1 INTRODUCTION

Foundation Models (FMs) [19] are models trained on broad data that can be adapted to a wide range of downstream tasks. These models

No
Foundation Model
"Product A is Title: Macbook Pro Price: $1,999
Product B is Title: Macbook Air Price: $899
Are product A and product B the same?"
Table 1
Title | Price
Macbook Pro | $1,999.00
Table 2
Title | Price
Macbook Air | $899.00

Figure 1: A large FM can address an entity matching task using prompting. Rows are serialized into text and passed to the FM with the question "Are products A and B the same?". The FM then generates a string "Yes" or "No" as the answer.

A natural question that arises is whether these advances can benefit hard classical *data tasks* (e.g. data cleaning and integration). While it is clear that FMs benefit text-intensive tasks, it is not clear whether these models can be applied to data tasks over structured data. The symbols commonly found in structured data (e.g. dates, numbers, alphanumeric codes) are less frequent in natural language text so it is unclear that FMs possess the ability to reason over them. Moreover, since FMs are trained to predict the next word, it is non-obvious that they can work out-of-the-box on complex data tasks. This paper explores the aforementioned question and introduces a new research vision for leveraging FMs for data management, focusing on data cleaning and integration tasks—two keys steps in

LLMs for Data Engineering: **What works?**

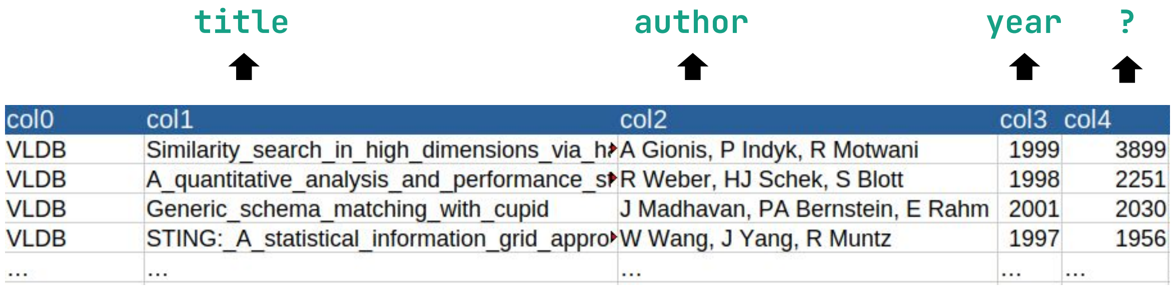
Overview over Data Engineering Tasks

- Semantic Column Type Annotation
- Data Imputation
- Error Detection
- Table Question Answering
- Text-to-SQL
- Data Transformation

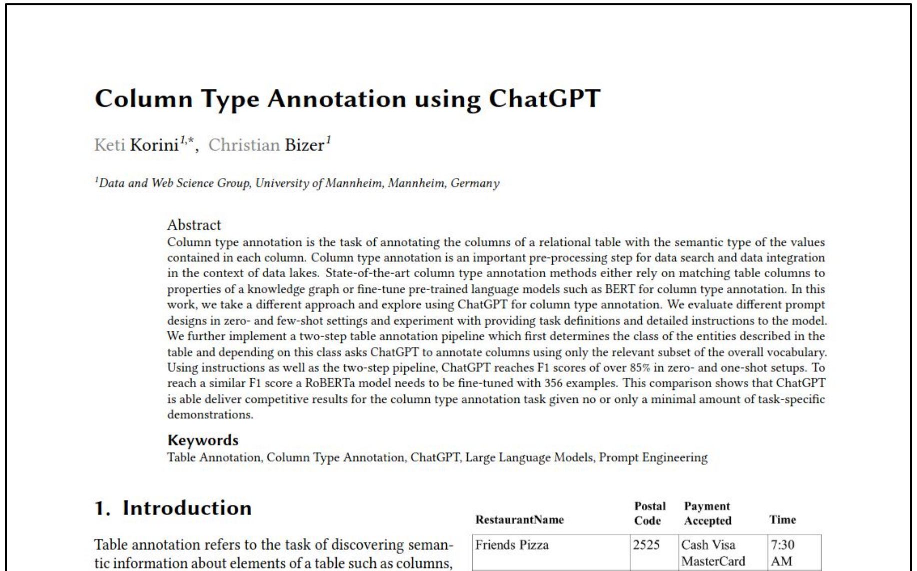
Suggesting column types

Column Type Annotation

= annotating table columns with *semantic types* from an ontology


title
author
year
?

col0	col1	col2	col3	col4
VLDB	Similarity_search_in_high_dimensions_via_h	A Gionis, P Indyk, R Motwani	1999	3899
VLDB	A_quantitative_analysis_and_performance_s	R Weber, HJ Schek, S Blott	1998	2251
VLDB	Generic_schema_matching_with_cupid	J Madhavan, PA Bernstein, E Rahm	2001	2030
VLDB	STING:_A_statistical_information_grid_appro	W Wang, J Yang, R Muntz	1997	1956
...	...	...	...	...



Column Type Annotation using ChatGPT

Keti Korini[1,*], Christian Bizer[1]

[1]Data and Web Science Group, University of Mannheim, Mannheim, Germany

Abstract

Column type annotation is the task of annotating the columns of a relational table with the semantic type of the values contained in each column. Column type annotation is an important pre-processing step for data search and data integration in the context of data lakes. State-of-the-art column type annotation methods either rely on matching table columns to properties of a knowledge graph or fine-tune pre-trained language models such as BERT for column type annotation. In this work, we take a different approach and explore using ChatGPT for column type annotation. We evaluate different prompt designs in zero- and few-shot settings and experiment with providing task definitions and detailed instructions to the model. We further implement a two-step table annotation pipeline which first determines the class of the entities described in the table and depending on this class asks ChatGPT to annotate columns using only the relevant subset of the overall vocabulary. Using instructions as well as the two-step pipeline, ChatGPT reaches F1 scores of over 85% in zero- and one-shot setups. To reach a similar F1 score a RoBERTa model needs to be fine-tuned with 356 examples. This comparison shows that ChatGPT is able deliver competitive results for the column type annotation task given no or only a minimal amount of task-specific demonstrations.

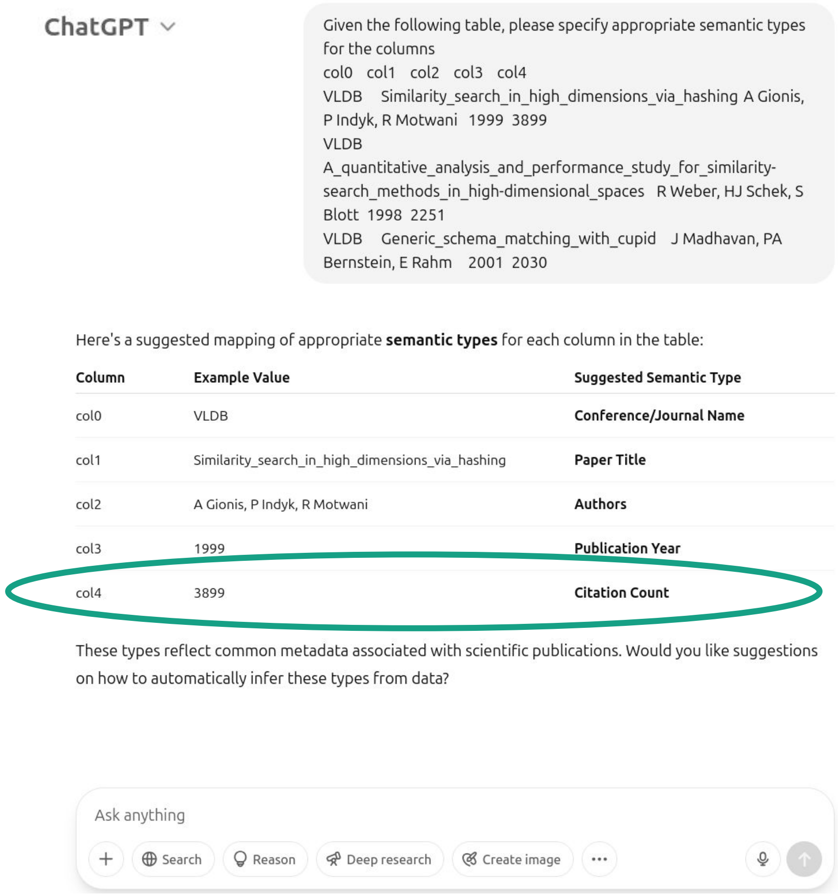
Keywords

Table Annotation, Column Type Annotation, ChatGPT, Large Language Models, Prompt Engineering

1. Introduction

Table annotation refers to the task of discovering semantic information about elements of a table such as columns,

RestaurantName	Postal Code	Payment Accepted	Time
Friends Pizza	2525	Cash Visa MasterCard	7:30 AM



Given the following table, please specify appropriate semantic types for the columns
col0 col1 col2 col3 col4
VLDB Similarity_search_in_high_dimensions_via_hashing A Gionis, P Indyk, R Motwani 1999 3899
VLDB A_quantitative_analysis_and_performance_study_for_similarity-search_methods_in_high-dimensional_spaces R Weber, HJ Schek, S Blott 1998 2251
VLDB Generic_schema_matching_with_cupid J Madhavan, PA Bernstein, E Rahm 2001 2030

Here's a suggested mapping of appropriate **semantic types** for each column in the table:

Column	Example Value	Suggested Semantic Type
col0	VLDB	**Conference/Journal Name**
col1	Similarity_search_in_high_dimensions_via_hashing	**Paper Title**
col2	A Gionis, P Indyk, R Motwani	**Authors**
col3	1999	**Publication Year**
col4	3899	**Citation Count**

These types reflect common metadata associated with scientific publications. Would you like suggestions on how to automatically infer these types from data?

Column Type Annotation - with examples

LLM Prompt for Column Type Annotation

Instruction

user: Predict the column types of the following tables. Provide just the column types as a JSON list without any introduction or explanation. Column types are: [“account type”, “clearing date”, …]

One-shot Example

user:

```
STAS
MANDT,STLTY,STLNR,STLAL,STLKN,STASZ,DATUV,TECHV,AENNR,...
1,F,47294573,0,8,21,20210304,,394729478,,20210301,...
1,F,93618467,0,9,14,20170121,,141834612,,20170120,...
1,F,34188479,0,21,34,20191123,,560289473,,20191119,...
```

assist: ["client", "bom category", "bill of material", ...]

Table to annotate

user:

```
BSEG
MANDT,BUKRS,BELNR,GJAHR,BUZEI,BUZID,AUGDT,AUGCP,AUGBL,...
1,D054,5930568205,2013,5,H,20140503,20140501,9836283674,...
1,D054,5829473293,2021,7,H,20221123,20221119,3485949047,...
1,D037,3168347239,2012,43,L,20120913,20120831,7554950694,...
```

Data Imputation & Error Detection

Filling **missing values** and detecting, if a cell **contains an error**

Data Imputation:
Row_ID: 001,
Country: England,
Capital: ?

"London"

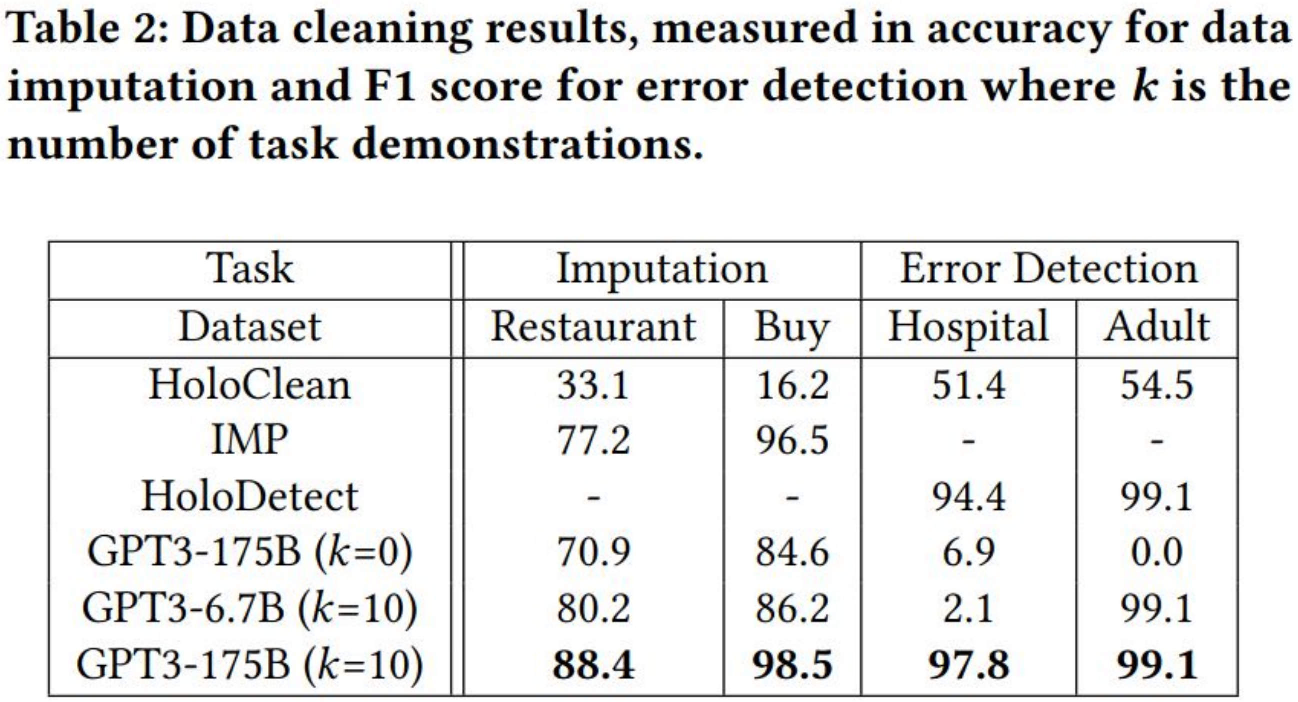


Table 2: Data cleaning results, measured in accuracy for data imputation and F1 score for error detection where *k* is the number of task demonstrations.

Task	Imputation		Error Detection	
Dataset	Restaurant	Buy	Hospital	Adult
HoloClean	33.1	16.2	51.4	54.5
IMP	77.2	96.5	-	-
HoloDetect	-	-	94.4	99.1
GPT3-175B (k=0)	70.9	84.6	6.9	0.0
GPT3-6.7B (k=10)	80.2	86.2	2.1	99.1
GPT3-175B (k=10)	**88.4**	**98.5**	**97.8**	**99.1**

Error Detection:
Is there an error in Country?

