Numerical information extraction in legal texts using open and closed Large Language Models

Bruno V. Bitelli[1,2,3], **Marcelo Finger**[1]

[1]Instituto de Matemática e Estatística – Universidade de São Paulo (IME-USP)

[2]Insper

[3]Artemis Technologies

bvbitelli@gmail.com, mfinger@ime.usp.br

Abstract. *This paper investigates numerical named-entity recognition in Portuguese legal texts using both closed and open-source decoder-only Large Language Models (LLMs). We conduct a quantitative and qualitative evaluation of two paradigms: (1) fine-tuning a modified version of LLaMA 2 via LoRA on over 600 new, manually annotated, judicial rulings, and (2) prompt engineering with closed models (OpenAI's GPT and Google's Gemini). We compare the performance of instruction tuning and prompt construction with closed models, with a parameter-efficient fine-tuning approach that bridges decoder-only LLMs with traditional encoder-only architectures. The results reveal that the modified, LoRA-tuned, LLaMA 2 achieves competitive entity-recognition performance while offering greater transparency and parameter-efficiency, whereas prompt-engineered closed models simplify deployment but incur limitations in consistency and fine-grained control.*

1. Introduction

The legal domain is a widely studied field within NLP and remains at the forefront of innovation and research. A key area within this field is jurisprudence, which can be defined as the body of documents and the cumulative impact of a court's repeated rulings on similar cases. In this context, NLP tasks like text classification assign documents to predefined categories based on extracted features, while Named Entity Recognition (NER) isolates and labels textual mentions of specific entities. Also, despite NER's prominence in NLP, it still struggles to accurately recognize and classify entities that consist of numerical expressions.

In recent years, new studies have emerged demonstrating the ability of pre-trained large language models (LLMs) to perform text classification and information extraction tasks with great success using little or no additional training or parameter tuning [Agrawal et al. 2022, Wei et al. 2023].

The goal of this paper is to investigate the recognition of number entities through the use of LLMs. To this end, we will quantitatively assess the performance of closed-source models, such as OpenAI's GPT and Google's Gemini, in zero-shot and few-shot settings, against Meta's open-source model, LLaMA 2, by applying a new method of fine-tuning using Low-Rank Adaptation (LoRA) for classification tasks. For the analysis, a new dataset made of a sample of approximately 607 civil court decisions was selected

from the São Paulo Court of Justice (TJSP) website. All documents in this sample were manually annotated for a NER task. We aim to evaluate how both paradigms of contemporary LLM usage, prompt engineering with selected examples and fine-tuning, perform when tackling a real-world problem in the legal domain in Brazil.

2. Related Works

2.1. Legal domain and Jurisprudence

Jurisprudence encompasses a wide range of legal documents. For this paper, we will focus on appellate decisions (or *acórdãos* in Portuguese). An *acórdão* follows the same structure prescribed for judgments in art. 489 of the Brazilian New Code of Civil Procedure (NCPC), comprising: *relatório*, which narrates the facts and principal occurrences of the case; *fundamentos*, in which the judge analyzes the issues set forth in the report and explains the reasons for the decision; and *dispositivo*, where the collegiate body's final decision on the case is presented. For this paper, we will utilize only the *dispositivo* section of the document.

Jurisprudence documents have already been employed for studies on various NLP tasks, ranging from clustering and classification [Furquim and de Lima 2012], multi-label classification [Serras and Finger 2021] to information extraction [Cabral et al. 2022]. To our knowledge, although there are other NER studies in this domain for the Portuguese language that explore the capabilities of encoder-only LLMs [Nunes et al. 2024], this paper would be the first to investigate the new approaches of prompt engineering and fine-tuning for the latest generations of decoder-only LLMs applied to this task.

2.2. Named and Number Entity Recognition

The field of NER involves the identification and classification of entities into specific categories. An entity is a text segment that represents a real-world object or concept that can be extracted and classified into a predefined category, such as *Person* (PER), *Organization* (ORG), *Location* (LOC), among others [Jurafsky and Martin 2024]. In other words, one way to model the recognition problem is as a multiclass classification task, with class definitions varying by application domain. Among the approaches to NER solutions, one of the most common is to use large labeled datasets in the format commonly referred to as BIO. In this format, B indicates the beginning of an entity, I indicates the inside of a multi-token entity, and O indicates a non-entity—that is, a token marked as not part of any entity. Even though numeric symbols account for approximately 5% of the symbols in a sentence [Naik et al. 2019], *number entities*, as [Sundararaman et al. 2022] writes, has not been a focus of NER studies. These entities may range from dates, ages, telephone numbers, to registration numbers and financial values. As the authors observe, numerical named entity recognition, *Number Entity Recognition* (NuER), as in the original paper, is still an emerging area of research.

