Pontifícia Universidade Católica
do Rio de Janeiro

Venícius Garcia Rego

Evaluating LLM In-Context Few-Shot Learning on Legal Entity Annotation Task

Dissertação de Mestrado

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática.

Advisor : Prof. Hélio Côrtes Vieira Lopes
Co-advisor: Prof. Fernando Alberto Correia dos Santos Junior

Rio de Janeiro
September 2024

Pontifícia Universidade Católica
do Rio de Janeiro

Venícius Garcia Rego

Evaluating LLM In-Context Few-Shot Learning on Legal Entity Annotation Task

Dissertation presented to the Programa de Pós–graduação em Informática of PUC-Rio in partial fulfillment of the requirements for the degree of Mestre em Informática. Approved by the Examination Committee:

Prof. Hélio Côrtes Vieira Lopes
Advisor
Departamento de Informática – PUC-Rio

Prof. Fernando Alberto Correia dos Santos Junior
Co-advisor
Departamento de Informática – PUC-Rio

Prof. Marcos Kalinowski
Departamento de Informática - PUC-Rio

Prof. Guilherme da Franca Couto Fernandes de Almeida
Insper

Prof. Jonatas dos Santos Grosman
Departamento de Informática - PUC-Rio

Rio de Janeiro, September 19th, 2024

Venícius Garcia Rego

Graduated in Computer Science by the Universidade Federal do Maranhão.

Bibliographic data

Rego, Venícius Garcia.

Evaluating LLM In-Context Few-Shot Learning on Legal Entity Annotation Task / Venícius Garcia Rego; advisor: Hélio Côrtes Vieira Lopes; co-advisor: Fernando Alberto Correia dos Santos Junior. – 2024.

78 f: il. color. ; 30 cm

Dissertação (mestrado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, 2024.

Inclui bibliografia

1. Informática – Teses. 2. LLM. 3. Direito. 4. Anotação de Entidades Legais. 5. Few-Shot Learning. I. Lopes, Hélio. II. A. Correia, Fernando. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. IV. Título.

CDD: 004

Acknowledgments

A heartfelt thank you to my family, who has always provided the emotional support I needed to continue researching and studying. Specifically, to my parents, Benilson and Hettie, for all the love and care they have given me and for living this dream with me, counting down the days until I finish my master's degree. To my dear sisters, Laynna and Alanna, for all the calls and laughter that made the tough times so much easier. Lastly, to my girlfriend, Glaucia, for being my partner through it all, for always caring about my health, and for all the love you've given me.

To my friends Arthur, Boaro, Laryssa, Pedro Thiago, and everyone else for all the advice and the fun moments we shared. A special thanks to my friend Carlos, who traveled with me to Rio de Janeiro with the same goal, for sharing all the tough moments of loneliness and homesickness, as well as the joyful times when we lived together.

To my advisor, thank you for believing in me and allowing me to be part of the lab. This greatly facilitated this journey and helped me grow as a researcher and professional. Thank you for all the advice and words of encouragement.

To my co-advisors, who are also friends, for their immense help in developing this work; I couldn't have done it without you. I owe you so much, and I hope to continue researching alongside you in the future.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Rego, Venícius Garcia.; Lopes, Hélio (Advisor); A. Correia, Fernando (Co-Advisor). **Evaluating LLM In-Context Few-Shot Learning on Legal Entity Annotation Task**. Rio de Janeiro, 2024. 78p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A considerable amount of legal documents is available on the Internet nowadays. Even so, knowledge extraction activities, such as Named Entity Recognition (NER), in the legal domain are still challenging, even more so when are not in English. One of the reasons is the low amount of annotated corpora available, combined with the burden and cost of developing a new one. The legal annotation task is itself challenging due to limitations on both time and human resources. The emergence of Large Language Models (LLMs) has attracted attention due to their capability of reasoning using only in-context information about the tasks. Recent studies present significant results regarding its usage in document annotation tasks; in some cases, the model is comparable to human annotators. Thus, in this work, we evaluate LLM's in-context few-shot learning capability on a legal NER, assessing its usage in an annotation task process with humans. To do so, our study is based on the data gathered along an annotation task previously conducted to produce a corpus of legal decisions written in Portuguese, published by Brazilian Supreme Federal Court (STF), dedicated to the NER, and annotated by law students. Our experiments showed that the LLM can produce highly accurate annotations, without any gradient update. Thus, may can assist annotators in the annotation process, reducing the amount of time and effort and making the annotation task more efficient.

Keywords

LLM; Legal; Legal Entity Annotation; Few-Shot Learning.

Resumo

Rego, Venícius Garcia.; Lopes, Hélio; A. Correia, Fernando. **Avaliando LLM na Tarefa de Anotação de Entidades Legais Utilizando Few-Shot Learning**. Rio de Janeiro, 2024. 78p. Dissertação de Mestrado – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Um número considerável de documentos no domínio do Direito estão disponíveis hoje na Internet. Mesmo assim, atividades de extração de informação, como Reconhecimento de Entidades Nomeadas (REN), no domínio do Direito, continuam desafiadoras, principalmente quando não são em Inglês. Um dos motivos é a escasses de corpus anotados, combinado com a dificuldade e custos de desenvolvimento. A tarefa de anotação de dados legais é custosa por limitações de tempo e de recursos humanos. O surgimento dos Modelos de Linguagem Grandes (LLMs) atraiu atenção por conta da capacidade de performar tarefas apenas com uma descrição ou exemplos de como realizar a atividade, em linguagem natural, passados no contexto. Estudos recentes apresentaram resultados significativos em relação a utilização de LLMs na tarefa de anotação de documentos, em alguns casos, a performance do modelo era comparável a de anotadores humanos. Portanto, neste trabalho, propomos avaliar a capacidade de LLMs na tarefa de anotação de entidades nomeadas em documentos do domínio do Direito utilizando *Few-shot Learning*, verificando sua utilização no processo de anotação junto com humanos. Para realizar a avaliação, utilizamos um corpus em Português dedicado ao REN contendo decisões do Supremo Tribunal Federal (STF) que foram previamente anotadas por estudantes de Direito. Os resultados obtidos mostram que LLMs são capazes de reconhecer corretamente as entidades presentes no texto e de produzir anotações precisas sem a necessidade de treinar novamente o modelo, portanto, podem auxiliar no processo de anotação, diminuindo a carga de trabalho dos anotadores e tornando a tarefa de anotação mais eficiente.

Palavras-chave

LLM; Direito; Anotação de Entidades Legais; Few-Shot Learning.

Table of contents

List of figures

List of Abreviations

BERT – Bidirectional Encoder Representations from Transformers

CRF – Conditional Random Fields

FFN – Feed Forward Network

GPT – Generative Pre-Trained Transformer

GRU – Gated Recurrent Units

HMM – Hidden Markov Models

LLaMA – Large Language Model Meta AI

LLM – Large Language Models

LSTM – Long Short-Term Memory

MoE – Mixture-of-Experts

NER – Named Entity Recognition

NL – Natural Language

NLP – Natural Language Processing

PII – Personally Identifiable Information

RNN – Recurrent Neural Networks

STF – Supreme Federal Court

"You must understand that there is more than one way to the top of the mountain."

Musashi Miyamoto, *Gorin No Sho.*

1
Introduction

Despite the large number of legal documents available on the Internet nowadays, the Named Entity Recognition (NER) tasks —a subtask from the field of Natural Language Processing (NLP)— in the legal domain are still challenging. One of the reasons is the low amount of annotated corpora available, combined with the burden and cost of developing a new one.

Addressing this gap for corpus in Portuguese, Correia et al. (2022) developed the most extensive known corpus for legal NER, containing 594 decisions issued by the Brazilian Supreme Court (STF) between 2009 and 2018. The annotation process employed was composed of two short training efforts and a longer final task. It was supported by the collaboration of 95 law students who performed the task over months under close supervision.

Some aspects of the annotation task in Correia et al. (2022) were based on Leitner, Rehm and Moreno-Schneider (2020). Similar studies also present annotated corpus in the legal domain Cao et al. (2022), Brito et al. (2023). In some of these works, the annotations were made by the authors themselves (LEITNER; REHM; MORENO-SCHNEIDER, 2020) or a small team of non-specialists (CAO et al., 2022), which, due to the data nature and volume, may overwhelm the annotators and increase the effort and time spent on the annotation task. An additional burden is annotation inconsistency and human subjectivity, which lead to long iterations for inspection and review. In Brito et al. (2023), the first annotation step took over two months, even with 36 legal experts (i.eg., public prosecutors and judges).

Thus, performing an annotation task in the legal domain is challenging due to limitations in terms of both time and specialized human resources. Furthermore, it is worth mentioning that none of these works mention the cost related to the annotation task, which makes it reasonable to assume that cost is also a limiting factor — the more annotators involved and the greater the expertise, the more expensive the annotation task is likely to be.

One of the recent changes in NLP was the shift from task-specific to task-agnostic models, where task-specific models are designed for particular tasks. In contrast, task-agnostic models aim to be versatile across a range of tasks. Those models gained popularity due to their effectiveness in learning general representations over unlabeled data that can later be fine-tuned to adapt a task-agnostic model to perform a desired task (BROWN et al., 2020). However, even this final step of fine-tuning may not be necessary, and a pre-

trained Language Model can be zero-shot transferred to perform standard NLP tasks (BROWN et al., 2020; RADFORD et al., 2019).

In this sense, Large Language Models (LLMs) demonstrate powerful in-context, few-shot learning capability on many text-annotation tasks, comparable with or even outperforming human annotators (GILARDI; ALIZADEH; KUBLI, 2023). Moreover, this prompting-based approach requires no large training sets, unlike fine-tuning a language model for a new task (WEI et al., 2022), which is useful when dealing with limited corpora scenarios. However, even with this powerful demonstration, the capabilities of LLMs on legal NER have not yet been truly explored.

1.1 Research Goal

Due to the powerful few-shot learning capability on many text-annotation tasks, LLMs could be added to the annotation task to aid the annotators in recognizing named entities in the text and speeding up the annotation task.

Given the remarkable performance of LLMs and the development gap of legal annotated corpora, our main research question is:

> ***MRQ**: How to use Large Language Models in the Legal Named Entity Recognition task?*

Moreover, the process developed to resolve the *MRQ* has to be reliable and replicable, and the resulting annotations must be capable of being validated.

Thus, seeking to answer the *MRQ*, this work presents a legal named entity recognition process with LLMs using Few-shot learning. Our study had as reference the annotation performed in Correia et al. (2022) and the corpus presented by them. A specific set of prompts was constructed and sent to every LLM in the study, considering the entity definitions and guidelines presented in their work. We also used the annotations from the two shorter efforts to create the examples database, performing the few-shot learning.

We also address the following research questions and sub-questions:

> ***RQ1**: How do we select a valuable set of examples for the prompt engineering process?*
>
> ***RQ1.a**: How do selected examples affect LLM's performance and generation of annotations?*
>
> ***RQ2**: What number of examples should be used to perform the task?*

***RQ2.a**: How does the number of examples affect LLM's performance and generation of annotations?*

***RQ3**: How do we evaluate the annotations generated by the LLMs?*

To answer the *RQ1* and *RQ2*, we assessed the annotation capabilities of six different LLMs, both open- and closed-source, at a coarser-grained level and in a strict- and relaxed-match way (LI et al., 2023b). Our experiments were split into two phases. In the first phase, the experiments were conducted on five documents (totaling 17,568 tokens, with 337 annotations covering 2,873 tokens). We also evaluated each model's sensitivity to in-context examples by changing the selection strategy and the number of examples in the prompt. Three example selection strategies were developed: random selection, similarity selection, based on the similarity between examples and the input text, and clustering selection, selecting the most representative set of examples. Further, in the second phase, we select the best model and configuration, based on the result of the prior phase, for annotating an additional 53 documents within 134,711 tokens and 2,585 annotations covering 21,311 tokens.

As a result, we find no significant differences in performance among the example selection strategies. Also, increasing the number of examples led to overall performance improvement in two of the six models. In contrast, for one model, it led to worse performance. The smaller models struggled to recognize and annotate the entities correctly. On the other hand, the larger models surpassed an F1 score of 0.70 in the first phase, and the best model achieved 0.76 in the second phase. Thus, the LLMs can generate good-quality annotations by recognizing most entities in the decisions. They may also be a valuable asset in helping the annotators with the annotation process.

1.2 Expected Contributions

- We present a new legal named entity recognition process that leverages LLMs' in-context few-shot learning capabilities.
- We present a reliable evaluation of the annotations generated, comparing them with those made by human annotators on the most extensive corpus known for legal NER.
- We assess the LLM's sensitivity to in-context examples by changing how they were selected and the impact of the number of examples included in the prompt.

1.3 Overview

This work is organized as follows: Chapter 2 describes the background and definitions providing the context and concepts required to understand this proposed process. Chapter 3 presents the related work that guided and founded this work. Chapter 4 presents the corpus acquired to evaluate the LLMs annotations capabilities, and also describes each entity included. Chapter 5 presents the proposed process for legal NER using LLMs, the examples database, prompt construction, and the example selection strategies developed. Chapter 6 describes the experiments executed to evaluate the LLMs in the proposed process and presents a detailed analysis of the results. Finally, Chapter 7 presents our conclusion of the proposed process, and reiterates the contributions and research questions that have been answered.

2
Background and Definitions

This chapter contextualizes the background of the NER tasks and Legal NER and provides the definitions of important concepts such as Transformer architectures. Finally, it also describes large language models and introduces the LLMs selected for this work.

2.1
Named Entity Recognition

Named Entity Recognition (NER) is a sub-task of Information Extraction (IR) that aims to identify and categorize pieces of text into predefined types of information elements called Named Entities (NE) (NADEAU; SEKINE, 2007; MARRERO et al., 2013), e.g., Person, Location, and Organization (GRISHMAN; SUNDHEIM, 1996). So, formally, given a sequence of tokens $s = \langle t_1, t_2, ..., t_{n-1}, t_n \rangle$ the output of a NER system will be a set of predefined labels $l = \langle l_1, l_1, l_2, ..., l_{k-1}, l_k \rangle$ and their corresponding tokens where they were mentioned. The NER task is also a fundamental pre-processing step for other tasks in natural language processing (NLP), such as text summarization, question answering, knowledge base construction, etc (LI et al., 2023b).

The three most common strategies employed in NER tasks are described in the following sub-sections.

2.1.1
Rule-based

The text is labeled using hand-coded rulings based on grammatical or syntactic-lexical patterns, which demands extreme expertise in the domain. These rules can be designed using domain-specific gazetteers, which link named features to their location and type (GOODCHILD; HILL, 2008), such as anchor links on Wikipedia (GATTANI et al., 2013; TORISAWA et al., 2007), using part-of-speech tagging, and synonym dictionaries (LI et al., 2023b).

This approach often has flaws due to limitations on the rules, like incomplete dictionaries, and due to domain-specific rules, they cannot be transferred to other domains. Combined with other techniques, the rule-based approach can also be used to extract information and post-process for classification using supervised, semi-supervised, or un-supervised models (FETAHU et al., 2021).