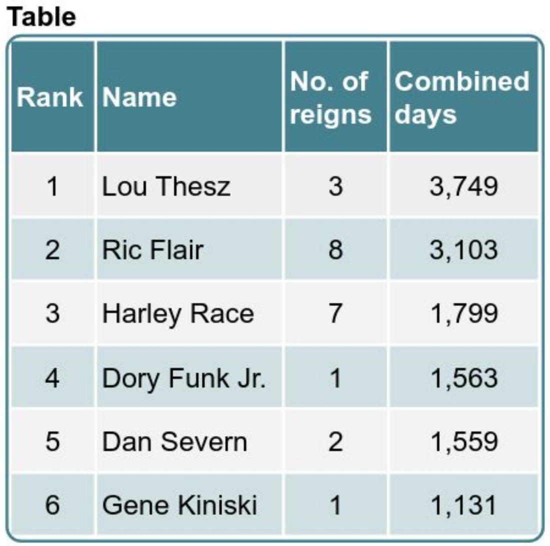
Row_ID: 004,
Country: USA,
City: Kyoto

yes / no

Table from Can Foundation Models Wrangle Your Data? Narayan et al. VLDB 2022

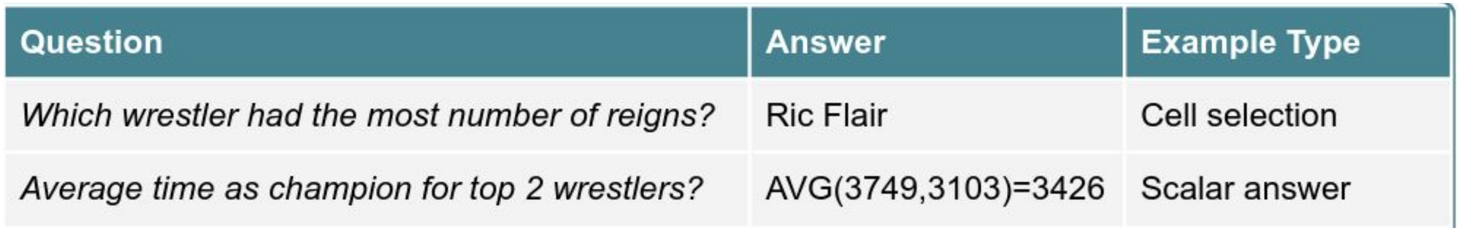
Table Question Answering

Answering questions over tabular data



Table

Rank	Name	No. of reigns	Combined days
1	Lou Thesz	3	3,749
2	Ric Flair	8	3,103
3	Harley Race	7	1,799
4	Dory Funk Jr.	1	1,563
5	Dan Severn	2	1,559
6	Gene Kiniski	1	1,131

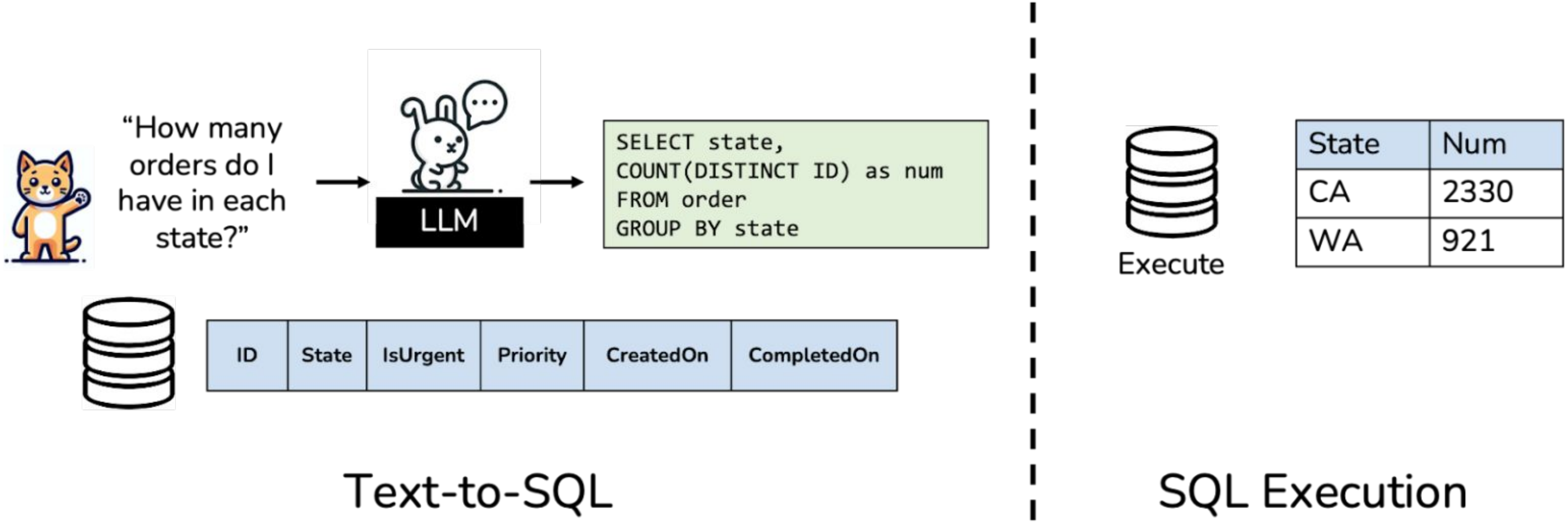


Question	Answer	Example Type
Which wrestler had the most number of reigns?	Ric Flair	Cell selection
Average time as champion for top 2 wrestlers?	AVG(3749,3103)=3426	Scalar answer

Image from [TAPAS: Weakly Supervised Table Parsing via Pre-training. Herzig et al. ACL 2020]

Text-to-SQL

Transforming natural language questions to SQL Queries



```
SELECT state,
COUNT(DISTINCT ID) as num
FROM order
GROUP BY state
```

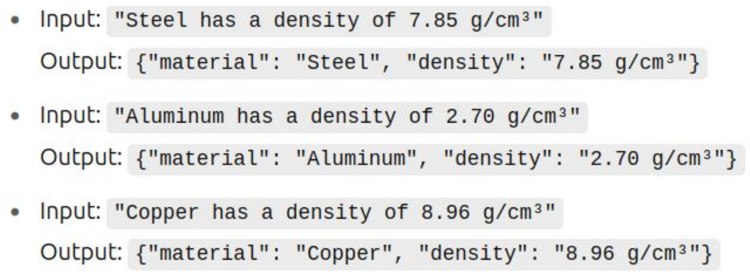
ID	State	IsUrgent	Priority	CreatedOn	CompletedOn

State	Num
CA	2330
WA	921

Image from https://www.numbersstation.ai/text-to-sql-that-isnt/

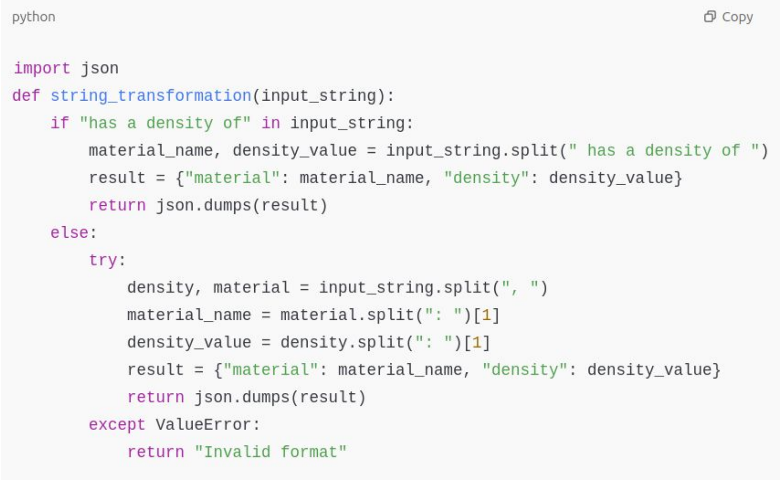
Wrangling data using Code Generation

Input examples

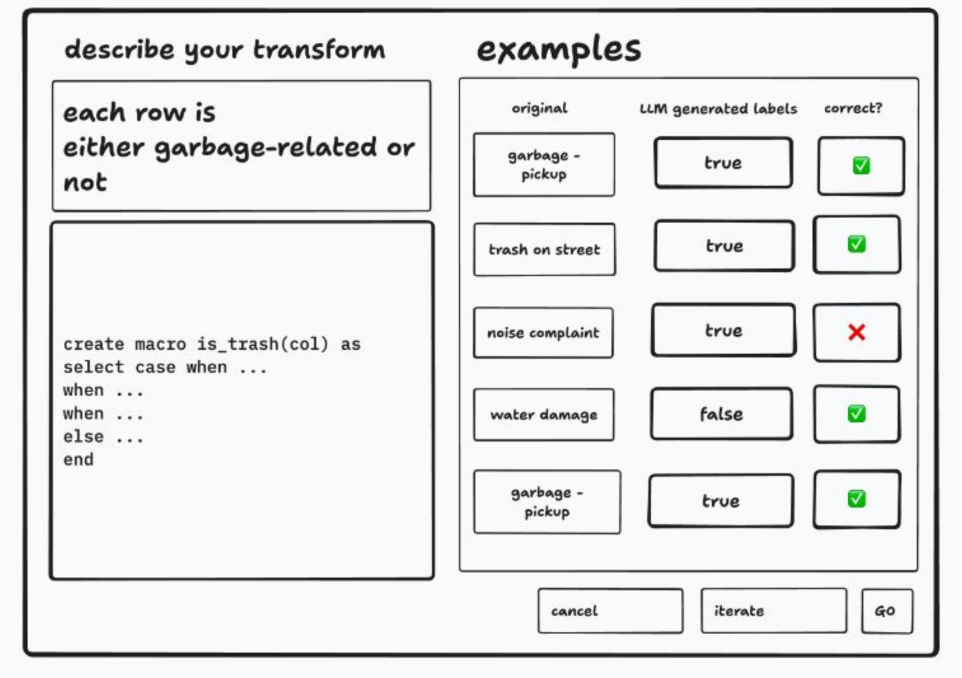


- Input: `"Steel has a density of 7.85 g/cm³"`
  Output: `{"material": "Steel", "density": "7.85 g/cm³"}`
- Input: `"Aluminum has a density of 2.70 g/cm³"`
  Output: `{"material": "Aluminum", "density": "2.70 g/cm³"}`
- Input: `"Copper has a density of 8.96 g/cm³"`
  Output: `{"material": "Copper", "density": "8.96 g/cm³"}`

Generated code:



```python
import json
def string_transformation(input_string):
    if "has a density of" in input_string:
        material_name, density_value = input_string.split(" has a density of ")
        result = {"material": material_name, "density": density_value}
        return json.dumps(result)
    else:
        try:
            density, material = input_string.split(", ")
            material_name = material.split(": ")[1]
            density_value = density.split(": ")[1]
            result = {"material": material_name, "density": density_value}
            return json.dumps(result)
        except ValueError:
            return "Invalid format"
```



each row is either garbage-related or not

```
create macro is_trash(col) as
select case when ...
when ...
when ...
else ...
end
```

original	LLM generated labels	correct?
garbage - pickup	true	✓
trash on street	true	✓
noise complaint	true	✗
water damage	false	✓
garbage - pickup	true	✓

Figure 4: An illustration on UX for human-in-the-loop code generation.

Figure 4 from [Towards Efficient Data Wrangling with LLMs using Code Generation. Li and Döhmen. DEEM@SIGMOD 2024](#)

Prompt engineering: Best practices

Role prompting

You are an expert Python user. I will give you a string transformation task. The task involves converting input strings to output strings.

Assuming all of the following python packages are installed: "regex", "fractions", "math", "pyproj", "BeautifulSoup", "geopy", "ummalqura", "mgrs", "pytz", "datetime", "calendar", "roman"], you can ```import``` them and please put all imports in the beginning of the generated python program. You can also import other packages if the listed ones are not working, but try using listed packages first.

Specify instructions & constraints

Please try to understand the intentions of the input-output examples while generating the function.

Please return only the python function and nothing else. Include any imports. Do not provide any comments.

Here is a very simple example:

Describe desired output in detail

Input: "john.doe@example.com"
Output: "Yes"

Input: "jane_doe@example.com"
Output: "Yes"

Input: "example.com"
Output: "No"

Generated function string:

Give examples

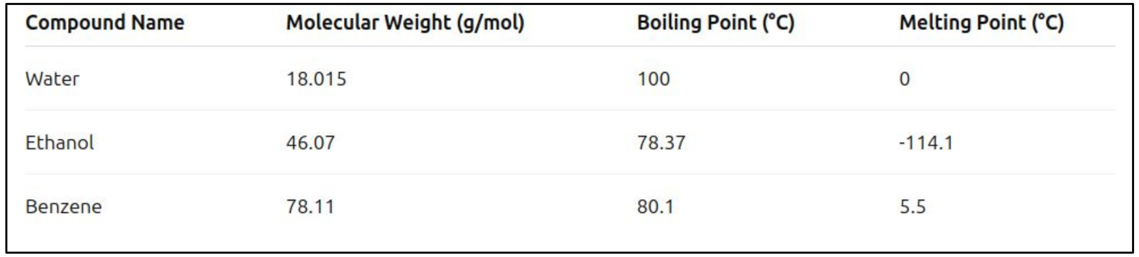
~~~text
```pythonimport re\ndef string_transformation(input_string):\n\tpattern = r'^[\w\.-]+@[\w\.-]+\.\w+$'\n\tif re.match(pattern, input_string):\n\t\treturn
"Yes"\n\telse: return\n\t\t"No"```\n
~~~

Now it's your turn. After generating the function string, please add escape symbol tab '\t' to format the code properly with indents like shown in the the example.

Promt from [Towards Efficient Data Wrangling with LLMs using Code Generation. Li and Döhmen. DEEM@SIGMOD 2024](https://github.com/effyli/efficient_llm_data_wrangling/blob/main/fast_ai_wrangler/promptsTemplate.py)
https://github.com/effyli/efficient_llm_data_wrangling/blob/main/fast_ai_wrangler/promptsTemplate.py

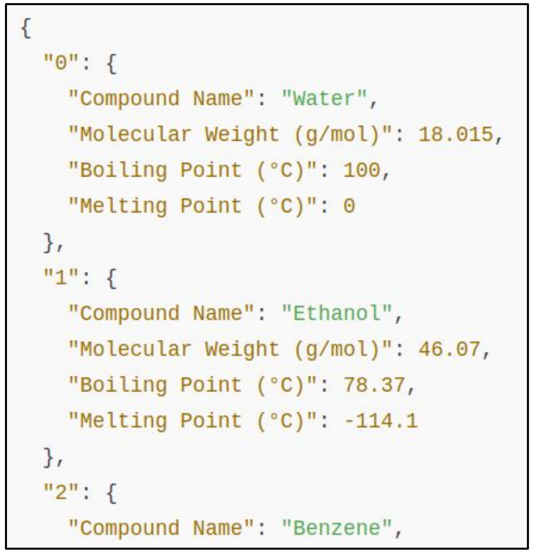
Linearization Techniques

Inputting tables into LLMs:



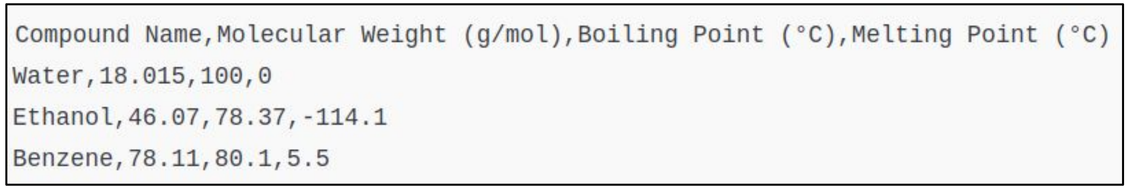
Compound Name	Molecular Weight (g/mol)	Boiling Point (°C)	Melting Point (°C)
Water	18.015	100	0
Ethanol	46.07	78.37	-114.1
Benzene	78.11	80.1	5.5

JSON:



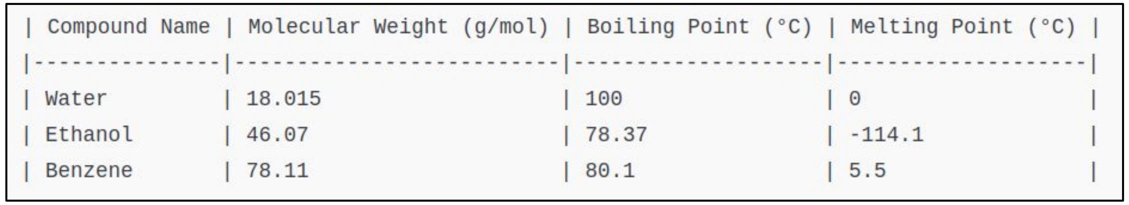
```
{
  "0": {
    "Compound Name": "Water",
    "Molecular Weight (g/mol)": 18.015,
    "Boiling Point (°C)": 100,
    "Melting Point (°C)": 0
  },
  "1": {
    "Compound Name": "Ethanol",
    "Molecular Weight (g/mol)": 46.07,
    "Boiling Point (°C)": 78.37,
    "Melting Point (°C)": -114.1
  },
  "2": {
    "Compound Name": "Benzene",
```

CSV:



