Procesamiento del Lenguaje Natural, Revista nº 73, septiembre de 2024, pp. 283-296 recibido 04-04-2024 revisado 09-05-2024 aceptado 17-05-2024

Towards Quality Benchmarking in Question Answering over Tabular Data in Spanish

Una Evaluación de Calidad en Preguntas y Respuestas sobre Datos Tabulares en Español

Jorge Osés Grijalba,[1,2] Luis Alfonso Ureña López,[2]
Jose Camacho-Collados,[3] Eugenio Martínez Cámara[2]
[1]Graphext
[2]SINAI Research Group. Advanced Studies Center in ICT (CEATIC)
Universidad de Jaén, Spain
[3]Cardiff University, UK
jorge@graphext.com, {laurena, emcamara}@ujaen.es, camachocolladosj@cardiff.ac.uk

Abstract: The rapid and incessant progress of language understanding and language generation capacity of large language models (LLMs) is followed by the discovery of new capabilities. The research community has to provide evaluation benchmarks to asses these emerging capabilities by studying, analysing and comparing different LLMs under fair and realistic settings. Question answering on tabular data is an important task to assess that lacks reliable evaluation benchmarks to assess LLMs in distinct scenarios, particularly for Spanish. Hence, in this paper we present Spa-DataBench, an evaluation benchmark composed of ten datasets about different topics of the Spanish society. Likewise, each dataset is linked to a set of questions written in Spanish and their corresponding answers. These questions are used to assess LLMs and analyse their capacity for answering questions that involve one single or multiple columns of different data types, and for generating source code to resolve the questions. We evaluate six LLMs on Spa-DataBench, and we compare their performance using both Spanish and English prompts. The results on Spa-DataBench show that LLMs are able to reason on tabular data, but their performance in Spanish is worse, which means that there is still room for improvement of LLMs in the Spanish language.
Keywords: Large language models, question answering, benchmark, tabular data.

Resumen: La evolución constante y veloz de la capacidad de compresión y generación de lenguaje de los modelos de lenguaje grandes (LLMs) va a acompañada del descrubimiento de nuevas habilidades. La evaluación de estas precisa de que la comunidad científica proporcione marcos de evaluación que permita el estudio, comparación y análisis de estas nuevas capacidades en diversos LLMs. La respuesta a preguntas a partir de datos en tablas es una de las nuevas capacidades de los LLMs, que aún carece de un *benchmark* de evaluación que permita analizarla en diferentes escenarios. Por tanto, en este trabajo se presenta Spa-DataBench, un *benchmark* de evaluación formado por diez conjuntos de datos sobre diferentes aspectos de la sociedad española. Cada conjunto de datos tiene asociado un conjunto de preguntas en español con sus respectivas respuestas, las cuales escrutan al LLM para estudiar su capacidad de responder preguntas que involucran una columa o varias sobre distintos tipos de datos, y de generar código fuente que permite la resolución de la pregunta. Se evalúan seis LLMs en Spa-DataBench, y se compara su rendimiento mediante el uso del mismo *prompt* escrito en inglés, debido a que los LLMs evaluados no han sido ajustados a usar *prompts* en español. Los resultados indican que los LLMs pueden razonar sobre datos tabulares, pero su rendimiento en español es inferior que en inglés, evidenciando que aún se debe seguir trabajando en mejorar el procesamiento del español de los LLMs.
Palabras clave: Modelos de lenguaje, respuesta a preguntas, *benchmark*, datos tabulares.

ISSN 1135-5948 DOI 10.26342/2024-73-21

1 Introduction

Recent work on Large Language Models (LLMs) has kickstarted a myriad of topics on natural language processing (NLP), especially since their scaling up as zero- and few-shot learners (Radford et al., 2019; Brown et al., 2020). These learning capabilities devoid of machine learning workflows enable the usage of objective-agnostic architectures to be employed in tasks such as sentiment analysis (Deng et al., 2023; Zhang et al., 2023c) or text summarization (Zhang et al., 2023b), to name a few. The release of general-purpose LLMs has contributed to this growth (Yang et al., 2023), leading to the discovery of emergent abilities (Wei et al., 2022). More recently, smaller open source models have become available which rival the capabilities of other bigger proprietary models (Jiang et al., 2023). Work in high quality large-scale benchmarks has not become yet as widespread for tasks that were considered niche prior to these emergent abilities.

Question Answering (QA) has traditionally focused on extracting answers from given questions in the context of plain text documents (Voorhees, 2001). QA has a number of different well established human-generated benchmarks like SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017), and NarrativeQA (Kočiskỳ et al., 2018). Question Generation (QG) as a related task refers to the task of extracting questions from a given text, usually from a list of given answers (Duan et al., 2017; Ushio, Alva-Manchego, and Camacho-Collados, 2022). Both tasks have traditionally yielded a heavy burden on human-generated benchmarks, and machine-generated QA has been incorporated successfully in parts of the task for benchmarking and even training models (Gururangan et al., 2018).