2.3. Large Language Models

The Transformer model, originally proposed in [Vaswani et al. 2017], is a neural-network architecture used to build sequence transduction algorithms, in which predictions for specific instances are made by leveraging other examples from the same domain. In general, transduction models employ an *encoder–decoder* structure, where the *encoder* maps an

input sequence to another sequence of continuous representations and the *decoder* produces an output sequence autoregressively, taking into consideration both the previously generated tokens and the encoder's outputs.

Encoder-only and *decoder-only* models both leverage Transformer architectures but serve complementary roles: encoder-only models (e.g. BERT [Devlin et al. 2018]) use bidirectional attention, making them ideal for labeling tasks like text classification or named-entity recognition, whereas decoder-only models rely on unidirectional, autoregressive generation, predicting each next token solely from preceding context, so they excel at tasks that require text production.

Closed LLMs, such as OpenAI's GPT series [Ouyang et al. 2022] and Google's Gemini [Anil et al. 2024], are commonly decoder-only Transformer models trained on privately curated data and accessible only via proprietary APIs, offering top-tier generative performance at the cost of transparency and customization. In contrast, open LLMs, like Meta's LLaMA 2 [Touvron et al. 2023], publish their full architectures and pretrained weights under permissive licenses, enabling researchers to inspect, fine-tune (e.g. via LoRA [Hu et al. 2021]), compress, and host these models, trading some of the immediate convenience of closed APIs for scientific openness.

2.4. Prompt-Based Learning

Prompt-Based Learning (PBL) uses natural-language prompts, combining a task description with input text, to enable LLMs to excel in few or zero-shot tasks, considering that their next-word prediction training across diverse contexts is implicitly teaching the model secondary abilities that support their primary objectives [Agrawal et al. 2022, Sanh et al. 2021]. For example, in entity recognition tasks, decoder-only models such as ChatGPT achieved notable F1 scores of approximately 85% and 67% on the BBN and CoNLL-2003 datasets respectively, especially when compared to BERT, one of the most widely used encoder-only models for these investigations, which obtained F1 scores of 80% and 91%, respectively [Li et al. 2023a].

PBL via Question Answering. As a very recent research area, different approaches have been proposed for prompt construction [Cui et al. 2021, Chen et al. 2021]. For this paper, prompt construction will follow the format originally proposed by [Wei et al. 2023], using a *multi-stage question-answering* (MQA) prompt structure. The MQA paradigm decomposes NER into a first stage that queries which predefined types appear in the input and a second stage that enumerates their mentions, which naturally extends to fine-grained NER (FG-NER) by chaining additional question stages to capture subtype distinctions.

2.5. Label Supervised LLaMA

Despite multiple studies reporting performance gains in recent years through improvements in prompt-construction steps [Wei et al. 2023, Li et al. 2023a] and via *instruction tuning* [Wang et al. 2023], most of the results obtained for NER and text-classification tasks fail to surpass those achieved by established benchmarks such as BERT and RoBERTa. Motivated by this limitation of decoder-only models and by the efficiency of architectures such as BERT for classification tasks, [Li et al. 2023b] propose a new

