UC-UCO-Plenitas Team - Exploring in the PRESTA 2025 challenge: Question Answering over Tabular Data in Spanish

Yoan Martínez-López[1,2,*], Mayte Guerra Saborit[3], Ansel Rodríguez-Gónzalez[4], Julio Madera[3], Ana Orquidea López Correoso[5], Carlos de Castro Lozano[1,2], Jose Miguel Ramírez Uceda[1,2] and José Carlos Arévalo Fernández[2]

[1]*Universidad de Córdoba, Córdoba, Spain*
[2]*Plénitas, C/ Le Corbusier s/n, 14005 Córdoba, Spain*
[3]*Universidad de Camaguey, Circunvalación Norte, Camino Viejo Km 5 y 1/2, Camaguey, Cuba*
[4]*CICESE-UT3, Nayarit, México*
[5]*IPVCE "Máximo Gómez Báez", Circunvalación Norte, Camaguey, Cuba*

Abstract

The UC-UCO-Plenitas team participated in the PRESTA 2025 challenge, focused on question answering over tabular data in Spanish using the DataBenchSPA benchmark. This benchmark is the first of its kind in the Spanish language and includes real-world tables with diverse data types, designed to evaluate system capabilities in answering natural language questions. The team implemented a solution leveraging GPT-4o, a multimodal large language model developed by OpenAI, known for its real-time, multi-input processing capabilities. GPT-4o was used to handle text-based question answering tasks, with the final system developed using under 150 lines of code, integrating the evaluation functions provided by the organizers. Among 23 competing teams, the UC-UCO-Plenitas team secured 7th place, achieving 66.0% accuracy, showcasing the model's potential and competitive performance against other state-of-the-art approaches. While not reaching the top three, the team's results highlight opportunities for further performance optimization through better prompt design and fine-tuning. The paper also provides insights into deep learning architectures, particularly transformers, and emphasizes the role of large language models (LLMs) in advancing natural language understanding over structured datasets.

Keywords

Question Answering, DataBenchSPA, Spanish Language Benchmark, GPT-4o, Large Language Models (LLMs)

1. Introduction

Lexical complexity relates to complexity of words. Its assessment can be beneficial in a number of fields, ranging from education to communication. For instance, lexical complexity studies can assist in providing language learners with learning materials suitable for their proficiency level or aid in text simplification [1]. These studies are also a central part of reading comprehension, as lexical complexity can predict which words might be difficult to understand and could hinder the readability of the text. Lexical complexity studies typically make use of Natural Language Processing and Machine Learning methods [2]. Previous similar studies focus on ComplexWord Identification (CWI), which is a process of identifying complex words in a text [3]. In this case, lexical complexity is assumed to be binary - words are either complex or not. LCP Shared Task 2021 addresses this limitation by introducing a new dataset designed for continuous rather than binary complexity prediction [4].In this paper, we present

IberLEF 2025, September 2025, Zaragoza, Spain

[*]Corresponding author.

yoan.martinez@plenitas.com (Y. Martínez-López); mayte.guerra@reduc.edu.cu (M. G. Saborit); ansel@cicese.edu.mx (A. Rodríguez-Gónzalez); julio.madera@reduc.edu.cu (J. Madera); analopezcorreoso@gmail.com (A. O. L. Correoso); carlosdecastrolozano@gmail.com (C. d. C. Lozano); p52raucj@uco.es (J. M. R. Uceda); josecarlos@plenitas.com (J. C. A. Fernández)

0000-0002-1950-567X (Y. Martínez-López); 0000-0002-9556-5869 (M. G. Saborit); 0000-0001-9971-0237 (A. Rodríguez-Gónzalez); 0000-0001-5551-690X (J. Madera); 0000-0001-6603-843X (C. d. C. Lozano); 0000-0002-5027-7521 (J. M. R. Uceda)

our participation in the PRESTA 2025: Question Answering over Tabular Data in Spanish, describing our methodological approach, model choices, and results across the different subtasks.

2. Methodology

Dataset The IberLEF 2025 shared task is centered on Question Answering over Tabular Data, making use of the newly developed DataBenchSPA benchmark [5, 6]. DataBenchSPA represents the first Spanish-language benchmark that features real-world tabular datasets with a substantial number of rows and columns [4, 3]. This benchmark includes a wide variety of data types, facilitating the evaluation of different question formats that are specific to each type of data. The task encourages participants to develop systems capable of answering questions based on daily-use datasets included in DataBenchSPA. The expected answers may be numerical values, categorical values, boolean outputs, or lists comprising elements of various types. While DataBenchSPA serves as the training and validation dataset, a separate test set is released specifically for the competition phase. Each system receives a set of (dataset, question) pairs and is required to return an answer that is then compared against a predefined gold standard. Participants are allowed to use any method of their choice to compute the answers. To facilitate participation, the organizers provide a Python library that enables a straightforward submission process using fewer than 150 lines of code. This library also includes the official evaluation function used during the competition, allowing teams to evaluate their systems locally on the development set [5].