Given the versatility of LLMs, QA on tabular data has become an approachable task. In regards to the tabular aspect of the task, its structured nature has originally yielded itself to interfacing with tables through programming languages like SQL, but more recently thanks to the aforementioned emerging capabilities of LLMs more research is being done in answering natural language questions for tabular data (Chen, 2023). Likewise, the progress on a new task, in this case on a new LLM ability, needs the support of varied and robust evaluation benchmarks that allows to assess models on different scenarios, which in this case involves to judge different table domains with dissimilar column size and data types, a large variety of questions types whose answer required the data from one column or the combination of several ones, and a miscellaneous of row sizes to assess this capacity of LLMs on small and large amount of data. However, most current available datasets are focused on Wikipedia tables (Kweon et al., 2023) constraining the evaluation of this new capacity of LLMs on a specific domain, or are only available for English (Osés-Grijalba et al., 2024).

Regarding the above mentioned lack of evaluation benchmarks for QA on tabular data, we present in this paper Spa-DataBench, which is a large benchmark for assessing the QA on tabular data ability of LLMs on the Spanish language. Spa-DataBench is composed of 10 datasets with (1) data from a ample range of topics of the Spanish society, (2) distinct number of rows and columns, and (3) a substantial variety of data types. Since the main purpose of Spa-DataBench is to provide an evaluation benchmark, each dataset is linked to a set of gold 20 hand-made questions, with a total number of 200 questions. The questions are categorized by the data type of the answer (i.e. true/false, categories from the dataset, numbers or lists), and they are accompanied by their corresponding gold standard answer. Moreover, the structure of Spa-Databench defines how to incorporate new multilingual datasets by adding the tuple of (*dataset*, *questions*, *answers*), facilitating the future expansion of the benchmark. Therefore, Spa-Databench contributes by providing a reliable and balanced benchmark to study, analyze and compare the QA on tabular data capacity of LLMs using the Spanish language.

Finally, we assess the utility of Spa-Databench on six different LLMs, and we compare their ability to respond questions from five different data types and with a varying number of columns. We used task code completion as a bridge, since it allows to process the large datasets of Spa-DataBench that may not be covered by the context size of the LLM. Although the LLMs used in the evaluation are multilingual, they are not tuned to process prompts written in Spanish. Hence, we compare the Spanish evalu-

ation with the same prompt written in English. The results reached by the Spanish and the English prompt follow a similar tendency, with a slightly higher performance in the English language in general. This result suggests that LLMs needs to be adapted or at least trained on Spanish data in order to reach a similar performance than in English. Therefore, the contribution of a novel evaluation benchmark focused on Spanish provides to the community a new tool to measure the progress of LLMs tackling the Spanish language, and potentially to be used as a training or fine-tuning tool.

The remainder of the paper is structured as follows: Section 2 describes the main related works aligned with our proposal. Section 3 presents the construction details of Spa-DataBench. We describe the experimental setting to assess the utility of Spa-Databench in Section 4, and we analyze the results in Section 5 and code errors in Section 6. Finally, we present the main conclusions of our study in Section 7 and outline some limitations of this study in Section 8.

2 Related Work

Given the novelty of the task of QA on tabular data with LLMs, in this section we review the main related works, by first presenting QA-related works, then works focused on QA on tabular data mainly based on using SQL, and we finally talk about benchmarking in NLP and specifically on QA on tabular data.

Question Generation and Question Answering Most research on QG and QA rely on crowd-sourced methods to reduce the cost of creating new collections (Joshi et al., 2017). The trend historically has consisted on generating a set of questions from a given set of correct answers. Since creating new collections is costly, some research has been done with proposed methods for generating synthetic data for question-answer generation (Labutov, Basu, and Vanderwende, 2015). Apart from making use of annotated linguistic features, these early approaches to question generation were primarily rule-based, generated more questions than needed and then ranking them using a variety of metrics (Heilman and Smith, 2010; Lindberg et al., 2013).

This trend eventually evolved into more neural-network based work (Du, Shao, and Cardie, 2017) that started seeing better results over these early approaches, even employing reinforcement learning (Ling, An, and Hasan, 2017). Nowadays QA has a number of different well established human-generated benchmarks like SQuAD (Rajpurkar et al., 2016), TriviaQA (Joshi et al., 2017) or NarrativeQA (Kočiskỳ et al., 2018).

QA over Tabular Data aims to answer questions asked in natural language from data stored in tables (Jin et al., 2022). Different approaches exist to retrieve the answer, including semantically parsing the question to programming languages like SQL which are then used to interface with the data stored in a database (Pasupat and Liang, 2015a; Zhong, Xiong, and Socher, 2017; Nan et al., 2022). Open QA is a related task that processes only large collections of databases to answer specifically factoid questions (Zhang et al., 2023a).

Evaluation The evaluation of language models started with early unification initiatives for dataset collections such as GLUE or SuperGLUE (Wang et al., 2018; Wang et al., 2019), integrating several NLP datasets into a single benchmarks which were then solved by more modern language models(Yang et al., 2019), as the initial tasks were very limited as to the current capabilities of what LLMs can do. Other newer benchmarks include MMMU (Hendrycks et al., 2021) and BIG-Bench (Srivastava et al., 2022), specifically for LLMs. None of them, however, are related to Tabular QA or reasoning over databases.