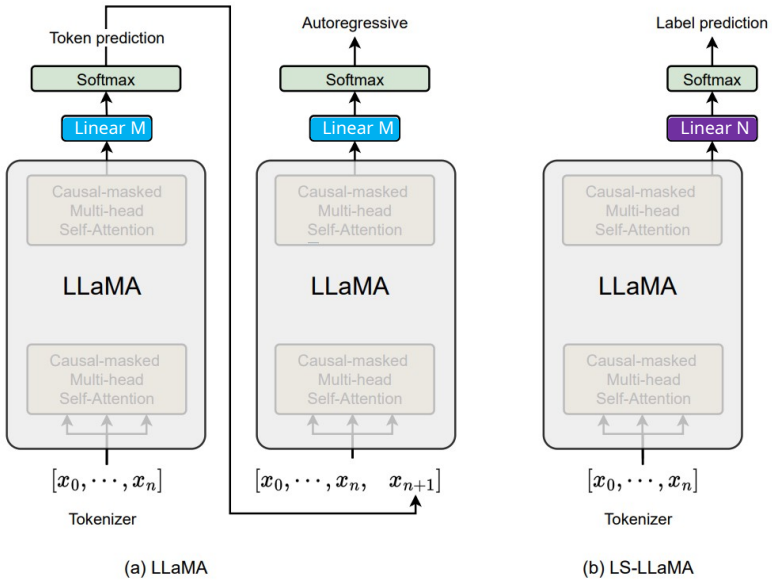


Figure 1. Comparison between the autoregressive LLaMA model and the LS-LLaMA model for classification tasks. Adapted from [Li et al. 2023b].

method in which LLMs undergo a supervised fine-tuning process with labeled data, named Label-Supervised LLaMA (LS-LLaMA).

Since LLaMA is inherently designed for autoregressive text-generation, its latent vectors cannot be used directly for classification tasks. They must first be mapped into a label-compatible format, like the representations produced by encoder-only models. In the LS-LLaMA framework, only the latent vector corresponding to the last token of the input sequence are extracted. Because the model is unidirectional, that final token is the only one whose representation has "seen" the entire context. If the vector for any earlier token were to be used, all subsequent context would be ignored. The vectors are then passed through simple neural network layers whose output dimension matches the number of target classes N. We then apply a softmax function to the resulting logits to obtain a normalized probability distribution over the classes. Finally, the class with the highest probability is selected as the predicted label, as shown in Fig. 1. After this step, the cross-entropy loss is computed and the model is fine-tuned using LoRA [Hu et al. 2021].

Since the objective of this paper is to compare the performance of closed LLMs in a zero-shot or few-shot setting against an open LLM architecture on an NER task, the proposed LS-LLaMA model emerges as the most compelling choice for this analysis.

3. Methodology

The data used in this project were extracted from the e-SAJ of the TJSP, which is a solution designed to facilitate access to information and streamline procedural operations via web services. Currently, there are multiple electronic case-management systems in Brazil, with e-SAJ being used in at least nine states, including São Paulo. In order to more precisely define the scope of this research project, only cases available on the São Paulo e-SAJ portal pertaining to the subject of Air Transportation were selected for data extraction, resulting in 600 second instance decisions.

3.1. NER Task

We defined five primary entity types, *Date* (D), *Percentage* (P), *Other Numbers* (ON), adapted from [Sundararaman et al. 2022], plus two conceptualized specifically for this study: *Financial Value* (VF), expressed in Brazilian reais, and *Indemnity Count* (CI), expressed in any other counting unit that the compensation is declared. VF and CI are then further subdivided in a second FG-NER stage into past (VFP/CIP), refering to indemnities assessed before the decision, versus final (VFF/CIF), refering to the final value decided in the document. Financial values that do not correspond to compensation are labeled as *Other Financial Value* (OVF). Table 1 provides a summary and examples of these subtypes as used in the recognition task of this study.

Table 1. Table of the classes defined for the FG-NuER task. Entities to be recognized are highlighted in bold in the example texts.

Class	Name	Example
VFF	Final Financial Value	"[. . .] majoro o valor da sentença de r$ 1000,00 para **dois mil reais**."
VFP	Past Financial Value	"[. . .] majoro o valor da sentença de **r$ 1000,00** para dois mil reais."
OVF	Other Financial Value	"[. . .] dólar se encontra no valor de r$ **3,50** [. . .]"
CIF	Final Indemnity Count	"[. . .] cabe a redução da indenização de 10 direitos especiais de saque para **5 des** [. . .]"
CIP	Past Indemnity Count	"[. . .] cabe a redução da indenização de **10 direitos especiais de saque** para 5 des [. . .]"
P	Percentage	"[. . .] mantenho os honorários em **15% do valor da causa** [. . .]"
D	Date	"[. . .] levando em consideração o ocorrido em **04 de junho de 2015** [. . .]"
ON	Other Numbers	"[. . .] considerando o **art. 14 do parágrafo 4 do N.C.P.C.** é justificável [. . .]"