Deep Learning Deep learning is a type of machine learning that uses artificial neural networks with many layers (hence "deep") to model complex patterns in data. It's especially good at handling unstructured data like images, text, and audio. Deep learning enables computers to learn from data much like the human brain does. Instead of being programmed with specific rules, a deep learning system figures out the rules on its own by training on large datasets[7, 8].

Transformers Transformers are a type of deep learning architecture introduced by [9]. They're built around a self-attention mechanism that lets the model weigh the importance of different tokens (words or subwords) in an input sequence when generating representations or predictions. Unlike recurrent or convolutional networks, Transformers process all tokens in parallel, which makes them highly efficient on modern hardware and capable of modeling long-range dependencies. Computes attention scores between every pair of positions in the input, producing a weighted sum of value vectors that captures contextual relationships. Runs several attention "heads" in parallel, letting the model attend to different types of relationships simultaneously. After attention, each position is passed through a small fully connected network (the same one for every position) to mix features. Since Transformers lack recurrence, they add sine/cosine or learned embeddings to each token to encode its position in the sequence. Each sub-layer (attention or feed-forward) is wrapped with skip connections and normalization for stable training. Transformers power many state-of-the-art models—BERT [10], GPT, T5, etc.—and have become the go-to architecture for NLP, and increasingly for vision and audio tasks, due to their scalability and strong performance.

Large Language Models (LLMs) Large Language Models are deep learning models trained on massive amounts of text data to understand, generate, and manipulate human language. These models use neural networks—typically transformer architectures—to learn patterns in text and perform tasks like Text generation, Translation, Summarization, Sentiment analysis, Question answering and Code generation. A Large Language Model is a type of artificial intelligence (AI) trained to understand, generate, and interact using human language. These models are built using deep learning techniques (usually transformers) and are trained on vast amounts of text data — including books, websites, and documents — to learn grammar, facts, reasoning, and context[11]. Examples of well-known LLMs include GPT (OpenAI) [12, 13], LLaMA (Meta)[14], Mistral (Mistral AI)[15, 16, 17], Claude (Anthropic)[18, 19], Qwen (Alibaba)[20, 21], DeepSeek(Deepseek AI)[22, 23] and Gemma (Google DeepMind)[24]. They are trained on trillions of words (from books, websites, etc.) using self-supervised learning. Most use the Transformer architecture, enabling them to understand context and relationships in language. Text is broken into "tokens" (words or subwords), and the model predicts the next token.

GPT4o GPT is a type of Large Language Model (LLM) developed by OpenAI [12, 13]. It's designed to understand and generate natural language text, making it capable of performing a wide range of language-based tasks such as writing, summarizing, translating, and answering questions. GPT in Simple Terms: 1) Generative: It can create new text based on a prompt. 2) Pre-trained: It's trained on a massive amount of text before being fine-tuned for specific tasks and 3)Transformer: It's built on a powerful deep learning architecture called the transformer, which allows it to understand context in long passages of text. Also, GPT-4o (pronounced "GPT-4 omni") is the latest multimodal model developed by OpenAI, released in May 2024 [25, 13]. It represents a major upgrade to the GPT-4 family. GPT-4o is natively multimodal [26], meaning it can handle: Text, Images, Audio, Video (limited interactions) and Real-time speech input/output. Unlike previous versions that used separate systems (like Whisper for audio and DALL·E for images)[27], GPT-4o processes all these inputs in a unified model. They abilities are:

- Hold natural conversations with tone and emotion.
- See and describe images, including charts and diagrams.
- Listen and respond to live speech.
- Translate, transcribe, summarize, and interpret audio in real-time.
- Solve math problems from handwritten notes or photos.

Setup and functionality The setup and functionality of an automated system designed to answer natural language questions about tabular datasets, in the context of the PRESTA 2025 (IberLEF) competition. The system uses OpenAI's GPT-4o language model to generate Python code that directly answers each question by manipulating the data using the pandas and numpy libraries. Its workflow is structured into four main stages: prompt generation, model query, code execution, and answer evaluation.