Benchmarking in Tabular QA has been characterized by a number of different collections like (Pasupat and Liang, 2015a; Nan et al., 2022; Zhong, Xiong, and Socher, 2017; Pasupat and Liang, 2015b; Kweon et al., 2023) that nonetheless share the same underlying data tables taken from Wikipedia. These tables share a general set of attributes (low data variety, few columns in general) which makes them not be ideal when compared against datasets usually encountered in day-to-day industry. The most similar approach to ours is an approach for constructing an English heterogeneous benchmark for QA over tabular data (Osés-Grijalba et al., 2024). We follow a similar approach and extend it for the Spanish language, as well as provide the foundations for the evaluation of

	Name	Rows	Columns	#QA	Source (Reference)
1	Encuesta de Igualidad (Equality survey)	2000	105	20	40dB (40dB, 2024a)
2	Calidad del Sueño (Sleep quality)	2000	80	20	40dB (40dB, 2024b)
3	Fusión Barómetros (Fusion surveys)	7430	161	20	CIS (CIS, 2023b)
4	Barómetro Andaluz (Andalusian survey)	5349	85	20	CEA (CEA, 2023)
5	Juventud (Youth)	1510	236	20	CRS (CRS, 2023)
6	Política Fiscal (Tax policy)	3011	198	20	CIS (CIS, 2023c)
7	Relaciones (Relations)	2491	186	20	CIS (CIS, 2023a)
8	Barómetro Mensual (Monthly survey)	2444	185	20	CIS (CIS, 2021a)
9	Percepción del Amor (Love perception)	2000	150	20	40dB (40dB, 2022)
10	Salud Mental (Mental health)	3083	354	20	CIS (CIS, 2021b)
	Total:	31318	1741	200	

Table 1: The 10 datasets in DataBench with their number of rows and columns, number of questions and answers (#QA), as well as their source reference.

Question	Answer	Type	Columns Used	Column Types
¿Están jubiladas más de dos terceras partes de las mujeres entrevistadas?	false	boolean	Edad, Ocupación	number, category
¿Cuál es la identificación subjetiva de clase más común de las personas sin estudios?	Clase baja	category	Clase Social	category
¿Cuántos años tiene el estudiante más mayor?	55	number	Edad	number
¿Cuáles son las dos formas de conocer a su pareja más comunes entre los votantes del partido con menos intención de voto?	['bar', 'discoteca o fiesta']	list[category]	Voto, Lugar	category, category
Para los que no recuerdan a quién votaron en 2019, ¿Cuáles son sus 3 niveles de ubicación ideológica más comunes?	[5, 4, 7]	list[number]	Escala ideológica	number

Table 2: Examples of Question-Answer pairs present in Spa-DataBench. The translation of the table is in Appendix A.

LLMs in languages other than English for which resources are not available.

3 The Spa-DataBench Benchmark

We describe in this section all the details of the datasets of Spa-DataBench (Section 3.1), as well as how we build it (Section 3.2), and the data types and the types of questions included (Section 3.3).[1].

3.1 Datasets

Spa-DataBench is composed of ten tabular datasets from the biggest survey agencies in Spain, namely CIS, CEA, CRS and 40dB. These were already publicly available, and we have unified and typed them according to our typing system (see Section 3.2), which ease their processing. Since the purpose of Spa-DataBench is to serve as an evaluation benchmark, each dataset is linked, so far, to twenty hand-crafted questions with their corresponding gold answers which we have created specifically for this work, totaling 200 questions and answers. This structure makes Spa-DataBench to resemble a set of tuples of the form of *(dataset, questions, answers)*, which facilitates the inclusion of new datasets.

Table 1 shows the datasets of Spa-DataBench, all of them related to different aspects of the Spanish society. All of them differ in the number of rows and columns, which makes Spa-DataBench diverse in terms of dataset sizes. Moreover, since all the datasets are related to real sociological studies, it is possible to define different kind of

[1]Spa-DataBench will be released publicly with an open license upon acceptance at `https://huggingface.co/datasets/SINAI/databenchSPA`

Type	Columns	Example
number	269	1
category	1464	banana
date	2	1979-01-01
text	1	A blue rabbit went to...
list[number]	1	[10,11,12]
list[category]	4	[banana, pineapple]

Table 3: Column types present in our datasets.

questions that may involve one column or several columns in order to elaborate the answer. All the datasets have been checked to get the necessary permissions to do so in their licensing, and all the questions and answers will be shared publicly along with the datasets and source code upon acceptance.

Spa-DataBench can be added to the English DataBench benchmark (Osés-Grijalba et al., 2024) in order to perform multilingual evaluation of QA in tabular data, since both benchmarks follow a similar structure.

3.2 Column Types

Tabular Data can be broadly defined as a series of records (or *rows*) that share a number of common attributes (or *columns*). This typing system consists of a human-adapted version of Apache Parquet's[2] typing system, and the open source library Lector (Buhrmann, 2023) makes it possible to automatically tag any given dataset with this typing system without human intervention. The type of data that these columns contains greatly influences how we interact with them, as we may need to perform mathematical operations on columns containing prices and performing natural language processing tasks on those containing texts such as reviews. Correctly tagging these columns will in turn provide us with a better picture of where our models are faring better or worse. Commonly found *data types* can be seen in Table 3.

