Question Answering over Tabular Data with DataBench: A Large-Scale Empirical Evaluation of LLMs

Jorge Osés Grijalba, Luis Alfonso Ureña-López, Eugenio Martínez Cámara and Jose Camacho-Collados
Graphext, University of Jaen, Cardiff University
jorge@graphext.com, {laurena, emcamara}@ujaen.es
camachocolladosj@cardiff.ac.uk

Abstract

Large Language Models (LLMs) are showing emerging abilities, and one of the latest recognized ones deals with their ability to reason and answer questions from tabular data. Although there are some available datasets to assess question answering systems on tabular data, they are not large and diverse enough to properly assess the capabilities of LLMs. To this end, we propose DataBench, a benchmark composed of 65 real-world datasets over several domains, including 20 human-generated questions per dataset, totaling 1300 questions and answers overall. Using this benchmark, we perform a large-scale empirical comparison of several open and closed source models, including both code-generating and in-context learning models. The results highlight the current gap between open-source and closed-source models, with all types of model having room for improvement even in simple boolean questions or involving a single column.

Keywords: question answering, tabular data, llms, benchmark

1. Introduction

The advent of the era of large language models (LLMs) has revolutionized the research on natural language processing (NLP), especially since their scaling up as zero- and few-shot learners (Radford et al., 2019; Brown et al., 2020). This capacity of learning without the need to follow the standard machine learning training workflow enables the usage of task-agnostic architectures to resolve a wide range of tasks such as sentiment analysis (Deng et al., 2023; Zhang et al., 2023c), machine translation (Jiao et al., 2023) or text summarisation (Zhang et al., 2023b), to name a few. This growth has been possible partially by the continuous release of general-purpose LLMs (Yang et al., 2023) and the discovery of emergent abilities of LLMs (Wei et al., 2022). However, this incessant flux of models has not been accompanied by the release of high-quality and large-scale benchmarks for evaluating and comparing specific capacities of LLMs.

Question answering (QA) is a longstanding NLP task focused on retrieving the most adequate answer for a question on unstructured or plain text documents (Voorhees, 2001). On the other hand, structured data encompasses a great bunch of knowledge whose query which has traditionally been linked to a programmatic access by SQL or SPARK queries. However, these languages make rigid assumptions about the structured data organized in tables and are not able to understand the semantics of the textual fields. Likewise, they do not allow to make questions in natural language.

Because of this, question answering in non-database tables, structured or tabular data has attracted the interest of the research community (Pasupat and Liang, 2015a; Aly et al., 2021; Nan et al., 2022), especially to leverage language models to generate appropriate queries from natural language questions (Herzig et al., 2020; Liu et al., 2022). Recently, tabular question answering has been shown as an emergent ability of LLMs (Chen, 2023). This new capacity, along with the public reliance on these models, highlight the need for a wide benchmark to reliably assess the performance of LLMs.

In this paper, we present DataBench, a large benchmark for the task of tabular question answering on structured or tabular data. We propose DataBench with the aim of providing a benchmark to evaluate and compare LLMs as tabular reasoners, but flexible to compare any other type of question answering model. Accordingly, DataBench is composed of 65 datasets from different domains, widely different numbers of rows and columns and heterogeneous data types. Moreover, DataBench has 20 hand-made questions per dataset, with a total number of 1300 questions. Questions are further split in different types depending on the type of answer (i.e., true/false, categories from the dataset, numbers or lists), and each question is accompanied by their corresponding gold standard answer. Finally, we use DataBench to evaluate the last-generation of LLMs over tabular data, including code-generating models. The results show that current models are still not fully reliable to be used on tabular data, and there is significant room for improvement for all types of question and domain.

LREC-COLING 2024, pages 13471–13488
20-25 May, 2024. © 2024 ELRA Language Resource Association: CC BY-NC 4.0

2. Related Work

2.1. Table question answering

Table question answering or question answering on tabular data is a spin-off task of QA, which aims to provide answers to natural language questions from data in tables (Jin et al., 2022). Given the diversity of the structure of tables, there are different approaches to retrieve the answer. One of those approaches is to transform the question by a semantic parsing strategy to a formal language as SQL or formal representation, in order to retrieve the data from a database, HTML pages or other kind of tables that also need to be processed (Pasupat and Liang, 2015a; Zhong et al., 2017; Nan et al., 2022). The tables may be surrounded by text, which implies to first identify the table, and them to extract the answer from the table. In this situation, there are works that extract the answer (Eisenschlos et al., 2021) and others that generate the answer (Chen et al., 2021b). Open QA is a related QA task that processes large amount of knowledge bases to answer factoid questions instead of only processing documents (Zhang et al., 2023a). We also find Open QA systems in tabular data, which have the ability of processing tabular data in the span of text relevant to the user question (Chen et al., 2020; Herzig et al., 2021; Ma et al., 2022).

2.2. Evaluation

The evaluation of language models has been an ongoing research endeavor in the community. This started with initial dataset unification initiatives such as GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019), which integrated a suite of heterogenous NLP datasets into a single benchmark. These benchmarks soon began to be saturated and solved by newer language models (Yang et al., 2019). Moreover, the tasks were limited in which the capabilities of current LMs are concerned. Because of these, newer benchmarks such as MMMU (Hendrycks et al., 2021) and BIG-Bench (Srivastava et al., 2022) have emerged. However, none of the tasks deal with the tabular reasoning ability of LLMs.

The evaluation of tabular QA systems is starred by Wikipedia-based datasets (Pasupat and Liang, 2015a; Zhong et al., 2017; Nan et al., 2022) and domain specific datasets as the financial domain (Zhu et al., 2021; Chen et al., 2021b). Additionally, there are some multi or open domain, nonetheless grounded in Wikipedia documents (Chen et al., 2020; Herzig et al., 2021). Currently, the efforts are focusing on making datasets for hard questions that require complex reasoning over tables, as the OpenWikiTables dataset (Kweon et al., 2023). Unlike previous efforts focusing on Wikipedia and domain-specific benchmarks, in this paper we put forward a diverse benchmark composed of a large set of datasets from different domains.

2.3. Limitations of Wikipedia-based Benchmarks

Given that most datasets represent Wikipedia tables, in this section we explain the main limitations of benchmark from this domain. For example, Open Wikitables (Kweon et al., 2023) is a representative of a Wikipedia-based benchmark composed of 2000 tables that is currently used to assess the performance of QA systems over tabular data.

Type of data Wikipedia tables consist essentially of numbers, categories and very short texts, but are severely lacking in other types that are common in today's real-world datasets. Some of these types that are entirely absent from Wikipedia's evaluation set include booleans, lists of numbers and lists of urls, and others such as *urls* only appear 5 times in the Open Wikitables test set.

Cleanliness Wikipedia tables do not usually contain nulls and are well formatted (Pasupat and Liang, 2015a), as opposed to real-world datasets.

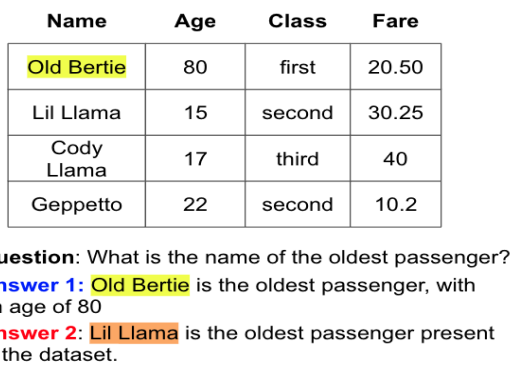
Size Some real-world datasets span millions of rows, as opposed to a maximum of hundreds generally found in Wikipedia tables, and potentially thousands of columns. For reference, the evaluation set of Open WikiTables has an average of 6 columns and 18 rows per table (Kweon et al., 2023), consisting of a total of 41,028 rows of data.

3. DataBench: QA Benchmark over Tabular Data

In this section, we present DataBench, a benchmark aiming to provide a realistic and diverse testing ground for question answering models over tabular data in the form of structured CSV-style files across rows and columns. DataBench is publicly available at `https://huggingface.co/datasets/cardiffnlp/databench`. A simplified overview of the task is shown in Figure 1.

3.1. Data Collection

First, we collected a wide variety of tabular datasets in English across various domains. We have compiled 65 publicly available datasets, which are summarized in Table 1 and Table 2. This corpus repre-



Name	Age	Class	Fare
Old Bertie	80	first	20.50
Lil Llama	15	second	30.25
Cody Llama	17	third	40
Geppetto	22	second	10.2

Question: What is the name of the oldest passenger?
Answer 1: Old Bertie is the oldest passenger, with an age of 80
Answer 2: Lil Llama is the oldest passenger present in the dataset.

Figure 1: A question asked over a given dataset.

sents different types of data belonging to various domains.