```
Compound Name,Molecular Weight (g/mol),Boiling Point (°C),Melting Point (°C)
Water,18.015,100,0
Ethanol,46.07,78.37,-114.1
Benzene,78.11,80.1,5.5
```

Markdown:



```
| Compound Name | Molecular Weight (g/mol) | Boiling Point (°C) | Melting Point (°C) |
|---------------|--------------------------|--------------------|--------------------|
| Water         | 18.015                   | 100                | 0                  |
| Ethanol       | 46.07                    | 78.37              | -114.1             |
| Benzene       | 78.11                    | 80.1               | 5.5                |
```

Text:

Compound Name is Water. Molecular Weight is 18.015 g/mol. Boiling Point is 100 °C. Melting Point is 0 °C.
Compound Name is Ethanol. Molecular Weight is 46.07 g/mol. Boiling Point is 78.37 °C. Melting Point is -114.1 °C.
Compound Name is Benzene. Molecular Weight is 78.11 g/mol. Boiling Point is 80.1 °C. Melting Point is 5.5 °C.

LLMs specifically for Tables

- **Table-GPT**
  [Li et al., 2024]
- **TableGPT**
  [Zha et al., 2023]
- **TableLlama**
  [Zhang et al., 2024]

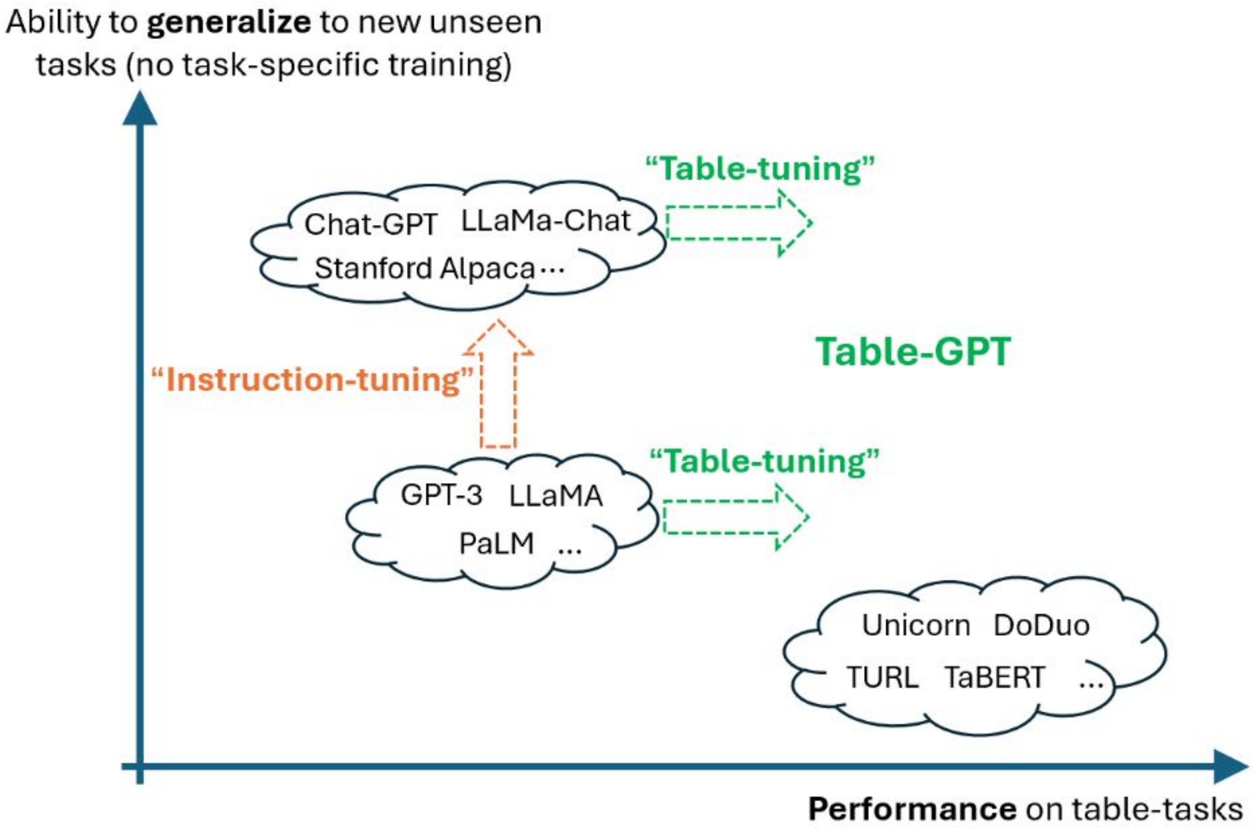
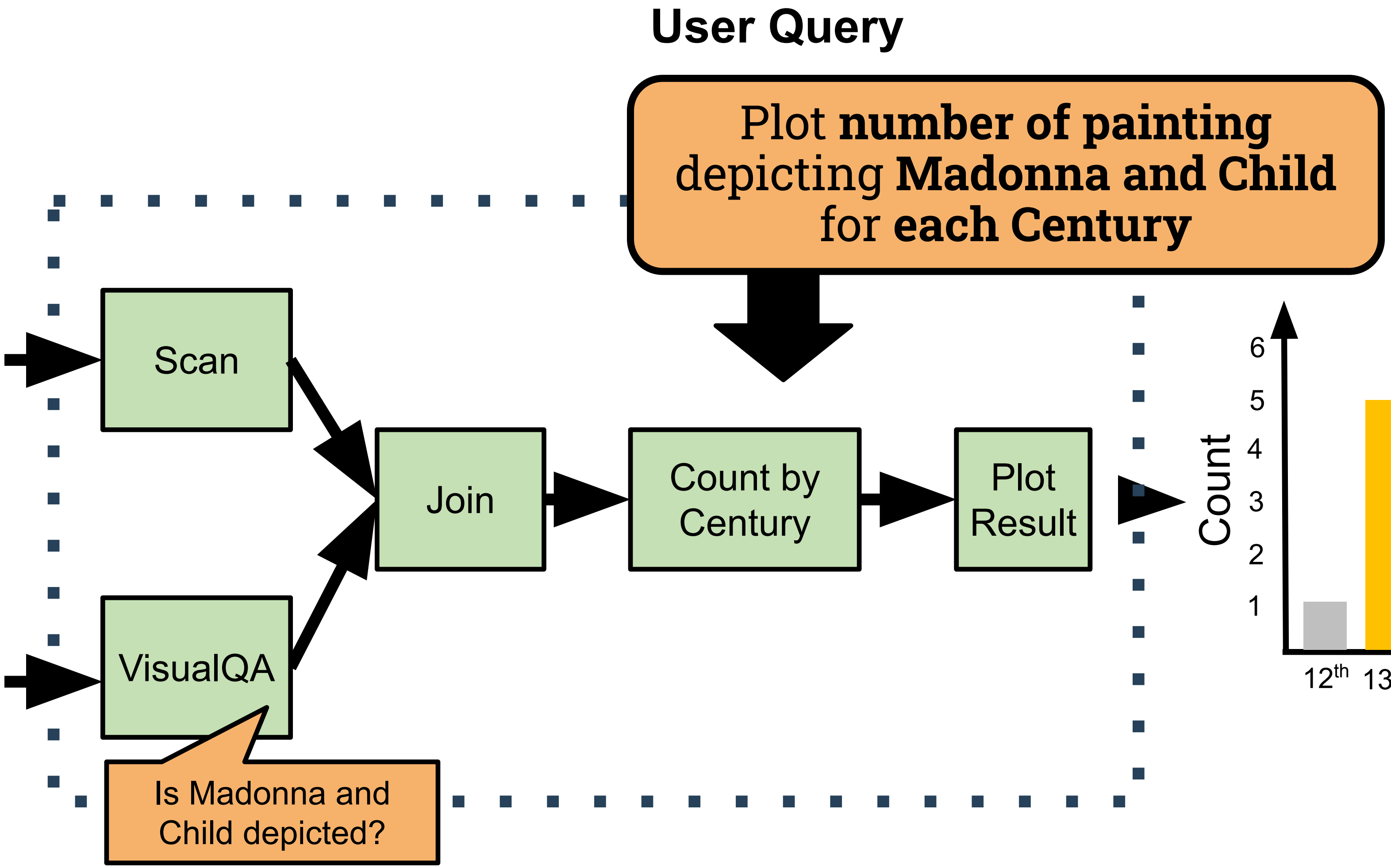


Image from [Table-GPT: Table Fine-tuned GPT for Diverse Table Tasks. Li et al. SIGMOD 2024]()

Multi-Modal Data Analytics [CAESURA]

metadata.csv

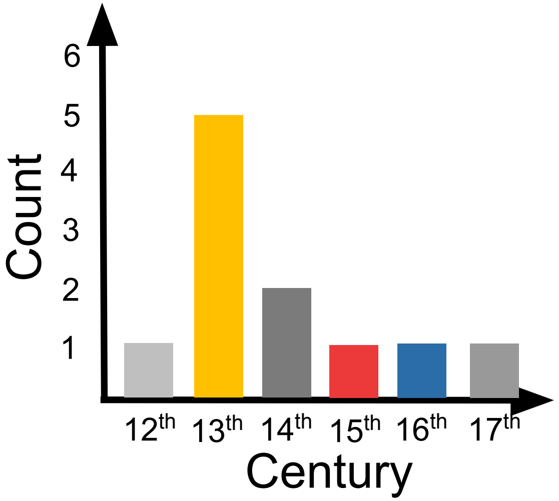
name, year, ..., img_path
Madonna, 1889, ..., i1.png
Irises, 1480, ..., i2.png
...
Scream, 1893, ..., in.png



User Query

Plot **number of painting** depicting **Madonna and Child** for **each Century**

Is Madonna and Child depicted?



From CAESURA: Language Models as Multi-Modal Query Planners. Urban and Binnig. CIDR 2024

LLMs for Data Engineering - What works?

🚀 Lots of **research** to support users with data engineering

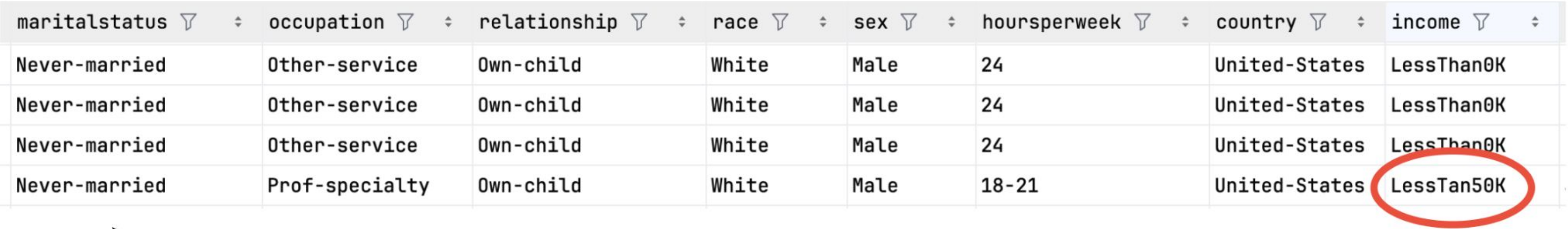
🤖 LLMs for data engineering seem to be a **promising avenue**

🛠️ Is everything **solved** already?

LLMs for Data Engineering:
What doesn’t work (yet)?

LLM Research Based On Web Tables

Adult Dataset for Error Correction



maritalstatus	occupation	relationship	race	sex	hoursperweek	country	income
Never-married	Other-service	Own-child	White	Male	24	United-States	LessThan0K
Never-married	Other-service	Own-child	White	Male	24	United-States	LessThan0K
Never-married	Other-service	Own-child	White	Male	24	United-States	LessThan0K
Never-married	Prof-specialty	Own-child	White	Male	18-21	United-States	LessTan50K

Simple table

Contrived examples

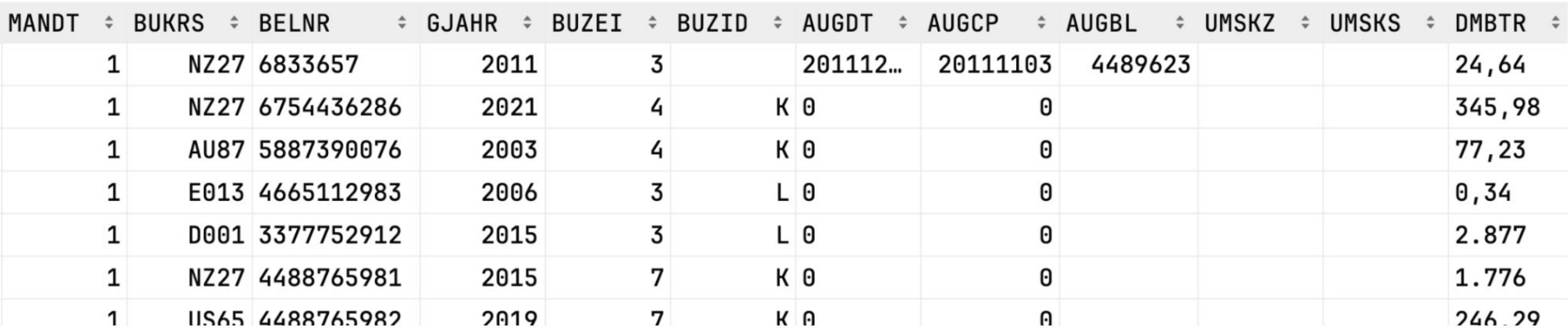
Real-world Data Looks Different

Example: Enterprise data from SAP

Schemas are not descriptive

Tables are substantially larger
(hundreds of columns and millions of rows)

SAP BSEG (Accounting Document Segment) Table with 425 Columns



MANDT	BUKRS	BELNR	GJAHR	BUZEI	BUZID	AUGDT	AUGCP	AUGBL	UMSKZ	UMSKS	DMBTR
1	NZ27	6833657	2011	3		201112…	20111103	4489623			24,64
1	NZ27	6754436286	2021	4	K	0	0				345,98
1	AU87	5887390076	2003	4	K	0	0				77,23
1	E013	4665112983	2006	3	L	0	0				0,34
1	D001	3377752912	2015	3	L	0	0				2.877
1	NZ27	4488765981	2015	7	K	0	0				1.776
1	US65	4488765982	2019	7	K	0	0				246,29

Values are not self-expressive

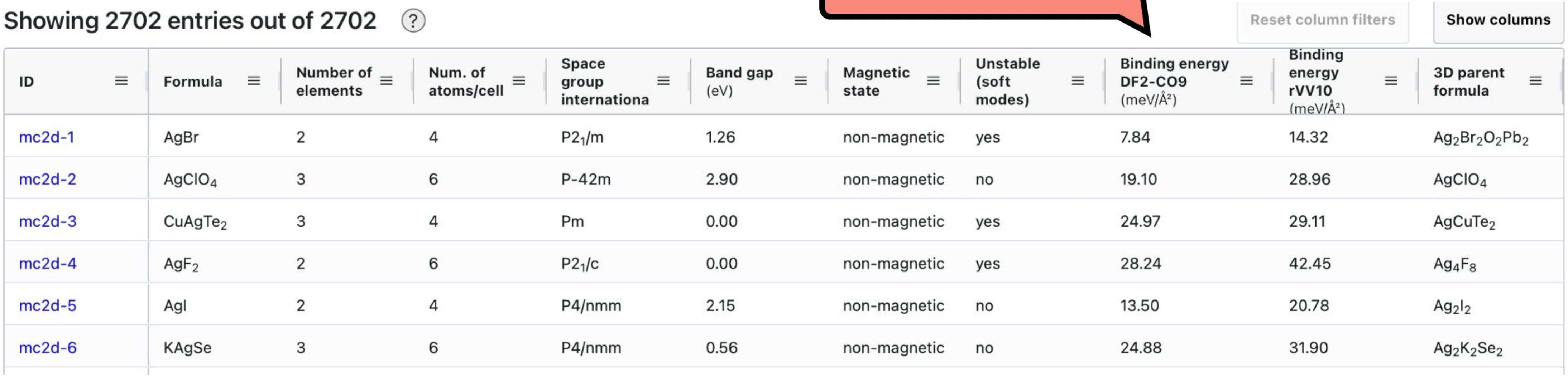
Tables are highly sparse
(43% of cells are empty)

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

Real-world Data Looks Different

Example: Research data from MaterialsCloud

Complex Units



Showing 2702 entries out of 2702 ?

ID	Formula	Number of elements	Num. of atoms/cell	Space group internationa	Band gap (eV)	Magnetic state	Unstable (soft modes)	Binding energy DF2-C09 (meV/Å²)	Binding energy rVV10 (meV/Å²)	3D parent formula
mc2d-1	AgBr	2	4	$P2_1/m$	1.26	non-magnetic	yes	7.84	14.32	$Ag_2Br_2O_2Pb_2$
mc2d-2	$AgClO_4$	3	6	P-42m	2.90	non-magnetic	no	19.10	28.96	$AgClO_4$
mc2d-3	$CuAgTe_2$	3	4	Pm	0.00	non-magnetic	yes	24.97	29.11	$AgCuTe_2$
mc2d-4	AgF_2	2	6	$P2_1/c$	0.00	non-magnetic	yes	28.24	42.45	Ag_4F_8
mc2d-5	AgI	2	4	P4/nmm	2.15	non-magnetic	no	13.50	20.78	Ag_2I_2
mc2d-6	KAgSe	3	6	P4/nmm	0.56	non-magnetic	no	24.88	31.90	$Ag_2K_2Se_2$

Numerical Data

Highly Symbolic Values

https://mc2d.materialscloud.org

How does this affect LLMs?

Real-world *enterprise data* vs. existing corpora

SAPCTA
real-world customer data
from SAP systems

GitTablesCTA
CSV files from GitHub
(Hulsebos et al. 2021)

SportsTables
web tables about sports
(Langenecker et al. 2023)

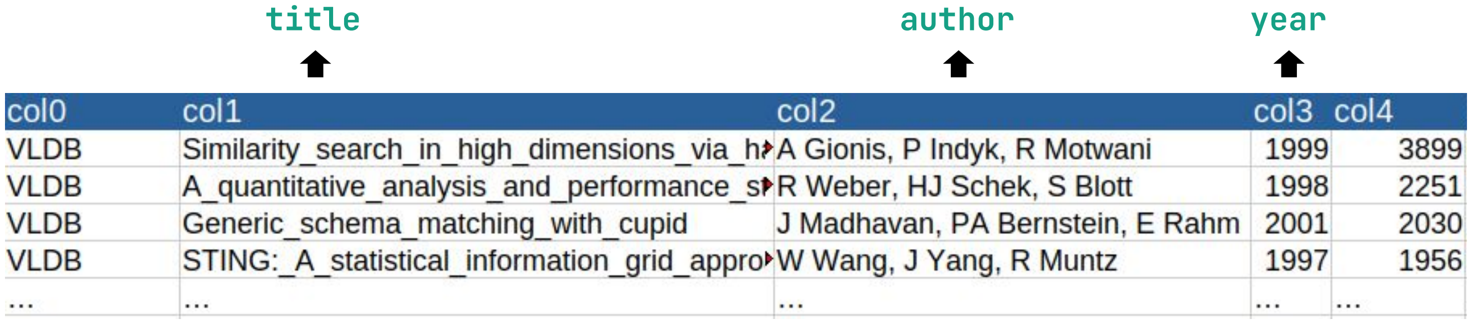
Drill-downs into data types, table size, and sparsity

Example task: Column Type Annotation (CTA)

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

Column Type Annotation (CTA)

= annotating table columns with *semantic types*


title
author
year

col0	col1	col2	col3	col4
VLDB	Similarity_search_in_high_dimensions_via_h	A Gionis, P Indyk, R Motwani	1999	3899
VLDB	A_quantitative_analysis_and_performance_s	R Weber, HJ Schek, S Blott	1998	2251
VLDB	Generic_schema_matching_with_cupid	J Madhavan, PA Bernstein, E Rahm	2001	2030
VLDB	STING:_A_statistical_information_grid_appro	W Wang, J Yang, R Muntz	1997	1956
…	…	…	…	…

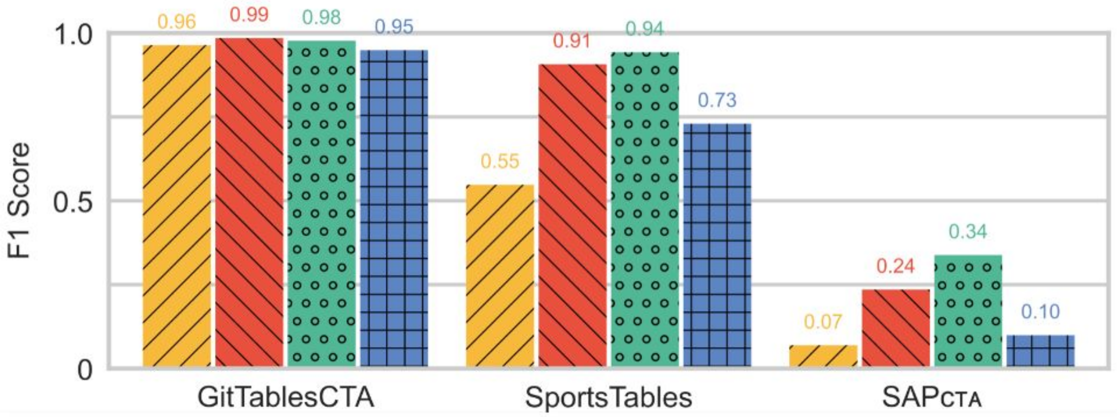
Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

Real-World Data Is Challenging

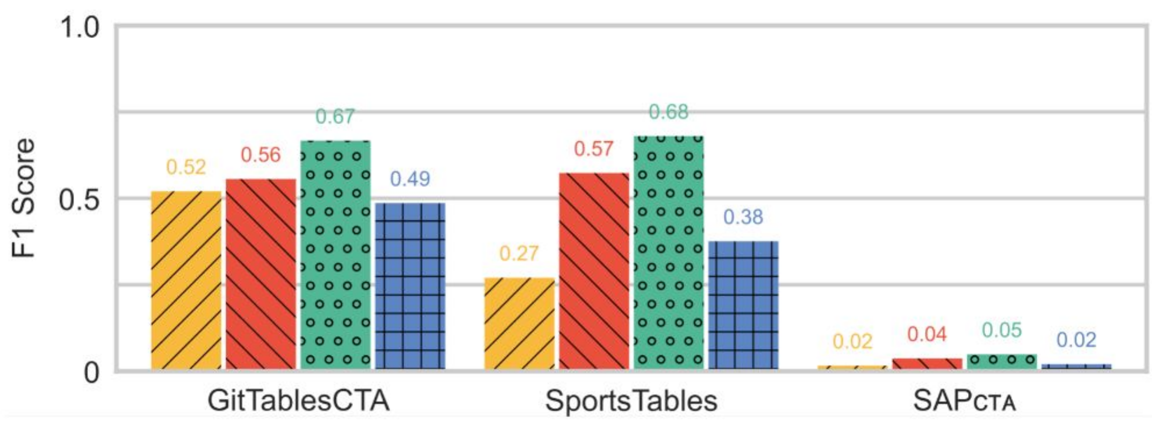
```
STAS
MANDT,STLTY,STLNR,STLAL,STLKN,STASZ,DATUV,TECHV,AENNR,…
1,F,47294573,0,8,21,20210304,,394729478,,20210301,...
1,F,93618467,0,9,14,20170121,,141834612,,20170120,...
1,F,34188479,0,21,34,20191123,,560289473,,20191119,...
```