3.3 Question Types

Data Types From the Column Type categorization stems the *question categorization* we have used. In order to develop a metric that can be evaluated easily, in the classical sense all 200 of our proposed QA pairs belong to a *factoid* categorization. We have further tagged each question with the type of answer, so we can better analyse where our models do better or worse. You can see examples of how the different types are tagged in Table 6

Complexity One way to roughly classify the complexity of a given question is to ask where the information they need to answer is contained. Those that require the information present only in one column, like asking for *how old is the oldest person* in a given set where we only have to check the *Age*, is inherently simpler than a question that requires us to answer *how old the oldest woman is*, where we may have to first perform a filter on *Gender* and then sort by *Age*. We have thus tagged each QA with the number of columns they require to be answered, and we will be analyzing how this relates to model performance in the analysis section.

4 Experimental Framework

The task of QA over tabular data has traditionally been conducted by generating source code, and in most of the cases SQL source code, because the aim was to access to data stored in relational databases. Indeed, the most recent dataset for QA on Tabular data, OpenWikitables (Kweon et al., 2023), is also designed to evaluate the generation of SQL code for querying Wikipedia data stored in relational databases. In our case, we used *code completion* as as bridge task because (1) we are working with large datasets that may not fit in the available context of the models used in the evaluation, and (2) to present a similar approach than others related works in QA in tabular data as the ones described in Section 2. Since our datasets are not stored in relational databases, we preferred to generate Python code instead of SQL code. Additionally, we also force LLM to use popular libreries for data science as Numpy and Pandas, because the questions require the manipulation of data, which can be easily conducted with those libraries.

As part of the task, we will first ask the models to complete a given function that receives a Pandas dataframe, representing the dataset that contains the answer to the user question. We also perform the renaming of the columns with the actual names, which gives the model enough information to access the columns and perform operations on them. We have encapsulated these behaviors in the prompt to resolve the questions, which Prompt 1 shows for the Spanish questions.

[2] `https://arrow.apache.org/`

```
import pandas as pd
import numpy as np

"""
Eres un asistente de código en Python. Debes completar la declaración de retorno de la función 'answer' para que responda la pregunta indicada en el comentario.
"""
def answer(df):
  """
  Esta función devuelve la respuesta a: ¿Cuál es la edad del jubilado más joven?
  """
  df.columns=['Edad', 'Ocupación']
```

Prompt 1: Code completion prompt example used for the task in Spanish.

```
import pandas as pd
import numpy as np

"""
You are a Python code assistant. You are to complete the function 'answer'
return statement so it answers the question stated in the comment.
"""
def answer(df):
  """
  This function returns the answer to: ¿Cuál es la edad del jubilado más joven?
  """
  df.columns=['Edad', 'Ocupación']
```

Prompt 2: Code completion prompt example used for the task in English.

Since the models are not tuned to Spanish, we also assess them with the equivalent questions in English (see Prompt 2).

For the purpose of simpler questions, one might argue that something like SQL or other query-specific languages are simpler and thus may work better, but given the advances of LLMs in general we think that developing interfaces that have access to full programming languages open up a wide array of possibilities that can pave the way for future research in the area. With this approach, advanced enough models would be able to perform tf-idf on columns of texts, for example, or retrieve URL content from urls and benefit from any third party library suited to fill the user's needs.

Models Our experiments are performed on a number of small open-source models, all of them containing 7 billion parameters except of codellama13, which has 13 billion parameters. Our goal for this evaluation is not to achieve the best results possible, but rather to analyze the impact of the language used in the instructions and to analyze the different types of errors encountered during our experiments, in addition to judge how these smaller models which can run on most consumer hardware are doing in regards of the task of QA over Tabular Data. In particular, the LLMs considered in the evaluation are:

- Codellama 7b & 13b (Rozière et al., 2023) finetuned versions of llama2 to run on code. We will use *CodeLlama-7B-Instruct* and *CodeLlama-13B-Instruct*, respectively.
- Mistral (Jiang et al., 2023), a 7-billion-parameter language model engineered for superior performance and efficiency. The version we'll be using is *Mistral-7B-Instruct-v0.2*, which is only trained in the English language.
- Zephyr (Tunstall et al., 2023) a smaller language model that is aligned to user intent. We will be using *zephyr-7B-beta*, trained in primarily english.
- OpenChat (Wang et al., 2023) fine-tuned on a diverse and high-quality dataset of multi-round conversations in English. We will make use of *openchat_3.5*.
- Deepseek-coder (Guo et al., 2024), which was designed for code generation, but with a mixture of English and Chinese in natural language. Our version will be *deepseek-coder-6.7B-instruct*.

For each model, we will use their 4bit-quantified version.

Evaluation Given the high volatility of these models, we will evaluate their average accuracy over 5 different executions on a low temperature setup to reduce variance. The main evaluation metric utilised is accuracy, in which we compare the model answer prediction with the gold standard answer.

Type Evaluation Aside from the general evaluation, we will require the models to output the desired types specifically. For example, if a model answers "one" instead of 1, we would not take it as a correct answer. The reason behind this decision, even though it hinders model performance on paper, is that the systems that employ these models need to be integrated into larger automated systems reliably, in order to perform more complex tasks. The evaluation scripts will be open sourced along with the rest of the code and the data.