To collect these datasets, we have partnered with Graphext, a data analytics company that gathers a collection of some of the most common public dataset types and analyses. These come from a variety of sources, and all of the information contained within is publicly available and with a compliant license. Table 3 contains a list of all datasets and their corresponding sources, including the individual number of rows and columns. All datasets are included in their original form. Some datasets may contain some degree of noise or redundancy, but we have decided to leave them as they are so as to test models in real-world settings in which datasets may not be perfectly cleaned.

Domain taxonomy Table 1 shows the number of datasets per domain, including the total number of rows and columns. We have categorized all the datasets in the following five domains: (1) **Health** public data related to diseases and health metrics; (2) **Business** data regarding money, transactions and finance related to different industries; (3) **Social Networks and Surveys** including X (formerly Twitter), surveys and similar; (4) **Sports and Entertainment** data for players, competitions and culturally relevant data; (5) **Travel and Locations** data related to travel locations, hotels or vacation rental properties.

Column types In an effort to map the stage for later analysis, we have categorized the columns by type. This information allows us to segment different kinds of data so that we can subsequently analyze the model's behavior on each column type separately. Table 2 lists all the data types assigned to each column, as well as the number of columns for each type. The most common data types are numbers and categories with 1336 columns of the total of 1615 included in DataBench (see Table 1). These are followed by some other more rare types as urls, booleans, dates or lists of elements.

Domain	Datasets	Rows	Columns
Business	26	1,156,538	534
Health	7	98,032	123
Social	16	1,189,476	508
Sports	6	398,778	177
Travel	10	427,151	273
Total	65	3,269,975	1615

Table 1: DataBench domain taxonomy.

Type	Columns	Example
number	788	55
category	548	apple
date	50	1970-01-01
text	46	A red fox ran...
url	31	google.com
boolean	18	True
list[number]	14	[1,2,3]
list[category]	112	[apple, orange, banana]
list[url]	8	[google.com, apple.com]

Table 2: Column types present in DataBench.

Simplified benchmark We have compiled a reduced version of each dataset to evaluate models that cannot process large datasets. The first version is the original DataBench benchmark described above, while the second one represents a smaller sample with the first 20 rows and will be called **DataBench_lite**.

3.2. Questions and Answers

Generation Process. We have produced a number of 20 hand-made questions per dataset, totaling 1300 questions. To generate the questions, we leveraged the partnership with Graphext in order to get an insight into some of the questions that are often asked while working with these datasets. This provided a basis of the main core of the questions, which we have then curated and expanded to complete the questions for all the desired answer types (see QA Types). This question expansion was done manually, generating the answers either assisted by the company's visual user interface or code to answer some of the most complex questions. As we mentioned in Section 3.1, we provided two versions for each dataset, one full and one reduced (DataBench_lite). Therefore, we also provided two answers for each question of each dataset's version.

QA Types. We have generated Question-Answer (QA) pairs of five different types: *boolean*, *category*, *number*, *list[category]* and *list[number]*.[1] This

[1]Note: types of columns (Table 2) are different from QA types (Table 4). The first refers to the data type contained within the column while the second deals with the answer format expected of a given question.

	Name	Rows	Cols	Domain	Source (Reference)
1	Forbes	2668	17	Business	Forbes (Forbes, 2022)
2	Titanic	887	8	Travel and Locations	Kaggle (Kaggle, 2021)
3	Love	373	35	Social Networks and Surveys	Graphext (Graphext, 2023a)
4	Taxi	100000	20	Travel and Locations	Kaggle (Risdal, 2017)
5	NYC Calls	100000	46	Business	City of New York (York, 2022)
6	London Airbnbs	75241	74	Travel and Locations	Kaggle (air, 2023)
7	Fifa	14620	59	Sports and Entertainment	Kaggle (fif, 2023)
8	Tornados	67558	14	Health	Kaggle (us, 2023)
9	Central Park	56245	6	Travel and Locations	Kaggle (nyc, 2022)
10	ECommerce Reviews	23486	10	Business	Kaggle (Agarap, 2018)
11	SF Police	713107	35	Social Networks and Surveys	US Gov (pol, 2018)
12	Heart Failure	918	12	Health	Kaggle (ML, 2021)
13	Roller Coasters	1087	56	Sports and Entertainment	Kaggle (Mulla, 2021)
14	Airbnb Madrid	20776	75	Travel and Locations	Inside Airbnb (Airbnb, Dec, 2022)
15	Food Names Embeddings	906	4	Business	Data World (Alexandra, 2018)
16	Holiday Package Sales	4888	20	Travel and Locations	Kaggle (Achary, 2021)
17	Hacker News	9429	20	Social Networks and Surveys	Kaggle (News, 2017)
18	Staff Satisfaction	14999	11	Business	Kaggle (Kaggle, 2023b)
19	Aircraft Accidents	23519	23	Health	Kaggle (avi, 2022)
20	Real Estate Madrid	26026	59	Business	Idealista (ide, 2023)
21	Telco Customer Churn	7043	21	Business	Kaggle (Jas, 2022)
22	Airbnbs Listings NY	37012	33	Travel and Locations	Kaggle (air, 2023)
23	Madrid Climate	36858	26	Travel and Locations	AEMET (AEMET, 2020)
24	Salary Survey Spain 2018	216726	29	Business	INE (INE, 2018)
25	Data Driven SEO	62	5	Business	Graphext (Graphext, 2021b)
26	Predicting Wine Quality	1599	12	Business	Kaggle (Cortez and Reis, 2009)
27	Supermarket Sales	1000	17	Business	Kaggle (Kaggle, 2023e)
28	Predict Diabetes	768	9	Health	Kaggle (of Vanderbilt, 2019)
29	NYTimes World In 2021	52588	5	Travel and Locations	New York Times(Times, 2021)
30	Professionals Kaggle Survey	19169	64	Business	Kaggle (Kaggle, 2023d)
31	TrustPilot Reviews	8020	6	Business	TrustPilot (Graphext)
32	Delicatessen Customers	2240	29	Business	Kaggle (Saldanha, 2020)
33	Employee Attrition	14999	11	Business	Kaggle (PavanKalyan, 2021)
34	World Happiness Report 2020	153	20	Social Networks and Surveys	World Happiness(WH, 2020)
35	Billboard Lyrics	5100	6	Sports and Entertainment	Brown University (of Brown, 2017)
36	US Migrations 2012-2016	288300	9	Social Networks and Surveys	US Census(cen, 2016)
37	Ted Talks	4005	19	Social Networks and Surveys	Kaggle (Jangra, 2021)
38	Stroke Likelihood	5110	12	Health	Kaggle (Pytlak, 2023)
39	Happy Moments	100535	11	Social Networks and Surveys	Kaggle (Labs, 2017)
40	Speed Dating	8378	123	Social Networks and Surveys	Kaggle (of Columbia, 2009)
41	Airline Mentions X	14640	15	Social Networks and Surveys	X (Graphext, 2021a)
42	Predict Student Performance	395	33	Business	Kaggle (Impapan, 2023)
43	Loan Defaults	83656	20	Business	SBA (Administration, 2021)
44	IMDb Movies	85855	22	Sports and Entertainment	Kaggle (Kaggle, 2023c)
45	Spotify Songs	21000	19	Sports and Entertainment	Spotify (Tomigelo, 2023)
46	120 Years Olympics	271116	15	Sports and Entertainment	Kaggle (Heesoo37, 2023)
47	Bank Customer Churn	7088	15	Business	Kaggle (Kaggle, 2023a)
48	Data Science Salary Data	742	28	Business	Kaggle (Ruchi798, 2023)
49	Boris Johnson UK PM Tweets	3220	34	Social Networks and Surveys	X (Graphext, 2022a)
50	ING 2019 X Mentions	7244	22	Social Networks and Surveys	X (Graphext, 2019)
51	Pokemon Feature Correlation	1072	13	Business	Kaggle (Banik, 2023)
52	Professional Map	1227	12	Business	Kern et al. (2019)
53	Google Patents	9999	20	Business	BigQuery (Google, 2021)
54	Joe Biden Tweets	491	34	Social Networks and Surveys	X (Graphext, 2022b)
55	German Loans	1000	18	Business	Kaggle (ML, 2016)
56	Emoji Diet	58	35	Health	Kaggle (kag, 2021)
57	Spain Survey 2015	20000	45	Social Networks and Surveys	CIS (CIS, 2015)
58	US Polls 2020	3523	52	Social Networks and Surveys	Brandwatch (Brandwatch, 2021)
59	Second Hand Cars	50000	21	Business	DataMarket (dat, 2021)
60	Bakery Purchases	20507	5	Business	Kaggle (García, 2020)
61	Disneyland Customer Reviews	42656	6	Travel and Locations	Kaggle (Chillar, 2023)
62	Trump Tweets	15039	20	Social Networks and Surveys	X (Graphext, 2020a)
63	Influencers	1039	14	Social Networks and Surveys	X (Graphext, 2020b)
64	Clustering Zoo Animals	101	18	Health	Kaggle (Daberger, 2023)
65	RFM Analysis	541909	8	Business	UCI ML (ML, 2015)

Table 3: The 65 datasets included in DataBench with their number of rows and columns, as well as their domain and source reference.