2.1.2 Unsupervised Learning Approach

In this approach, mainly a Clustering-based system, the named entities are extracted using a semantic vector to group the unlabeled data by context similarity (LI et al., 2023b). For example, Zhang and Elhadad (2013) developed an unsupervised biomedical entity recognition system for clinical and biological text by similarity between the entity class and the candidates from the corpus. The entity classes and the text from the corpus were represented by signature vectors using an inverse document frequency (IDF) based technique and cosine similarity to categorize the text.

2.1.3 Supervised Learning Approach

In this approach, supervised models are trained using annotated data samples to perform a multi-class classification or a sequence label task. From the data features, these models learn to recognize similar patterns from unlabeled data. Some of the models that have been applied to supervised NER are Hidden Markov Models (HMM) (EDDY, 1996), Decision Trees (QUINLAN, 1986), Conditional Random Fields (LAFFERTY; MCCALLUM; PEREIRA, 2001) or more recently deep architecture like BERT models (DEVLIN, 2018).

The supervised learning approach requires a prior step of manual annotation, and this process is often costly. Moreover, a large training set is necessary for deep architectures, so this approach may not be feasible for limited-source scenarios.

2.1.4 Named Entity Recognition in the Legal Domain

Implementing and applying artificial intelligence approaches in the legal domain to reduce the exhaustive and redundant work for legal professionals, turning the legal system more efficient, has become a research hot-spot nowadays (ZHANG et al., 2023).

Furthermore, as many of the resources in this field are presented in text form, such as decisions, contracts, and legal opinions, most of the legal tasks are based on NLP (ZHONG et al., 2020). Therefore, information extraction tasks, including NER, can greatly benefit the legal domain.

Some of the applications enabled by NER in legal texts are providing links to cited laws and legal cases, clustering similar documents, extraction of events and relationship of documents, legal search and summarizing, quantifying

citation relevance, legal judgment prediction, and so on (ARAUJO et al., 2018; ZHANG et al., 2023; CORREIA et al., 2022).

Most of the NER systems focus only on the flat annotations, e.g., person, location, and organization. Still, legal texts are more complex and specific compared to general fields (ZHANG et al., 2023). The same person entity in the legal context may be referencing the judge who wrote the ruling, lawyers, the person accused, the name of the author of an academic citation cited in the decision, etc. All this semantic information will be lost considering only a flat annotation.

To this end, many research studies that aim to develop annotated corpora for legal NER work with the idea of nested named entities to capture these details on legal documents, such as Leitner, Rehm and Moreno-Schneider (2020), Cao et al. (2022), Correia et al. (2022), Brito et al. (2023). Moreover, as suggested by Ringland et al. (2019), the benefit of the nested entity recognition approach is the capture of important information such as entity-entity relationships, entity attribute values, and part-whole relationships.

In Leitner, Rehm and Moreno-Schneider (2020), the legal entities are described as either designations, the title of legal documents, and consist of a long title, short title, and an abbreviation or references to acts, norms, or contracts.

2.2 Transformer

Recurrent neural models, like LSTM and GRU, present a natural constraint of sequential computation due to their generation process, where they generate a sequence of hidden states h_t based on previous hidden states h_{t-1} and the input position t. This constraint hinders parallelization, becoming critical at longer sequence lengths (VASWANI, 2017).

To this end, Vaswani (2017) developed the Transformer model architecture that eschews the recurrence process and relies entirely on an attention mechanism to compute representations of input and output. This mechanism relates different positions of the sequences, creating dependencies even for distant positions, without sequence-aligned RNNs or convolution, then allowing significantly more parallelization. Figure 2.1 shows the proposed Transformer model architecture.

The Transformer consists of two stacks: an encoder and a decoder.

– **Encoder**: is composed of a stack of 6 identical layers, and each layer has two sub-layers; the first is a Multi-Head Attention layer, and the second is a position-wise fully connected feed-forward network.

– **Decoder**: is another six identical layers stack, as the encoder, with the addition of one more Multi-Head Attention layer, that operates over the encoder output. Moreover, the self-attention sub-layer is also modified to prevent positions from obtaining subsequential information.

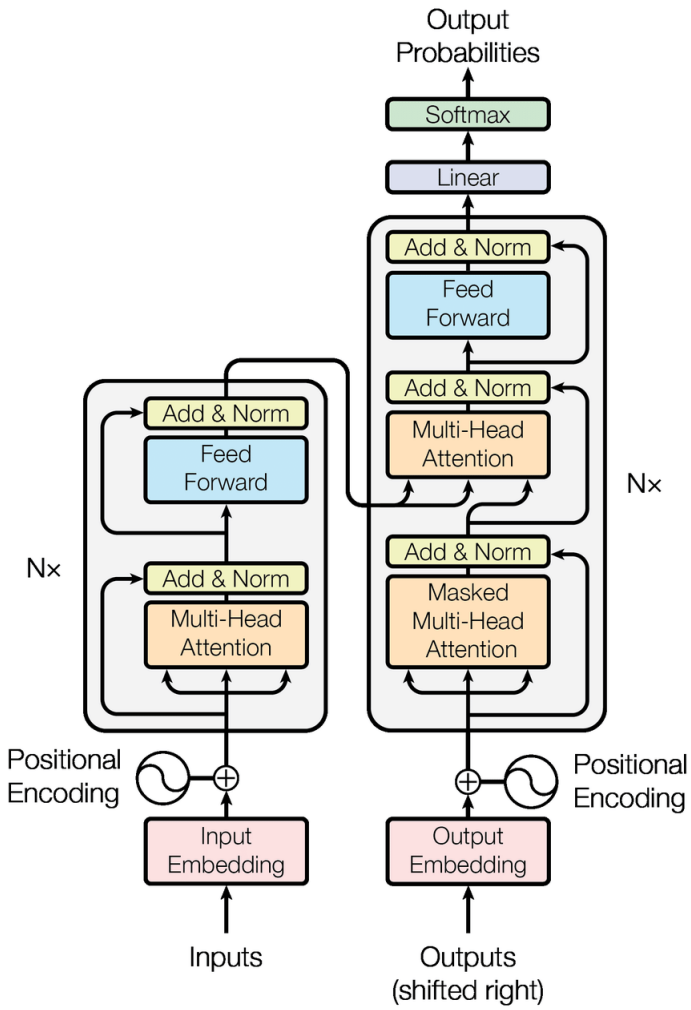


Figure 2.1: The Transformer model architecture extracted from Vaswani (2017)

The Multi-Head Attention consists of several scaled dot-product attention layers, running in parallel, between queries and keys of dimension d_k, and values of dimension d_v, as presented in Figure 2.2. The dot-product of the query with all keys is scaled by dividing for $\sqrt{d_k}$, and then applying a softmax function to obtain the weights on the values.

The Equation 2-1 presents the attention function computed on a set of queries, keys, and values packed together in the matrices Q, K, and V.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2-1}$$

Each attention result is concatenated and once again projected. Thus, the Multi-Head Attention can be described as follows:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^o \quad (2\text{-}2)$$

$$Attention = (QW_i^q, KW_i^k, VW_i^v) \quad (2\text{-}3)$$

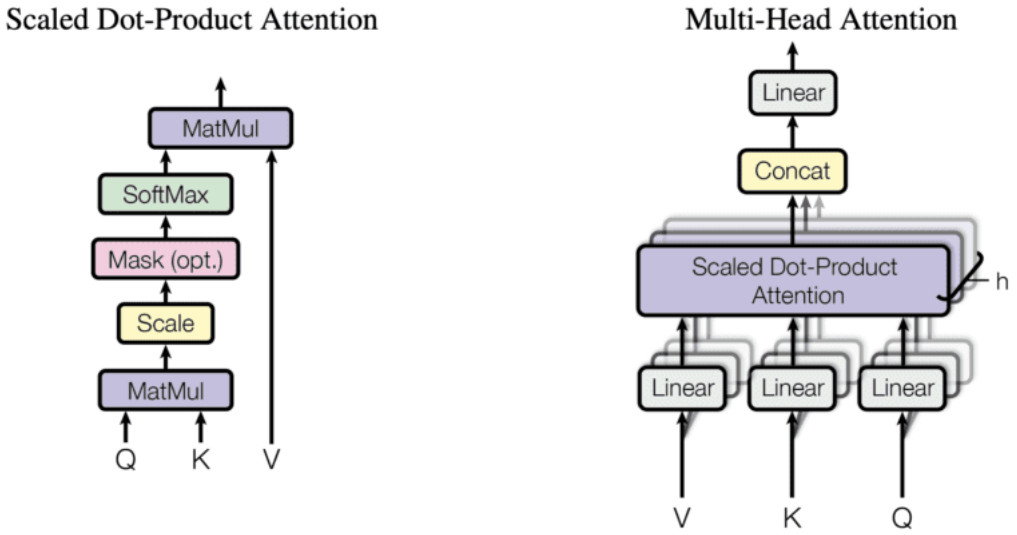


Figure 2.2: Scaled Dot-Product Attention and Multi-Head Attention extracted from Vaswani (2017)

2.2.1 BERT

Proposed in Devlin et al. (2018), Bidirectional Encoder Representations from Transformer, or BERT, are multi-layer bidirectional Transformer-based models designed to pretrain deep bidirectional representations from the unlabeled text. The BERT models can be easily transferred to another task by including an additional output layer and then fine-tuning the model without architectural changes. Figure 2.3 shows the two-step framework for BERT models.

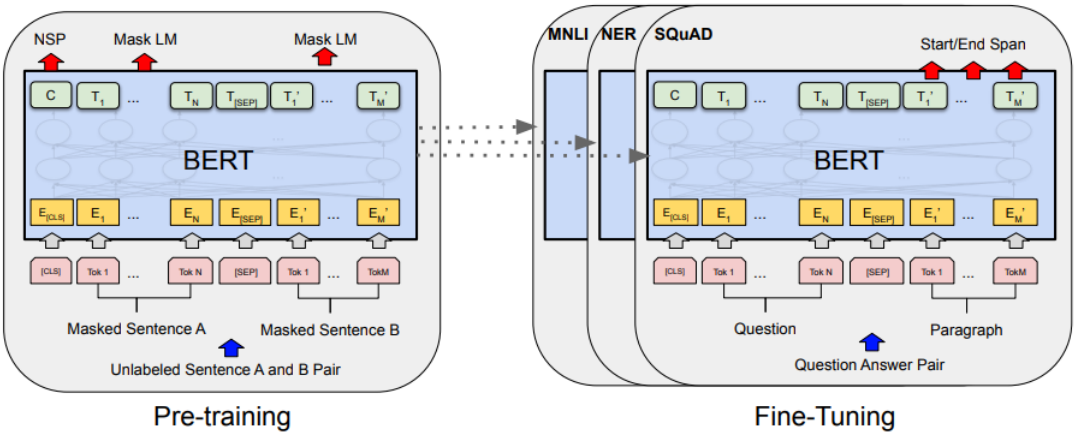


Figure 2.3: BERT procedures for pre-training and fine-tuning extracted from Devlin (2018)

To adapt to many downstream tasks, the BERT input representation can be both a single sentence and a pair of sentences in one token sequence. In the input sequence, the first token is always the special token *[CLS]* used for classification tasks, and the *[SEP]* special token is used to separate the pair of sentences. The special token *[MASK]* is only used during the deep bidirectional representations training, where some random percentage of the input sequence is masked for further prediction.

2.3 Large Language Models

Large Language Models (LLM) refer to Transformer-based (VASWANI, 2017) models containing billions of parameters that were pre-trained on a massive volume of unlabeled corpora. The LLMs emerged from the pivoting of NLP task-specific models to general-purpose models, such as BERT, with the pre-training and fine-tuning paradigm and the findings of capacity improvement on downstream tasks by scaling up pre-trained language model and data size Brown et al. (2020).

The evolution process of language models came through to only specific task helpers or language modeling to complex task solvers (ZHAO et al., 2023) nowadays with the latest LLMs development by OpenAI (2023a), Google (2023), Anthropic (2023), Meta (2024). To this end, these foundation models are pre-trained using next-word prediction to learn the language structure and representations. Finally, to follow instructions and behave as expected, they are post-trained in a supervised fine-tuning process with paired instructions and responses and human feedback (DEEPMIND, 2024; DUBEY et al., 2024; OPENAI et al., 2024; CHRISTIANO et al., 2017).

However, due to their size, they require larger computing resources for training, which makes it difficult to adopt LLMs; for example, for training, the new Llama 3.1 requires 16 thousand GPUs, each one with 80 GB and 240 PB for storage (DUBEY et al., 2024). Some architecture optimizations emerged to minimize these costs, such as the Mixture-of-Experts (MoE), implemented in the DeepSeek V2 model (DEEPSEEK-AI et al., 2024), which can save 42% of training costs.

2.3.1 Prompt

The prompt is an instruction in natural language that guides the model to predict the desired output (DONG et al., 2024). An instruction can be described as the bridging of different combinations of an input X, single or a group of sentences, for classification, textual entailment, NER, etc; and an output Y, which can be one or multiple labels, or any sequence for text generation tasks (LOU; ZHANG; YIN, 2024).

The process of finding and constructing the prompt that improves the performance of LLMs for the downstream tasks is called prompt engineering. In this process, several strategies and tactics are applied to leverage the LLM's capabilities of resolving tasks given demonstrations (OPENAI, 2023b; LIU et al., 2023). One kind of prompt strategy is the Few-shot in-context learning, in which a set of supervised demonstrations is included with the instruction to condition to desired output; this strategy does not require parameter update and is performed on pre-trained LLMs (DONG et al., 2024).

2.3.2 LLMs Inference Parameters

The LLMs have parameters that control how the model generates responses. With these parameters, the models can generate different responses based on different values provided. The most commonly used are:

- **Max output tokens**: Controls the maximum number of tokens generated on response.
- **Temperature**: Controls the randomness of the sequence generated, with values close to 0, tokens with high probability tend to be chosen, generating more concise and precise responses; values close to 1 lead to more diverse or creative results.
- **topK** - Controls how the model selects the tokens for output. A topK of 1 means that the token selected is the most probable among all the tokens,

and a topK of 3, for example, means that the next token is selected from among the 3 most probable using the temperature.

– **topP** - Controls how the model selects the tokens for output, to the most to least probable, until the sum of their probabilities equals the topP value. When the sum of their probabilities reaches the topP value, one of the sampled tokens is selected using the temperature parameter.

The LLMs used in this work are presented in the following sections.

2.3.3 Gemini 1.5

Gemini 1.5 Pro and Gemini 1.5 Flash are the latest generation of multimodal models from Deepmind (2024). These models included major advances in sparse and dense scaling, training, and distillation and are capable of recalling fine-grained information from millions of tokens of context (DEEPMIND, 2024), and were pre-trained across many different domains, including web documents, codes, image, audio and video content.