5 Result Analysis

We analysis the results according to different aspects as the types of data, the language of the prompt and the number of columns need to resolve the question. Additionally, we compare the results with an English benchmark for QA on Tabular data. However, we first give a general picture of the results, which are summarised in Table 4.

The best model is codellama-13b with a 23% overall accuracy score, while the best among the 7b ones remains the *deepseek* model in English (21% accuracy) while its performance falls off using a Spanish prompt. The worst results are achieved by *zephyr* both in English and Spanish. Nonetheless, all results are relatively similar, between 15% and 23% accuracy scores. In general, this suggests that there is a clear room for improvement for all models.

Types & Accuracy Boolean questions seem to perform around or below the 50% baseline, but this should not be surprising since the operations required to answer them are similar to the rest of the types.

Performance on numbers and categories seems to be around the same, with lists lagging behind on all models producing worse accuracy results and lower proportions of executable code. Lower performance on lists is to be expected, since these put extra requirements, we need to extract certain subsets and make sure it is in the expected order, thus have an inherent complexity to them that simpler types do not.

Language & Accuracy There were small differences between Spanish and English instructions, especially for some models like *deepseek*. This is especially relevant for *deepseek* since it was only trained in English and Chinese coding snippets, so it seems to lose a lot of performance in English. This gap is not as large for other models that were not trained using Spanish data but were trained on English natural language.

Number of Columns We can observe that for all models questions where the information necessary to answer them is included in one column are easier to answer. For example, the best model codellama13 using the Spanish prompt achieves an accuracy score of 30.2% in questions that require a single column, while the accuracy lowers to 14.4% for questions requiring multiple columns.

Comparison with English benchmark For completeness, we compare with the results achieved at the sibling benchmark DataBench (Osés-Grijalba et al., 2024), which is composed of English data.While the models compared are not exactly the same, the results on English data are stronger than with the Spanish data (33.1% accuracy for the best 13B codellama model in comparison with 30.2% in Spa-DataBench), which suggests the Spanish language needs specific LLMs, or at least fined-tuned to better understand Spanish.

6 Code Errors

One way we can approximate how far our models are from getting high accuracy scores is checking the percentage of code that is generated correctly. In order to generate an answer first, the models have to be able to generate code than can be run without errors. Table 4 shows the percentage of code generate that can be executed successfully in parenthesis. Subsequently, we describe the main errors found in the source code generated by the models.

Number of Columns & Code Errors Questions asked over a single column have considerably higher rates of successful code production. This is expected and goes back to these questions being inherently simpler

prompt,model	AVG	boolean	category	number	list[category]	list[number]	single col	multiple cols
Spanish								
codellama	17.5 (35.5)	47.5 (27.5)	10.0 (47.5)	15.0 (40.0)	5.0 (37.5)	10.0 (25.0)	22.8 (28.53)	11.1 (43.85)
codellama13	**23.0** (28.0)	52.5 (12.5)	15.0 (42.5)	20.0 (35.0)	10.0 (32.5)	**17.5** (17.5)	29.2 (20.2)	**15.5** (37.2)
mistral	19.0 (41.5)	47.5 (25.0)	12.5 (47.5)	15.0 (60.0)	7.5 (32.5)	12.5 (42.5)	22.8 (35.9)	14.4 (48.2)
zephyr	15.0 (53.5)	25.0 (40.0)	15.0 (55.0)	**22.5** (57.5)	5.0 (55.0)	7.5 (60.0)	17.3 (45.9)	12.2 (72.5)
openchat	18.0 (44.0)	45.0 (25.0)	12.5 (42.5)	15.0 (47.5)	7.5 (47.5)	10.0 (57.5)	24.7 (35.9)	10.3 (53.7)
deepseek	14.5 (63.5)	25.0 (42.5)	12.5 (52.5)	12.5 (70.0)	12.5 (77.5)	10.0 (75.0)	15.5 (56.9)	13.3 (71.3)
English								
codellama	19.5 (30.0)	52.5 (15.0)	17.5 (47.5)	15.0 (20.0)	7.5 (37.5)	5.0 (30.0)	25.6 (18.5)	12.2 (43.8)
codellama13	23.0 (28.5)	**55.0** (15.0)	17.5 (42.5)	20.0 (32.5)	10.0 (32.5)	12.5 (20.0)	**30.2** (32.2)	14.4 (36.1)
mistral	18.0 (38.5)	40.0(30.0)	12.5 (45.0)	15.0 (45.0)	12.5 (42.5)	10.0 (40.0)	21.9 (32.2)	13.3 (46.0)
zephyr	18.5 (39.5)	45.0 (35.0)	12.5 (35.0)	15.0 (42.5)	10.5 (42.5)	9.5 (42.5)	23.8 (34.9)	12.2 (54.9)
openchat	19.0 (38.0)	45.0 (35.0)	12.5 (30.0)	27.5 (42.5)	5.0 (45.0)	5.0 (47.5)	26.5 (30.3)	10.0 (47.1)
deepseek	21.0 (30.0)	35.0 (35.0)	**17.5** (37.5)	20.0 (27.5)	**15.0** (22.5)	17.5 (27.5)	25.6 (25.8)	15.5 (35.0)

Table 4: Accuracy by type of answer and number of columns used, for Spanish questions when providing the instructions in Spanish or English respectively. Total code error percentages between parentheses.