Question	Answer	Type	Columns Used	Column Types
Is Lil Llama the oldest passenger?	false	boolean	Name, Age	category, number
What's the class of the oldest passenger?	first	category	Name, Age	category, number
What's the lowest fare paid?	10.2	number	Fare	number
Who are the passengers under 30?	[Lil Lama, Cody Lama]	list[category]	Name, Age	category, number
What are the fares paid by passengers under 30?	[30.25, 10.2]	list[number]	Age, Fare	number, category

Table 4: Types of Question-Answer pairs present in our benchmark

categorization of QA pairs in different types of answer has the aim of making it simpler to diagnose and evaluate a model's performance. In total we have generated 4 hand-made questions for each

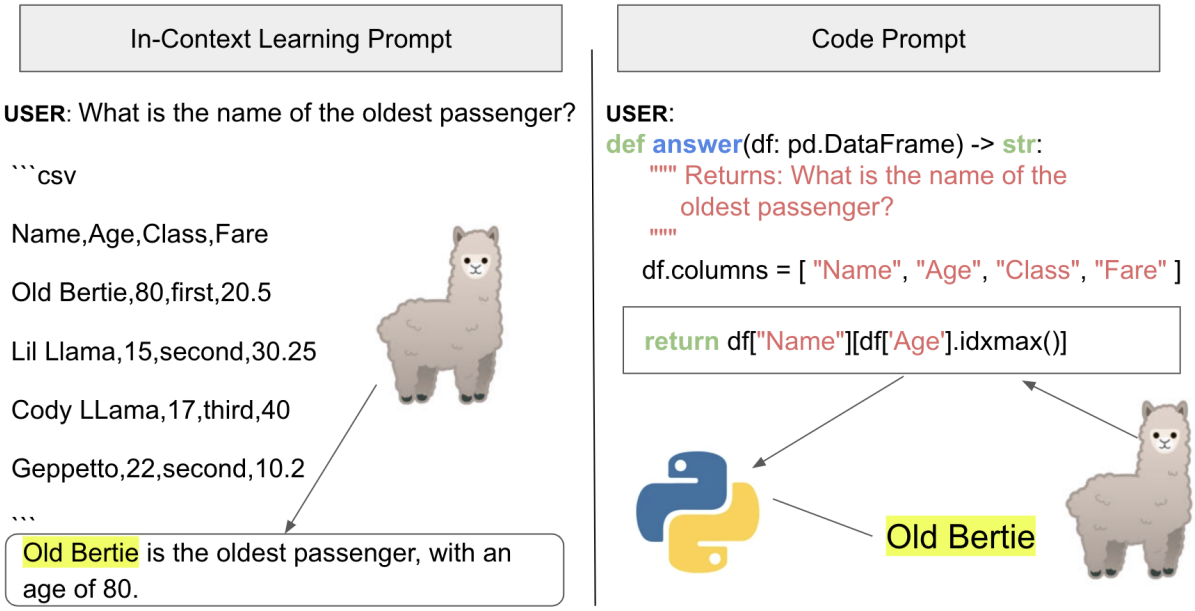


Figure 2: An overview of the two types of prompt included in the evaluation.

of the five types per dataset, with a total of 260 questions per type in DataBench. We have also tagged each question with the number of columns used to answer it, as well as the types of those columns. For example, in Table 4, to answer the question *Is Lil LLama the oldest passenger?* we need the information contained within the *Age* and *Name* (of types *category* and *number* respectively) columns, with a *boolean* answer. Examples for question and answers for the five different types, including the column required to answer the questions and number of questions and answers per type can be found in Table 4. This information is then used in our analysis (see Section 6).

Note that the only type of answer that has a predefined set of valid answers is the *boolean* type, namely *true* or *false*. The answers to the other types are essentially either statistics computed from the dataset or values found within the dataset. This in turn ensures that if the question is not ambiguous, it is guaranteed to have a factoid answer that we can easily validate against.

4. Experimental Setting

To test LLMs on our proposed benchmark, we present the following experimental setting.

4.1. Comparison models

Given the dependency of the performance of LLMs to how prompts query them, we propose two different kind of prompts to evaluate the tabular reasoning capacity of LLMs with DataBench.

The two prompts differ in the information given to the LLMs, and the definition of the format of the answer. In the following sections we present the details of each of the prompts, which we refer to as "In-Context" (Section 4.1.1) and "Code-based" (Section 4.1.2). Figure 2 provides an overview of the two approaches, which are further detailed in the following.

4.1.1. Zero-shot In-Context Learning

In Zero-shot In-Context Learning (Z-ICL), models are provided with the description of the task and data in the same prompt. In this case, we present the dataset in the form of a Comma Separated Values (CSV) format, which is one of the most common file formats used to store tabular data. We use markdown notation to signal the model that this is a CSV-formatted dataset we are using, and that's all the context it gets to then provide us with an answer.

We then request the model to structure the response around a JSON object with two or three keys. In the case of requesting two keys (Z-ICL Prompt 1), we request the actual answer and the columns that the model claims to have used to answer the question. We also add an alternative second prompt (Z-ICL Prompt 2) asking for a third field containing for an explanation of the answer given. The main advantage is that is given all available information about the dataset, at the cost of being less scalable as it is being limited by the window size. In the following we present the prompt provided for the model.

The decision to require the model to output a JSON file was motivated by suboptimal early results

that we got trying to automatically validate some plain text-based answers. These early experiment results have been included in Table 5 under *Z-ICL Plain Text*.

Models For the In-Context learning models, we compare both open- and closed-source models. As open model we include *llama-2-chat* in its 7B and 13B versions (Touvron et al., 2023b). We also include ChatGPT, a closed model, in particular *chatgpt3.5-turbo-0613* (Brown et al., 2020), in its *August 28th*, 2023 version. In this case, no further information has been given to chatgpt in order to answer the prompt. Note that for the open models we added the *INST* token delimiters as recommended in Touvron et al. (2023b).

Zero Shot In-Context Learning (Z-ICL) Prompt 2. Z-ICL Prompt 1 is similar but excludes the explanation.

You are an assistant tasked with answering the questions asked of a given CSV in JSON format. You must answer in a single JSON with three fields:
* "answer": answer using information from the provided CSV only.
* "columns_used": list of columns from the CSV used to get the answer.
* "explanation": A short explanation on why you gave that answer.
Requirements:
* Only respond with the JSON.
In the following CSV
“‘csv
passenger,wealth($)
value1,value2.
“‘
USER: What is the name of the richest passenger?
ASSISTANT: {"answer":

4.1.2. Code-based

The prompts provided in the previous section have the drawback of requesting a specific format for the answers, which can be challenging for certain types with our particular approach, especially lists. This shortcoming may be overcome by using a more formal language as a bridge between the user and the model, usually a programming language. The model will generate a chunk of code, which the user can follow if they are technical enough, and also will be easier to ask for the desired format of the answer, since we are not dealing with the verbosity of natural language. The potential limitation of this approach, however, is that it needs a third actor, in our case a *Python* interpreter, to extract the actual answer, which may not be inherent to all LLMs.

In particular, the prompt includes a chunk of code in *Python*, and the header of a function with one extra line, which then the model is trained to complete by generating the last code line. The prompt provides the format the model will receive the data in, a pandas dataframe, as well as the names of the columns, but not the whole dataset. Another advantage of this is the ability to scale this approach to larger datasets that might not fit in a prompt, since we are not including the whole dataset in a prompt, which may limite LLMs with a short context window. Moreover, by using a formal language such as Python that we understand, the black-box nature of LLMs is mitigated to a certain extent, as we can then easily analyse the type of coding errors, as we will see in Section 6.3.

The prompt provided encapsulates the minimal dataset information needed to answer the question, as well as the question to be answered (we will call this prompt *Code Prompt 1*). As an alternative, we also include a version that contains additional typing information of the columns included in the dataset (*Code Prompt 2*). In particular, we will be using pandas' basic *dtypes* without further prompt engineering. Listing 1 shows the actual prompts given as input to the models.

In both cases, for code completion we also import the external libraries that the model will be allowed to use, and include them in the prompt. In our case, these libraries are *pandas* and *numpy*.

```
import pandas as pd
import numpy as np

def answer(df) -> bool:
  '''The df dtypes are:
  {
    'Age': dtype('int8'),
    'Name': dtype('O'),
    'Class': dtype('O'),
    'Fare': dtype('float16')
  }
  Returns: Is the oldest
     passenger Old Bertie?
  '''
  df.columns=['Name', 'Age',
    'Class', 'Fare']
```

Listing 1: Code Prompt 2. Code Prompt 1 is similar but excludes the dtypes.