The Gemini 1.5 Pro is a sparse mixture-of-expert (MoE) Transformer-based model that learns to direct inputs to a subset of the model's parameters for processing; this conditional computation allows the growth of the model and data size while keeping only one or few experts running for a given input. Moreover, the Gemini 1.5 Flash, a lightweight variant, is a transformer decoder distilled from the Gemini 1.5 Pro model, with the same multimodal capabilities and lower latency, designed for efficiency with minimal regression in quality.

The Gemini 1.5 Pro model has an input token limit of two million tokens and an output of eight thousand; the Gemini 1.5 Flash has an input limit of one million tokens and the same output limit as the Pro version.(DEEPMIND, 2024).

2.3.4 Llama 3.1

The Llama 3.1 developed by Dubey et al. (2024) is a dense Transformer-based model with a context window of up to 128 thousand tokens, the overview of Llama 3.1 parameters is present in Table 2.1. The largest model has 405 billion parameters. The Llama 3.1 was pre-trained on a corpus of about 15 trillion multilingual tokens, compared to 1.8T tokens from the previous Llama 2.

The dataset developed in (DUBEY et al., 2024) for language model pre-training was made from a variety of data sources, most obtained from the web,

containing knowledge up to the end of 2023. Furthermore, the collected data was submitted to several cleaning processes to ensure high-quality training data. Some of the cleaning processes were:

- **Safety Filtering**: to remove domains ranked as harmful or containing high volumes of Personally Identifiable Information (PII).
- **De-duplication**: on URL-level, to keep only the most recent version of pages relative to each URL, document-level removing near duplicate documents, and line-level to remove lines that appeared more than 6 times in each bucket of 30M documents.
- **Heuristic Filtering**: heuristics to remove additional low-quality documents, outliers, and documents with excessive repetitions.
-

The final dataset contains 50% of tokens corresponding to general knowledge, 25% of mathematical and reasoning tokens, 17% code tokens, and 8% multilingual tokens.

Table 2.1: Overview of the key hyperparameters of Llama 3.1 extracted from Dubey et al. (2024)

	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8		
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE($\theta = 500,000$)		

The results reported by Dubey et al. (2024) suggest that Llama 3.1 405B performs on par with leading solutions such as GPT-4 across a variety of tasks. On the other hand, the smaller versions with 8B and 70B parameters outperform alternative models with a similar number of parameters, such as Mistral 7B, Mixtral 8x22B, and GPT-3.5 Turbo.

2.3.5 GPT-4o mini

The GPT-4o mini is the most cost-efficient small model from OpenAI, more than 60% cheaper than the GPT-3.5 Turbo. Moreover, it supports text, image, video, and audio inputs and has an input context window of 128 thousand tokens and up to 16 thousand output tokens per request. This model was pre-trained with data collected up to October 2023. Furthermore, the GPT-4o mini surpasses other small models on academic benchmarks across both textual and multimodal.

2.3.6 DeepSeek V2

The DeepSeek V2 developed by DeepSeek-AI et al. (2024) is an economical training and efficient inference MoE model, composed of 236B parameters, of which 21B are activated for each token, with an input context window of 128 thousand tokens. The DeepSeek V2 also adopts a modification in the Multi-Head Attention proposed in Devlin (2018), a Multi-Head Latent Attention (MLA), which compresses and caches part of the computation in MLA, ensuring efficient inference.

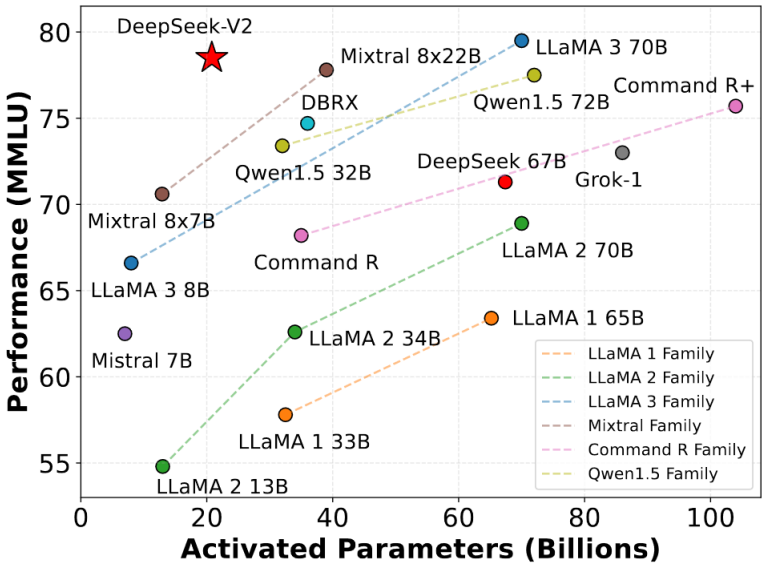


Figure 2.4: DeepSeek V2 performance on Multi-task Language Understanding on MMLU benchmark extracted from DeepSeek-AI et al. (2024)

The model was pre-trained on high-quality data and in a variety of source corpus containing 8.1T tokens. Further, the model was submitted to a supervised fine-tuning process and reinforcement learning to improve the instruction-following and align with human preference. Compared with

the previous DeepSeek version with 67B parameters, it obtained a stronger performance, as shown in Figure 2.4, while saving 42.5% of training costs and generating the tokens 5.76 times faster.

3
Related Work

The emergence of LLMs has attracted the scientific community's attention due to their powerful in-context, few-shot learning capability, and many works have appeared trying to explore and evaluate their performances in information extraction for data annotation tasks, aiming to discover time-saving and cost-effective approaches(ALDEEN et al., 2023; XIE et al., 2023; WANG et al., 2023; LI et al., 2023a).

Aldeen et al. (2023) evaluates LLM data annotation capabilities across ten different datasets with a diverse number of classes using three prompt strategies. The datasets used for evaluation were Website Classification, Question Classification, Banking Queries, Sarcasm Headlines, Mental Health, Emotions, Spam Messages, News Headlines, Amazon Reviews, and Twitter Topic Classification. Furthermore, the prompt strategies implemented are as follows: (1) a baseline straightforward approach, with only a task description without the label's context; (2) to improve the annotations' accuracy and context understanding, a description of each label is provided; and finally in (3), was attributed a role of specialist annotator, instructing the LLM to use the expertise in data annotation and labeling, combining with the prompt (1). The three prompt strategies aim to assess the LLM's behavior due to their significant influence on generated responses. GPT-3.5 and GPT-4 models were evaluated on the ten datasets. Moreover, for the GPT-3.5 two values of temperature parameter were tested: 0.25 and 1. As a result, the GPT-4 model was the most proficient model across many datasets. Moreover, some prompts worked better for certain tasks, and their effectiveness can vary depending on the task and the nature of the data. Finally, no significant differences were observed in GPT-3.5 performance between using 0.25 or 1.0 temperature, indicating that adjusting this parameter only, had minimal impact on the datasets tested.

Our work is similar to Aldeen et al. (2023) regarding improving the annotations by providing a better context understanding describing each label and the task. Still, we opted not to implement the other approaches; we believe the straightforward approach is too shallow to help in a more challenging domain.

The temperature results observed in Aldeen et al. (2023) are corroborated by Renze and Guven (2024), which also investigates the effect of temperature sampling. A total of nine LLMs, Claude 3 Opus, Command R+, Gemini 1.0 Pro, Gemini 1.5 Pro, GPT-3.5 Turbo, GPT-4, Llama 2 7 and 70B, and

Mistral Large, were tested on a multiple-choice question-and-answer (MCQA) exam developed using LLMs benchmark problems randomly sampled. The experiment results suggest again that sampling temperature from 0.0 to 1.0 does not produce statistically significant differences in performance on MCQA. Due to observed results in Aldeen et al. (2023) and Renze and Guven (2024), our work set the temperature parameter to 0. in all experiments to provide a more constant output while not affecting the performance.

Regarding the use of LLMs on NER tasks, Xie et al. (2023) explores zero-shot learning as Aldeen et al. (2023) and few-shot learning. Xie et al. (2023) enables the model to extract relevant information by analyzing the syntactic structure of the input text. The extraction can either made by LLM itself by providing syntactic hints, '*First, let's perform Parf-of-Speech tagging. Then, we recognize named entities based on the Part-of-Speech tags*', in the input instruction or using a parsing tool. The syntactic information contained noun phrases, Part-of-Speech tags, constituency and dependency trees, and word segmentation only for Chinese. Moreover, as it is challenging to recognize all entities at the same time, even when the label size is large, or the data is from an out-of-distribution domain, it adopts a decomposed strategy where the NER task is broken down into simpler sub-problems; the recognition process is a multi-turn dialogue where the LLM recognizes one type of entity per dialogue iteration. GPT-3.5 was the main subject of the evaluation, but GPT-3 and Llama2 were also evaluated. The GPT-3.5 was evaluated across seven benchmarks from general-domain as ACE05, ACE04, OntoNotes 4, MSRA, Weibo NER, and for domain-specific PowerPlantFlat and PowerPlantNested containing either flat entities or nested. As a result, the proposed strategies improved zero-shot and few-shot NER across the seven benchmarks. The decomposed strategy for zero-shot achieves a significant improvement of 9.22% of F1 for domain-specific and 3.82% for general-domain compared to the base method which does not divide. Moreover, using the parsing tool exhibits consistent improvements across six datasets. The strategies also improved the few-shot, which achieves 0.57 of F1 on Ontonotes 4 using 10-shots and 0.42 of F1 on PowerPlantFlat using 10-shots.

Due to the legal domain's natural difficulty even for legal professionals, our work adopted a decomposed strategy, dividing the legal NER tasks into sub-problems. Our work differs from the Xie et al. (2023) because our recognition process is not a multi-turn dialogue. The annotations are made entirely separated, and further, with all annotations for each entity generated, we combine the annotations in a single document. Besides, we also implemented a few-shot strategy, as corroborated by Xie et al. (2023). This

approach demonstrates significant improvement compared to zero-shot.

Furthermore, Wang et al. (2023) adapt LLMs to NER by transforming the sequence labeling task to a generation task, instructing them to generate labeled sequences by surrounding the entities with special marks, and also proposes a self-verification to review the extracted entities to handle the hallucination problem. (WANG et al., 2023) conducted the experiments using GPT-3 on both flat and nested NER datasets, and the LLM achieved comparable performances to supervised baselines, demonstrating remarkable ability in a low-resource scenario.

Similar to Wang et al. (2023), our work adapts the legal NER tasks to a sequence labeling task. Through prompt engineering, we conduct the LLM to generate the same sequence, adding special symbols to mark the legal entities.

On the other hand, works like Li et al. (2023c) focused on creating a framework of collaboration between human annotators and LLMs, allocating some data for an LLM to annotate based on the quantified confidence of how well the model can annotate these data points. The results showed that the confidence scores generated by LLM are well-calibrated and can achieve more efficient and accurate work allocation using this score than the random allocation baseline. We deeply believe in the collaboration of LLMs and human annotators, and similar to Li et al. (2023c), this work proposes a process that leverages the LLMs' annotations capabilities to help the annotators by reducing the workload.

Deferring from previous presented works, which focused mainly on only one LLM, our work evaluates six LLMs, both open- and closed-source, to better assess and comprehend the actual scenario of LLM's in-context few-shot learning capabilities on the NER task.

4
Data Acquisition

The dataset used in our experiments developed in Correia et al. (2022) has 594 decisions published by STF between 2009 and 2018, divided into two classes of documents: monocratic single justice decisions (261 documents) and collegiate decisions (333 documents). Moreover, each decision has two levels of nested legal entity annotation: four coarser legal named entities and twenty-four nested ones (fine-grained). The four coarser named entities on the first level are Academic Citations, Legislative References, Persons, and Precedents.

4.1
Academic Citations

Represents a direct citation of books, articles, and journals written by judges on the decision, often used to support arguments in a ruling. Although academic citations are rare regarding the other entities, they can offer valuable information of the influence of certain authors in the legal debate (CORREIA et al., 2022). Correia et al. (2022) mapped six possible fine-grained entities within an academic citation. An example of an academic citation is shown in Figure 4.1.

> Nesse ponto, destaco que Celso Antônio Bandeira de Mello elege como pedras de toque do Regime Jurídico Administrativo os princípios da supremacia do interesse público sobre o interesse privado e o princípio da indisponibilidade do interesse público (BANDEIRA DE MELLO, Celso Antônio. Curso de Direito Administrativo, 25ª Edição. São Paulo: Malheiros, 2008. p. 55), do que decorre o princípio constitucional da legalidade para a Administração Pública.

Figure 4.1: Academic Citation example with the author, title, and publisher information

	Total	Per excerpt				
		min	max	average	std	median
Sentences	62,933	3	551	105.97	81.33	93.0
Tokens	1,782,395	121	16,087	3,000.66	2,501.79	2,692.5
Coarser-grained	33,055	1	267	55.65	42.44	47.0
Fine-grained	57,573	0	507	96.92	69.21	81.0

Table 4.1: Counting of sentences, tokens, coarse and fine-grained annotations collected from Correia et al. (2022).

4.2 Legislative references

Citations to legislative references, a fundamental part of legal reasoning, consist of articles, laws, sections, and constitution references mentioned in the decisions. This entity has seven possible fine-grained entities. An example of legislative reference is present in Figure 4.2

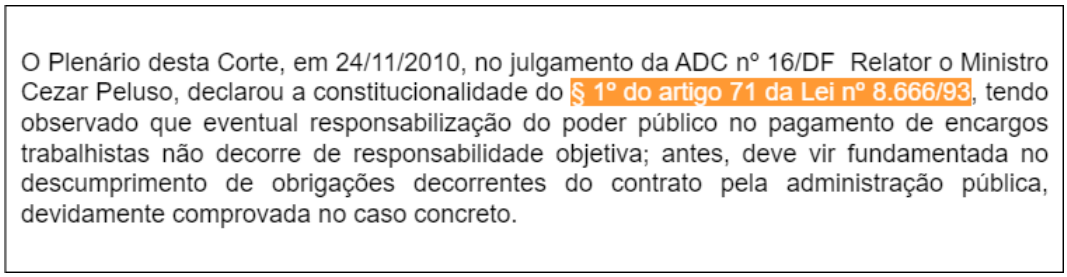

O Plenário desta Corte, em 24/11/2010, no julgamento da ADC nº 16/DF Relator o Ministro Cezar Peluso, declarou a constitucionalidade do § 1º do artigo 71 da Lei nº 8.666/93, tendo observado que eventual responsabilização do poder público no pagamento de encargos trabalhistas não decorre de responsabilidade objetiva; antes, deve vir fundamentada no descumprimento de obrigações decorrentes do contrato pela administração pública, devidamente comprovada no caso concreto.

Figure 4.2: Legislative Reference example with the author, title, and publisher information

4.3 Persons

This entity represents name, surname, titles, and treatment pronouns as long as they are followed by the name and not within other entities. The primary purpose is disambiguation regarding personal identification on the other entities. Also, this is the only coarser-grained entity in the corpus with no fine-grained elements linked to it. The Figure 4.3 presents an example of a Persons entity.