To get started, Python 3.8 or higher is required, along with the installation of key dependencies: pandas, numpy, openai, nest_asyncio, and the databench_eval library, the latter provided as part of the official evaluation setup. Once the environment is ready, a valid OpenAI API key must be configured. Although the base code defines it as a constant, best practice is to store the key in an environment variable using export OPEN_AI_KEY="sk-..." and access it within the script using os.getenv("OPEN_AI_KEY"). It is also recommended to download the datasets in Parquet format and store them under the path ˙/databenchSPAdatasetall.parquet, although HuggingFace datasets can also be used via the "hf:// prefix".

The system loads the development question set (qa_dev) using the load_qa function. For each question, it dynamically constructs a prompt that instructs the model to generate a single line of code within a predefined answer(df) function. This line should directly return the answer to the question using only the provided DataFrame. For example, for the question "What is the average number of bedrooms?" on the airbnb dataset, the system generates a prompt like:

"You are a pandas code generator. Your goal is to complete the function provided... def answer(df: pd.DataFrame): return".

The model might respond with a line such as df['bedrooms'].mean(), which the system executes dynamically using exec() to compute the final answer. This technique allows the automatic evaluation of the model's reasoning capabilities on tabular data. The generated results are saved in a file named predictions.txt, which can then be submitted as an official run. Additionally, performance is measured by comparing the generated responses to ground truth labels using the Evaluator module. This baseline system achieves approximately 49% accuracy on the development set.

An optional utility function, column_generator(), is also included. It generates prompts to filter and select only the most relevant columns needed to answer each question. This is useful for reducing input size when dealing with models that have context limitations. Despite its effectiveness, this system carries potential security risks due to the use of exec() for code execution, and should only be used in controlled environments.

Evaluation An automated evaluation function is currently provided to handle most of the assessment process. When a participant uploads a submission, the default evaluation function from the databench_eval package is executed, comparing the submission against the ground truth set. This function has been modified to be less strict than the one used in the initial experiment. The adjustment accommodates slight variations in formatting, allowing smaller models to avoid penalties for minor errors. Given the heuristic nature of the evaluation, the characteristics of the models being used, and the open-ended nature of the task, the organizers will manually review the top-scoring submissions before selecting a winner.

Types of Answers Expected

According to the expected answer types:

- Boolean: Valid answers include True/False, Y/N, Yes/No (all case insensitive).
- Category: A value from a cell (or a substring of a cell) in the dataset.
- Number: A numerical value from a cell in the dataset, which may represent a computed statistic (e.g., average, maximum, minimum).
- List[category]: A list containing a fixed number of categories. The expected format is: "['cat', 'dog']". Pay attention to the wording of the question to determine if uniqueness is required or if repeated values are allowed.
- List[number]: Similar to List[category], but with numbers as its elements.

3. Results

Competition phase In the competition, our team employed GPT-4o as the primary method for solving the tasks, achieving the following results: In the competition phase, out of 23 total participants, the UC-UCO-Plenitas team secured 7th place with an accuracy of 66.0% . This result reflects a solid performance considering the number of competitors, although there is clear room for improvement to reach the top positions. The top three teams — itunlp , sonrobok4 , and hcerezo — achieved significantly higher accuracies (85–87%), indicating a high level of performance in the task. Teams ranked from 4th to 6th (e.g., LyS Group , quang3010 , and ScottyPoseidon) also outperformed UC-UCO-Plenitas by a noticeable margin. Despite not reaching the podium, UC-UCO-Plenitas demonstrated competitive capability, especially when compared to other mid-to-lower-ranked teams. With further refinement, particularly in precision and fine-tuning of classification strategies, the model could potentially move up in future rankings. See table 1.

Table 1
Top 7 Participants in PRESTA 2025 Ranked by Accuracy

Rank	Model	Participant	Accuracy (%)
1		itunlp	87.0
2		sonrobok4	87.0
3		hcerezo	85.0
4		LyS Group	78.0
5		quang3010	75.0
6		ScottyPoseidon	73.0
7	**GPT-4o**	**UC-UCO-Plenitas**	**66.0**

4. Conclusions

The participation of the UC-UCO-Plenitas team in the PRESTA 2025 competition demonstrated the growing applicability of Large Language Models, particularly GPT-4o, in the domain of question answering over tabular data. Achieving a 66.0% accuracy rate, the team's performance places them in the top third of participating systems and highlights the feasibility of using GPT-4o in data-intensive Spanish-language NLP tasks. The results validate the use of minimal coding approaches with powerful pre-trained models and point toward the potential of improved performance through enhanced prompt engineering and fine-tuning strategies. The competition served as a valuable benchmark for evaluating the capabilities of modern LLMs in multilingual and structured data contexts. Future efforts will focus on improving system generalization and interpretability to reach the performance levels of the top-ranking teams.