3.2. Labeling

For the numerical entity recognition task, because final outcome values are not guaranteed to appear explicitly in the text, we extracted entities directly from the *dispositivo* section. As there is no explicit marker delimiting the end of the previous section and the start of the *dispositivo* section, we adopted an arbitrary rule of selecting the last four paragraphs of each document after preprocessing them for possible mistakes in the PDF file. This heuristic was determined by examining hundreds of decisions, in all of which the outcome description fell within this interval. All annotations were encoded using the BIO format.

The entire labeling task was carried out by a single individual, the first author of this work, over a four month period. A detailed discussion of this tool's development and the specific labeling guidelines can be found in the first Appendix of the authors' dissertation [Bitelli and Finger 2024]. At the end of the labeling phase, 607 documents of the appellate decisions had been annotated, each containing at least one marked entity.

3.3. Implementation

The implementation for the tasks are described in detail in [Bitelli and Finger 2024]. We can divide the most relevant characteristics of implementation into two stages. First, a version of the LS-LLaMA model will be implemented, then the same tasks will be performed via the PBL format with multi-stage question answering (MQA), using the GPT-3.5 Turbo and Gemini 1.0 Pro models. Since the objective of this paper is to compare two different LLM architectures for the same task, no implementation using more common methodologies (e.g., BERT-based approaches) was included. This additional evaluation is proposed in the Future Work section and discussed in greater detail in the author's dissertation [Bitelli and Finger 2024]. All research steps were conducted in Python.

LS-LLaMA Implementation. The authors of the LS-LLaMA have made their code available in an open repository under the MIT license[1]. This allows the focus of the implementation to be limited to data preprocessing steps and adaptations for this specific use case. The default class from the Transformers library, *AutoTokenizer* , was selected as the tokenizer, since it already provides an interface for LLaMA-family models. For LoRA's implementation, we used the Parameter-Efficient Fine-Tuning (PEFT) library.

The model selected for the experiment was LLaMA-2-7B. Although the original experiment conducted by [Li et al. 2023b] used an NVIDIA GeForce RTX 4090, the input sequences in our experiment are substantially longer, with a maximum token length of 1024 versus 64 in the original. This increase is due to the large size of the documents, which renders the VRAM on the RTX 4090 insufficient. Accordingly, the NVIDIA A40 GPU, with 48 GB of VRAM, was chosen for this experiment. All the parameters were maintained from the original paper, with the exception of the learning rate, which was reduced to $4e$-5 to prevent out-of-memory (OOM) errors, even when using an A40 GPU.

To ensure an approximately consistent distribution of entity types across the different datasets, we adapted a version of the *stratify_train_test_split_multi_label* function from the Deep Utils library . The training set was allocated 70% of the data, the validation set 15%, and the test set 15%.

Multi-stage Question Answering Implementation. All calls to the closed-source models from OpenAI and Google were made through their respective APIs. The GPT-3.5 Turbo and Gemini 1.0 Pro models were selected for this paper, both configured with a temperature of 0 to ensure maximally deterministic outputs. During this same preliminary testing phase, we observed that the model tended to return entities with missing interior content, even though it correctly identified the entity's start and end. To reduce this type of error, instead of passing the sentences directly, we reconstructed them using the same tokenizer employed in the labeling process, concatenating tokens with the "—" separator.

The proposed MQA approach employs a two-stage flow, first detecting entities of types D, P, ON, VF, and CI, then, if VF or CI entities are present, triggering specific prompts to classify them as VFP, VFF, OVF or CIP, CIF. Because the expected output in the token-level labeling BIO scheme produced non-deterministic hallucinations, such as incomplete annotations and spurious tokens, even after setting the temperature to 0, we adopted the concise, dictionary-based output format shown in Fig. 2, which restricts responses to actual input tokens and dramatically reduces hallucinations.

We curated five out-of-dataset example sentences per prompt type to illustrate entity variations and conducted six full MQA test iterations, one zero-shot, and five few-shot with progressively added examples (further details into the prompt-construction process can be found in the second Appendix of [Bitelli and Finger 2024]).

3.4. Performance Metrics

Because the predominance of "O" labels in BIO-formatted tasks leads to excessive true negatives, inflating the accuracy, we follow established LLM comparison studies by evaluating zero- and few-shot performance using precision, recall, and the micro F1-Score.

[1] `https://github.com/4AI/LS-LLaMA/tree/main`. Accessed April 19, 2025.