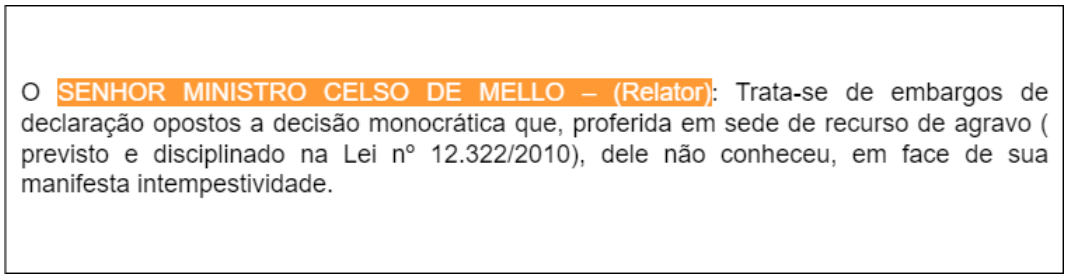

O SENHOR MINISTRO CELSO DE MELLO – (Relator): Trata-se de embargos de declaração opostos a decisão monocrática que, proferida em sede de recurso de agravo (previsto e disciplinado na Lei nº 12.322/2010), dele não conheceu, em face de sua manifesta intempestividade.

Figure 4.3: Persons example

4.4 Precedents

Citations to prior court decisions in the ruling. As cited in Leibon et al. (2018), precedent citation undoubtedly has great value in common-law-based judicial systems, such as in the United States and Canada, where courts are bound to their previous rulings. The precedent references in STF do not follow

Coarser-grained Entity	Per Excerpt	Total	Average Tokens
Academic Citation	2.99	1,775	24.60
Precedents	15.33	9,108	10.27
Legislative Reference	17.22	10,229	8.41
Person	20.11	11,943	3.38

Table 4.2: Number of annotations by each Coarser-grained Entity collected from Correia et al. (2022).

a formal standard; they may appear with a legal procedure identification or temporal element like the judgment date or the decision's publication date to identify which decision is being referenced, or even both. The precedent citations can tell the importance and relevance of a given legal procedure to the court (CORREIA et al., 2022). An example of precedent is present in Figure 4.4.

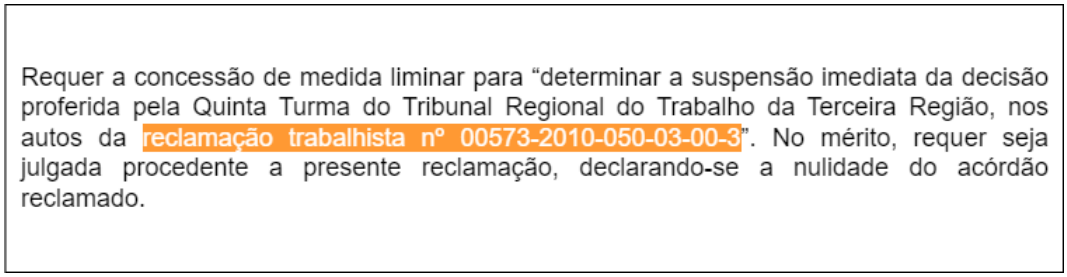


Figure 4.4: Precedent example in orange with the legal procedure class and number information

Table 4.1 presents the overall counting of sentences, tokens, coarser and fine-grained annotations and by each excerpt. The corpus has a variability of excerpt lengths; some excerpts are very small, with only a few tokens, while others reach over 16,087. Moreover, the number of coarser-grained presents in the excerpt also has great amplitude, with a minimum of one and a maximum registered of 267 annotations and an average of 55 annotations.

The number of inner elements (fine-grained elements) linked to each coarser-grained element illustrates the complexity related to the recognition task. Furthermore, Table 4.2 presents each coarser-grained element's total occurrences and its average size (in number of tokens), where we suppose that the larger and rarer the element, the more challenger its recognition will be.

5
Methodology

The proposed process uses a minimal set of manually annotated documents, the Minimal Golden Dataset (MGD), to extract the examples used in the prompt engineering step. Three strategies were developed for selecting these examples based on random selection, clustering selection, which chooses the most representative subset of examples, and similarity selection, based on the similarity distance between input text and examples. The resulting prompt provides task-specific context, including definitions for every named entity and a carefully chosen set of examples illustrating how to perform the annotations.

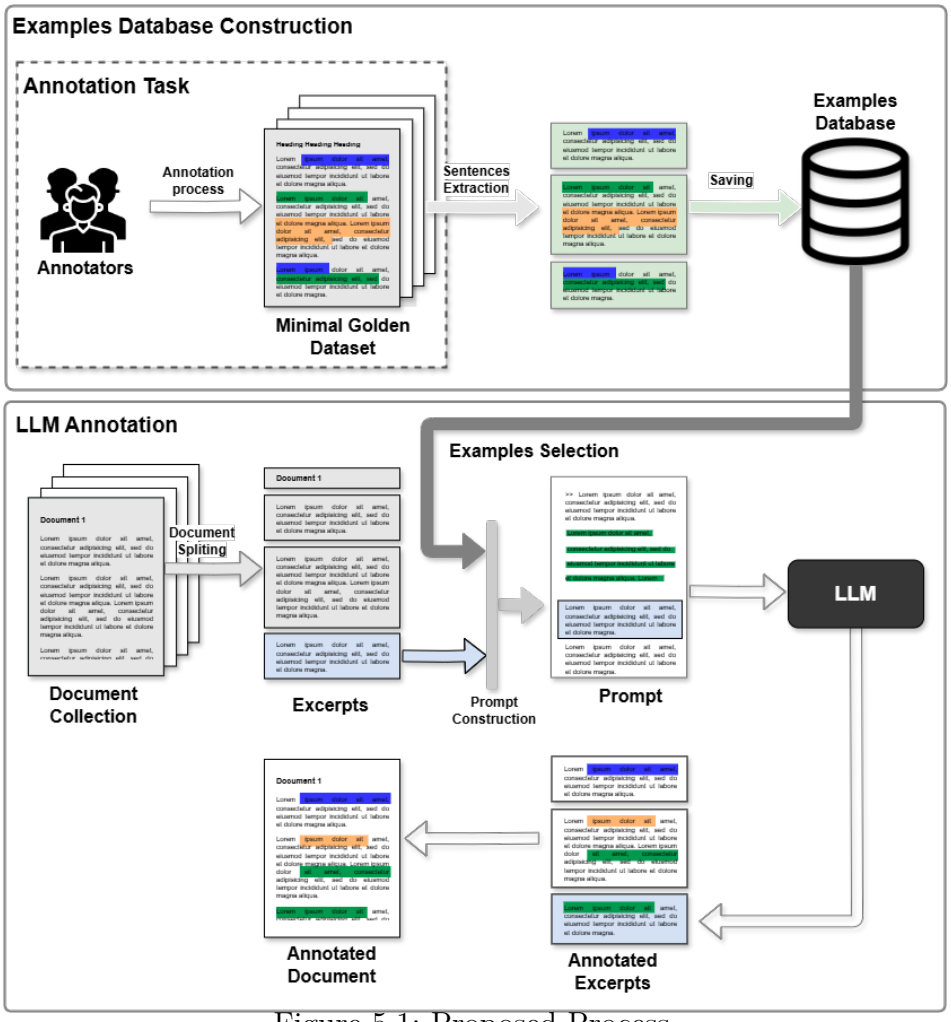


Figure 5.1: Proposed Process.

Figure 5.1 presents an overview of the proposed process, a straightforward process composed of two subprocesses: the first for the examples gathering and the second for the annotation task itself. The first subprocess begins with the annotation task to build the Minimal Golden Dataset. In the following step, we extract sentences from the MGD containing at least one of the defined entities

to compose the Example Database, where these sentences are then encoded and stored.

In the second subprocess, we have as input the collection of documents to be annotated. The first step consists of breaking every document into a set of excerpts. For every excerpt, we selected a set of examples, using one of the three selection strategies stated before, and included them in the prompt provided to the LLM for individual annotation of each entity. Finally, we compile all the responses into a single document, consolidating the annotations for all entities. Suppose two or more annotations of different entities collide in the same tokens. In that case, a heuristic treatment is applied, where each entity has a priority level, to the least to most: Person, Legislative Reference, Precedent, and Academic Citation; as an example, if a person annotation collides with an academic citation, the tokens of academic citation will be selected.

In the following subsections, we present a detailed description regarding the MGD, the construction of the examples database, the prompt description, and the strategies for example selection.

5.1 Minimal Golden Dataset (MGD)

An MGD is required to create the examples database, which provides the context needed to perform the LLM‘s few-shot learning for the annotation generation process. This minimal golden dataset represents an annotated corpus that should contain a few manually annotated documents and include all entities' examples. Since the MGD significantly impacts the examples collection, its quality reflects on the LLM performance. That's why domain specialists must participate in this annotation step.

The size of the MGD, in terms of the number of documents and tokens, may vary according to the annotation task domain. In this study, we have evaluated the impact of the number of examples on the LLMs' performance, and our results (described in Section 6.3) have shown that a few dozen examples are enough for this annotation task. So, due to its small size, the effort, time, and human resources required for its production will be lower.

5.2 Examples Database

The Examples Database is a collection of examples of each entity extracted from the MGD after compiling the annotations made by each annotator and defining the annotation classes. At this point, several steps of

cleaning and selection can be applied to ensure a high-quality set of examples. See the Section 6.1.2 for the details.

In summary, the examples are sentences containing at least one annotation for each entity. Thus, each entity e_k has its own examples set $Se_k = \{s_1, s_2, ..., s_{n-1}, s_n\}$, and the size of Se_k is limited by the number of sentences where the e_k is present, without repetition.

Furthermore, each entity example s_n in Se_k is masked, removing the annotation class information or any special mark, which will used to represent the input text. Thus, each entity e_k also has a masked example set $Me_k = \{m_1, m_2, ..., m_{n-1}, m_n\}$.

Finally, the Examples Database can be more rigorously described as a collection of tuples $\langle e_k, s_{e_n}, m_{e_n} \rangle$ where e_k is the labeled class, s_{e_n} is the n-th example of e_k, and m_{e_n} the respective mask for the n-th example.

5.3 Prompt Construction

The developed prompt described in Figure 5.2, first provides information about the task's context on the top level of the prompt, followed by an explicit description of the entity extracted from the guidelines in (CORREIA et al., 2022), improving the comprehension of the entity. Moreover, it describes how to perform the annotation task, assigning the beginning tag (@@) and the end tag (##) for every annotated entity (WANG et al., 2023).

Furthermore, we also include a few input-output examples demonstrating the task in the prompt. For example, for the excerpt:

"As Súmulas 282 e 356 do STF dispõem respectivamente"

we have the following response:

"As @@Súmulas 282 e 356 do STF## dispõem respectivamente"

The process to select the examples used in the prompt is described in Section 5.4.

Finally, we give the sentence that needs to be annotated and expect the LLM response with the same text and the annotations' special markers addition.

We are analyzing decisions published by Supremo Tribunal Federal
Academic Citations are text references to books, journals, and articles the judge writes on the ruling. An Academic Citation is composed of the following information:
Title - Identifies the paper title
First Author - Identifies the name of the first author
Co-Author - If it exists, identify the name of the other authors
Vehicle - Identifies the means through which the paper was published, such as journals, ISBN codes, etc.
Publisher - The name of the company responsible for literary works, discography, printed media publication, and distribution
Year of Publication - The year in which the paper was published
Your task is to annotate the academic citation entities contained in the text. The response must be a complete copy of the text with the addition of the annotation markers if it exist.

Some examples of the task are below:
Text: [...] a parte deve opor embargos declaratórios . Caso não o faça , não poderá invocar essa questão não apreciada na decisão recorrida (RTJ 56/70 ; v . Súmula 356 do STF e Súmula 211 do STJ ; Nelson Luiz Pinto , Manual dos Recursos Cíveis , Malheiros Editores , 1999 , p . 234 ; Carlos Mário Velloso , Temas de Direito Público , p . 236) .
Response: [...] a parte deve opor embargos declaratórios . Caso não o faça , não poderá invocar essa questão não apreciada na decisão recorrida (RTJ 56/70 ; v . Súmula 356 do STF e Súmula 211 do STJ ; @@Nelson Luiz Pinto , Manual dos Recursos Cíveis , Malheiros Editores , 1999 , p . 234## ; @@Carlos Mário Velloso , Temas de Direito Público , p . 236##) .

Text: [...] interesse público sobre o interesse privado e o princípio da indisponibilidade do interesse público (BANDEIRA DE MELLO , Celso Antônio . Curso de Direito Administrativo , 25ª Edição . São Paulo : Malheiros , 2008 . p . 55) , do que decorre o princípio constitucional da legalidade para a Administração Pública [...]
Response: LLM response

Figure 5.2: Prompt structure for Academic Citation entity annotation. The text in dashed lines is passed as a system message, and the continuous text is passed as a user message. The text marked in red is the task and entity description, in yellow the few-shot, and in blue the input sentence. Finally, in green, the LLM completion.

5.4 Example Selection Strategies

For LLMs few-shot evaluation, we adopted three different ways for example-selection: (1) Randomized, in this strategy, we select k sentences randomly, and each sentence has an equal chance to be selected; (2) Clustering, the sentences are grouped into k groups, where k is equal to the length of the few-shot set. Further, the centroid of each group is selected and used as an example. Since the centroids had to be actual sentences, we used the *K-Medoids* model to create the clusters; (3) Similarity, k sentences are selected based on cosine similarity with the input (LIU et al., 2021).

To increase the retrieval results for both (1) and (2) example strategy selection, the sentences were embedded using a task-related encoder (LIU et al., 2021), in this case, the Legal-BERTimbau[1] model, a BERT model fine-tuned with over 30,000 Portuguese legal documents available online.

[1]https://huggingface.co/rufimelo/Legal-BERTimbau-base

6
Experiments

This chapter describes the experiments executed to evaluate the LLMs in the proposed process and presents a detailed analysis of the results. This chapter is organized as follows: Section 6.1 presents the data, LLMs, metrics, and other configurations used in the evaluation. Section 6.2 describes the experiment conducted to estimate the best configuration of the number of examples and selection strategy for each LLM and discusses the results. Finally, Section 6.3 presents the results of the best model on the prior experiment on large data.

6.1
Experimental Setup

A total of six LLMs were selected, open- and closed-source. We tested the Gemini 1.5 Pro variant model and 1.5 Flash through the Gemini API[1], GPT-4 O mini from OpenAI[2], DeepSeek Chat V2 using DeepSeek API[3], Llama 3.1 405B and Llama 3.1 70B through DeepInfra platform[4], the pricing for each LLM is presented in the Table 6.1.