Models As open models we include the smaller versions of *CodeLLaMa* (Rozière et al., 2023), with 7B and 13B parameters. These are versions of LLaMa 2 (Touvron et al., 2023b) trained to handle a variety of tasks involving code. As in the previous case, we also include ChatGPT. In the case

of ChatGPT, given its more generalist nature, an additional prefix has been added to each prompt simply stating that the task is to complete the *answer* function. The rest of the prompt remains the same in both cases.

4.2. Data and Evaluation

Hardware and Model Versions For this experiment we have run the 4-bit quantized versions of the LLaMa models on consumer hardware, a *16 GB M2 2022 Macbook Air* using *llama.cpp*'s *Metal* optimization to run on CPU with GPU acceleration (Gerganov, 2023). We have also relied on OpenAI's API for the *gpt3.5* evaluation.

Data In order to be able to compare the models described in Section 4.1, the data needs to fit within the prompt for the models where all the data is given to the model. Because of this, our evaluation focuses on the reduced version of DataBench (i.e., DataBench_lite – see Section 3.1).

Type Suffixes We have also added a *type suffix* to each prompt explaining the desired format to the model for each type of question. For example, for *boolean* questions, we prompted the model to *Answer true or false* while for the categories we include *Answer with a category present in the dataset* or *Answer with a single number* for number-based questions. In the case of the Z-ICL JSON prompts, these formats were asked to be applied to the *answer* field within the JSON.

Relaxed evaluation One of the problems with LLMs is that the answers often do not follow a predictable formatting pattern. Because of this, we have relaxed the conditions for a match to allow for small format drifts to still be considered as a correct answer. For example, if a model answers *"true,"* or *"True."* or *"Yes"* they will all be considered to be valid answer for a boolean question that is supposed to be *true*. This has a smaller effect for the Code-based models. For lists, we have not accounted for order of the elements when comparing to the ground truth, which in some cases may be relevant. Despite their simplicity, these measures based on the type of answer have enabled a high enough level of automation so that the results can be evaluated without human intervention. This can help scale up the evaluation process to more datasets and questions.

Evaluation metric For the evaluation we focus on accuracy and further split by question types and other factors. We can see in Appendix A the full accuracy splits by domain and types of questions.

Format Errors In addition to accuracy, we also provide the percentage of format errors given by each model between parentheses. This format error comes from the inherently open nature of most modern LLMs, whose output may potentially be any text token. In our prompts, we ask the model to answer in the form of an object with some required attributes or some snippet of executable python code. In the case the JSON object returned cannot be successfully parsed as the specified response, due to for example missing braces or incorrect attributes, we mark it as a format error. We do the same for code-based prompts, if the code returned cannot be compiled and executed successfully without errors. By providing this metric we hope to encapsulate the dual nature of the task required of the models. First, models need to answer in a recognizable format, which is essential in order to mix the process with other automated processes that may feed off of their outputs. Then, models need answer correctly given a specific question. For the exact information on the code used to perform this parsing please refer to the repository of our dataset.

5. Experimental Results

In this section, we present the main experimental results of LLMs on *DataBench*. Table 5 shows the accuracy results split by question type, including the percentage of answers generated by the models that have an incorrect output format according to our evaluation script (see Section 4.2). This number is relevant because it allows us to check the capability of LLMs to follow the instructions, as well as isolating the failures of format that the model makes from the cases where the format is correct but the answer is nonetheless *wrong*.

In general, chatgpt3.5 achieves the best overall results, with accuracy numbers over 50% in all types of question, and 63% using the code-based functionality and the first prompt. These results clearly outperform the best open models, in this case codellama-13b with a 33.1% using the second prompt. While these chatgpt3.5 results highlight the progress of LLMs in this task, there is clear room for improvement, especially when it comes to open models. In the following section, we perform a more in-depth analysis of the results.

6. Analysis

In order to fully understand the quantitative results, we analyse them from four different perspectives.

6.1. Type of Prompt

When comparing code-based and Z-ICL prompts (see Table 5), the former prove more competitive

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	27.4	45.8 (37.8)	16.8 (63.0)	43.3 (36.8)	14.2 (41.0)	17.2 (32.4)	33.8 (39.2)	18.5 (46.5)
codellama-13b	31.0	53.4 (29.8)	25.2 (62.6)	46.7 (32.2)	18.8 (44.4)	11.1 (40.1)	37.2 (36.0)	22.3 (50.0)
chatgpt3.5	**63.0**	52.7 (6.1)	**73.3** (12.6)	**75.9** (8.0)	**56.7** (6.9)	**56.5** (11.1)	**67.0** (7.6)	**57.4** (10.9)
Code Prompt 2								
codellama-7b	30.3	45.0 (38.9)	23.3 (55.7)	49.8 (32.2)	16.5 (34.9)	16.8 (36.3)	37.3 (35.4)	20.3 (45.6)
codellama-13b	33.1	54.6 (25.2)	27.1 (58.0)	50.6 (32.6)	16.9 (38.7)	16.4 (34.0)	38.9 (33.3)	24.9 (43.9)
chatgpt3.5	55.7	46.6 (14.5)	64.5 (21.4)	74.3 (14.6)	47.1 (22.6)	45.8 (26.7)	62.9 (13.2)	45.4 (29.5)
Z-ICL Prompt 1								
llama-2-7b	14.4	38.0 (13.2)	19.4 (17.1)	10.5 (14.8)	3.1 (34.6)	0.8 (23.6)	16.2 (19.6)	11.8 (22.1)
llama-2–13b	19.3	56.6 (14.0)	21.7 (27.1)	13.6 (14.4)	3.9 (54.5)	0.8 (36.4)	20.5 (26.7)	17.7 (32.8)
chatgpt3.5	32.7	**67.4** (8.9)	34.5 (12.0)	34.2 (10.5)	13.2 (10.5)	14.0 (9.7)	40.3 (10.1)	22.1 (10.7)
Z-ICL Prompt 2								
llama-2-7b	14.8	38.4 (11.2)	21.7 (17.8)	8.9 (12.1)	4.3 (16.0)	0.8 (15.9)	16.5 (14.3)	12.5 (14.9)
llama-2–13b	20.7	60.9 (12.8)	23.3 (23.6)	14.8 (12.8)	2.7 (55.3)	1.6 (23.6)	23.2 (23.1)	17.2 (29.2)
chatgpt3.5	33.4	65.5 (9.3)	36.8 (12.4)	31.5 (8.2)	18.7 (8.6)	14.3 (8.5)	39.7 (8.3)	24.7 (10.9)
Z-ICL Plain Text								
llama-2-7b	14.4	33.9 (18.9)	4.2 (89.1)	5.0 (70.0)	-	-	-	-
llama-2–13b	20.0	54.8 (18.2)	1.6 (95.3)	3.8 (92.2)	-	-	-	-

Table 5: Accuracy by type of answer and number of columns used, with type format errors in parentheses.

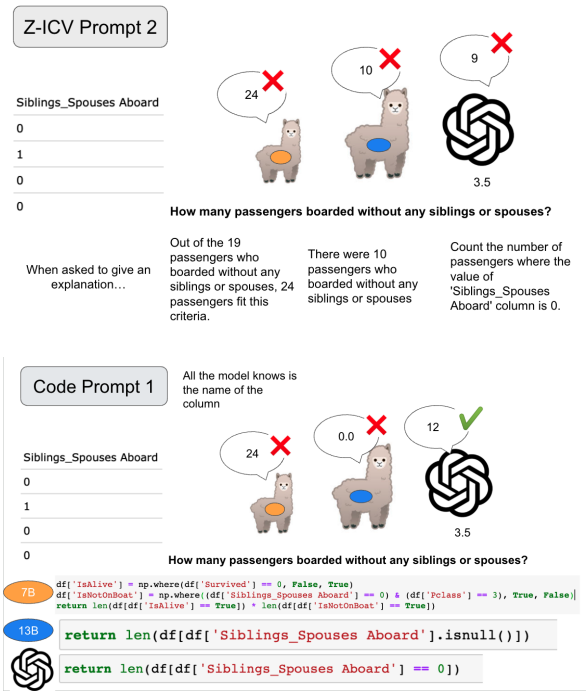


Figure 3: Sample answers of the three comparison models with Z-ICL (above) and code (below) prompts. The correct answer is 12.

overall. In general, they manage to better capture the format and achieve better performance as a result. Figure 3 provides two illustrative examples for both types of prompt in which all models output incorrect answers except for code-based ChatGPT. They also show how explanations may be incoherent or wrong, as we will see in Section 6.4.

In addition to the prompts tested requiring a JSON output, we also analysed the open models with a prompt called *plain text*, where only a plain-text answer was required (see Section 4.1.1). The results in Table 5 illustrate the very low success rates with this approach for categories and numbers in comparison with the JSON approach, which also makes it easier to process.

6.2. Type of Answer

As we can see in Table 5, the smaller models that use code are generally better at answering numerical than categorical answers, and are generally unreliable for answers that require a list as an answer. This effect is not as pronounced in ChatGPT, which fares generally better at almost any task. The models that rely on a Z-ICL prompt tend to be better at answering categorical questions than those that rely on code, but they struggle with numerical-based questions. They also fare notably better for boolean questions.