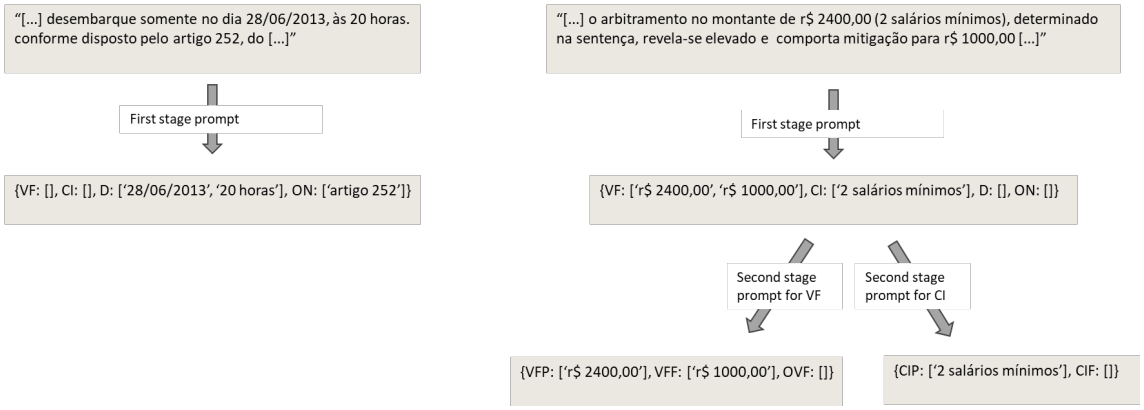
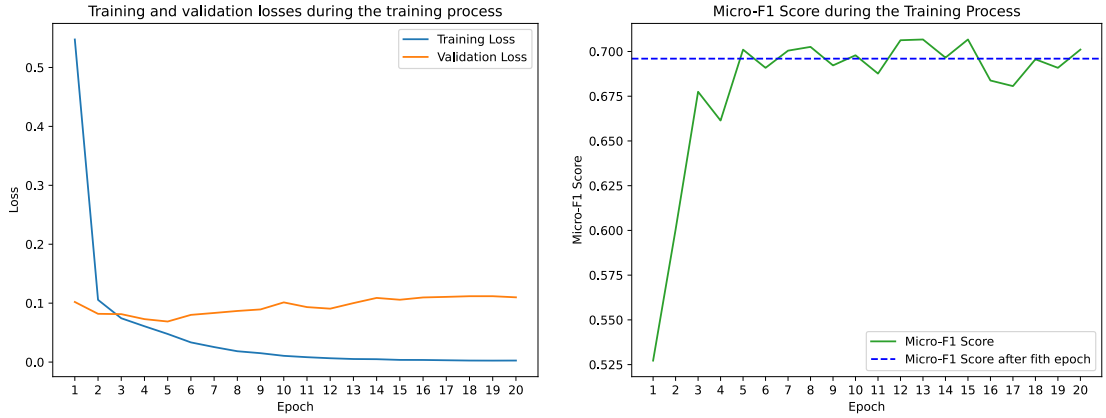


Figure 2. (Left) MQA system's input-output example where no VF or CI entities are recognized, stopping at the first QA stage. (Right) Example where VF and CI entities are recognized, triggering a second QA stage.



(a) Training and validation loss **(b) Micro F1-Score**

Figure 3. Loss and micro F1-Score results for over 20 epochs of training.

To count as a correct classification, there must be an **exact match** of the entire sequence of tokens that comprise the entity. For the implementation of the performance metric in the LS-LLaMA experiments, we employed the *strict* mode of the *seqeval* package from the Evaluate library, which considers only BIO-formatted entities with full-span matching. For the MQA experiments, however, we had to develop a custom solution, since the output formatting is nonstandard and was created specifically for this project.

4. Results

For the LS-LLaMA experiment, 20 epochs were run, ten more than in the experiments conducted by [Li et al. 2023b]. Training loss, evaluation loss, and the validation micro F1-Score were recorded at each epoch, as seen in Fig. 3. The model with the lowest loss observed during training was chosen.

4.1. Zero-Shot and Few-Shot

Fig. 4 presents each test's model performance, measured by the micro F1-Score, where the examples used were randomly selected. As expected, adding examples to the instruction yields a substantial performance increase for both models, with micro F1-Scores rising

from approximately 0.07 to 0.55. Another notable observation is the similarity of results between GPT and Gemini, particularly in the zero and one-example settings. A possible interpretation is that, with only a few examples, the models correctly identify only the clearest or easiest entities, lacking reference cases for the less obvious ones.