Error Type	**English**	**Spanish**	**Single**	**Multi**	**Short explanation**
AttributeError	4.77	3.67	4.78	3.69	Object lacks the attribute.
FileNotFoundError	3.28	0.73	2.31	1.70	Specified file or directory not found.
IndentationError	1.79	0.88	1.08	1.56	Bad code indentation.
IndexError	3.58	3.96	2.01	5.39	Invalid sequence index
KeyError	25.78	23.31	21.60	27.23	Dictionary lacks specified key.
ModuleNotFoundError	5.96	2.35	6.02	2.41	Tries to import external module.
NameError	1.34	1.32	1.39	1.28	Undefined variable or function name.
SyntaxError	**30.25**	**43.99**	**39.20**	**35.32**	Code syntax is incorrect.
TypeError	20.12	15.54	18.36	17.30	Inappropriate type usage.
ValueError	3.13	4.11	3.24	3.97	Inappropriate value for argument.
ZeroDivisionError	0.00	0.15	0.00	0.14	Division by zero attempted.

Table 5: Percentage of errors for each type within the errors generated by all models.

than those that involve more than one column to elaborate the answer.

Language & Code Errors The instructions provided in Spanish obtain a higher percentage of errors in all models. This is specially true for *deepseek*, where it goes from 40% to 73.5%. Hence, the models show some grade of weakness in the processing of prompt written in Spanish at least in the bridge task of code completion.

Python errors As can be seen in Table 5, there are a number of Python errors generated that are more present throughout the dataset. In the table we can find a short description of what each type of error means, as well as the percentage of that error within the total errors generated for each subset of our tests. We can observe a difference in errors produced in English vs Spanish and for single and multi-column approaches. In particular, syntax errors (*SyntaxError*) appear to be more prevalent in Spanish, where they make up almost 44% of the error count, while only 30% in English. Other differences include those in *KeyErrors* where they are 6% higher in multi-column questions compared to single-column questions.

In general, the pattern of error proportions remains shockingly consistent, with models tending to fail with similar proportions for the given set of questions. It may be related to the task at hand presenting unique hardships for the models that could be further studied, but that is would be subject for future work.

6.1 Error examples

In order to illustrate the kind of errors that we're encountering in our analysis, we will take a look at two cases.

FileNotFoundError A typical error made by the models consists on acting as if the function structure to answer was not provided, and instead try to load the data from a csv file. Loading the data is something very commonly found in code snippets all over the internet, so it makes sense that

```python
import pandas as pd
import numpy as np

# df = pd.read_csv("myfile.csv")
"""
Eres un asistente de código en Python. Debes completar la declaración de retorno de la función 'answer' para que responda a la pregunta indicada en el comentario.
"""
def answer(df: pd.DataFrame):
    '''Devuelve la respuesta a: ¿Es 18-24 el grupo de edad más frecuente entre los hombres?'''
    df.columns = [...] # column names are provided, excluded for readability
    # from here on: generated code
    if __name__ == "__main__":
        df = pd.read_csv("myfile.csv")
```

Prompt 3: Syntax error example.

```python
import pandas as pd
import numpy as np

# df = pd.read_csv("myfile.csv")
"""
Eres un asistente de código en Python. Debes completar la declaración de retorno de la función 'answer' para que responda a la pregunta indicada en el comentario.
"""
def answer(df: pd.DataFrame):
    '''Devuelve la respuesta a: Para los que no recuerdan a quién votaron en 2019, ¿Cuáles son sus 3 niveles de ubicación ideológica más comunes?'''
    df.columns = [...] # column names are provided, excluded for readability
    # from here on: generated code
    return df[['Más afecto', 'Vecinos_as', 'Familiares', 'Supervisor']].mean().tolist()
```

Prompt 4: Type error example.

they're biased towards it. Nevertheless it happens infrequently, and the models should understand by the context provided that their task does not involve loading any data from disk. We can see an example of this in Prompt 3.

TypeError Another commonly found error is the TypeError, where the model tries to use a function for a data type that is not correct. We can see this in action in Prompt 4, which yields exactly *TypeError: 'Categorical' with dtype category does not support reduction 'mean'*

7 Conclusion and Future Work

In this paper, we have contributed with a novel benchmark for QA over Tabular Data and provided a reliable framework to perform evaluation in Spanish. The results highlight the the currently weak results achieved with open source models. This is correlated with the smaller performance obtained with English prompts over English data, so the challenge remains in developing robust systems capable of answering these questions reliably, especially for open-source models. Our analysis also shows some universal results that also apply to Spanish, such as questions that need several columns' information are indeed harder to answer than those that require the information contained in only one column, and that list-like questions present a harder challenge. When the instructions provided are in Spanish models tend to generate significantly more syntax-related errors and could impact some models negatively.

Finally, more work is still needed to develop a robust benchmark for QA over Tabular Data in Spanish and other languages, especially other than English. The costly nature of QA tagged sets makes the process of exploring other options, like procedurally generating automated questions heuristically based on datasets' structure, worth explor-

ing.