```
1,F,47294573,0,8,21,20210304,,394729478,,20210301,...
1,F,93618467,0,9,14,20170121,,141834612,,20170120,...
1,F,34188479,0,21,34,20191123,,560289473,,20191119,...
```

With column names in prompt



Without column names in prompt



Results on real-world data are substantially worse

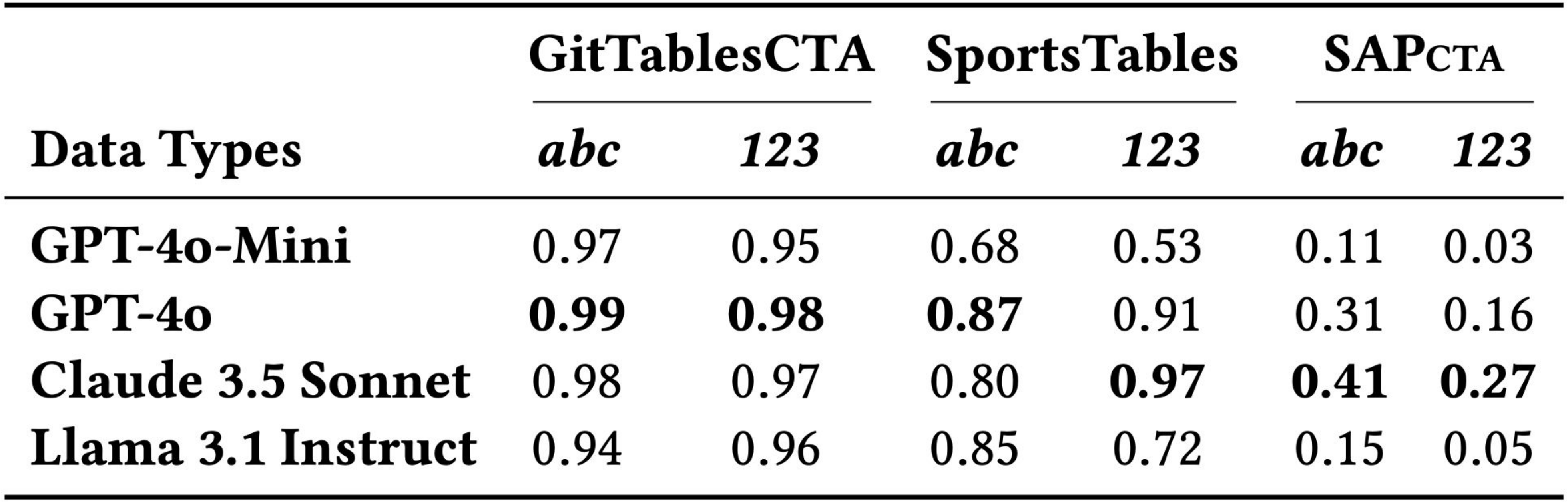
Signal comes mainly from column names

▨ GPT-4o-Mini ▨ GPT-4o ▨ Claude 3.5 Sonnet ▦ Llama 3.1 Instruct

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

What causes this performance drop?

Numerical data vs. non-numerical data



	GitTablesCTA		SportsTables		SAPCTA	
Data Types	*abc*	*123*	*abc*	*123*	*abc*	*123*
GPT-4o-Mini	0.97	0.95	0.68	0.53	0.11	0.03
GPT-4o	**0.99**	**0.98**	**0.87**	0.91	0.31	0.16
Claude 3.5 Sonnet	0.98	0.97	0.80	**0.97**	**0.41**	**0.27**
Llama 3.1 Instruct	0.94	0.96	0.85	0.72	0.15	0.05

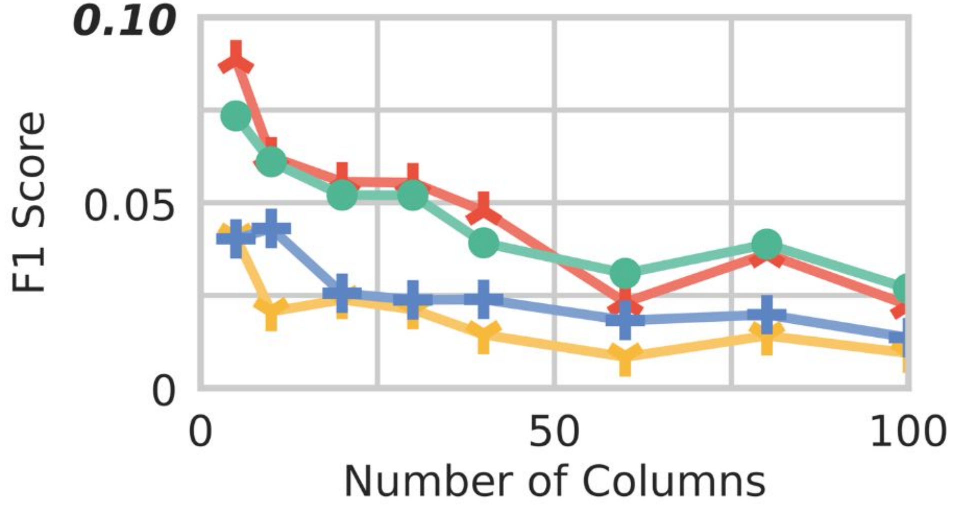
Real-world numerical data is substantially harder

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

What causes this performance drop?

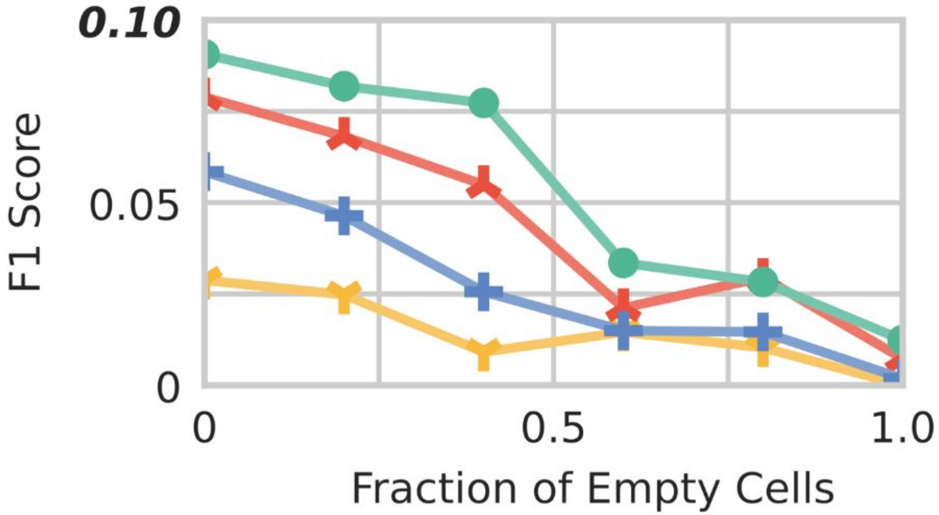
NHR4CES NHR for Computational Engineering Science

Table size



Wide tables cause problems

Sparsity



Sparsity causes problems

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

Real-world Data Is Challenging

Large performance gap between benchmarks and real-world use cases!

Causes:
- Non-descriptive schemas
- Large and wide tables
- Non-expressive values
- Sparsity

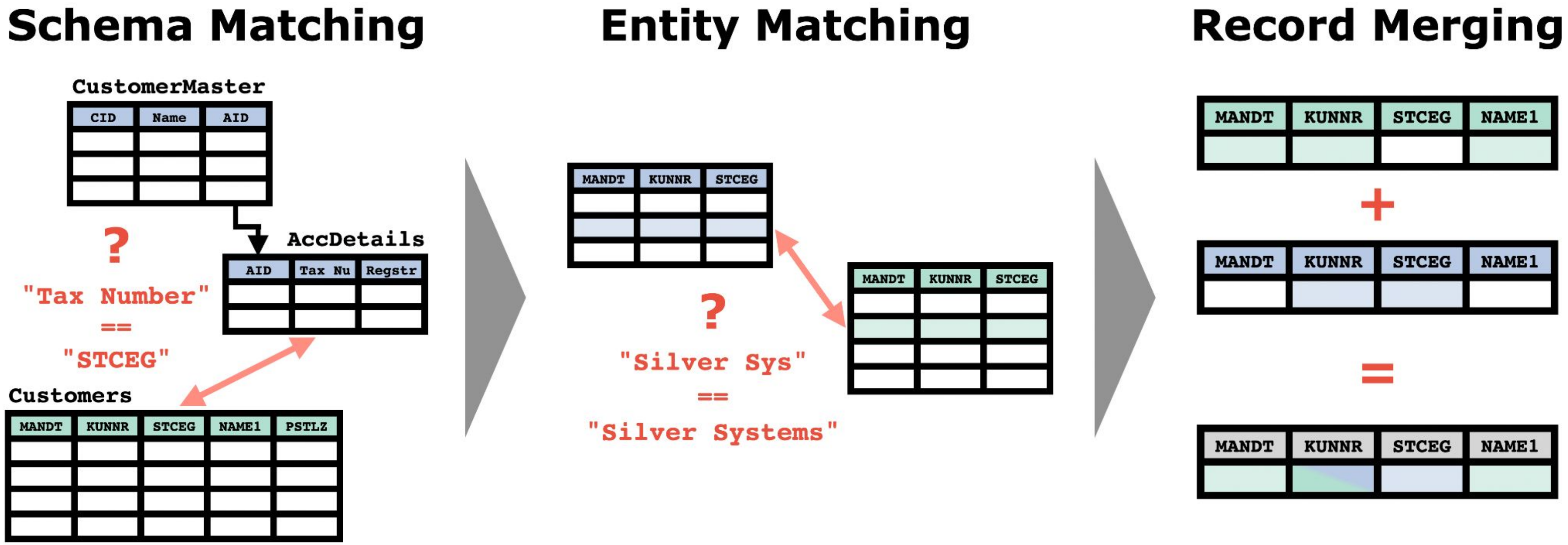
Next: real-world tasks

Bodensohn et al. (2024) LLMs for Data Engineering on Enterprise Data

Real-world Tasks Are Also Challenging

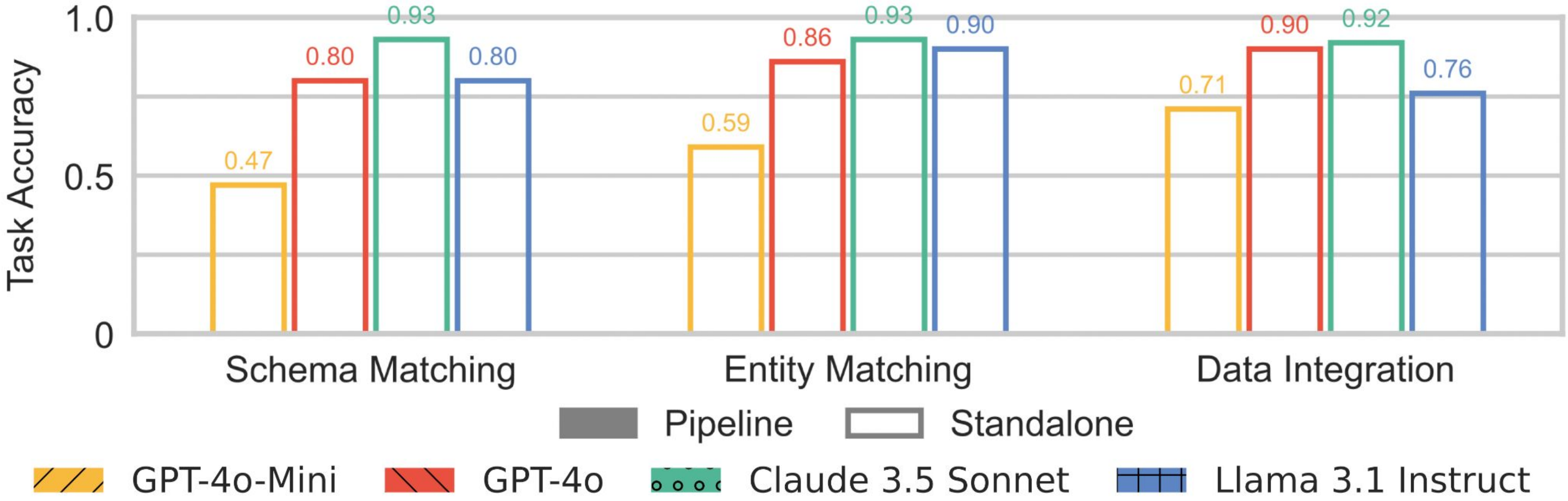
Existing research looks at isolated problems:
Column type annotation, error detection, missing value imputation, ...

Real-world problems are often compound tasks with multiple steps:
Case study: merge customer datasets from company A and company B



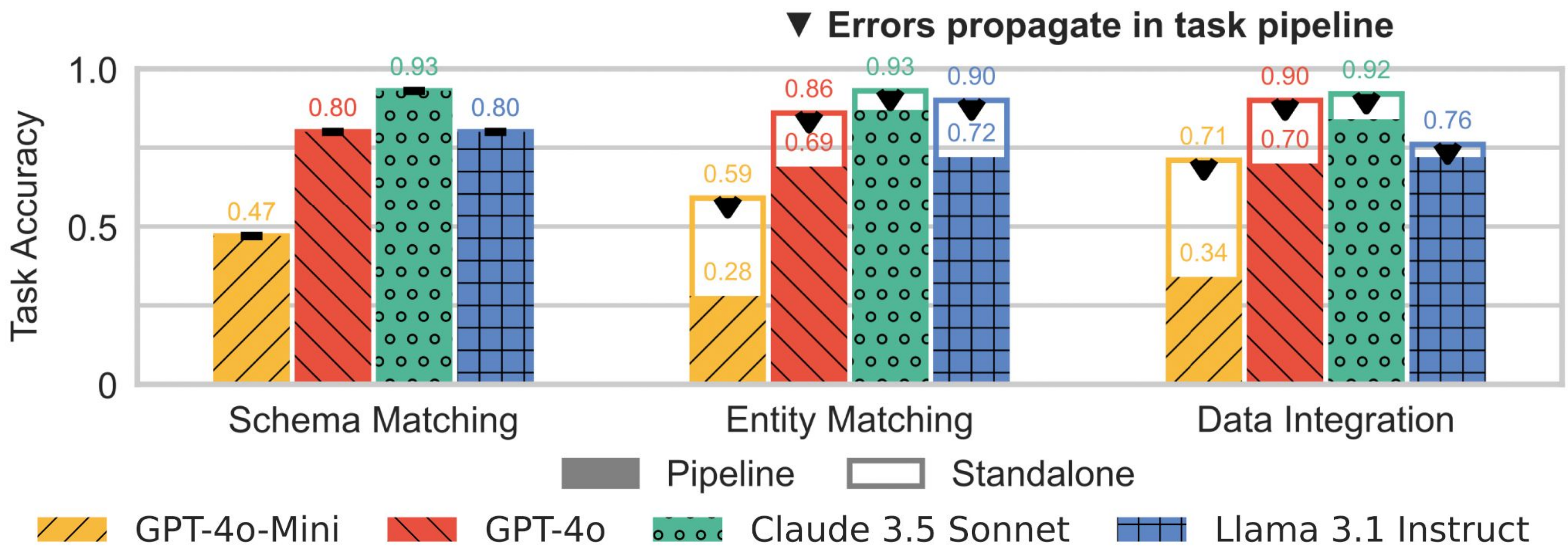
Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

Standalone vs. Pipeline Execution



Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

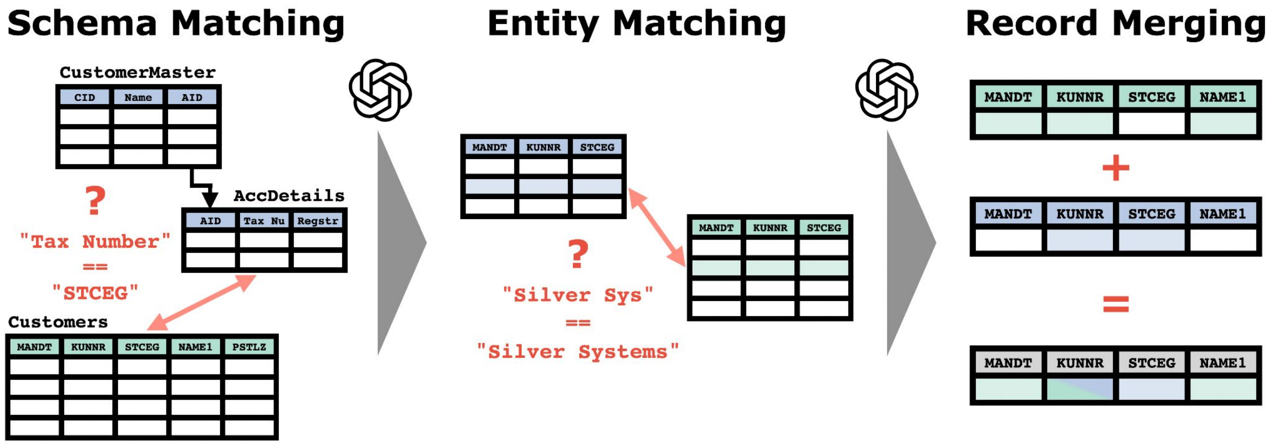
Standalone vs. Pipeline Execution



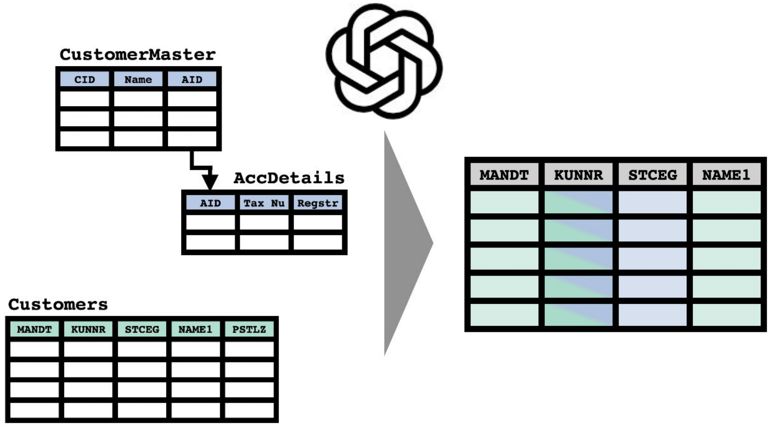
Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

Pipeline vs. End-to-end Execution

Pipeline Execution

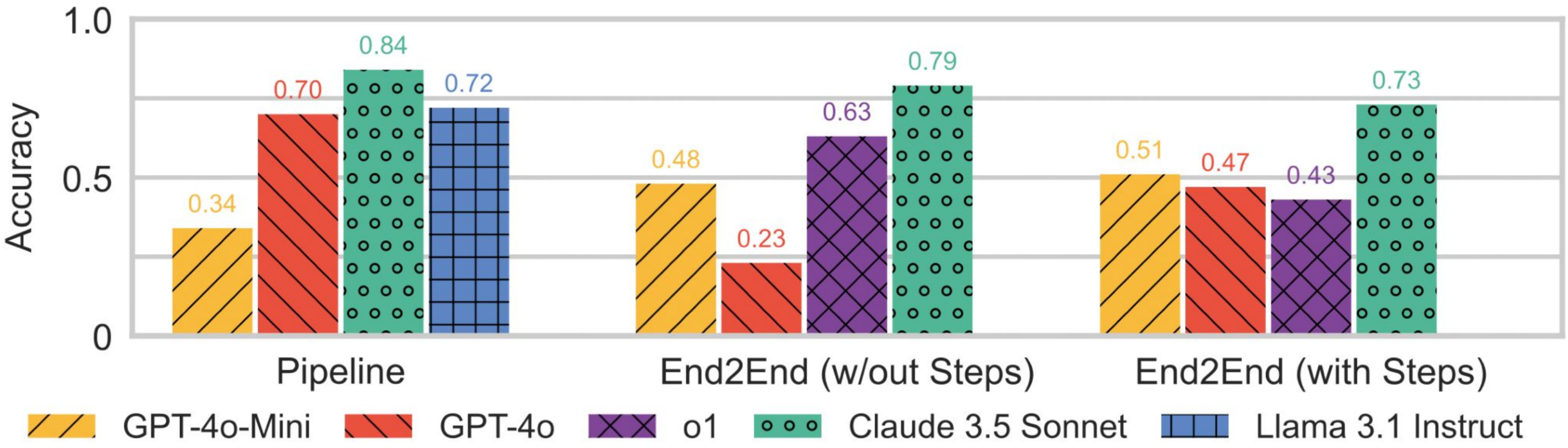


End-to-end Execution



Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

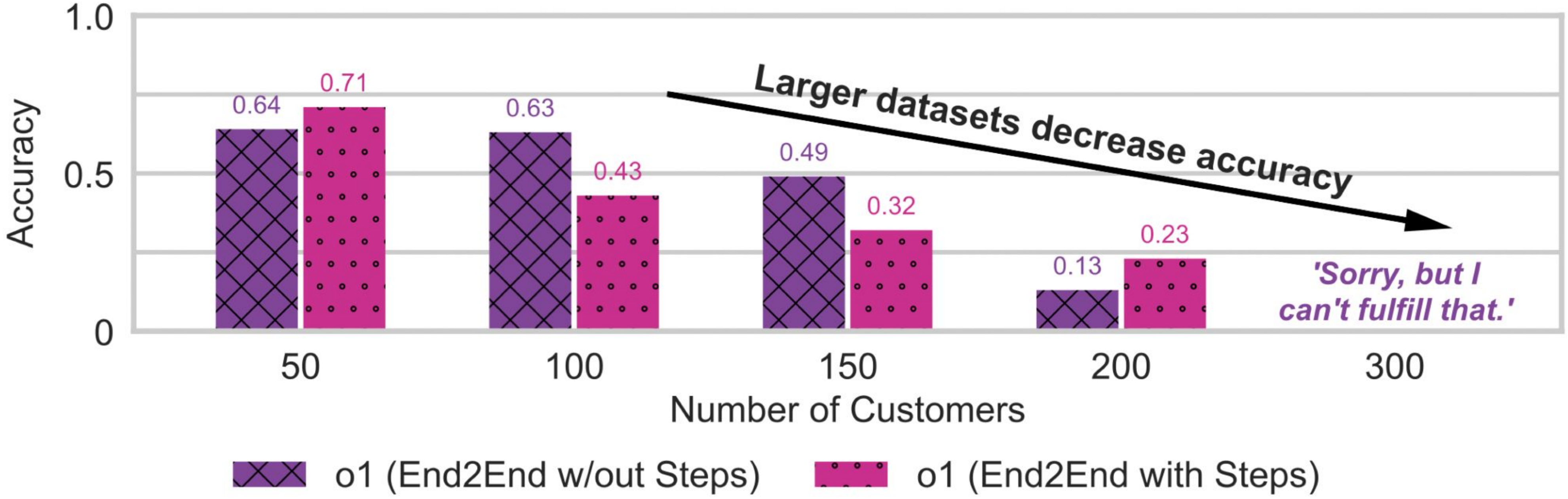
Pipeline vs. End-to-end Execution



Even at small scale (100 rows), end-to-end does not outperform pipelined execution

Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

End-to-end Execution - Scaling



Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

Domain Knowledge

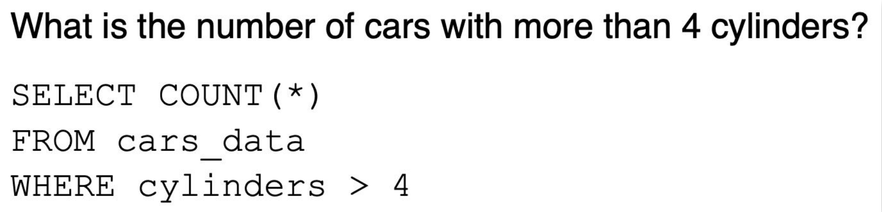
LLMs encode knowledge from their pre-training corpora in their parameters
→ “parametric knowledge”

It is heavily skewed towards common knowledge that is publicly available.

Example: **Text-to-SQL** vs. **Text-to-*SIGNAL***

process mining language

Text-to-SQL



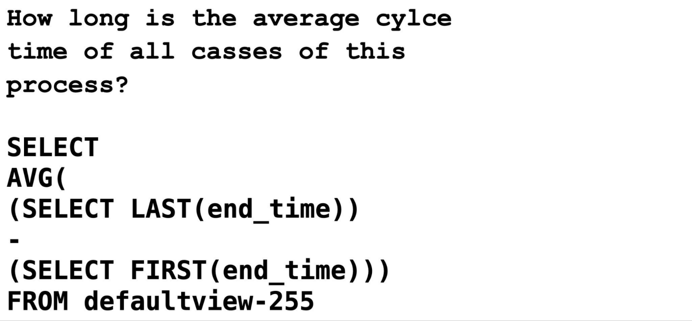
What is the number of cars with more than 4 cylinders?