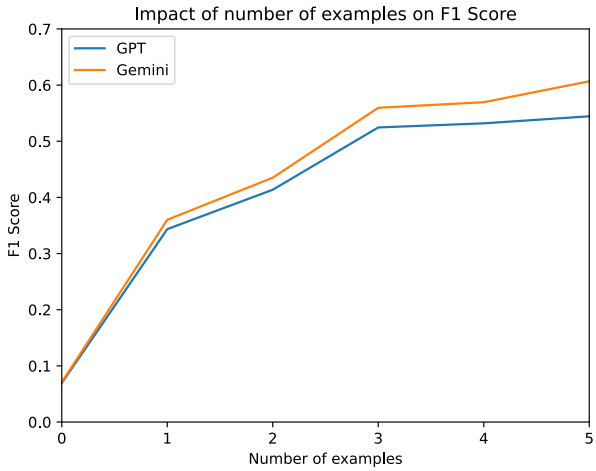
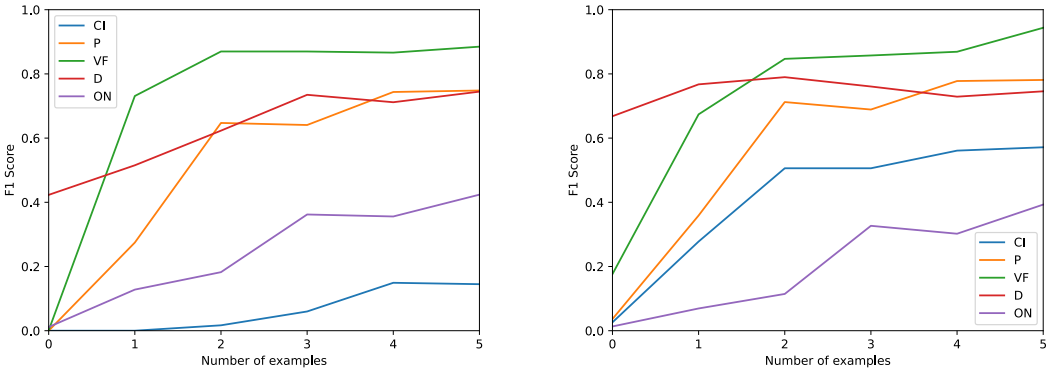


Figure 4. Comparison of the GPT-3.5 Turbo and Gemini 1.0 Pro models using the micro F1-Score for the NER task across different numbers of examples.

In Fig. 5, the performance gains for the first stage entities in the MQA tests can be observed. It is clear that some entity types benefit from the inclusion of examples more than others. For instance, Date (D) already achieve relatively high performance, even with no or few examples. These high initial scores can be partly explained by the more straightforward nature of their usage, which demands less contextual interpretation than that required for other entities, such as Financial Values (VF), which exhibit the largest performance improvements.

For some types, such as Indemnity Count (CI), the presence of examples can lead to significant performance improvements, but their results still lag behind those of other entity types, even with five examples. One hypothesis is that in the texts these entities often resemble Other Number (ON) entities. For example, in the sentence "[...] o valor de 2 salários mínimos está abaixo do teto previsto na lei de 10 salários [...]" correct extraction requires recognizing "2 salários mínimos" as CI and "10 salários" as ON, since the latter is merely a citation of the legal limit, not an compensatory count for the case.

A curious effect observed for the D and ON entities is the sporadic decline in performance as more examples are added. One possible explanation is that the crafted examples incorporate different formats and contexts for these entities, which can lead to confusion, since there are many instances in which D and ON appear in similar forms. For example, in the input sentence: "[...] decisão presente no diário de justiça 04.12.2003, página 298 [...]", the classification of the extracted entity "diário de justiça 04.12.2003, página 298" should be ON, but the registration number of the gazette, "04.12.2003", also serves as the gazette's date, and thus can be easily misclassified as a D entity.



(a) GPT. (b) Gemini.

Figure 5. Comparison of the F1 metric across varying numbers of examples for each entity type in the first stage of the MQA task.