8 Limitations

The relatively small size of the data does not allow us to jump to definite conclusions with respect to the results. Our evaluation was performed on a limited set of models on the smaller end of the LLMs and of around the same size, but a wider variety of models could be evaluated, as well as different settings and prompts not explored on this paper. Finally, the benchmark is limited to 200 questions and answers, which are limited to be used for training (or fine-tuning) and evaluation.

Acknowledgements

This work was partly supported by the grants FedDAP (PID2020-116118GA-I00), MODERATES (TED2021-130145B-I00), SocialTOX (PDC2022-133146-C21) and CONSENSO (PID2021-122263OB- C21) funded by MCIN/AEI/10.13039/501100011033, "ERDF A way of making Europe" and "European Union NextGenerationEU/PRTR". Jose Camacho-Collados is supported by a UKRI Future Leaders Fellowship.

References

40dB, E. P. 2022. Percepción del amor. https://elpais.com/sociedad/2022-06-05/consulte-todos-los-datos-internos-de-la-encuesta-de-el-pais-sobre-la-percepcion-del-amor-cuestionarios-y-respuestas-individuales.html.

40dB, E. P. 2024a. Encuesta de igualdad marzo 2024. https://elpais.com/espana/2024-03-11/consulte-todos-los-datos-internos-de-la-encuesta-de-el-pais-de-marzo-cuestionarios-cruces-y-respuestas.html.

40dB, E. P. 2024b. Encuesta sobre el sueño. https://elpais.com/ciencia/2024-02-25/consulte-todos-los-datos-internos-del-barometro-de-el-pais-cuestionarios-cruces-y-respuestas-individuales.html.

Brown, T. B., B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. 2020. Language models are few-shot learners.

Buhrmann, T. 2023. Lector, dec.

CEA. 2023. Barómetro andaluz septiembre 2023. https://www.centrodeestudiosandaluces.es/barometro/barometro-andaluz-de-septiembre-2023.

Chen, W. 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, May. Association for Computational Linguistics.

CIS. 2021a. Salud mental durante la pandemia. https://www.cis.es/es/detalle-ficha-estudio?idEstudio=14676.

CIS. 2021b. Salud mental durante la pandemia. https://datos.gob.es/es/catalogo/ea0022266-2193comportamiento-de-los-espanoles-ante-las-vacaciones-iii.

CIS. 2023a. Cis - relaciones afectivas pospandemia iii. https://www.cis.es/detalle-ficha-estudio?origen=estudio&idEstudio=14702.

CIS. 2023b. Fusión barómetros enero-marzo 2023. https://www.cis.es/es/detalle-ficha-estudio?idEstudio=14707.

CIS. 2023c. Opinión pública y política fiscal julio 2023. https://www.cis.es/detalle-ficha-estudio?origen=estudio&idEstudio=14741.

CRS. 2023. Barómetro juventud, salud y bienestar 2023. https://www.centroreinasofia.org/publicacion/barometro-salud-2023/.

Deng, X., V. Bashlovkina, F. Han, S. Baumgartner, and M. 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.

Du, X., J. Shao, and C. Cardie. 2017. Learning to ask: Neural question generation for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1342–1352,

Vancouver, Canada, July. Association for Computational Linguistics.

Duan, N., D. Tang, P. Chen, and M. Zhou. 2017. Question generation for question answering. In M. Palmer, R. Hwa, and S. Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark, September. Association for Computational Linguistics.

Guo, D., Q. Zhu, D. Yang, Z. Xie, K. Dong, W. Zhang, G. Chen, X. Bi, Y. Wu, Y. Li, F. Luo, Y. Xiong, and W. Liang. 2024. Deepseek-coder: When the large language model meets programming – the rise of code intelligence.

Gururangan, S., S. Swayamdipta, O. Levy, R. Schwartz, S. Bowman, and N. A. Smith. 2018. Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June. Association for Computational Linguistics.

Heilman, M. and N. A. Smith. 2010. Good question! statistical ranking for question generation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 609–617, Los Angeles, California, June. Association for Computational Linguistics.

Hendrycks, D., C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. 2021. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Jiang, A. Q., A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. de las Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier, L. R. Lavaud, M.-A. Lachaux, P. Stock, T. L. Scao, T. Lavril, T. Wang, T. Lacroix, and W. E. Sayed. 2023. Mistral 7b.

Jin, N., J. Siebert, D. Li, and Q. Chen. 2022. A survey on table question answering: Recent advances. In M. Sun, G. Qi, K. Liu, J. Ren, B. Xu, Y. Feng, Y. Liu, and Y. Chen, editors, *Knowledge Graph and Semantic Computing: Knowledge Graph Empowers the Digital Economy*, pages 174–186, Singapore. Springer Nature Singapore.

Joshi, M., E. Choi, D. S. Weld, and L. Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611.

Kočiskỳ, T., J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette. 2018. The narrativeqa reading comprehension challenge. *Transactions of the Association for Computational Linguistics*, 6:317–328.

Kweon, S., Y. Kwon, S. Cho, Y. Jo, and E. Choi. 2023. Open-WikiTable : Dataset for open domain question answering with complex reasoning over table. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8285–8297, Toronto, Canada, July. Association for Computational Linguistics.

Labutov, I., S. Basu, and L. Vanderwende. 2015. Deep questions without deep understanding. 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 889–898, Beijing, China, July. Association for Computational Linguistics.