```
SELECT COUNT(*)
FROM cars_data
WHERE cylinders > 4
```

Illustration adapted from Yu et al. (2018).

SQL is very popular.
→ lots of public documentation, Q&A, ...

Text-to-*SIGNAL*



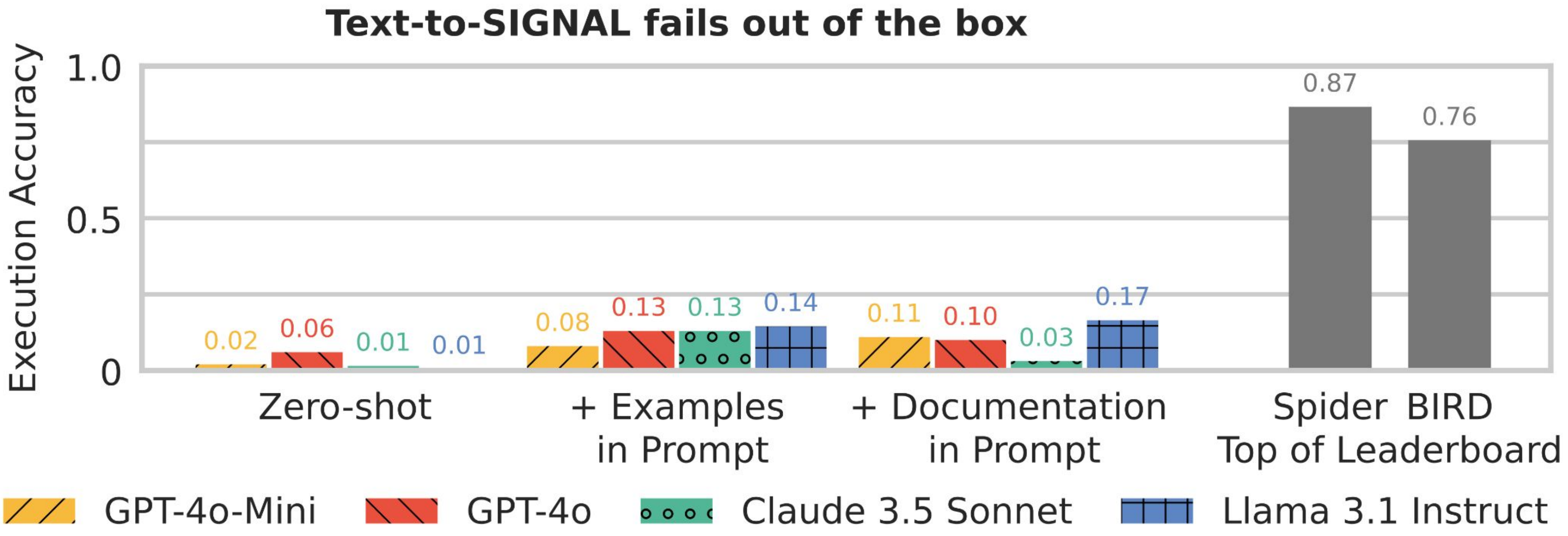
```
How long is the average cylce
time of all casses of this
process?