One possible explanation for the boolean behaviour may be similar to that observed in Lin et al. (2022), where the largest models were generally the less truthful. In this case, the accuracy remains roughly stable across the different types because the decreasing rate of code errors is compensated by a decreasing performance. In any case, boolean-based true-or-false questions seem to be harder to crack for these models than standard categorical/numerical questions, despite their reduced number of options. Surprisingly, not even ChatGPT attain results that are significantly higher than a random baseline in boolean questions. For code-based attempts in particular, this might be due to the inherent difficulty to generate code to check a boolean proposition, which may not be significantly lower than the code generated to check a numerical or categorical value. For example, com-

paring the question to compute the mean of a column to a question asking whether the mean of a column is above a certain value, the code for the second proposition is bound to be more complex even though it has only two possible outcomes. Performance on list-based questions is in general lower than the others, which is to be expected given the additional complexity they usually entail and the additional complexity their automated evaluation entails. In general, these differences between question types reinforce our initial categorization of the datasets, and can help further analyse the performance of models in a more targeted manner.

6.3. Code Errors

As can be observed in Figure 4, the smaller models generate invalid code in a rate much higher than ChatGPT. The figure also includes the different types of Python error. The most common errors are *KeyError* which happens when accessing a none existing value in a dictionary, or others such as *TypeError* that occur when an operation is performed on an unsupported type. The number of code errors differ from the two prompts, with the more complete second prompt being more reliable in LLaMA models compared to ChatGPT, which shows the opposite trend.

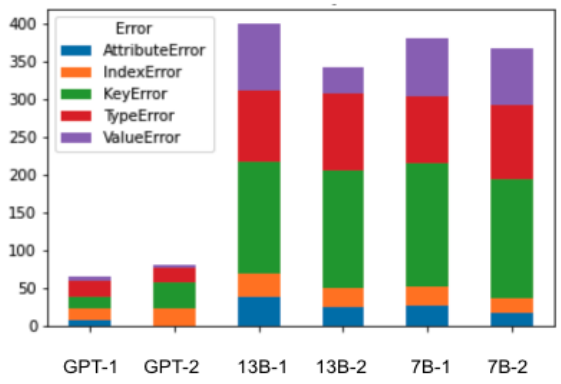


Figure 4: Most common code error types per model (chatgpt3.5, codellama-13b, codellama-7b) and Code Prompts (1 and 2).

6.4. Number of Columns

In Table 5 we included the averaged results depending on the number of columns required to answer a question, i.e., a single column or more than one. For questions that require the model to use more than one column, the results tend to lower than when using a single column. This result may be explained by the fact that using multiple columns requires a more advanced level of reasoning, which is something that LLMs may struggle with. In general, we observe both a decrease in accuracy and higher formatting errors when using multiple columns.

Columns used. For the Z-ICL JSON prompts we have also asked the model to respond with the columns used to get the answer in addition to the answer itself, which we analyse in Table 6. The first column represents the proportion of questions where the models have gotten the columns to use *wrong*, with the accuracy of the answer provided in such cases between parentheses. The second column shows the same for the cases the models got them *right*, and the last column represents the cases where the model failed to provide columns used or presented them in the wrong format for us to validate. As we can see, if the models have rightly identified the columns to extract the answer from, they are more likely to answer correctly. This jump in accuracy happens across all models with the two prompts tested. What is perhaps more surprising is that there is a non-negligible number of cases where the models are getting the right answer from, on their own account, the wrong columns. This is in line with LLMs hallucinating behaviour (Lin et al., 2022). A more detailed exploration of these and other explanations is however left for future work.

model	wrong cols	right cols	format error
Z-ICL Prompt 1			
llama-2-7b	63.4 (16.2)	18.5 (22.3)	18.2, (0.0)
llama-2-13b	51.2 (24.1)	22.7 (30.7)	26.1 (0.0)
chatgpt3.5	**31.6** (31.9)	**59.5** (**37.9**)	8.9 (0.0)
Z-ICL Prompt 2			
llama-2-7b	69.6 (15.5)	19.0 (21.2)	11.3, (0.0)
llama-2-13b	54.0 (23.3)	24.5 (32.9)	21.5, (0.0)
chatgpt3.5	34.9 (**35.0**)	56.7 (37.4)	**8.5** (**0.0**)

Table 6: Accuracy when predicting columns to use to get the answer, with answer accuracy for that case in parentheses.

7. Conclusion

In this paper, we release DataBench, a question answering benchmark over tabular data. The benchmark provides a useful tool to evaluate QA tabular data systems. This evaluation is complemented by domain, column and question categorizations that facilitate a simple diagnosis and analysis of the results. Our evaluation shows how current models, especially smaller ones, have significant room for improvement, especially when it comes to handling questions that require information from various columns. Finally, while in this paper we have focused on zero-shot QA answering evaluation, our benchmark can also serve as a basis to further trained, fine-tune or specialize models on QA over tabular data. We leave this further experimentation for future work.

Limitations

Our evaluation comes with a number of limitations. First, the number of tested models may be limited in number. For this evaluation we have focused on two of the latest general-purpose LLMs that have been shown a strong performance across tasks (namely LLaMA and GPT-based models), but may not be necessarily the best for this specific type of question answering over tabular data task. There are also recent models that could have been tested and may have provided different conclusions. Also, given computational constraints, we have not been able to run the largest models, which may provide a better performance. Second, the evaluation of generative LLMs in open question answering is notoriously difficult, as it is not always straightforward to enforce the desired output into the model. Because of this, we have tested various possibilities including enforcing the model to output an structured answer in the form of JSON. We acknowledge that this may not be the optimal solution as we are not only testing the capabilities of the model to answer the question correctly, but also their ability to generate JSON-style answers. Third, we evaluate LLMs only in a zero-shot context. They may perform better with examples or even fine-tuned on the task, which we leave for future work. Fourth, most of our evaluation is centered on the reduced version of DataBench with only 20 rows per datasets. This is done mainly given the limitations of current LLMs in handling larger amounts of data. Therefore, the numbers here may not reflect the real limitations of these models. The full results are available in Appendix A

With the full and open release of DataBench, we hope that this can facilitate further research in the future, including the development of new evaluation protocols over DataBench or similar datasets. Finally, our only focus in this paper is English, which may be limiting in nature given the variety of tabular data available in other languages. Our initial effort with DataBench should be extended to other languages, including low-resource ones which may pose additional challenges to current models.

Ethical Statement

All datasets included in this paper are obtained from public sources and under non-restrictive license. We have attributed the credit of all datasets accordingly (see Table 3) wihtout modifying them. We have included datasets without offensive or controversial language. While the models tested in our evaluation may be biased, we did check the output of models to ensure that the outputs did not include malicious language.

Acknowledgements

This work was partly supported by the grants PID2020-116118GA-I00 and TED2021-130145B-I00 funded by MCIN/AEI/10.13039/501100011033. Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

8. Bibliographical References

2016. County-to-county migration flows 2012-2016. https://www.census.gov/topics/population/migration/guidance/county-to-county-migration-flows.html. Accessed on 2023-10-20.

2018. Police department incident reports 2018 to present. data.gov, catalog.data.gov/dataset/police-department-incident-reports-2018-to-present. Data.gov dataset. URL: catalog.data.gov/dataset/police-department-incident-reports-2018-to-present.

2021. Kaggle dataset: Emoji diet nutritional data sr28. https://www.kaggle.com/datasets/ofrancisco/emoji-diet-nutritional-data-sr28. Accessed on 2023-10-20.

2021. Kaggle dataset: Venta de coches. https://www.kaggle.com/datasets/datamarket/venta-de-coches. Accessed on 2023-10-20.

2022. Aviation accidents history (1919 - april 2022) dataset. Kaggle, www.kaggle.com/datasets/ramjasmaurya/aviation-accidents-history1919-april-2022. Kaggle dataset. URL: www.kaggle.com/datasets/ramjasmaurya/aviation-accidents-history1919-april-2022.

2022. New york city weather 1869-2022 dataset. Kaggle, www.kaggle.com/datasets/danbraswell/new-york-city-weather-18692022. Kaggle dataset. URL: www.kaggle.com/datasets/danbraswell/new-york-city-weather-18692022.

2023. Airbnb listings in new york city dataset. Kaggle, www.kaggle.com/datasets/labdmitriy/airbnb. Kaggle dataset. URL: www.kaggle.com/datasets/labdmitriy/airbnb.

2023. Data scraped from idealista. Graphext.

2023. Fifa 21 complete player dataset. Kaggle, www.kaggle.com/datasets/stefanoleone992/fifa-21-complete-player-dataset.

Kaggle dataset. URL: www.kaggle.com/datasets/stefanoleone992/fifa-21-complete-player-dataset.