Lindberg, D., F. Popowich, J. Nesbit, and P. Winne. 2013. Generating natural language questions to support learning online. In *Proceedings of the 14th European Workshop on Natural Language Generation*, pages 105–114, Sofia, Bulgaria, August. Association for Computational Linguistics.

Ling, Y., Y. An, and S. Hasan. 2017. Improving clinical diagnosis inference through integration of structured and unstructured knowledge. In *Proceedings of the 1st Workshop on Sense, Concept and*

Entity Representations and their Applications, pages 31–36, Valencia, Spain, April. Association for Computational Linguistics.

Nan, L., C. Hsieh, Z. Mao, X. V. Lin, N. Verma, R. Zhang, W. Kryściński, H. Schoelkopf, R. Kong, X. Tang, M. Mutuma, B. Rosand, I. Trindade, R. Bandaru, J. Cunningham, C. Xiong, D. Radev, and D. Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Osés-Grijalba, J., L. A. Ureña-López, E. M. Cámara, and J. Camacho-Collados. 2024. Question answering over tabular data with databench: A large-scale empirical evaluation of llms. In *Proceedings of LREC-COLING 2024*, Turin, Italy.

Pasupat, P. and P. 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, July. Association for Computational Linguistics.

Pasupat, P. and P. Liang. 2015b. Compositional semantic parsing on semi-structured tables.

Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, et al. 2019. Language models are unsupervised multitask learners. Technical report, OpenAI.

Rajpurkar, P., J. Zhang, K. Lopyrev, and P. Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas, November. Association for Computational Linguistics.

Rozière, B., J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin, A. Kozhevnikov, I. Evtimov, J. Bitton, M. Bhatt, C. C. Ferrer, A. Grattafiori, W. Xiong, A. Défossez, J. Copet, F. Azhar, H. Touvron, L. Martin, N. Usunier, T. Scialom, and G. Synnaeve. 2023. Code llama: Open foundation models for code.

Srivastava, A., A. Rastogi, A. Rao, A. A. M. Shoeb, A. Abid, A. Fisch, A. R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al. 2022. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *arXiv preprint arXiv:2206.04615*.

Tunstall, L., E. Beeching, N. Lambert, N. Rajani, K. Rasul, Y. Belkada, S. Huang, L. von Werra, C. Fourrier, N. Habib, N. Sarrazin, O. Sanseviero, A. M. Rush, and T. Wolf. 2023. Zephyr: Direct distillation of lm alignment.

Ushio, A., F. Alva-Manchego, and J. Camacho-Collados. 2022. Generative language models for paragraph-level question generation. In Y. Goldberg, Z. Kozareva, and Y. Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 670–688, Abu Dhabi, United Arab Emirates, December. Association for Computational Linguistics.

Voorhees, E. M. 2001. The trec question answering track. *Natural Language Engineering*, 7(4):361–378.

Wang, A., Y. Pruksachatkun, N. Nangia, A. Singh, J. Michael, F. Hill, O. Levy, and S. 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.

Wang, A., A. Singh, J. Michael, F. Hill, O. Levy, and S. 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, November. Association for Computational Linguistics.

Wang, G., S. Cheng, Q. Yu, and C. Liu. 2023. OpenLLMs: Less is More for Open-source Models, 7.

Wei, J., Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler, E. H. Chi, T. Hashimoto, O. Vinyals, P. Liang, J. Dean, and W. Fedus. 2022. Emergent

abilities of large language models. *Transactions on Machine Learning Research*. Survey Certification.

Yang, J., H. Jin, R. Tang, X. Han, Q. Feng, H. Jiang, B. Yin, and X. Hu. 2023. Harnessing the power of llms in practice: A survey on chatgpt and beyond. *arXiv preprint arXiv:2304.13712*.

Yang, Z., Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhang, Q., S. Chen, D. Xu, Q. Cao, X. Chen, T. Cohn, and M. 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, July. Association for Computational Linguistics.

Zhang, T., F. Ladhak, E. Durmus, P. Liang, K. McKeown, and T. B. Hashimoto. 2023b. Benchmarking large language models for news summarization. *arXiv preprint arXiv:2301.13848*.

Zhang, W., Y. Deng, B. Liu, S. Jialin Pan, and L. Bing. 2023c. Sentiment analysis in the era of large language models: A reality check. *arXiv e-prints*, pages arXiv–2305.

Zhong, V., C. Xiong, and R. Socher. 2017. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint arXiv:1709.00103*.

A English translation of Questions-Answer pairs

We show the English translation of Table 6.

Question	Answer	Type	Columns Used	Column Types
Are more than two-thirds of the women interviewed retired?	false	boolean	Age, Ocupation	number, category
What is the most common subjective social class identification of uneducated people?	Lower class	category	Social class	category
How old are the oldest student?	55	number	Age	number
What are the two most common ways of meeting your partner among voters of the party with the least intention to vote?	['bar', 'disco or party']	list[category]	Vote, Place	category, category
For those who don't remember who they voted for in 2019, what are your 3 most common levels of ideological location?	[5, 4, 7]	list[number]	Ideology scale	number

Table 6: Examples of Question-Answer pairs present in Spa-DataBench.