Because of the LLMs' cost and the combinatory number of examples, selection strategies, and LLMs, we needed to reduce the number of documents used for the evaluation. To do so, we performed a stratified selection of approximately 10% of the dataset, totaling 58 documents, to create the validation set with five documents and the test set with 53 documents. Table

[1]https://ai.google.dev/gemini-api
[2]https://platform.openai.com/
[3]https://platform.deepseek.com/
[4]https://deepinfra.com/

Table 6.1: LLMs pricing in US dollars per one million tokens. Accessed on July 25, 2024

Model	Input Price	Output Price
DeepSeek V2	$ 0,14	$ 0,28
Gemini 1.5 Pro	$ 3,50	$ 10,50
Gemini 1.5 Flash	$ 0,35	$ 1,05
GPT-4o mini	$ 0,50	$ 1,50
Llama 3.1 405B	$ 2,70	$ 2,70
Llama 3.1 70B	$ 0,52	$ 0,75

6.2 presents the number of annotations and tokens for each entity on the validation set and for the test set.

Table 6.2: Number of annotations and tokens for each named entity on the validation set

Entity	Validation Set		Test Set	
	Annotations	Tokens	Annotations	Tokens
Person	144	505	916	3,137
Legislative Ref.	95	823	824	6,867
Precedent	80	1,013	693	7,410
Academic Cit.	18	532	152	3,897
	337	2,873	2,585	21,311

For all experiments, we set the temperature parameter to zero, which alters the level of randomness of the generation, providing a more constant output, for reproducibility purposes, and as suggested by Renze and Guven (2024), Aldeen et al. (2023), changes in temperature from 0.0 to 1.0 do not have a statistically significant impact on LLM's performances, so we discard changes in this parameter.

Moreover, we split the decisions from both validation and test sets by sentences with a minimum length of 2000 tokens.

Since our purpose is to provide a reliable evaluation of LLM's performance in the legal named entity annotation task, we used the annotations on the validation and test set as ground truth and measured the performances of LLMs in an exact-match and relaxed-match way (LI et al., 2023b):

- **Exact-match**: the LLM annotations boundaries and type must match the golden annotation.
- **Relaxed-Match**: the LLM annotations must assign the correct entity and overlap the ground truth, regardless of its boundaries.

allowing us to calculate the precision, recall, and F1-score metrics.

6.1.1 MGD Setup

Based on (CORREIA et al., 2022) work, we got the annotations obtained during the two shorter training sessions of the annotators in the annotation activity and used them as MGD[5]. Thus, the MGD contains a total of ten

[5]The annotation data built in these training sessions were kindly provided and anonymized by the authors of the referenced work (CORREIA et al., 2022).

annotated documents. During these shorter trainings, all annotators had the same set of decision excerpts. Meanwhile, in the final annotation session, every excerpt was annotated on average by 5.4 students. So, by selecting only documents from the training session instead of the other documents from the corpus, we reduced the chances of contaminating the examples database, as the LLMs would only have access to annotations produced before the long final annotation effort, making our results more reliable.

6.1.2 Examples Database Setup

From the MGD, we calculated the number of annotations for each annotator, excluding the annotators with none or very few annotations, and then the annotation's percentage of votes for the most voted class. Figure 6.1 presents the distribution of the annotations' percentage of votes for each entity after the annotator's removal.

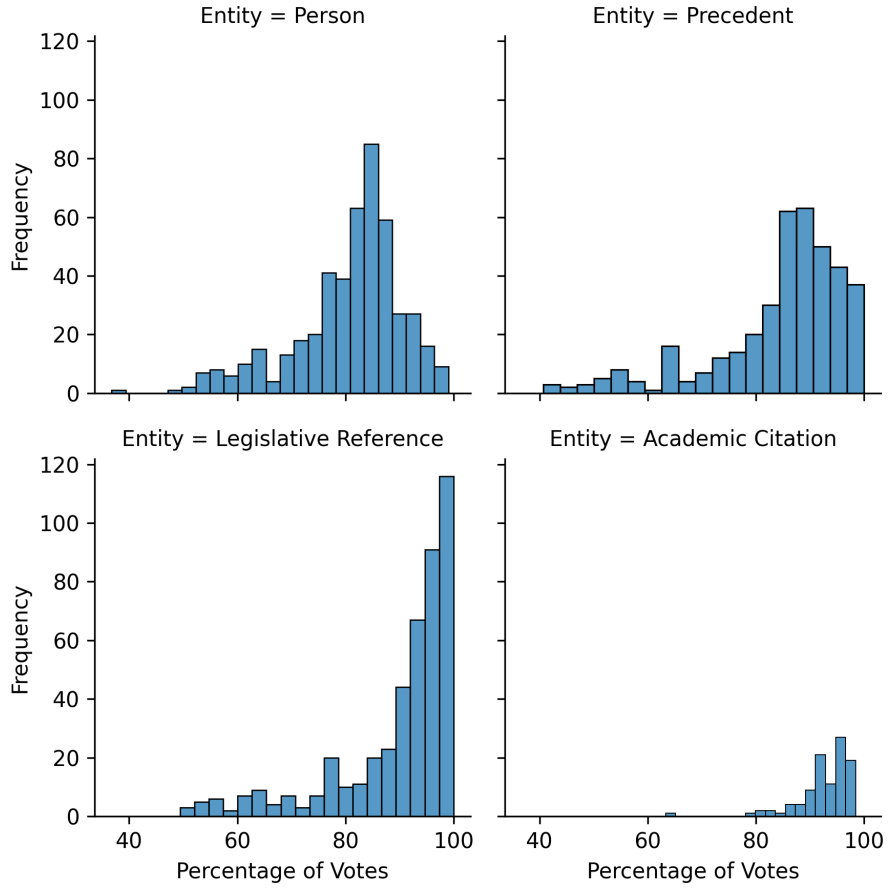


Figure 6.1: Pertange of votes for each annotation per entity.

After defining the annotations classes, we extracted the sentences containing at least one annotation for each entity. Figure 6.2 shows the distribution of the number of tokens in the sentences collected for each entity. The sentences

containing academic citations tend to be more extensive regarding other entities. This is related to the length of academic citation, with more tokens on average, as present in Table 4.2. The sentences' length median for the rest of the entities is below 100 tokens, but sometimes the number of tokens reaches over 700.

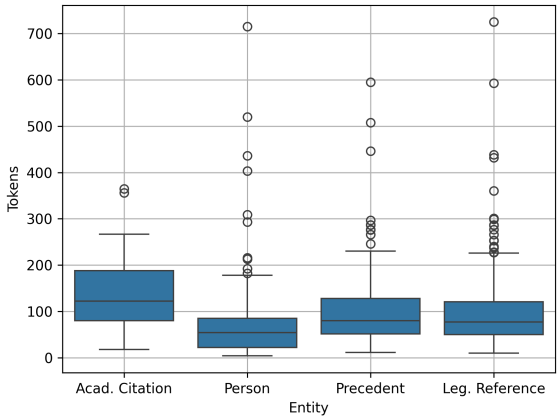


Figure 6.2: Number of tokens in the sentence per entity.

Although these more extensive sentences may appear commonly on a ruling, including them in the examples database will greatly increase the context sent to the LLMs, sometimes leading to an unpractical and unaffordable price. Thus, to turn around this problem and uniform the length of the sentences for both four entities, we apply a filter in the academic citations sentences by median and the persons, precedents, and legislative references on the third percentile. The resulting distributions are present in the Figure 6.3.

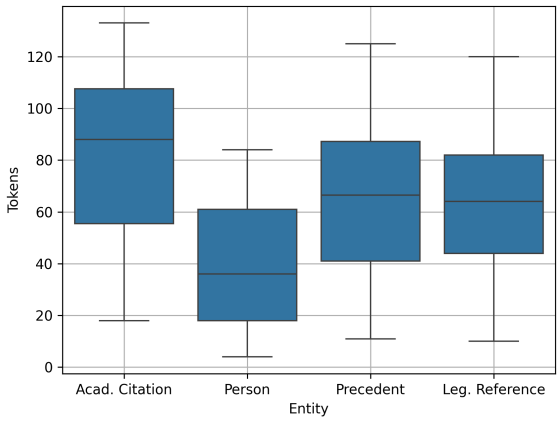


Figure 6.3: Number of tokens in the sentence per entity after filtering process.

Some observations regarding the annotations on MGD are as follows: some entities are easier to identify and distinguish from others. Most Academic Citation annotations receive over 80% of the annotator's vote — even so, there's no annotation with 100% of votes. The percentage of votes for the

others is more spread. For the Person entity, some annotations received less than 40% of the votes, showing that most of the annotators did not recognize or distinguish it from the rest of the entities.

Moreover, the precedent annotations were mostly wrongly labeled as a person, followed by legislative references and academic citations. In legislative reference annotations, the precedent was the most common entity wrongly assigned, in second the person. The person appears again as the most wrongly assigned to the academic citation. See the Appendix 9.2 for more details.

Therefore, even if the Person entity was created for disambiguation as presented in Section 4.3, this entity often led to misinterpretation and showed difficulty annotating. Consequently, it potentially will also be a challenge to the annotation process using LLMs.

6.2 Tuning Experiment

We conducted an exhaustive search on the validation set to find the best configuration for the number of examples and selection method. As stated before, we used three selection methods: random, similarity, and clustering, and changed the number of examples between 4, 8, 16, and 32. This analysis was applied to each of the six selected models. Additionally, each combination was executed five times with different seeds, and the same examples were provided to all models. Figure 6.4 provides an overview of the models' performances.

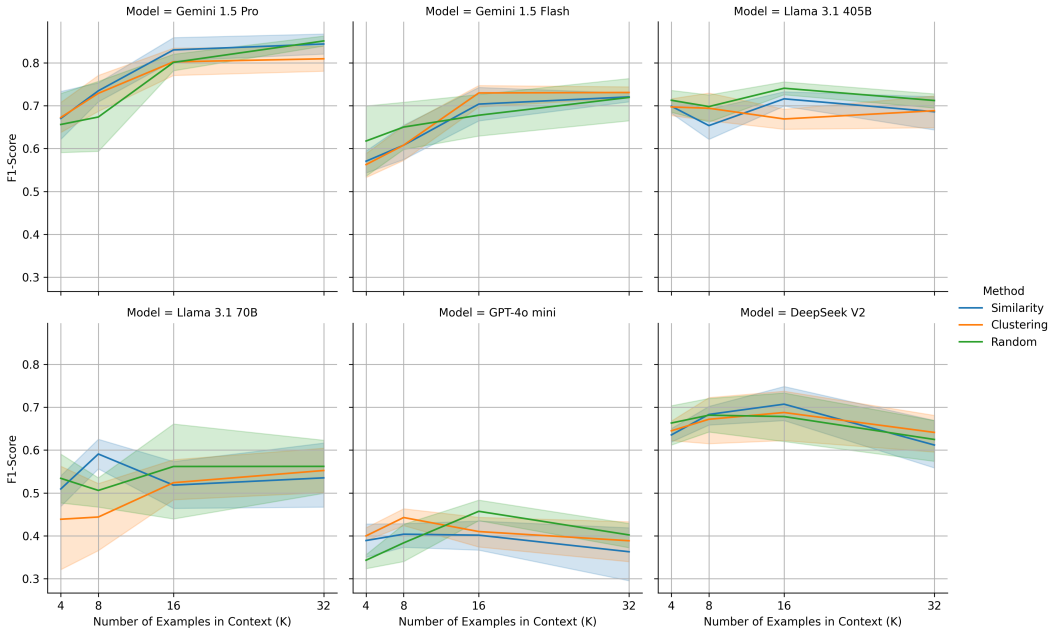


Figure 6.4: LLMs' average performances on validation set with strict-match.

It was observed that the Gemini 1.5 Pro achieved the best result among

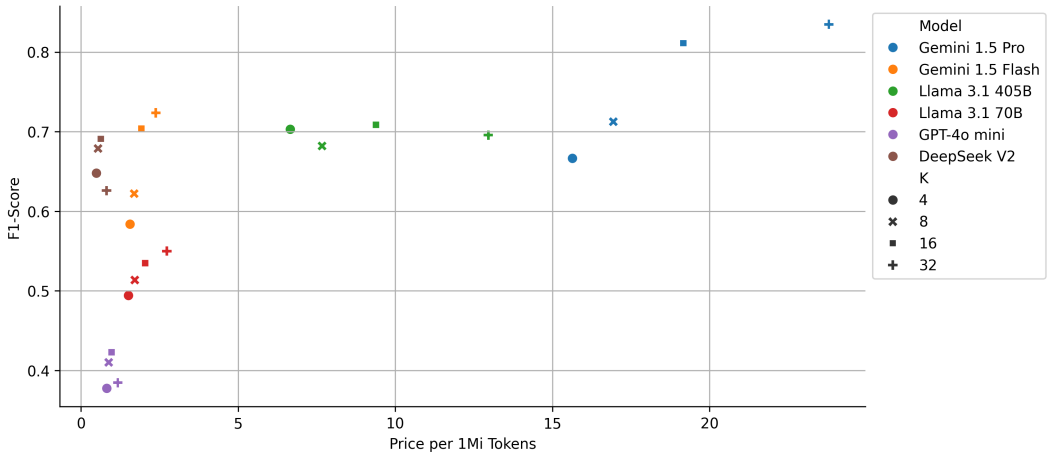


Figure 6.5: LLMs' Performance Cost-benefit on the validation set with strict match

all LLMs, followed by its smaller version, Gemini 1.5 Flash, Llama 3.1 405B, and DeepSeek V2. Lastly, the Llama 3.1 70B and GPT-4o mini struggled to annotate a significant portion of the entities. We also calculate the cost-benefit ratio of each model by the number of examples in the context. To each K, we find the ratio between the response and the prompt, which includes the entity description, examples, and input text information. Finally, we calculate the costs for processing 1 million tokens for each LLM using the pricing presented in Section 6.1. As a result, alongside DeepSeek V2, the Gemini 1.5 Flash offers the best cost-benefit ratio with 16 examples, delivering quality annotations at a lower price, as depicted in Figure 6.5.

Overall, there were no significant differences between the developed selection methods, indicating that LLMs can generalize even when using random examples (see Appendix 9.4 for more details). Regarding the number of examples, both the Gemini Pro and Flash improved their annotations with an increased number of examples. At the same time, no significant differences were observed in the Llama 3.1 405B, 70B, and GPT-4o mini. However, DeepSeek V2's performance worsened with the increased context of using 32 examples compared to 16.