4.2. Comparing Few-Shot MQA and LS-LLaMA

The micro F1-Score results for the MQA zero-shot method, the few-shot method with five examples (5-shot), and the LS-LLaMA method are presented in Table 2. Although both MQA approaches exhibit similar performance, the LS-LLaMA method achieves a markedly superior result, with a micro F1-Score more than twice as high. The only class that showed a lesser performance was the CIP, as this subtype is hampered by its low frequency in the test set, with a singular occurrence.

In the MQA experiments, adding five examples to the prompt markedly increases precision, as seen in Table 3, by reducing false positives with the model better identifying what qualifies as a positive instance for that entity type. However, it only modestly improves recall, as seen in Table 4, reflecting the model's ongoing difficulty in generalizing to novel entity instances not presented in the examples.

Table 2. F1-Scores for the different entity types in the experiment dataset. Columns labeled "5-Shot" indicate the use of five examples in the prompt construction process.

Classes	GPT Zero-Shot	Gemini Zero-Shot	GPT 5-Shot	Gemini 5-Shot	LS-LLAMA
ON	0.01	0.01	0.42	0.39	**0.72**
P	0.00	0.04	0.75	0.78	**0.90**
VFP	0.00	0.00	0.09	0.34	**0.80**
CIF	0.00	0.00	0.12	0.48	**0.67**
D	0.42	0.67	0.74	0.75	**0.80**
VFF	0.00	0.00	0.70	0.79	**0.86**
CIP	0.00	0.00	0.07	**0.17**	0.00
OVF	0.00	0.00	0.12	0.39	**0.40**
MICRO	0.07	0.07	0.54	0.61	**0.79**

It is interesting to note the low performance of the zero-shot tests for almost all classes. The clearest explanation for this difference lies in the fact that they are entities with novel definitions. Consequently, in the zero-shot setting, the model must infer the entities' definitions and their correct application entirely from the instruction, which can be challenging and result in a lower performance. The only class that demonstrated significant performance in the zero-shot tests was the Date entity, which is the only type

Table 3. Precision values for the different entity types in the experiment dataset.

Classes	GPT Zero-Shot	Gemini Zero-Shot	GPT 5-Shot	Gemini 5-Shot	LS-LLAMA
ON	0.02	0.11	0.59	**0.70**	0.68
P	0.00	0.04	0.78	0.80	**0.91**
VFP	0.00	0.00	0.30	**0.78**	0.76
CIF	0.00	0.00	0.07	0.37	**0.67**
D	0.29	0.69	0.71	0.84	**0.89**
VFF	0.00	0.00	0.62	0.70	**0.84**
CIP	0.00	0.00	0.04	**0.12**	0.00
OVF	0.00	0.00	0.12	0.33	**0.75**
MICRO	0.08	0.09	0.59	0.70	**0.77**

Table 4. Recall values for the different entity types in the experiment dataset.

Classes	GPT Zero-Shot	Gemini Zero-Shot	GPT 5-Shot	Gemini 5-Shot	LS-LLAMA
ON	0.01	0.01	0.33	0.27	**0.76**
P	0.00	0.03	0.72	0.76	**0.89**
VFP	0.00	0.00	0.05	0.21	**0.84**
CIF	0.00	0.00	0.45	**0.70**	0.67
D	0.75	0.65	**0.78**	0.67	0.73
VFF	0.00	0.00	0.80	**0.90**	0.89
CIP	0.00	0.00	0.14	**0.29**	0.00
OVF	0.00	0.01	0.12	**0.47**	0.27
MICRO	0.06	0.06	0.51	0.53	**0.81**

common to other datasets and NER tasks, making it likely that the model was already exposed to its definitions and examples within its extensive training data.

5. Conclusion and Future Works

The LS-LLaMA fine-tuning method demonstrated superior performance for numerical entity recognition in the Portuguese legal domain over prompt-engineering techniques, by employing a methodology that diverges from the current literature's standard for LLM parameter tuning, restructuring the algorithm to more closely resemble encoder-only architectures. Notably, the positive outcomes achieved doesn't depend on prompt design, as is typical for prompt-engineering, or also, in instruction tuning.

Taking this into account, the results of this project are in line with the findings by [Li et al. 2023b], which reveal the potential, although seemingly paradoxical at first, of current decoder-only LLMs to function as highly capable text encoders. Their latent vector representations can be applied to a range of tasks beyond autoregressive text prediction when properly trained, thereby obviating the need for elaborate prompt-engineering systems to perform tasks traditionally handled by encoder-only models, such as text classification and NER.