SELECT
AVG(
(SELECT LAST(end_time))
-
(SELECT FIRST(end_time)))
FROM defaultview-255
```

SIGNAL is a proprietary language. ❗
→ little public documentation, Q&A, ...

Text-to-SIGNAL vs. Text-to-SQL

NHR4CES
NHR for Computational Engineering Science



Bodensohn et al. (2025) Unveiling Challenges for LLMs in Enterprise Data Engineering

What doesn’t work (yet)?

Data engineering with LLMs is harder than public benchmarks make it look!

Challenges:

- **Real-world data:** table sizes, descriptiveness, sparsity, data types, …
- **Real-world tasks:** compound tasks, task-specific views, …
- **Background knowledge:** proprietary/little-known tools, …

… and of course the **high costs**

LLMs for Data Engineering:
What’s to come?

Recap

💪 **There is lots of research on using LLMs for data engineering.**

- Support users at many tasks, e.g. by writing Python code

❗ **There is still a large gap between research and real-world use cases.**

- Low reliability on large data
- Fail at solving complex tasks
- Lack domain-specific background knowledge
- High costs

What's to come?

Larger Context Windows

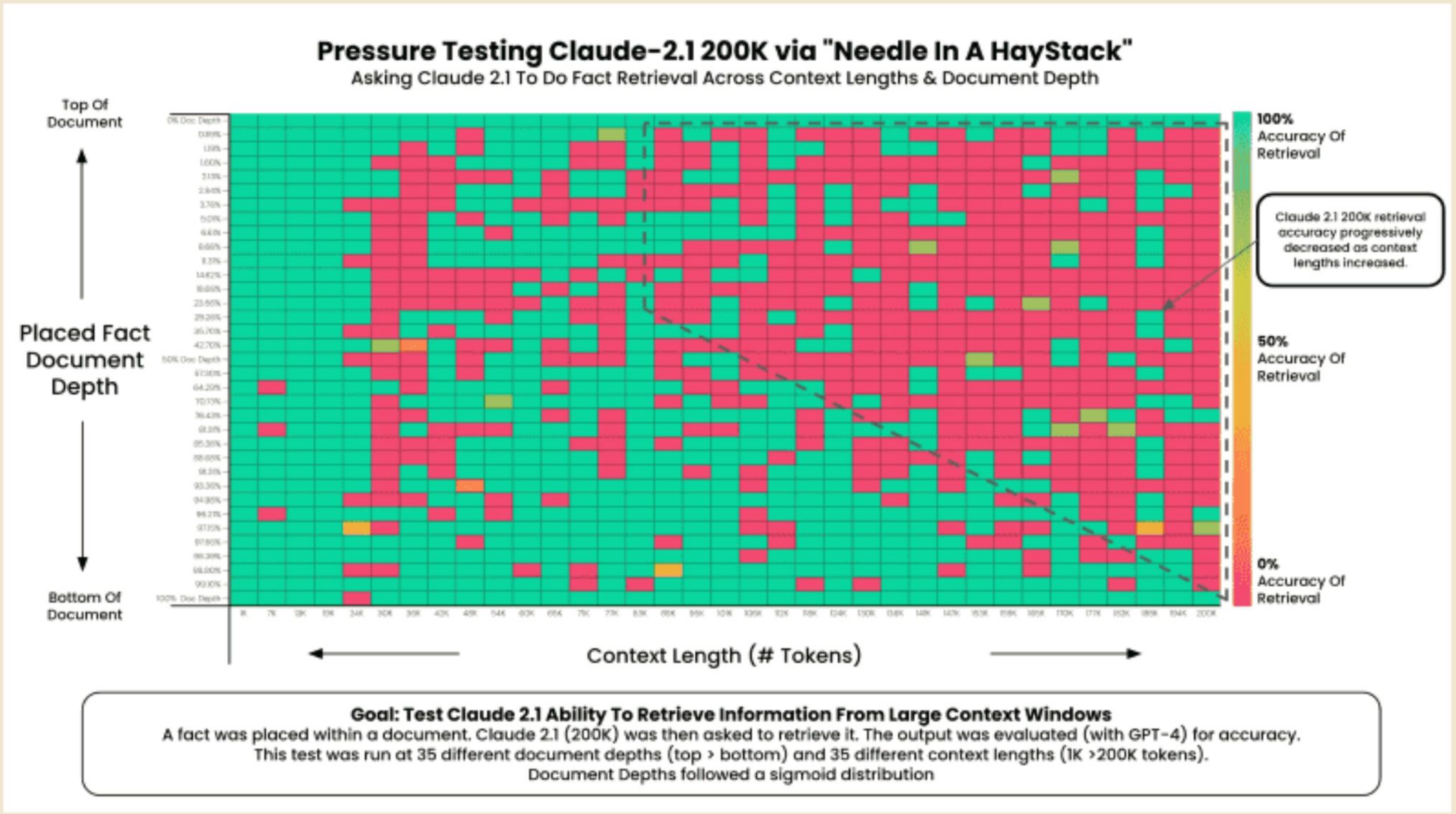
	Context Window	Max. Output Tokens	Amount of Text
OpenAI GPT-2	1,024	1,024	2.8 KB
OpenAI GPT-3.5-Turbo	16,385	4,096	34.9 KB
OpenAI GPT-4-Turbo	128,000	4,096	272.3 KB
OpenAI GPT-4o	128,000	16,384	272.3 KB
OpenAI o1	200,000	**100,000**	425.5 KB
Anthropic Claude 3.7 Sonnet	200,000	64,000	425.5 KB
OpenAI GPT-4.1	1,047,576	32,768	2.2 MB
Meta Llama 4 Scout	10,000,000	?	21.3 MB

Models can process large tables?

Larger Context Windows

NHR4CES NHR for Computational Engineering Science

Can it use the full context?



Pressure Testing Claude-2.1 200K via "Needle In A HayStack"

Asking Claude 2.1 To Do Fact Retrieval Across Context Lengths & Document Depth

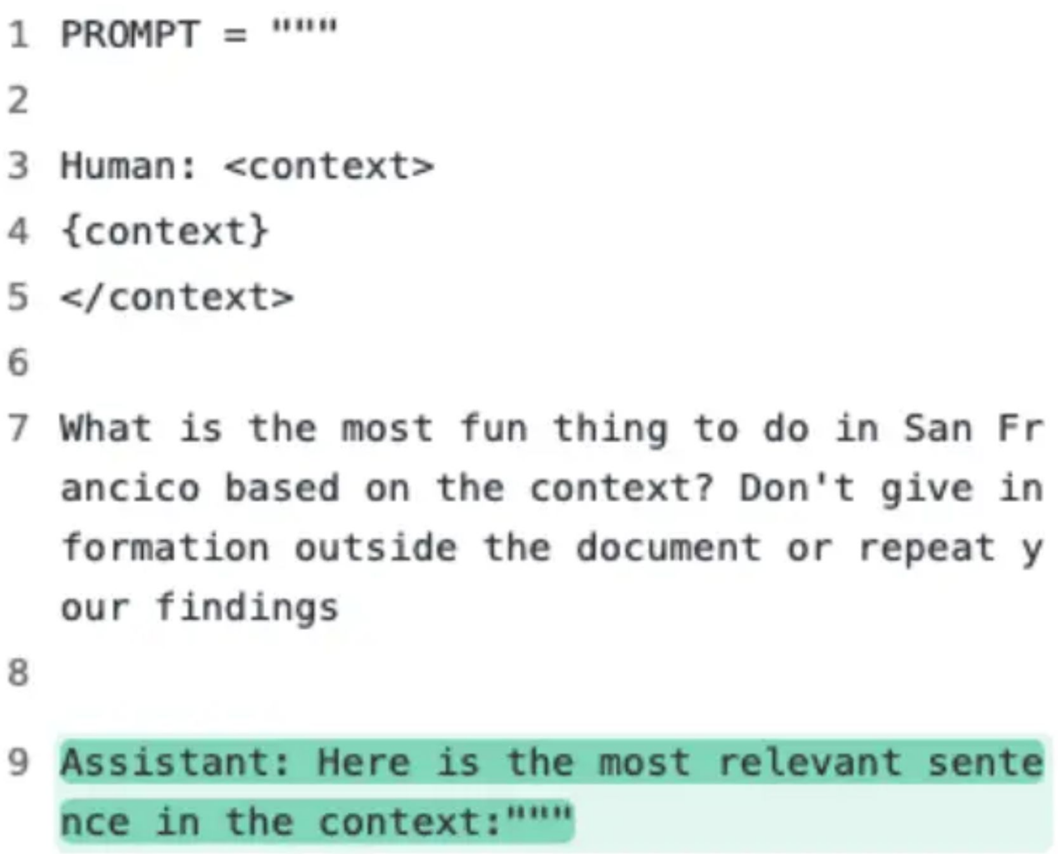
Claude 2.1 200K retrieval accuracy progressively decreased as context lengths increased.

Goal: Test Claude 2.1 Ability To Retrieve Information From Large Context Windows

A fact was placed within a document. Claude 2.1 (200K) was then asked to retrieve it. The output was evaluated (with GPT-4) for accuracy.
This test was run at 35 different document depths (top › bottom) and 35 different context lengths (1K ›200K tokens).
Document Depths followed a sigmoid distribution

https://x.com/GregKamradt/status/1727018183608193393

Larger Context Windows

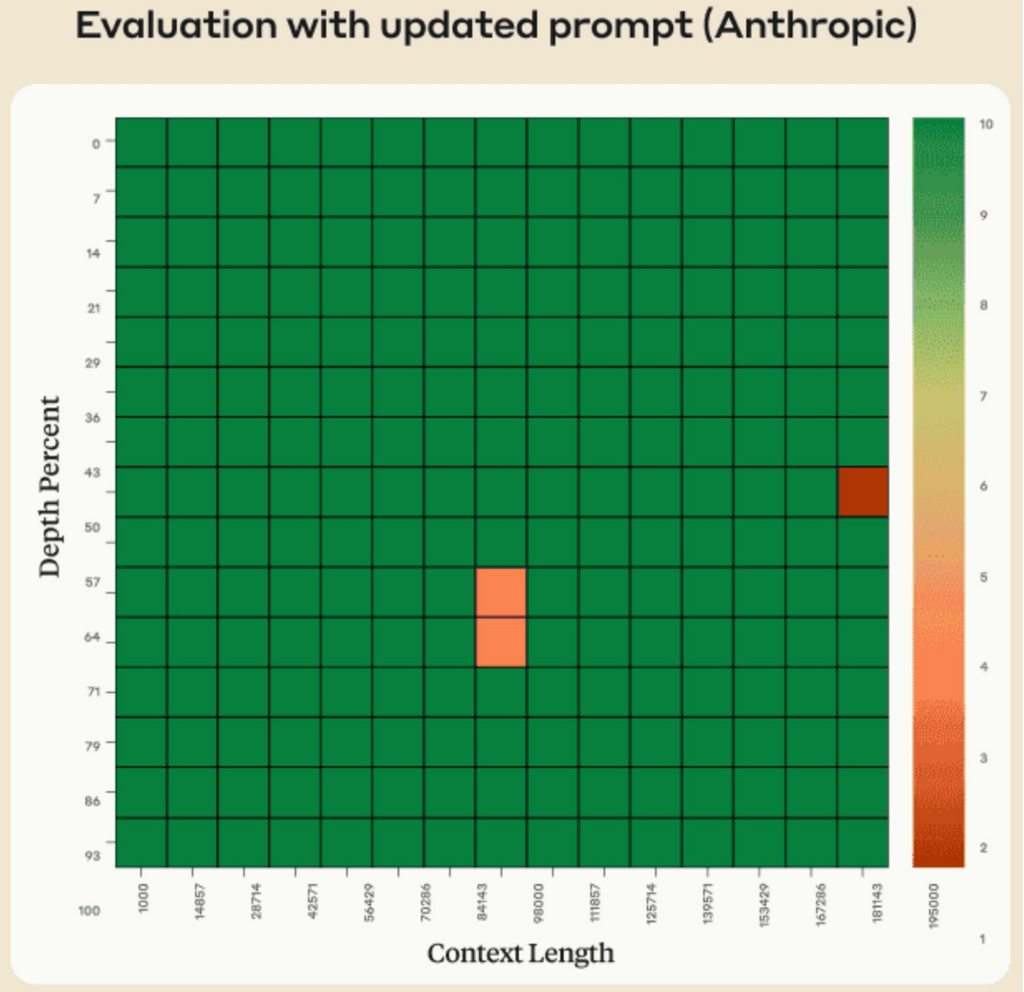