Based on the significance testing and cost-benefit analysis, the optimal selection strategy for all models is random selection, as no significant differences were found among the three strategies; thus, it is not required to implement similar-retrieval-based strategies. Moreover, for Gemini 1.5 Pro and Gemini 1.5 Flash, 16 examples are the outstanding number, as they provide similar performance to 32 examples but at a lower cost. For Llama 3.1 405B, Llama 3.1 70B, and GPT-4o Mini, four examples are sufficient; there were no significant differences in performance, and it is way cheaper. Finally, DeepSeek performed

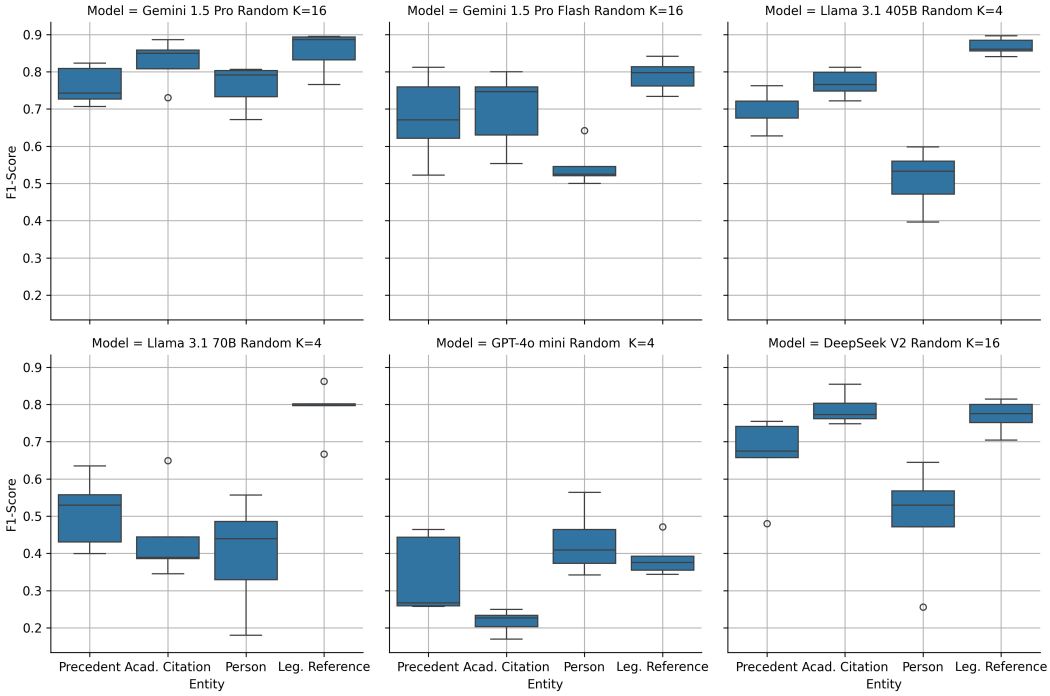


Figure 6.6: LLMs' average performances with strict-match on the validation set for each entity using the best configuration of the number of examples and selection strategy.

better with 16 examples. Table 6.3 summarizes these conclusions.

Table 6.3: Best configuration for each model considering significance tests and cost-benefit analysis

Model	Selection Strategy	Number of Examples
Gemini 1.5 Pro	Random	16
Gemini 1.5 Flash	Random	16
Llama 405B	Random	4
Llama 70B	Random	4
GPT-4o mini	Random	4
DeepSeek V2	Random	16

Figure 6.6 shows the individual analysis for each entity and model using the best configuration. The Gemini 1.5 Pro once again achieved the best result, with over 0.70 F1-Score for all entities, including the rarer ones like Academic Citation. The other models achieved results close to the Gemini 1.5 Pro in the individual analysis, except for the Person entity, which the other five models struggled with. For the entity Precedent, Legislative References, and Academic Citations, the Llama 3.1 405B, DeepSeek V2, and Gemini 1.5 Flash models exchanged positions for best performance. Thus, it is possible to develop a

multi-LLM approach and target the models that performed best for each entity, improving costs and time for generating annotations.

6.2.1 Gemini 1.5 Pro

The Gemini 1.5 Pro achieves the best performance of all models and presents an improvement with increasing examples in the context. With 16 or more examples, the model registered a remarkable score of over 0.80 F1 among all selection strategies. However, the model reached 0.65 of F1 when only four examples were used, as described in Figure 6.9.

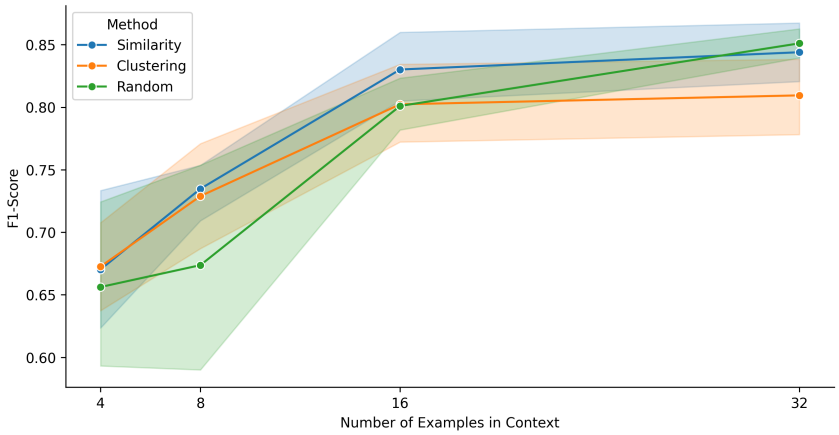
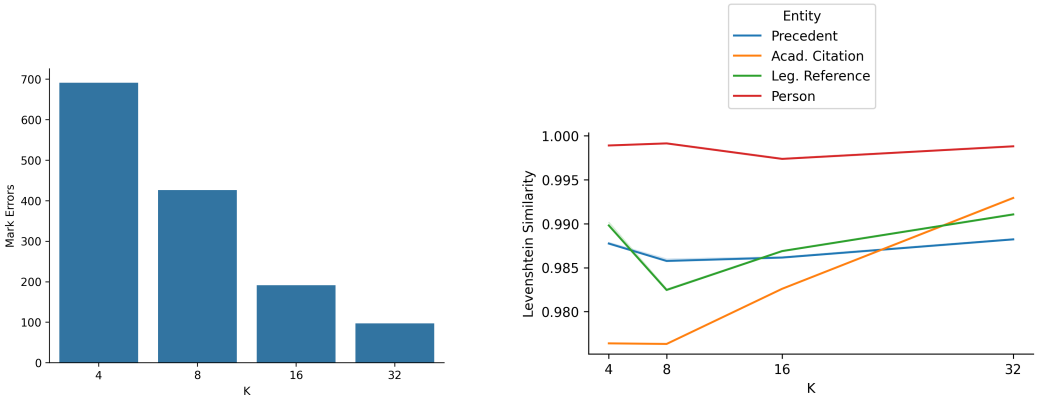


Figure 6.7: Gemini 1.5 Pro performance on validation set with strict-match.

The performance for each entity was balanced, and the model got a high-quality annotation for all four entities. The entity where the model scored most was Legislative Reference, with a median close to 0.90. It was followed by Academic Citation, which, due to its length, is more challenging and obtained a median of approximately 0.85 F1; the third was the Person, and lastly, the Precedent, as seen in Figure 6.6.



(a) Number of marker errors by number of examples in context

(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.8: Gemini 1.5 Pro generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

Regarding the generation capabilities, as shown in Figure 6.8, the model generated a sequence mostly identical to the input, despite the special marks, across all numbers of K. However, with only four examples, the model could not complete 700 annotations, which matches the performance observed in Figure 6.7, and each time when the number of examples increases, the model generates fewer marker errors.

6.2.2 Gemini 1.5 Flash

The Gemini 1.5 Flash, a smaller variant of the 1.5 Pro, also performed well, achieving over 0.70 F1-Score. The model also demonstrated similar improvement as observed in the Gemini 1.5 Pro, with the number of examples increasing, as depicted in Figure 6.9.

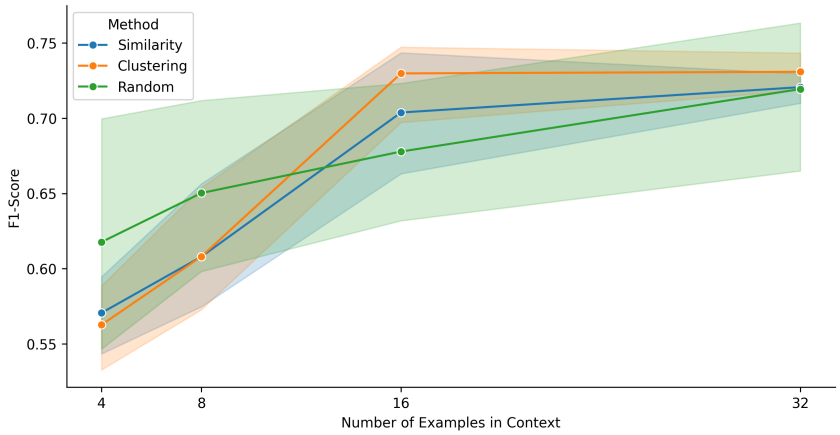
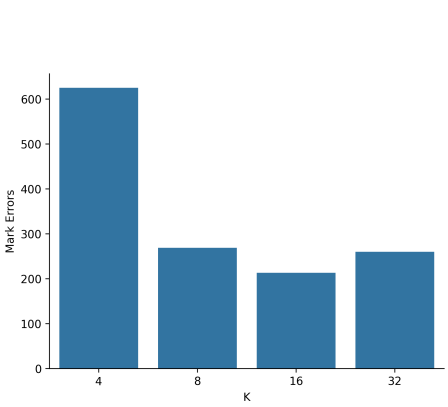
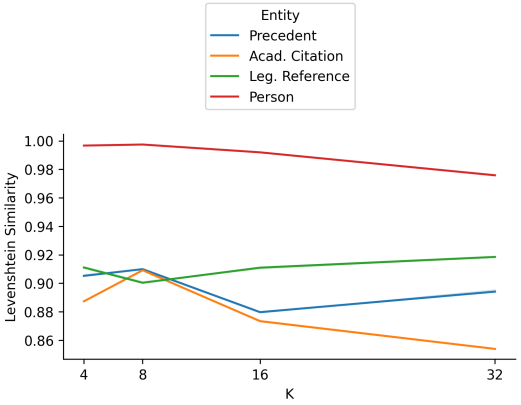


Figure 6.9: Gemini 1.5 Flash performance on validation set with strict-match.

The best performance was for the Legislative Reference entity, followed by Academic Citation, Precedent, and lastly, the Person entity with a median below 0.6, which is compatible with the difficulty to distinguish and recognize associated as we saw in Figure 6.6.



(a) Number of marker errors by number of examples in context



(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.10: Gemini 1.5 Flash generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

Moreover, as depicted in Figure 6.10, the model also generated more marker errors when submitted to fewer examples, and with the increase, the errors were minimized. Also, the model generates more distant responses than the Gemini 1.5 Pro, which could be associated with generation interruptions when the model does not finish the sequences or extra information. This distance is greater for the academic citation annotations.

6.2.3 Llama 3.1 405B

The Llama 3.1 405B model performed stable across all combinations of the number of examples and the selection strategy, with no significant differences, scoring around 0.70 of F1 as shown by Figure 6.11. Demonstrating a strong performance even with fewer examples demonstrating the task.

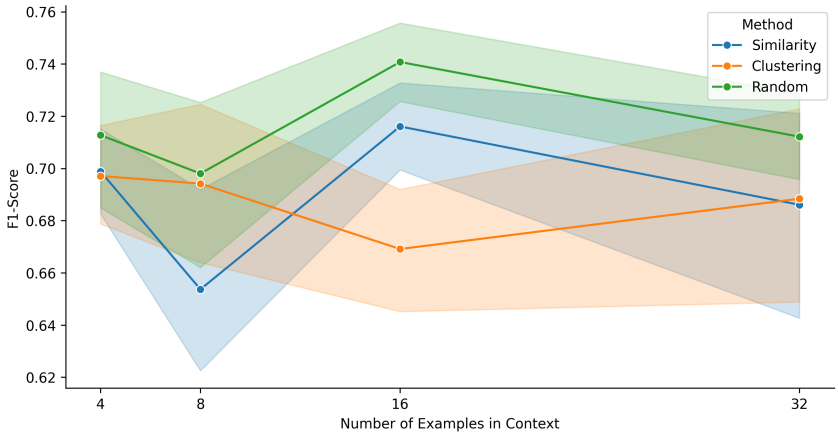
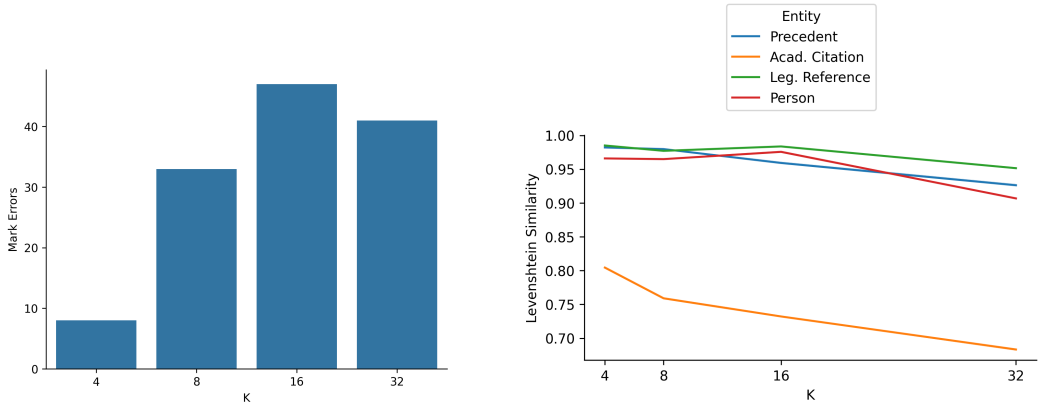


Figure 6.11: Llama 3.1 405B performance on validation set with strict-match.

Furthermore, the Llama 3.1 405B model presents minimal marker errors and generates sequences that closely align with the input for Precedent, Legislative Reference, and Person entities (as presented in Figure 6.12). For academic citation, the model exhibits the same behavior observed for the Gemini 1.5 Flash, indicating that adhering to the instruction to respond with the same text and add special marks proves more challenging for this entity.

However, despite these large differences, the model shows remarkable performance for academic citation, all above 0.70 of F1 score, using four examples and strategy random, as shown in Figure 6.6. The model also achieves the best performance for legislative reference annotation and the worst for the person entity.

Although the Gemini 1.5 Pro scored the highest performance, the Llama 3.1 405B can offer more reliability in the results because it is an open-source model, as cited in (CHEN; ZAHARIA; ZOU, 2024) the GPT-3.5 and GPT-4 capabilities to follow user instructions worsened over time, occasioning many behavior drifts. These results may be related to other closed-source models, making the Llama 3.1 405B the most trustworthy model.



(a) Number of marker errors by number of examples in context

(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.12: Llama 405B generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

6.2.4 Llama 3.1 70B

The Llama 3.1 70B obtained a moderate performance, close to 0.55 of F1, as shown by Figure 6.13. The results in Brown et al. (2020) that scaling up the model size significantly impacts the model's performance and larger models that are more proficient at in-context learning are reinforced by the observed results. The Llama 3.1 405B was significantly better than Llama 3.1 70B, similar to Gemini 1.5 Pro and Gemini 1.5 Flash.