2023. Us tornado dataset 1950-2021. Kaggle, www.kaggle.com/datasets/danbraswell/us-tornado-dataset-1950-2021. Kaggle dataset. URL: www.kaggle.com/datasets/danbraswell/us-tornado-dataset-1950-2021.

Susant Achary. 2021. Holiday package purchase prediction dataset.

US Small Business Administration. 2021. Should this loan be approved or denied? Accessed on 2023-10-20.

AEMET. 2020. Últimos datos de observación de temperatura en madrid. Accessed on: 2020.

Abien Fred Agarap. 2018. Statistical analysis on e-commerce reviews, with sentiment classification using bidirectional recurrent neural network (rnn). *arXiv preprint arXiv:1805.03687*.

Inside Airbnb. Dec, 2022. Airbnb listings in madrid, spain (december, 2022). Dataset, Inside Airbnb. URL: http://insideairbnb.com/get-the-data/.

Alexandra. 2018. Generic food database. data.world, data.world/alexandra/generic-food-database. Data.world dataset. URL: data.world/alexandra/generic-food-database.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Rounak Banik. 2023. Pokemon feature correlation dataset. Kaggle.

Brandwatch. 2021. Us election raw polling data. https://www.brandwatch.com/p/us-election-raw-polling-data/. Accessed on 2023-10-20.

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. 2020. Language models are few-shot learners.

BSI. 1973a. *Natural Fibre Twines*, 3rd edition. British Standards Institution, London. BS 2570.

BSI. 1973b. Natural fibre twines. BS 2570, British Standards Institution, London. 3rd. edn.

A. Castor and L. E. Pollux. 1992. The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1):37–53.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Kaijie Zhu, Hao Chen, Linyi Yang, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. 2023. A survey on evaluation of large language models. *arXiv preprint arXiv:2307.03109*.

Lingjiao Chen, Matei Zaharia, and James Zou. 2023. How is chatgpt's behavior changing over time?

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021a. Evaluating large language models trained on code.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Yang Wang, and William W Cohen. 2020. Open question answering over tables and text. In *International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao

Huang, Bryan Routledge, and William Yang Wang. 2021b. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

J.L. Chercheur. 1994. *Case-Based Reasoning*, 2nd edition. Morgan Kaufman Publishers, San Mateo, CA.

Arush Chillar. 2023. Disneyland customer reviews dataset. Kaggle.

N. Chomsky. 1973. Conditions on transformations. In *A festschrift for Morris Halle*, New York. Holt, Rinehart & Winston.

CIS. 2015. 2015 spain political polls cis. https://public.graphext.com/90ca7539b160fdfa/index.html?section=data. Accessed on 2023-10-20.

Cerdeira A. Almeida F. Matos T. Cortez, Paulo and J. Reis. 2009. Wine Quality. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C56S3T.

Jirka Daberger. 2023. Clustering zoo animals. Kaggle.

Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. 2023. Llms to the moon? reddit market sentiment analysis with large language models. In *Companion Proceedings of the ACM Web Conference 2023*, WWW '23 Companion, page 1014–1019, New York, NY, USA. Association for Computing Machinery.

Umberto Eco. 1990. *The Limits of Interpretation*. Indian University Press.

Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. 2021. MATE: Multi-view attention for table transformer efficiency. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Forbes. 2022. Forbes billionaires. https://www.forbes.com/billionaires/. Accessed on 2023-10-20.

Xavier Vivancos García. 2020. Bakery market basket analysis dataset. Kaggle.

G. Gerganov. 2023. llama.cpp: Low-latency audio streaming library for c++. https://github.com/ggerganov/llama.cpp. Accessed: Sep 20, 2023.

Siavash Golkar, Mariel Pettee, Michael Eickenberg, Alberto Bietti, Miles Cranmer, Geraud Krawezik, Francois Lanusse, Michael McCabe, Ruben Ohana, Liam Parker, Bruno Régaldo-Saint Blancard, Tiberiu Tesileanu, Kyunghyun Cho, and Shirley Ho. 2023. xval: A continuous number encoding for large language models.

Google. 2021. Bigquery dataset: Patents. https://www.kaggle.com/datasets/bigquery/patents/data. Accessed on 2023-10-20.

Graphext. Trustpilot: Wise vs n26 reviews - scraped by graphext. https://public.graphext.com/367e29432331fbfd/index.html?section=data. Accessed on 2023-10-20.

Graphext. 2019. 2019 ing twitter mentions - scraped by graphext. https://public.graphext.com/075030310aa702c6/index.html. Accessed on 2023-10-20.

Graphext. 2020a. Trump tweets - scraped by graphext. https://public.graphext.com/be903c098a90e46f/index.html?section=data. Accessed on 2023-10-20.

Graphext. 2020b. X influencer analysis - scraped by graphext. https://public.graphext.com/e097f1ea03d761a9/index.html. Accessed on 2023-10-20.

Graphext. 2021a. Airline mentions on x - scraped by graphext. https://public.graphext.com/7e6999327d1f83fd/index.html. Accessed on 2023-10-20.

Graphext. 2021b. Data-driven seo: A keyword optimization guide using web scraping, co-occurrence analysis (graphext, deepnote, adwords). https://www.graphext.com/post/data-driven-seo-a-keyword-optimization-guide-using-web-scraping-co-occurrence-analysis-graphext-deepnote-adwords. Accessed on 2023-10-20.

Graphext. 2022a. Boris johnson tweets as pm - scraped by graphext. https://public.graphext.com/f6623a1ca0f41c8e/index.html. Accessed on 2023-10-20.

Graphext. 2022b. Joe biden tweets (scraped from x). https://public.graphext.com/339cee259f0a9b32/index.html?section=data. Accessed on 2023-10-20.

Graphext. 2023a. Love survey. Accessed on 2023-10-20.

Graphext. 2023b. Professional map. Graphext.

Heesoo37. 2023. 120 years of olympic history: Athletes and results dataset. Kaggle.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. 2021. Open domain question answering over tables via dense retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online. Association for Computational Linguistics.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Seattle, Washington, United States.

Paul Gerhard Hoel. 1971a. *Elementary Statistics*, 3rd edition. Wiley series in probability and mathematical statistics. Wiley, New York, Chichester. ISBN 0 471 40300.

Paul Gerhard Hoel. 1971b. *Elementary Statistics*, 3rd edition, Wiley series in probability and mathematical statistics, pages 19–33. Wiley, New York, Chichester. ISBN 0 471 40300.

Impapan. 2023. Predict student performance dataset. Kaggle.

INE. 2018. Encuesta cuatrienal de estructura salarial. INE Website.

Ashish Jangra. 2021. Ted talks - scraped from ted website. Kaggle.

Ram Jas. 2022. Telco customer churn dataset. Kaggle, www.kaggle.com/datasets/blastchar/telco-customer-churn. Kaggle dataset. URL: www.kaggle.com/datasets/blastchar/telco-customer-churn.

Otto Jespersen. 1922. *Language: Its Nature, Development, and Origin*. Allen and Unwin.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, and Zhaopeng Tu. 2023. Is chatgpt a good translator? a preliminary study. In *ArXiv*.

Nengzheng Jin, Joanna Siebert, Dongfang Li, and Qingcai Chen. 2022. A survey on table question answering: Recent advances. In *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers the Digital Economy*, pages 174–186, Singapore. Springer Nature Singapore.

Kaggle. 2021. Titanic dataset. Kaggle, www.kaggle.com/c/titanic. Dataset available on Kaggle. URL: www.kaggle.com/c/titanic.

Kaggle. 2023a. Bank customer churn. Kaggle.

Kaggle. 2023b. Employee satisfaction index dataset. Kaggle.

Kaggle. 2023c. Imdb movies. Kaggle.

Kaggle. 2023d. Professionals kaggle survey. Kaggle.

Kaggle. 2023e. Supermarket sales. Kaggle.

Margaret L. Kern, Paul X. McCarthy, Deepanjan Chakrabarty, and Marian-Andrei Rizoiu. 2019. Social media-predicted personality traits and values can help match people to their ideal jobs. *Proceedings of the National Academy of Sciences*, 116(52):26459–64.

Sunjun Kweon, Yeonsu Kwon, Seonhee Cho, Yohan Jo, and Edward Choi. 2023. Open-wikitable: Dataset for open domain question answering with complex reasoning over table.

Megagon Labs. 2017. Happy moments dataset. Kaggle.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods.

Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. 2022. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*.

Kaixin Ma, Hao Cheng, Xiaodong Liu, Eric Nyberg, and Jianfeng Gao. 2022. Open-domain question answering via chain of reasoning over heterogeneous knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5360–5374, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

UCI ML. 2015. Online Retail. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5BW33.

UCI ML. 2016. German credit. https://www.kaggle.com/datasets/uciml/german-credit/data. Accessed on 2023-10-20.