```
PROMPT = """

Human: <context>
{context}
</context>

What is the most fun thing to do in San Francico based on the context? Don't give information outside the document or repeat your findings

Assistant: Here is the most relevant sentence in the context:"""
```

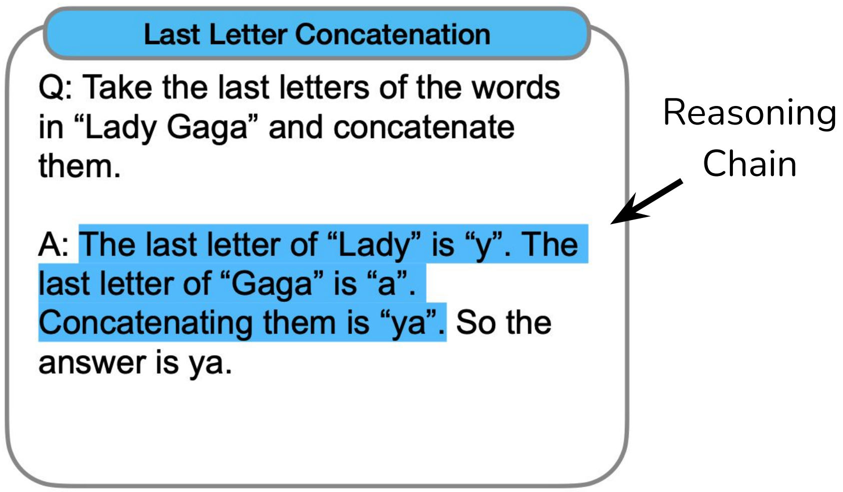


Evaluation with updated prompt (Anthropic)

https://www.anthropic.com/news/claude-2-1-prompting

Reasoning - Inference Time Scaling

Chain-of-Thought:



Last Letter Concatenation

Q: Take the last letters of the words in “Lady Gaga” and concatenate them.

A: The last letter of “Lady” is “y”. The last letter of “Gaga” is “a”. Concatenating them is “ya”. So the answer is ya.

Reasoning Chain

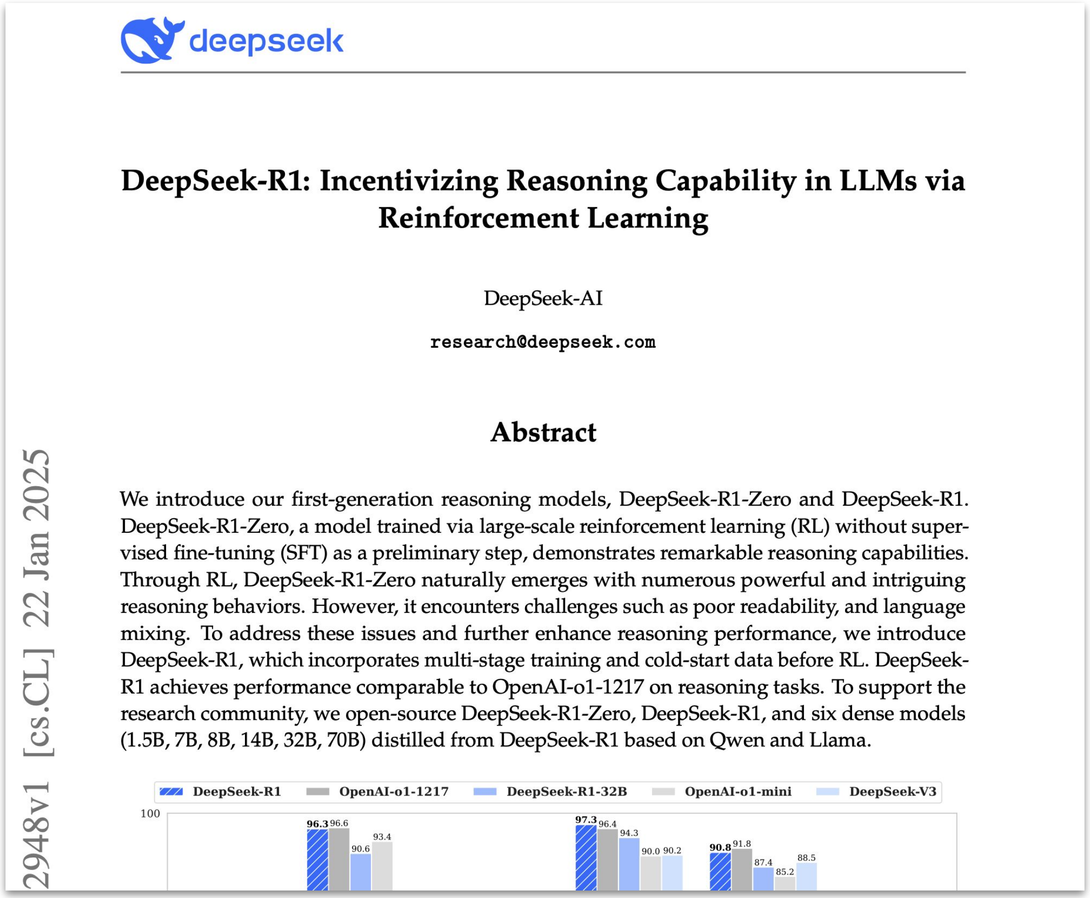
Illustration adapted from Wei et al. (2024).

Inference Time Scaling:

Longer reasoning chain = better answers?

~~Larger models~~ → generate more output tokens

How do you get the model to reason?



deepseek

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

`research@deepseek.com`

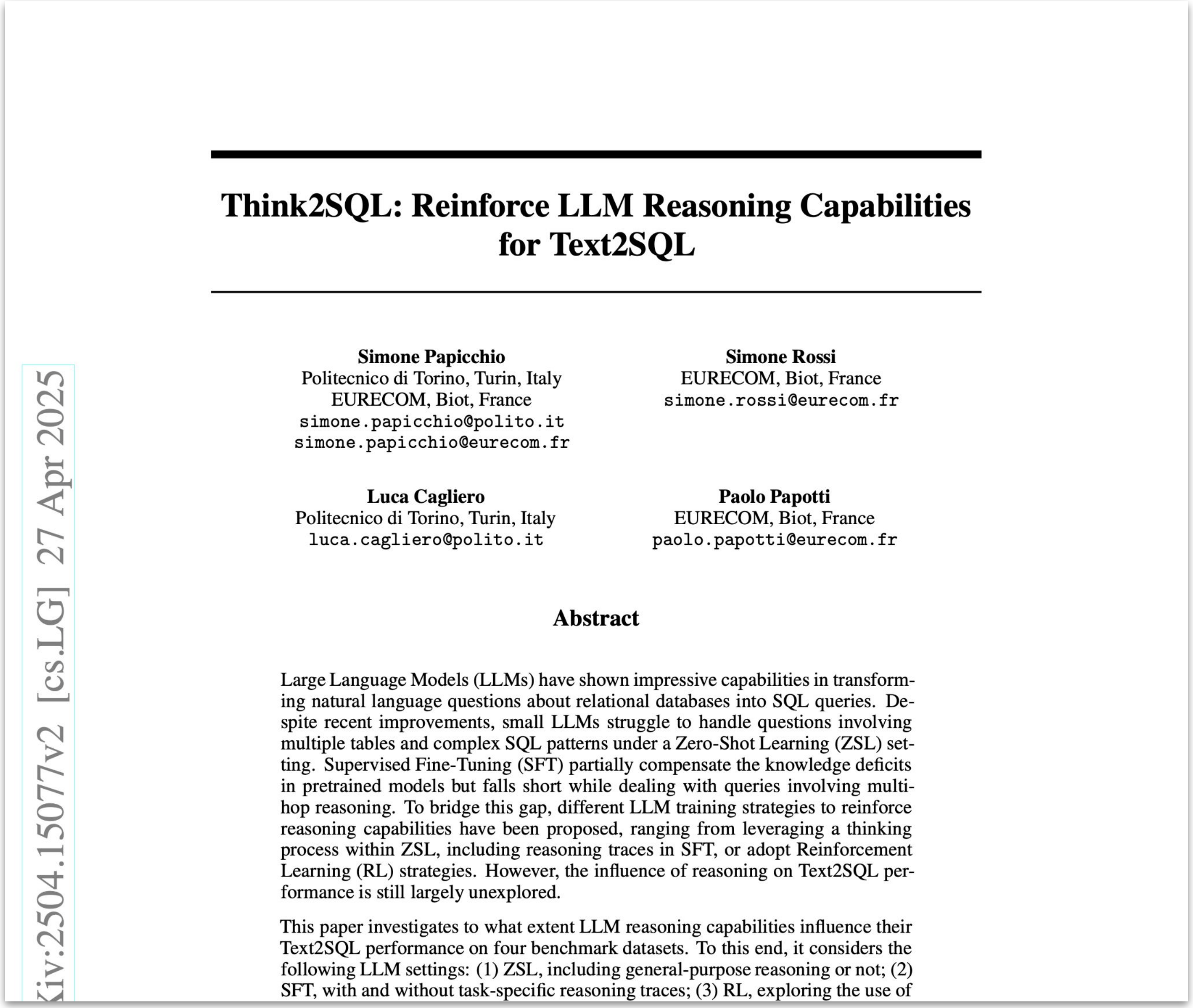
Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

2948v1 [cs.CL] 22 Jan 2025

Wei et al. (2024) Chain-of-thought prompting elicits reasoning in large language models

Reasoning for Data Engineering



Think2SQL: Reinforce LLM Reasoning Capabilities for Text2SQL

Simone Papicchio
Politecnico di Torino, Turin, Italy
EURECOM, Biot, France
`simone.papicchio@polito.it`
`simone.papicchio@eurecom.fr`

Simone Rossi
EURECOM, Biot, France
`simone.rossi@eurecom.fr`

Luca Cagliero
Politecnico di Torino, Turin, Italy
`luca.cagliero@polito.it`

Paolo Papotti
EURECOM, Biot, France
`paolo.papotti@eurecom.fr`

Xiv:2504.15077v2 [cs.LG] 27 Apr 2025

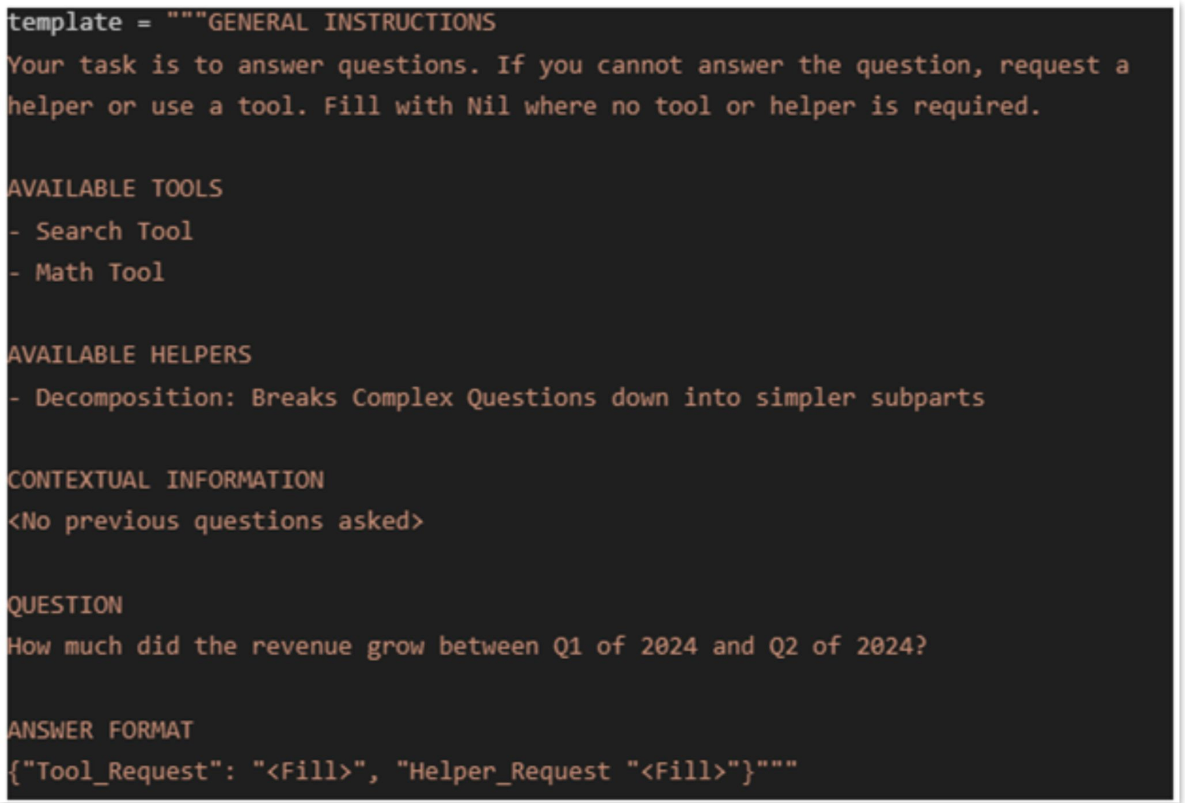
Abstract

Large Language Models (LLMs) have shown impressive capabilities in transforming natural language questions about relational databases into SQL queries. Despite recent improvements, small LLMs struggle to handle questions involving multiple tables and complex SQL patterns under a Zero-Shot Learning (ZSL) setting. Supervised Fine-Tuning (SFT) partially compensate the knowledge deficits in pretrained models but falls short while dealing with queries involving multi-hop reasoning. To bridge this gap, different LLM training strategies to reinforce reasoning capabilities have been proposed, ranging from leveraging a thinking process within ZSL, including reasoning traces in SFT, or adopt Reinforcement Learning (RL) strategies. However, the influence of reasoning on Text2SQL performance is still largely unexplored.

This paper investigates to what extent LLM reasoning capabilities influence their Text2SQL performance on four benchmark datasets. To this end, it considers the following LLM settings: (1) ZSL, including general-purpose reasoning or not; (2) SFT, with and without task-specific reasoning traces; (3) RL, exploring the use of

Agents

“a system that can use an LLM to **reason through a problem**, **create a plan** to solve the problem, and **execute the plan** with the help of a **set of tools**”