Declaration on Generative AI

During the preparation of this work, the author(s) used GPT-4 and Grammarly in order to: Grammar and spelling check. After using these tool(s)/service(s), the author(s) reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] A. Siddharthan, A survey of research on text simplification. itl-international journal of applied linguistics. special issue on readability and text simplification, 2014.

[2] G. Paetzold, L. Specia, Inferring psycholinguistic properties of words, in: Proceedings of the 2016 conference of the north american chapter of the association for computational linguistics: Human language technologies, 2016, pp. 435–440.

[3] M. Shardlow, A comparison of techniques to automatically identify complex words., in: 51st annual meeting of the association for computational linguistics proceedings of the student research workshop, 2013, pp. 103–109.

[4] M. Shardlow, R. Evans, G. H. Paetzold, M. Zampieri, Semeval-2021 task 1: Lexical complexity prediction, arXiv preprint arXiv:2106.00473 (2021).

[5] J. Á. González-Barba, L. Chiruzzo, S. M. Jiménez-Zafra, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[6] J. Osés-Grijalba, L. A. Ureña-López, E. M. Cámara, J. Camacho-Collados, Overview of PRESTA at IberLEF 2025: Question Answering Over Tabular Data In Spanish, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, nature 521 (2015) 436–444.

[8] N. Rusk, Deep learning, Nature Methods 13 (2016) 35–35.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).

[10] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers), 2019, pp. 4171–4186.

[11] M. Johnsen, Large language models (LLMs), Maria Johnsen, 2024.

[12] M. Zhang, J. Li, A commentary of gpt-3 in mit technology review 2021, Fundamental Research 1 (2021) 831–833.

[13] K. I. Roumeliotis, N. D. Tselikas, Chatgpt and open-ai models: A preliminary review, Future Internet 15 (2023) 192.

[14] O. Analytica, Meta llama leak raises risk of ai-linked harms, Emerald Expert Briefings (2023).

[15] O. Aydin, E. Karaarslan, F. S. Erenay, N. Bacanin, Generative ai in academic writing: A comparison of deepseek, qwen, chatgpt, gemini, llama, mistral, and gemma, arXiv preprint arXiv:2503.04765 (2025).

[16] F. Hamzah, N. Sulaiman, Multimodal integration in large language models: A case study with mistral llm (2024).

[17] A. Mistral, Mixtral of experts, Fecha de Publicación 11 (2023).

[18] A. Priyanshu, Y. Maurya, Z. Hong, Ai governance and accountability: An analysis of anthropic's claude, arXiv preprint arXiv:2407.01557 (2024).

[19] A. J. Adetayo, M. O. Aborisade, B. A. Sanni, Microsoft copilot and anthropic claude ai in education and library service, Library Hi Tech News (2024).

[20] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, et al., Qwen technical report, arXiv preprint arXiv:2309.16609 (2023).

[21] A. Cloud, Qwen 2.5 (2024).

[22] A. Liu, B. Feng, B. Xue, B. Wang, B. Wu, C. Lu, C. Zhao, C. Deng, C. Zhang, C. Ruan, et al., Deepseek-v3 technical report, arXiv preprint arXiv:2412.19437 (2024).

[23] L. Xiong, H. Wang, X. Chen, L. Sheng, Y. Xiong, J. Liu, Y. Xiao, H. Chen, Q.-L. Han, Y. Tang, Deepseek: Paradigm shifts and technical evolution in large ai models, IEEE/CAA Journal of Automatica Sinica 12 (2025) 841–858.

[24] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al., Gemma: Open models based on gemini research and technology, arXiv preprint arXiv:2403.08295 (2024).

[25] K. I. Roumeliotis, N. D. Tselikas, D. K. Nasiopoulos, Leveraging large language models in tourism: A comparative study of the latest gpt omni models and bert nlp for customer review classification and sentiment analysis, Information 15 (2024) 792.

[26] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, J. Zou, Mixture-of-agents enhances large language model capabilities, arXiv preprint arXiv:2406.04692 (2024).

[27] F. B. Kern, C.-T. Wu, Z. C. Chao, Assessing novelty, feasibility and value of creative ideas with an unsupervised approach using gpt-4, British Journal of Psychology (2024).

A. Online Resources

The results are available via

- PRESTA Codabench,
- PRESTA Dataset.