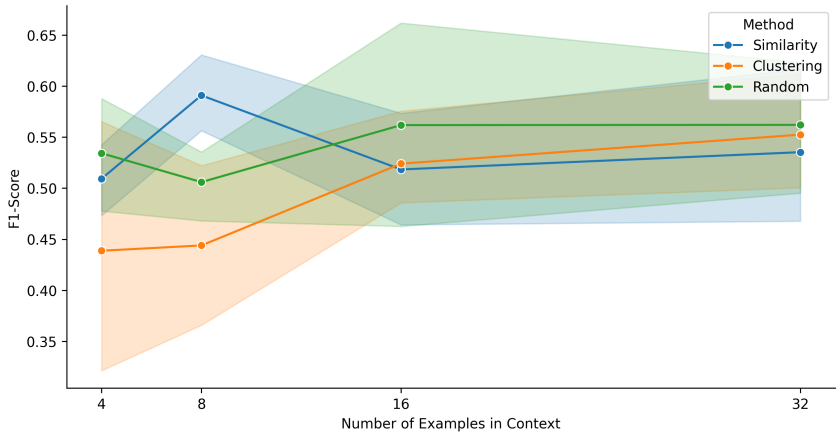
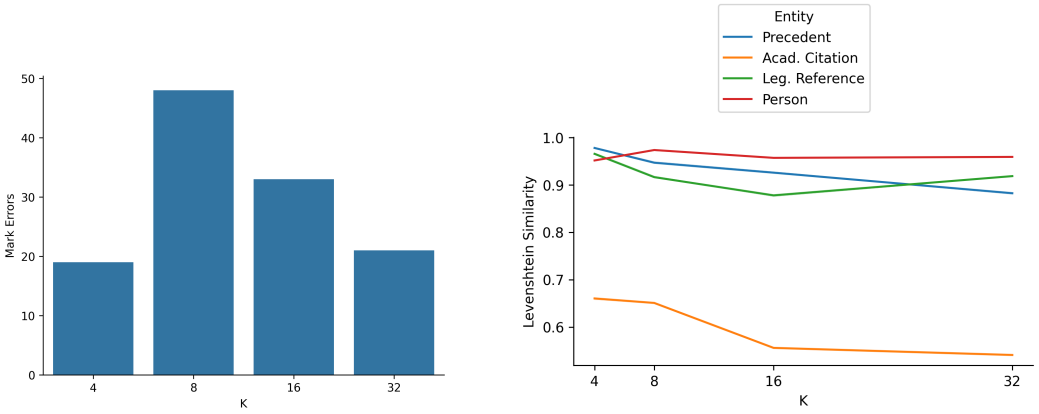


Figure 6.13: Llama 3.1 70B performance on validation set with strict-match.

Moreover, the model demonstrates similar capabilities as Llama 405B in following the annotation instructions and does not produce many marker errors. However, the model presented even more severe deviations for text

generation for academic citation than Llama 405B and Gemini 1.5 Flash, as described in Figure 6.14, which may have interfered negatively, as observed in the individual entity analysis (Figure 6.6).



(a) Number of marker errors by number of examples in context

(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.14: Llama 70B generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

Besides, with only four examples, the Llama 3.1 70B demonstrated strong legislative reference annotation capabilities, even more than larger models such as DeepSeek V2, and in summary, performed better than the GPT-4o mini.

6.2.5 GPT-4o mini

The GPT-4o mini obtained the worst performance among all models. It could not correctly identify and annotate most entities present using the portion and examples selected across the three strategies, not surpassing the 0.50 F1-Score mark. The LLM seems to follow the instructions correctly, generating coherent results. Still, with serious flaws in adding the annotations' special markers, reaching more than 350 marker errors even with 32 examples (Figure 6.16), and adding up the values, it was the model that generated the most marking errors, which matches the results observed in Figure 6.15.

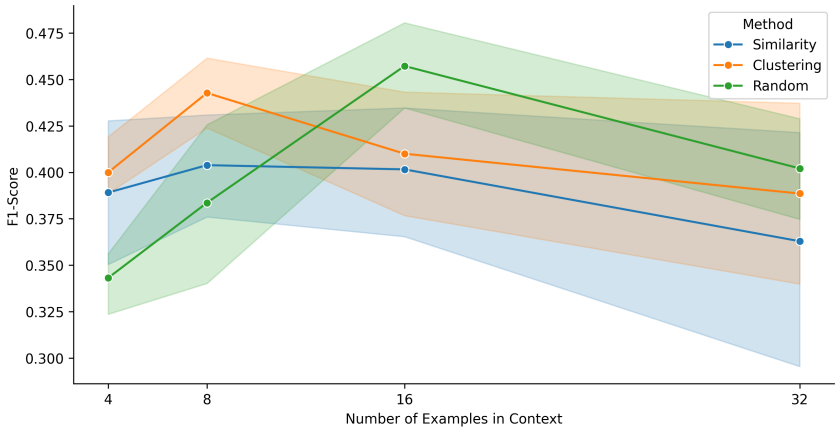
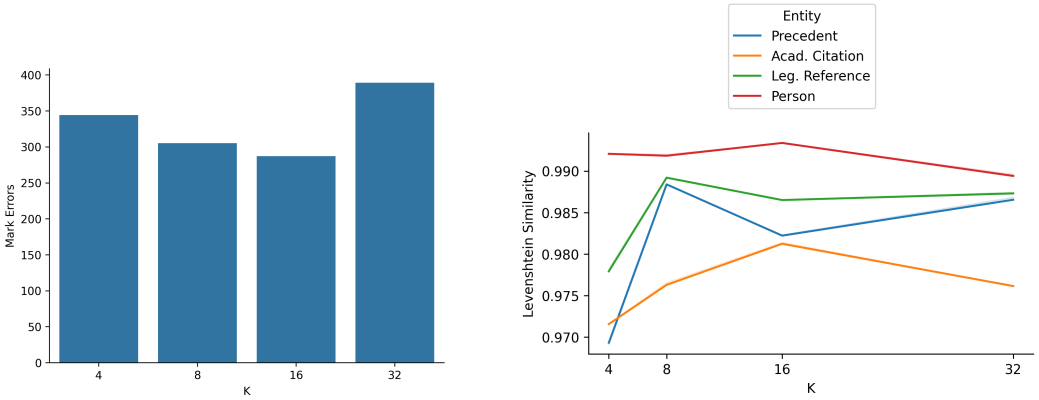


Figure 6.15: GPT-4o mini performance on validation set with strict-match.

Based on our prior results and in Aldeen et al. (2023), we can presumably see that GPT-4o will perform better than GPT-o, given that it is a larger version. However, we cannot precisely estimate the extent of the improvement. Still, due to GPT-4o mini results, the GPT-4o may be slightly below the Gemini 1.5 Pro, which is the direct concurrent among the larger models.



(a) Number of marker errors by number of examples in context

(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.16: GPT-4o mini generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

6.2.6 DeepSeek V2

The DeepSeek V2 model also achieved remarkable performance, as presented in Figure 6.17, close to or exceeding 0.70 of F1. It fulfilled the instructions of the annotation special marks, was the model with fewer marker

errors, only six errors including all five executions using four examples, and generated mostly identical sequences to input, similar to Gemini 1.5 Pro, depicted in Figure 6.18. Also, the LLM is one of the most cost-benefit models, with only $ 0,14 per million tokens, the lowest price among all examined models. This price is mostly due to the economical MoE architecture developed in DeepSeek V2, which can save 42% of training costs.

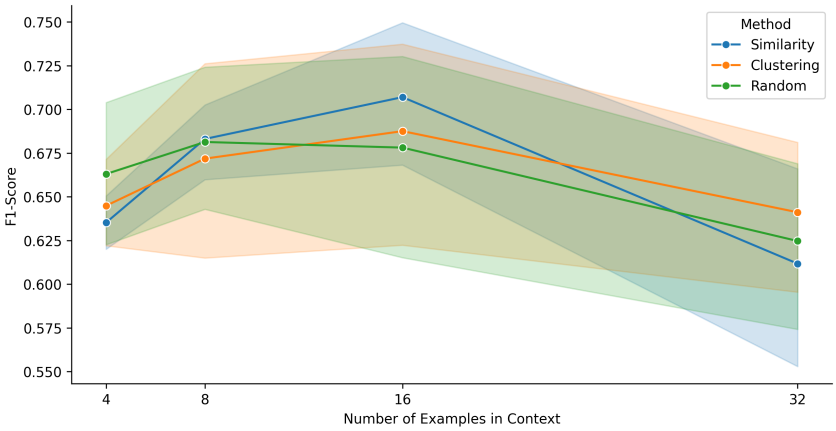
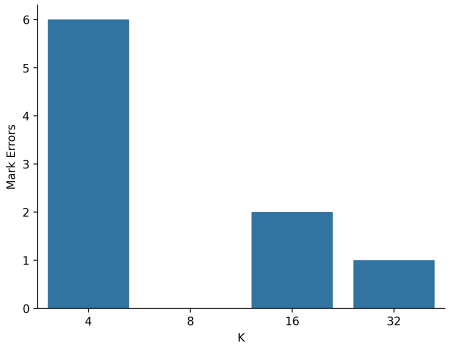
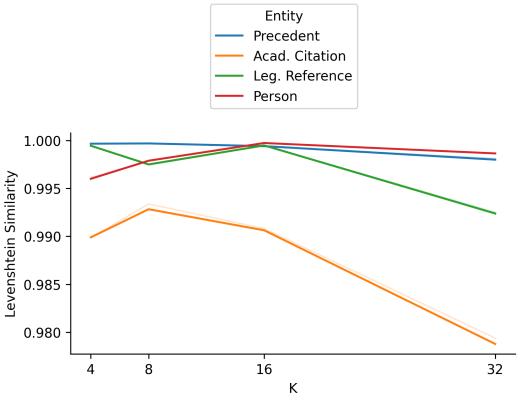


Figure 6.17: DeepSeek V2 performance on validation set with strict-match.



(a) Number of marker errors by number of examples in context



(b) Distance between the sequence generated and the input by the number of examples in context

Figure 6.18: DeepSeek V2 generation capabilities. (a) denotes the number of errors in the generation of special marks, @@ without closing and ## without opening up with @@. (b) shows the similarity of the response generated with the input using Levenshtein Similarity.

Besides, Gemini 1.5 Pro and Flash improved by constantly increasing the number of samples, while the DeepSeek V2 performed worse with 32 examples compared to 16. The additional examples may act as a noise, or the information vanishes with the context increase, which oddly contradicts the results reported

by DeepSeek-AI et al. (2024) in the Haystack experiment (Figure 6.19), which measures the models' capabilities to retrieve information at any in-context depth and context length.

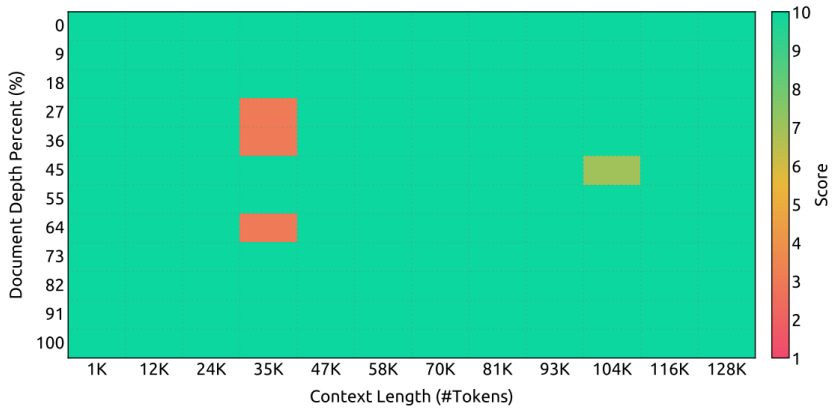


Figure 6.19: Evaluation results on the "Needle In A Haystack" (NIAH) tests for DeepSeek V2 extracted from (DEEPSEEK-AI et al., 2024)

6.3 Benchmarking of Optimal Model

To validate the experiment's results in the validation set, the Gemini 1.5 Pro was executed five times, with different seeds, on the test set using 16 examples and the random selection strategy. The results obtained with strict-match and for relaxed-match are shown in Table 6.4.

With strict-match, the Gemini 1.5 Pro achieved an average F1-Score of 0.66, and with relaxed-match 0.76, a significant improvement of 10% on average.

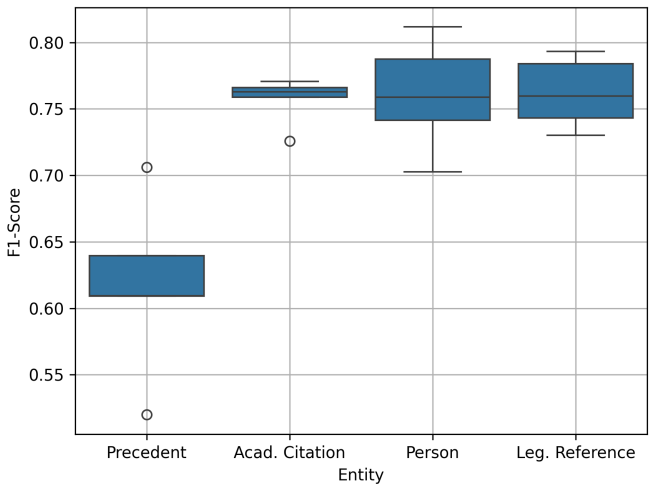


Figure 6.20: Gemini 1.5 Pro performance on evaluation experiment for each entity with relaxed-match

Moreover, the annotation of Precedent entities was significantly worse. As shown in Figure 6.20, the median of all the other entities is above 0.75, while the Precedent's median is close to 0.60. This result is related to the lack of a formal standard of precedent references in the STF.

However, the model performed well overall in the largest set, even with the increased variability of the entities.

Table 6.4: Gemini 1.5 Pro results on Test set with strict- and relaxed-match

It.		Strict-Match			Relaxed-Match		
		Precision	Recall	F1-Score	Precision	Recall	F1-Score
1	Acad. Citation	0.69	0.75	0.71	0.72	0.83	0.77
	Leg. Reference	0.77	0.70	0.73	0.82	0.75	0.78
	Person	0.6	0.59	0.60	0.71	0.69	0.70
	Precedent	0.73	0.63	0.68	0.76	0.66	0.70
		0.7	0.67	0.68	0.75	0.73	0.73
2	Acad. Citation	0.63	0.71	0.67	0.66	0.81	0.72
	Leg. Reference	0.76	0.73	0.74	0.81	0.78	0.79
	Person	0.67	0.64	0.66	0.76	0.73	0.74
	Precedent	0.69	0.54	0.61	0.72	0.58	0.64
		0.69	0.65	0.67	0.74	0.72	0.72
3	Acad. Citation	0.68	0.71	0.69	0.73	0.80	0.76
	Leg. Reference	0.69	0.65	0.67	0.75	0.71	0.73
	Person	0.67	0.64	0.66	0.78	0.74	0.76
	Precedent	0.57	0.48	0.52	0.58	0.47	0.52
		0.65	0.62	0.63	0.71	0.68	0.69
4	Acad. Citation	0.69	0.69	0.69	0.75	0.77	0.76
	Leg. Reference	0.71	0.66	0.69	0.77	0.71	0.74
	Person	0.7	0.66	0.68	0.82	0.76	0.79
	Precedent	0.62	0.56	0.59	0.64	0.58	0.61
		0.68	0.64	0.66	0.74	0.7	0.72
5	Acad. Citation	0.69	0.74	0.71	0.72	0.82	0.77
	Leg. Reference	0.71	0.69	0.7	0.77	0.74	0.76
	Person	0.72	0.69	0.7	0.83	0.79	0.81
	Precedent	0.59	0.58	0.59	0.59	0.63	0.61
		0.68	0.67	0.67	0.73	0.74	0.74

6.3.1 Multi-LLM Approach

Based on the results observed so far, we have also employed a cost-benefit multi-LLM approach for entity annotation. In this approach, we mix up the models Llama 405B for legislative reference annotation using four examples, DeepSeekV2 for academic citation using 16 examples, and Gemini 1.5 Flash for precedent annotation, following the individual analysis on Figure 6.6. For all models, the examples were selected using the random strategy.