UCI ML. 2021. Heart failure prediction dataset. Kaggle, www.kaggle.com/datasets/fedesoriano/heart-failure-prediction. Originally published by UCI ML at https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease.

Rob Mulla. 2021. Roller coaster scraped from wikipedia. Kaggle, www.kaggle.com/datasets/robikscube/rollercoaster-database. Kaggle dataset. URL: www.kaggle.com/datasets/robikscube/rollercoaster-database.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, Dragomir Radev, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Hacker News. 2017. Hacker news dataset. Kaggle, www.kaggle.com/datasets/hacker-news/hacker-news. Kaggle dataset. URL: www.kaggle.com/datasets/hacker-news/hacker-news.

University of Brown. 2017. Billboard lyrics. https://www.kaggle.com/code/djohnbar/text-mining-of-billboard-lyrics-1964-2015.

University of Columbia. 2009. Speed dating.

University of Vanderbilt. 2019. Predict diabetes. Kaggle.

OpenAI. 2023. Gpt-4 technical report.

Panupong Pasupat and Percy Liang. 2015a. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015b. Compositional semantic parsing on semi-structured tables. *CoRR*, abs/1508.00305.

PavanKalyan. 2021. Predicting employee attrition. https://www.kaggle.com/datasets/pavan9065/predicting-employee-attrition. Accessed on 2023-10-20.

Kamil Pytlak. 2023. Stroke likelihood dataset. Kaggle.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Meg Risdal. 2017. New york city taxi trip duration.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code.

Ruchi798. 2023. Data science job salaries dataset. Kaggle.

Rodolfo Saldanha. 2020. Marketing campaign. https://www.kaggle.com/datasets/rodsaldanha/arketing-campaign. Accessed on 2023-10-20.

Charles Joseph Singer, E. J. Holmyard, and A. R. Hall, editors. 1954–58. *A history of technology*. Oxford University Press, London. 5 vol.

Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Nathaneal Scharli, Aakanksha Chowdhery, Philip Mansfield, Blaise Aguera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. 2022. Large language models encode clinical knowledge.

Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Jannik Strötgen and Michael Gertz. 2012. Temporal tagging on different domains: Challenges, strategies, and gold standards. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, pages 3746–3753, Istanbul, Turkey. European Language Resource Association (ELRA).

S. Superman, B. Batman, C. Catwoman, and S. Spiderman. 2000. *Superheroes experiences with*

books, 20th edition. The Phantom Editors Associates, Gotham City.

New York Times. 2021. Redivis dataset: Nyt world 2021. https://redivis.com/datasets/prrj-e3mazx6p3. Accessed on 2023-10-20. Sampled for 2021.

Tomigelo. 2023. Spotify audio features dataset. Kaggle.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models.

Omar Olivares Urrutia. 2018. Emoji diet. Kaggle.

Ellen M Voorhees. 2001. The trec question answering track. *Natural Language Engineering*, ftem7(4):361–378.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Superglue: a stickier benchmark for general-purpose language understanding systems. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, pages 3266–3280.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

WH. 2020. World happiness report 2020 data. https://worldhappiness.report/data. Accessed on 2023-10-20.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. 2023. Bloomberggpt: A large language model for finance.

Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

City Of New York. 2022. Nyc 311 data. https://data.cityofnewyork.us/Social-Services/NYC-311-Data/jrb2-thup. Accessed on 2023-10-20.

Qin Zhang, Shangsi Chen, Dongkuan Xu, Qingqing Cao, Xiaojun Chen, Trevor Cohn, and Meng Fang. 2023a. A survey for efficient open domain question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14447–14465, Toronto, Canada. Association for Computational Linguistics.

Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. 2023b. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Wenxuan Zhang, Yue Deng, Bing Liu, Sinno Jialin Pan, and Lidong Bing. 2023c. Sentiment analysis in the era of large language models: A reality check. *arXiv e-prints*, pages arXiv–2305.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Kaijie Zhu, Jindong Wang, Jiaheng Zhou, Zichen Wang, Hao Chen, Yidong Wang, Linyi Yang, Wei Ye, Neil Zhenqiang Gong, Yue Zhang, et al. 2023. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv preprint arXiv:2306.04528*.

A. Experimental Results by Domain

Tables Table 7 through Table 11 represent accuracy by type of answer and number of columns used, with type format errors in parentheses for each domain. They are equivalent to the results in Table 5 but only taking into account the datasets belonging to each domain. Although we can see some variations, in general there are only small changes and the analysis described in section 6 is also applicable here. The best model results in general come from *chatgpt3.5* with code prompts, while the *Z-ICL* prompts are better for boolean questions.

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	29.1	46.7 (37.1)	15.2 (67.6)	45.7 (35.2)	15.0 (38.3)	23.3 (23.3)	34.6 (38.3)	20.4 (43.8)
codellama-13b	29.9	49.5 (32.4)	27.6 (62.9)	42.9 (36.2)	18.7 (43.9)	10.7 (42.7)	33.3 (39.8)	24.4 (49.8)
chatgpt3.5	**60.4**	53.3 (4.8)	66.7 (13.3)	**78.1** (7.6)	**50.5** (5.6)	**53.4** (10.7)	**64.5** (8.0)	**53.7** (9.0)
Code Prompt 2								
codellama-7b	30.9	42.9 (39.0)	21.9 (54.3)	55.2 (26.7)	18.7 (37.4)	15.5 (31.1)	37.7 (34.3)	19.9 (43.3)
codellama-13b	33.9	52.4 (23.8)	29.5 (53.3)	53.3 (28.6)	17.8 (30.8)	16.5 (29.1)	38.9 (30.6)	25.9 (37.3)
chatgpt3.5	54.1	43.8 (15.2)	**66.7** (16.2)	76.2 (11.4)	42.1 (21.5)	41.7 (20.4)	60.8 (10.8)	43.3 (26.9)
Z-ICL Prompt 1								
llama-2-7b	13.9	37.6 (15.8)	17.8 (21.8)	8.9 (17.8)	5.0 (19.0)	0.0 (19.8)	16.5 (20.8)	11.2 (16.9)
llama-2–13b	19.0	55.4 (17.8)	19.8 (26.7)	13.9 (16.8)	4.0 (52.0)	2.0 (25.7)	21.6 (28.2)	16.5 (27.3)
chatgpt3.5	30.0	**58.4** (12.9)	30.7 (13.9)	28.7 (12.9)	22.0 (13.0)	9.9 (12.9)	36.5 (16.5)	23.3 (9.6)
Z-ICL Prompt 2								
llama-2-7b	13.3	40.6 (16.8)	14.9 (19.8)	8.9 (20.8)	2.0 (36.0)	0.0 (24.8)	15.7 (24.7)	10.8 (22.5)
llama-2–13b	18.1	52.5 (17.8)	16.8 (30.7)	13.9 (20.8)	6.0 (55.0)	1.0 (38.6)	19.2 (32.5)	16.9 (32.5)
chatgpt3.5	28.0	56.4 (13.9)	23.8 (14.9)	33.7 (14.9)	16.0 (15.0)	9.9 (12.9)	38.4 (18.8)	17.3 (9.6)

Table 7: Accuracy by type of answer and number of columns used, with type format errors in parentheses for domain Business

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	26.0	52.5 (35.0)	17.5 (57.5)	37.5 (30.0)	10.0 (40.0)	12.5 (30.0)	29.0 (34.7)	21.1 (44.7)
codellama-13b	34.0	72.5 (10.0)	22.5 (57.5)	52.5 (22.5)	12.5 (47.5)	10.0 (42.5)	39.5 (28.2)	25.0 (48.7)
chatgpt3.5	**59.0**	52.5 (5.0)	**70.0** (20.0)	**72.5** (0.0)	**60.0** (7.5)	**40.0** (17.5)	**60.5** (4.8)	**56.6** (18.4)
Code Prompt 2								
codellama-7b	25.0	47.5 (42.5)	25.0 (62.5)	37.5 (32.5)	7.5 (35.0)	7.5 (45.0)	29.8 (38.7)	17.1 (51.3)
codellama-13b	31.5	62.5 (17.5)	22.5 (67.5)	47.5 (25.0)	17.5 (37.5)	7.5 (37.5)	36.3 (33.1)	23.7 (43.4)
chatgpt3.5	44.5	60.0 (7.5)	42.5 (40.0)	57.5 (17.5)	32.5 (40.0)	30.0 (40.0)	48.4 (17.7)	38.2 (47.4)
Z-ICL Prompt 1								
llama-2-7b	18.5	52.5 (0.0)	27.5 (25.0)	5.0 (0.0)	5.0 (12.5)	2.5 (7.5)	19.3 (7.3)	16.0 (14.0)
llama-2–13b	23.0	67.5 (0.0)	32.5 (27.5)	15.0 (0.0)	0.0 (37.5)	0.0 (20.0)	22.0 (14.7)	26.0 (24.0)
chatgpt3.5	36.5	72.5 (0.0)	50.0 (12.5)	35.0 (0.0)	10.0 (0.0)	15.0 (0.0)	38.7 (0.7)	30.0 (8.0)
Z-ICL Prompt 2								
llama-2-7b	14.5	40.0 (10.0)	20.0 (25.0)	7.5 (2.5)	2.5 (37.5)	2.5 (20.0)	16.0 (15.3)	10.0 (30.0)
llama-2–13b	22.5	65.0 (5.0)	35.0 (25.0)	10.0 (2.5)	2.5 (55.0)	0.0 (35.0)	21.3 (18.7)	26.0 (42.0)
chatgpt3.5	36.5	**77.5** (0.0)	50.0 (12.5)	32.5 (5.0)	10.0 (2.5)	12.5 (5.0)	36.7 (3.3)	36.0 (10.0)