The MQA results demonstrate that downstream tasks on novel datasets can be performed with only a few examples. Applying this same MQA approach to newer closed models, such as GPT-4 and Gemini 2.0, could highlight the evolution of their capabilities, especially compared to open models that still require dedicated training datasets and programming expertise to leverage the libraries used in this study. In addition to evaluating these new closed models, it would also be valuable to assess other open models beyond the LLaMA family and compare any performance gains relative to their model sizes.

Future work should explore prompt engineering that allows partial and inexact matches, considering that one possible reason zero-shot tests perform so poorly is the difficulty of exact output formatting and boundary delimitation. Another suggestion for

future work is to benchmark LS-LLaMA against encoder-only models such as BERT and RoBERTa and to evaluate the LS-unLLaMA variant, where causal self-attention masks are removed, to assess its performance on the same dataset.

References

Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., and Sontag, D. (2022). Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022.

Anil, R., Borgeaud, S., Alayrac, J.-B., and et al., J. Y. (2024). Gemini: A family of highly capable multimodal models.

Bitelli, B. and Finger, M. (2024). Numerical information extraction in legal texts using large language models. Master's thesis, Universidade de São Paulo.

Cabral, B., Souza, M., and Claro, D. B. (2022). Portnoie: A neural framework for open information extraction for the portuguese language. In *Computational Processing of the Portuguese Language*, pages 243–255, Cham. Springer International Publishing.

Chen, X., Li, L., Deng, S., Tan, C., Xu, C., Huang, F., Si, L., Chen, H., and Zhang, N. (2021). Lightner: a lightweight tuning paradigm for low-resource ner via pluggable prompting. *arXiv preprint arXiv:2109.00720*.

Cui, L., Wu, Y., Liu, J., Yang, S., and Zhang, Y. (2021). Template-based named entity recognition using bart. *arXiv preprint arXiv:2106.01760*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Furquim, L. O. d. C. and de Lima, V. L. S. (2012). Clustering and categorization of brazilian portuguese legal documents. In *Computational Processing of the Portuguese Language*, pages 272–283, Berlin, Heidelberg. Springer Berlin Heidelberg.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). Lora: Low-rank adaptation of large language models.

Jurafsky, D. and Martin, J. H. (2024). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. 3rd edition. Online manuscript released August 20, 2024.

Li, B., Fang, G., Yang, Y., Wang, Q., Ye, W., Zhao, W., and Zhang, S. (2023a). Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness.

Li, Z., Li, X., Liu, Y., Xie, H., Li, J., lee Wang, F., Li, Q., and Zhong, X. (2023b). Label supervised llama finetuning.

Naik, A., Ravichander, A., Rose, C., and Hovy, E. (2019). Exploring numeracy in word embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3374–3380.

Nunes, R. O., Spritzer, A. S., Freitas, C. M. D. S., and Balreira, D. G. (2024). Reconhecimento de entidades nomeadas e vazamento de dados em textos legislativos. *Linguamática*, 16(2):preprint–preprint.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Sanh, V., Webson, A., Raffel, C., Bach, S. H., Sutawika, L., Alyafeai, Z., Chaffin, A., Stiegler, A., Scao, T. L., Raja, A., et al. (2021). Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

Serras, F. R. and Finger, M. (2021). verbert: Automating brazilian case law document multi-label categorization using bert. In *Anais do XIII Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana*, pages 237–246. SBC.

Sundararaman, D., Subramanian, V., Wang, G., Xu, L., and Carin, L. (2022). Number entity recognition. *arXiv preprint arXiv:2205.03559*.

Touvron, H., Martin, L., Stone, K., and et al., P. A. (2023). Llama 2: Open foundation and fine-tuned chat models.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Wang, X., Zhou, W., Zu, C., Xia, H., Chen, T., Zhang, Y., Zheng, R., Ye, J., Zhang, Q., Gui, T., et al. (2023). Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv e-prints*, pages arXiv–2304.

Wei, X., Cui, X., Cheng, N., Wang, X., Zhang, X., Huang, S., Xie, P., Xu, J., Chen, Y., Zhang, M., et al. (2023). Zero-shot information extraction via chatting with chatgpt. *arXiv e-prints*, pages arXiv–2302.