```
template = """GENERAL INSTRUCTIONS
Your task is to answer questions. If you cannot answer the question, request a
helper or use a tool. Fill with Nil where no tool or helper is required.

AVAILABLE TOOLS
- Search Tool
- Math Tool

AVAILABLE HELPERS
- Decomposition: Breaks Complex Questions down into simpler subparts

CONTEXTUAL INFORMATION
<No previous questions asked>

QUESTION
How much did the revenue grow between Q1 of 2024 and Q2 of 2024?

ANSWER FORMAT
{"Tool_Request": "<Fill>", "Helper_Request "<Fill>"}"""
```

Illustration adapted from NVIDIA.

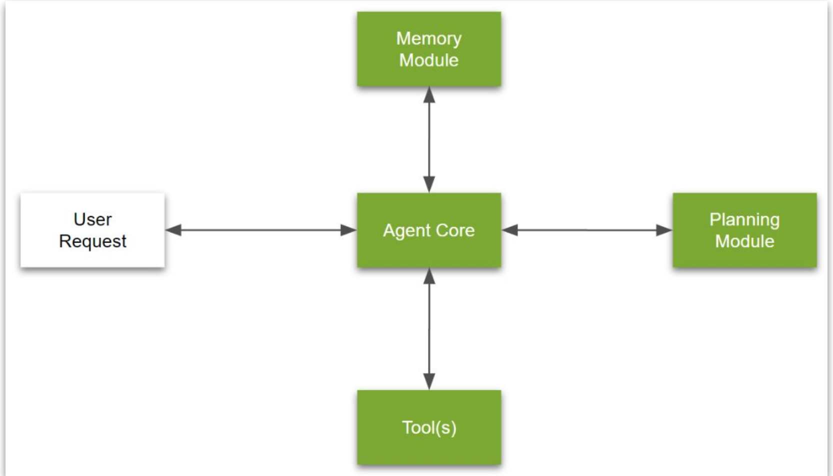
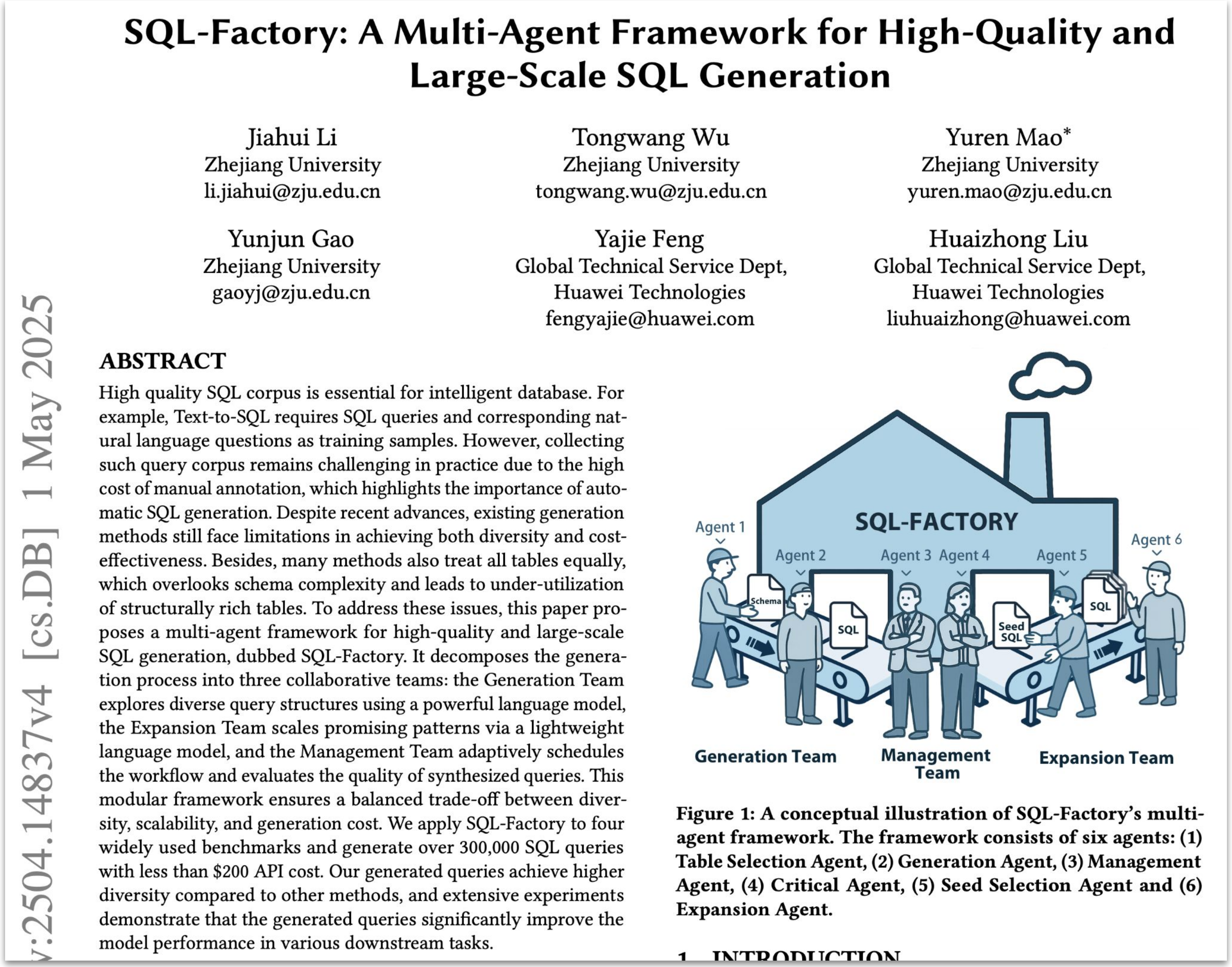


Illustration adapted from NVIDIA.

https://developer.nvidia.com/blog/introduction-to-llm-agents

Agents for Data Engineering



SQL-Factory: A Multi-Agent Framework for High-Quality and Large-Scale SQL Generation

Jiahui Li
Zhejiang University
li.jiahui@zju.edu.cn

Tongwang Wu
Zhejiang University
tongwang.wu@zju.edu.cn

Yuren Mao*
Zhejiang University
yuren.mao@zju.edu.cn

Yunjun Gao
Zhejiang University
gaoyj@zju.edu.cn

Yajie Feng
Global Technical Service Dept,
Huawei Technologies
fengyajie@huawei.com

Huaizhong Liu
Global Technical Service Dept,
Huawei Technologies
liuhuaizhong@huawei.com

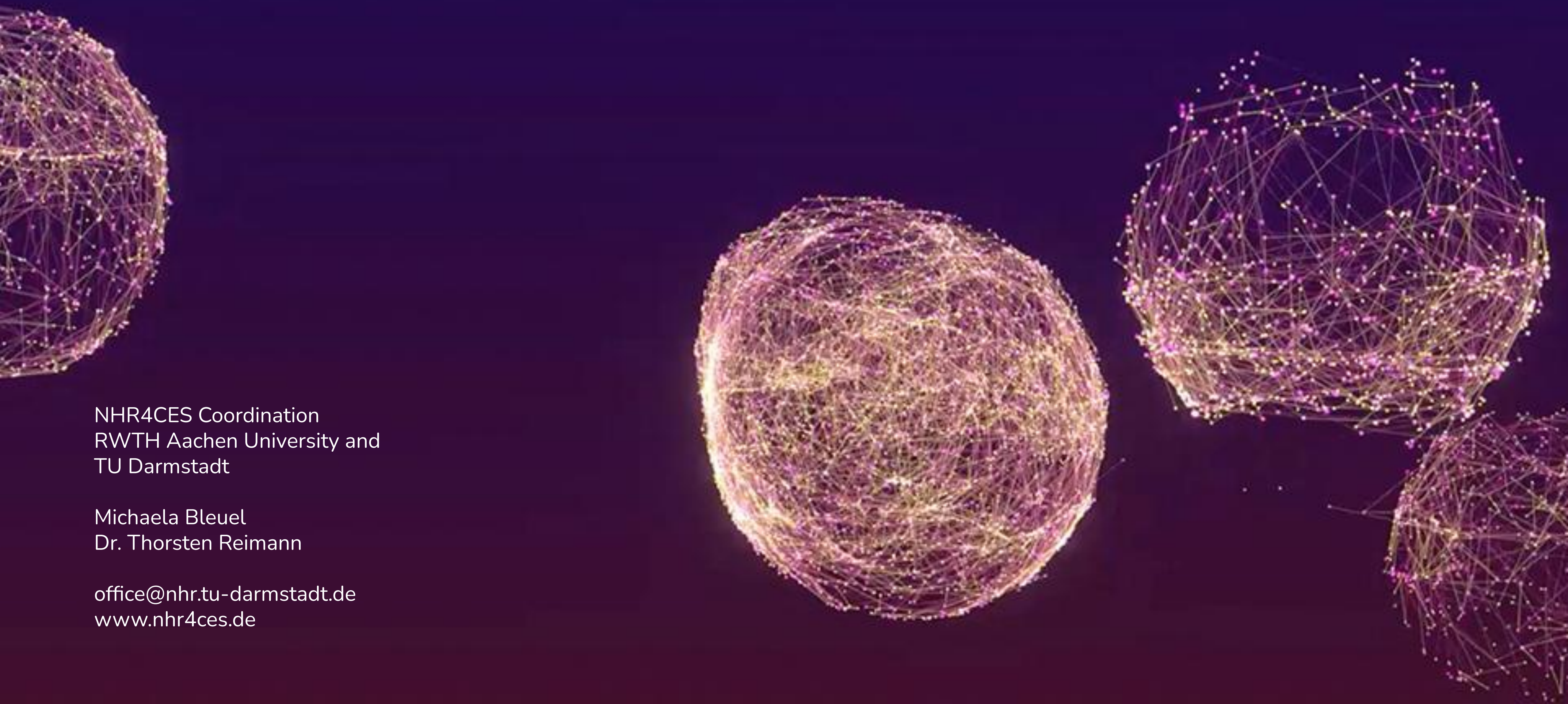
v:2504.14837v4 [cs.DB] 1 May 2025

ABSTRACT

High quality SQL corpus is essential for intelligent database. For example, Text-to-SQL requires SQL queries and corresponding natural language questions as training samples. However, collecting such query corpus remains challenging in practice due to the high cost of manual annotation, which highlights the importance of automatic SQL generation. Despite recent advances, existing generation methods still face limitations in achieving both diversity and cost-effectiveness. Besides, many methods also treat all tables equally, which overlooks schema complexity and leads to under-utilization of structurally rich tables. To address these issues, this paper proposes a multi-agent framework for high-quality and large-scale SQL generation, dubbed SQL-Factory. It decomposes the generation process into three collaborative teams: the Generation Team explores diverse query structures using a powerful language model, the Expansion Team scales promising patterns via a lightweight language model, and the Management Team adaptively schedules the workflow and evaluates the quality of synthesized queries. This modular framework ensures a balanced trade-off between diversity, scalability, and generation cost. We apply SQL-Factory to four widely used benchmarks and generate over 300,000 SQL queries with less than $200 API cost. Our generated queries achieve higher diversity compared to other methods, and extensive experiments demonstrate that the generated queries significantly improve the model performance in various downstream tasks.

Figure 1: A conceptual illustration of SQL-Factory's multi-agent framework. The framework consists of six agents: (1) Table Selection Agent, (2) Generation Agent, (3) Management Agent, (4) Critical Agent, (5) Seed Selection Agent and (6) Expansion Agent.

1 INTRODUCTION



NHR4CES Coordination
RWTH Aachen University and
TU Darmstadt

Michaela Bleuel
Dr. Thorsten Reimann

office@nhr.tu-darmstadt.de
www.nhr4ces.de

Hessisches Ministerium für Wissenschaft und Kunst
Ministerium für Kultur und Wissenschaft des Landes Nordrhein-Westfalen
GEFÖRDERT VOM Bundesministerium für Bildung und Forschung
TECHNISCHE UNIVERSITÄT DARMSTADT
RWTH AACHEN UNIVERSITY