Table 8: Accuracy by type of answer and number of columns used, with type format errors in parentheses for Travel and Locations

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	23.8	42.2 (43.8)	15.6 (64.1)	39.1 (48.4)	11.5 (44.3)	10.6 (45.5)	31.6 (47.7)	11.9 (51.6)
codellama-13b	30.1	51.6 (34.4)	21.9 (65.6)	50.0 (35.9)	19.7 (44.3)	7.6 (40.9)	38.3 (36.8)	17.5 (55.6)
chatgpt3.5	**64.6**	53.1 (7.8)	**82.8** (6.2)	71.9 (10.9)	50.8 (8.2)	**63.6** (4.5)	69.9 (7.3)	**56.3** (7.9)
Code Prompt 2								
codellama-7b	27.3	43.8 (37.5)	26.6 (51.6)	37.5 (40.6)	13.1 (32.8)	15.2 (45.5)	33.2 (37.8)	18.3 (47.6)
codellama-13b	29.2	54.7 (29.7)	20.3 (64.1)	40.6 (43.8)	11.5 (54.1)	18.2 (39.4)	34.7 (42.0)	20.6 (52.4)
chatgpt3.5	61.1	45.3 (18.8)	75.0 (10.9)	**79.7** (12.5)	**50.8** (18.0)	54.5 (22.7)	**71.5** (11.4)	45.2 (24.6)
Z-ICL Prompt 1								
llama-2-7b	16.4	46.2 (7.7)	23.1 (15.4)	7.8 (7.8)	3.1 (13.8)	1.5 (15.4)	17.6 (6.8)	14.9 (18.2)
llama-2–13b	22.8	69.2 (9.2)	26.2 (26.2)	15.6 (7.8)	3.1 (61.5)	0.0 (26.2)	28.4 (17.0)	16.2 (37.2)
chatgpt3.5	32.7	**72.3** (10.8)	36.9 (10.8)	25.0 (6.2)	13.8 (7.7)	15.4 (7.7)	39.8 (2.3)	24.3 (16.2)
Z-ICL Prompt 2								
llama-2-7b	16.7	44.6 (7.7)	21.5 (15.4)	10.9 (10.9)	4.6 (32.3)	1.5 (24.6)	19.3 (12.5)	13.5 (25.0)
llama-2–13b	20.4	61.5 (12.3)	21.5 (33.8)	14.1 (6.2)	3.1 (49.2)	1.5 (33.8)	22.7 (23.3)	17.6 (31.8)
chatgpt3.5	32.7	69.2 (7.7)	43.1 (9.2)	28.1 (6.2)	7.7 (9.2)	15.4 (7.7)	40.3 (2.3)	23.6 (14.9)

Table 9: Accuracy by type of answer and number of columns used, with type format errors in parentheses for Social Networks and Surveys

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	36.4	50.0 (25.0)	28.6 (42.9)	60.7 (21.4)	21.4 (35.7)	21.4 (28.6)	45.7 (23.5)	23.7 (40.7)
codellama-13b	36.4	60.7 (25.0)	32.1 (57.1)	46.4 (25.0)	32.1 (32.1)	10.7 (35.7)	44.4 (28.4)	25.4 (44.1)
chatgpt3.5	**72.1**	53.6 (7.1)	**85.7** (14.3)	82.1 (10.7)	**78.6** (3.6)	60.7 (21.4)	70.4 (12.3)	**74.6** (10.2)
Code Prompt 2								
codellama-7b	38.6	46.4 (39.3)	28.6 (53.6)	75.0 (25.0)	17.9 (28.6)	25.0 (25.0)	54.3 (28.4)	16.9 (42.4)
codellama-13b	37.9	57.1 (17.9)	35.7 (57.1)	64.3 (28.6)	14.3 (39.3)	17.9 (35.7)	50.6 (24.7)	20.3 (50.8)
chatgpt3.5	66.4	39.3 (7.1)	67.9 (28.6)	**89.3** (10.7)	71.4 (3.6)	**64.3** (21.4)	**70.4** (13.6)	61.0 (15.3)
Z-ICL Prompt 1								
llama-2-7b	11.4	21.4 (14.3)	25.0 (7.1)	10.7 (14.3)	0.0 (14.3)	0.0 (14.3)	10.1 (18.0)	13.7 (3.9)
llama-2–13b	19.3	53.6 (14.3)	25.0 (7.1)	7.1 (25.0)	3.6 (71.4)	7.1 (21.4)	21.3 (28.1)	15.7 (27.5)
chatgpt3.5	40.0	71.4 (0.0)	42.9 (3.6)	39.3 (0.0)	25.0 (0.0)	21.4 (0.0)	43.8 (0.0)	33.3 (2.0)
Z-ICL Prompt 2								
llama-2-7b	14.3	25.0 (14.3)	28.6 (3.6)	14.3 (17.9)	3.6 (39.3)	0.0 (25.0)	12.4 (23.6)	17.6 (13.7)
llama-2–13b	20.7	60.7 (14.3)	25.0 (10.7)	14.3 (25.0)	3.6 (71.4)	0.0 (39.3)	22.5 (31.5)	17.6 (33.3)
chatgpt3.5	42.1	**85.7** (0.0)	39.3 (0.0)	42.9 (7.1)	21.4 (0.0)	21.4 (3.6)	49.4 (2.2)	29.4 (2.0)

Table 10: Accuracy by type of answer and number of columns used, with type format errors in parentheses for Health datasets

prompt,model	avg	boolean	category	number	list[category]	list[number]	single col	multiple cols
Code Prompt 1								
codellama-7b	21.8	36.0 (44.0)	12.0 (72.0)	33.3 (41.7)	16.0 (52.0)	12.0 (44.0)	29.5 (50.0)	17.5 (51.2)
codellama-13b	27.4	36.0 (44.0)	20.0 (68.0)	45.8 (29.2)	12.0 (56.0)	24.0 (28.0)	40.9 (40.9)	20.0 (47.5)
chatgpt3.5	**66.1**	48.0 (8.0)	**68.0** (12.0)	**75.0** (12.5)	**68.0** (12.0)	**72.0** (8.0)	**84.1** (4.5)	**56.2** (13.8)
Code Prompt 2								
codellama-7b	34.7	52.0 (36.0)	12.0 (64.0)	50.0 (41.7)	28.0 (36.0)	32.0 (32.0)	43.2 (36.4)	30.0 (45.0)
codellama-13b	37.1	48.0 (40.0)	32.0 (48.0)	54.2 (37.5)	28.0 (36.0)	24.0 (32.0)	43.2 (31.8)	33.8 (42.5)
chatgpt3.5	54.0	48.0 (20.0)	60.0 (32.0)	62.5 (33.3)	56.0 (32.0)	44.0 (48.0)	68.2 (25.0)	46.2 (37.5)
Z-ICL Prompt 1								
llama-2-7b	12.5	16.7 (16.7)	20.8 (8.3)	16.7 (16.7)	8.3 (16.7)	0.0 (16.7)	15.8 (19.7)	6.8 (6.8)
llama-2–13b	19.2	58.3 (20.8)	12.5 (16.7)	25.0 (16.7)	0.0 (62.5)	0.0 (16.7)	21.1 (30.3)	15.9 (20.5)
chatgpt3.5	36.7	58.3 (16.7)	33.3 (20.8)	45.8 (16.7)	25.0 (16.7)	20.8 (16.7)	47.4 (19.7)	18.2 (13.6)
Z-ICL Prompt 2								
llama-2-7b	12.5	20.8 (16.7)	20.8 (12.5)	16.7 (16.7)	4.2 (25.0)	0.0 (20.8)	15.8 (22.4)	6.8 (11.4)
llama-2–13b	15.0	41.7 (16.7)	16.7 (16.7)	16.7 (16.7)	0.0 (45.8)	0.0 (33.3)	15.8 (25.0)	13.6 (27.3)
chatgpt3.5	35.0	**70.8** (16.7)	25.0 (20.8)	45.8 (16.7)	12.5 (20.8)	20.8 (16.7)	43.4 (21.1)	20.5 (13.6)

Table 11: Accuracy by type of answer and number of columns used, with type format errors in parentheses for Sports and Entertainment