Gemini 1.5 Flash's performance in Precedent annotation was close to Llama 405B, so we decided to keep it for diversity. Furthermore, as none of the three models performed well in annotating the person entity, we used the Gemini 1.5 Pro results.

The strict- and relaxed-match results are in Table 6.5. The approach using multiple LLMs yielded good results, surpassing the approach using only Gemini 1.5 Pro in both strict- and relaxed-match, achieving F1 scores of 0.69 and 0.75, respectively. Llama 3.1 also performed well, similar to Gemini 1.5 Pro, with a median F1 score of 0.75, as shown in Figure 6.21, which is impressive considering only four examples were used. Additionally, DeepSeek V2 also demonstrated good performance on the test set, similar to what was observed on the validation set, with an average F1 score of 0.74 in the relaxed-match, reaching an F1 score of 0.81. Gemini 1.5 Flash exceeded its validation set performance and was also better than Gemini 1.5 Pro, showing a high capacity to generalize using the selected examples.

Table 6.5: Multi-LLM Approach results on Test set with strict- and relaxed-match

		Strict-Match			Relaxed-Match		
It.		Precision	Recall	F1-Score	Precision	Recall	F1-Score
1	Acad. Citation	0.65	0.78	0.71	0.65	0.86	0.74
	Leg. Reference	0.73	0.60	0.65	0.77	0.70	0.73
	Person	0.61	0.60	0.60	0.71	0.70	0.70
	Precedent	0.81	0.67	0.73	0.84	0.68	0.75
		0.69	0.63	0.68	0.74	0.74	0.73
2	Acad. Citation	0.70	0.72	0.71	0.80	0.82	0.81
	Leg. Reference	0.82	0.60	0.69	0.90	0.66	0.76
	Person	0.72	0.70	0.71	0.82	0.78	0.80
	Precedent	0.75	0.66	0.70	0.75	0.70	0.73
		0.74	0.68	0.70	0.82	0.74	0.77
3	Acad. Citation	0.63	0.71	0.66	0.66	0.81	0.73
	Leg. Reference	0.76	0.66	0.70	0.79	0.72	0.75
	Person	0.66	0.64	0.65	0.77	0.73	0.75
	Precedent	0.76	0.63	0.69	0.79	0.61	0.69
		0.71	0.65	0.67	0.75	0.72	0.73
4	Acad. Citation	0.62	0.74	0.67	0.62	0.84	0.71
	Leg. Reference	0.75	0.59	0.66	0.77	0.67	0.72
	Person	0.70	0.65	0.67	0.81	0.74	0.77
	Precedent	0.78	0.64	0.70	0.83	0.67	0.74
		0.72	0.65	0.67	0.76	0.73	0.74
5	Acad. Citation	0.64	0.72	0.67	0.67	0.83	0.74
	Leg. Reference	0.82	0.63	0.71	0.91	0.69	0.78
	Person	0.73	0.69	0.71	0.84	0.80	0.82
	Precedent	0.78	0.69	0.73	0.82	0.74	0.78
		0.75	0.69	0.71	0.81	0.76	0.78

The differences observed in person entity annotation are due to the treatment of heuristic collisions: with fewer collisions occurring, more person annotations were preserved. These results highlight the importance of analyzing and utilizing different LLMs to build cost-effective solutions, targeting the best models for each entity. Additionally, it demonstrates that the process developed for annotating legal entities supports the addition of one or more LLMs during the annotation phase.

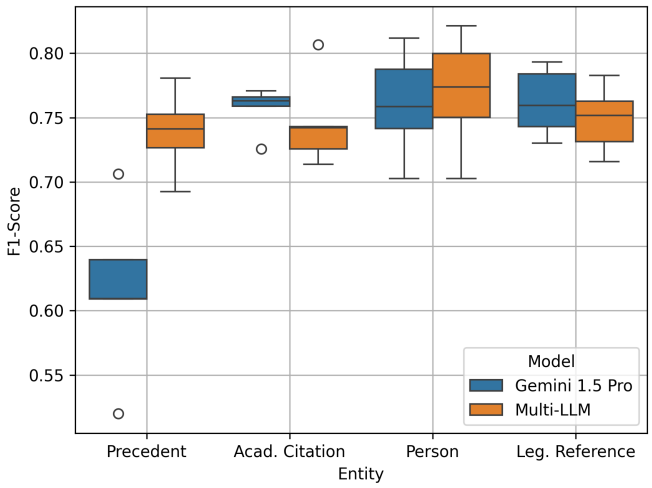


Figure 6.21: Multi-LLM approach performance on evaluation experiment for each entity with relaxed-match

6.3.2 Concluding Remarks

Performing a few-shot NER in such a real-world problem is challenging and requires strong generalization abilities from the models. Moreover, the data on which the models were submitted in training has a substantial impact on the recognition, as observed, different models were better in different entities. Although, in summary, the larger models were better at recognizing and generating the labeled sequences than the smaller models tested, we can stand this and conclude that lightweight models with less than 70B parameters could not be able to annotate most of the entities.

Furthermore, there are no significant differences between the three developed example selection strategies. In a limited-source scenario, the examples could be selected randomly and do not require semantically similar retrieval solutions. Moreover, regarding the number of examples included in the prompt, some models improved by constantly increasing the number of samples, while others worsened or did not present any differences. Thus, the impact of a number of examples depends on each LLM architecture or the pre-trained data.

Therefore, Table 6.6 shows the relaxed-match results on the test set of Gemini 1.5 Pro, which was the best model on the tunning experiment, and the Multi-LLM approach. The two approaches obtained similar F1 scores, 0.76 for Gemini 1.5 Pro and 0.75 for Multi-LLM. The Gemini 1.5 Pro annotated more tokens, while the Multi-LLM was more accurate. Both solutions obtained a remarkable performance on the academic citation, legislative reference, and person annotations, all above 0.73 of F1. However, the Gemini 1.5 achieves

Table 6.6: Gemini 1.5 Pro and Multi-LLM approach median performance on the test set.

Entity	Gemini 1.5 Pro			Multi-LLM		
	Precision	Recall	F1-Score	Precision	Recall	F1
Academic Cit.	**0,72**	0,81	**0,76**	0,66	**0,83**	0,74
Legislative Ref.	0,77	**0,75**	**0,76**	**0,79**	0,69	0,75
Person	0,78	**0,74**	0,76	**0,81**	**0,74**	**0,77**
Precedent	0,64	0,58	0,61	**0,82**	**0,68**	**0,74**
	0,75	**0,74**	**0,76**	**0,80**	0,72	0,75

moderate performance for precedent, not generalizing as well as the Multi-LLM solution, as presented in Figure 6.21.

7
Conclusion

In this work, seeking to answer the **MRQ**, we developed a process for legal entity annotation using LLM in-context learning capabilities, as presented in Chapter 5.

To answer the **RQ1**, we developed three different strategies to select the examples for the prompt engineering process: a randomized selection, a similarity-based selection between the input and the examples, and finally, a clustering selection to choose the most representative subset of examples. As a result, we found no significant difference between the strategies implemented. Thus, the LLMs could generalized even with randomly sampled examples and do not require similar-retrieval-based methods, as seen in Section 6.2.

We also evaluated the number of examples that should be used to perform the task, answering the **RQ2**, and all six models achieved a remarkable performance with less than 32 examples. For the Llama 3.1 405B, only four examples were sufficient to annotate 70% of the legislative references, as presented in Section 6.3.1. In Section 6.2, we answered the **RQ2.a** and found that some models improve their performance by increasing the examples included in the prompt, such as Gemini 1.5 Pro and Gemini 1.5 Flash. On the other hand, the DeepSeek V2 worsened with increasing from 16 to 32 examples, suggesting that additional examples vanished or acted as noise, which oddly contradicts the experiments in DeepSeek-AI et al. (2024). Therefore, the impact of the number of examples depends on the models' architecture and pre-trained data — some models will perform better with more or fewer examples.

Regarding the **RQ3**, we have compiled each entity annotation excerpt on a single document and compare them with the ground truth annotations manually annotated by specialists in Correia et al. (2022), in two ways: in a strict-match manner when the type and boundaries of the generated annotation must match the ground-truth, and in a relaxed-match where only the types must match, despite the boundaries, as mentioned in Section 6.1. The first manner provides a more rigorous evaluation than relaxed-match.

Finally, our findings show that LLMs are indeed capable of making high-quality annotations, even for those rarer entities as academic citations. Our best results are 0.76 F1 on average, scored by Gemini 1.5 Pro with only an entity description and a set of examples demonstrating the task. These results are close to the proposed Multi-LLM approach with a 0.75 F1 score on average. Thus, due to the observed results, the LLMs could assist the annotators in

the annotation process by highlighting legal entities in the text, reducing the burden and mitigating inconsistencies and subjective problems.

7.1 Future Work

For future work, we plan to evaluate more LLMs, such as GPT-4o and Claude 3.5 Sonnet, both top-notch models for OpenAI (2023a) and Anthropic (2023) to better assess the results shown by Gemini 1.5 Pro. We also plan to run the experiments and analysis using the remaining documents from the corpus to evaluate the LLMs in an even more diverse scenario.

Moreover, we intend to review the generated annotations deeply and investigate the DeepSeek V2 performance with 32 examples to understand the behavior shown in Section 6.2. And so on, assess the proposed process in another legal corpus, or even in a different domain, to ensure that it can be transferred and whether LLMs produce again highly accurate annotations.

We also plan to create a new version of the corpus used in this work by implementing an LLM-in-the-loop annotation process with humans. These annotators will have access to the prior annotations and the LLMs' suggestions to refine and increment the corpus. We aim to create an annotation process that reduces the annotators' workload while mitigating subjectivity and inconsistency.

8
Bibliography

ALDEEN, M. et al. Chatgpt vs. human annotators: A comprehensive analysis of chatgpt for text annotation. In: IEEE. **2023 International Conference on Machine Learning and Applications (ICMLA)**. [S.l.], 2023. p. 602–609.

ANTHROPIC. **Introducing Claude**. 2023. <https://www.anthropic.com/news/introducing-claude>. Accessed: 2024-09-08.

ARAUJO, P. H. Luz de et al. Lener-br: A dataset for named entity recognition in brazilian legal text. In: VILLAVICENCIO, A. et al. (Ed.). **Computational Processing of the Portuguese Language**. Cham: Springer International Publishing, 2018. p. 313–323. ISBN 978-3-319-99722-3.

BRITO, M. et al. Cdjur-br-uma coleção dourada do judiciário brasileiro com entidades nomeadas refinadas. In: SBC. **Anais do XIV Simpósio Brasileiro de Tecnologia da Informação e da Linguagem Humana**. [S.l.], 2023. p. 177–186.

BROWN, T. et al. Language models are few-shot learners. **Advances in neural information processing systems**, v. 33, p. 1877–1901, 2020.

CAO, Y. et al. Cailie 1.0: A dataset for challenge of ai in law - information extraction v1.0. **AI Open**, v. 3, p. 208–212, 2022. ISSN 2666-6510. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S2666651022000237>.

CHEN, L.; ZAHARIA, M.; ZOU, J. How Is ChatGPT's Behavior Changing Over Time? **Harvard Data Science Review**, The MIT Press, v. 6, n. 2, mar 12 2024. Https://hdsr.mitpress.mit.edu/pub/y95zitmz.

CHRISTIANO, P. F. et al. Deep reinforcement learning from human preferences. In: GUYON, I. et al. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Available from Internet: <https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf>.

CORREIA, F. A. et al. Fine-grained legal entity annotation: A case study on the brazilian supreme court. **Information Processing Management**, v. 59, n. 1, p. 102794, 2022. ISSN 0306-4573. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0306457321002727>.

DEEPMIND. **Gemini 1.5 Technical Report**. 2024. <https://goo.gle/GeminiV1-5>. Accessed: 2024-09-08.

DEEPSEEK-AI et al. **DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model**. 2024. Available from Internet: <https://arxiv.org/abs/2405.04434>.

DEVLIN, J. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2018.

DEVLIN, J. et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 10 2018.

DONG, Q. et al. **A Survey on In-context Learning**. 2024. Available from Internet: <https://arxiv.org/abs/2301.00234>.

DUBEY, A. et al. **The Llama 3 Herd of Models**. 2024. Available from Internet: <https://arxiv.org/abs/2407.21783>.

EDDY, S. R. Hidden markov models. **Current Opinion in Structural Biology**, v. 6, n. 3, p. 361–365, 1996. ISSN 0959-440X. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0959440X9680056X>.

FETAHU, B. et al. Gazetteer enhanced named entity recognition for code-mixed web queries. In: **Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: Association for Computing Machinery, 2021. (SIGIR '21), p. 1677–1681. ISBN 9781450380379. Available from Internet: <https://doi.org/10.1145/3404835.3463102>.

GATTANI, A. et al. Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. **Proc. VLDB Endow.**, VLDB Endowment, v. 6, n. 11, p. 1126–1137, aug 2013. ISSN 2150-8097. Available from Internet: <https://doi.org/10.14778/2536222.2536237>.

GILARDI, F.; ALIZADEH, M.; KUBLI, M. Chatgpt outperforms crowd workers for text-annotation tasks. **Proceedings of the National Academy of Sciences**, National Acad Sciences, v. 120, n. 30, p. e2305016120, 2023.

GOODCHILD, M. F.; HILL, L. L. Introduction to digital gazetteer research. **International Journal of Geographical Information Science**, Taylor & Francis, v. 22, n. 10, p. 1039–1044, 2008.

GOOGLE. **Introducing Gemini: our largest and most capable AI model**. 2023. <https://blog.google/technology/ai/google-gemini-ai/>. Accessed: 2024-09-08.

GRISHMAN, R.; SUNDHEIM, B. Message Understanding Conference- 6: A brief history. In: **COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics**. [s.n.], 1996. Available from Internet: <https://aclanthology.org/C96-1079>.

LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: **Proceedings of the Eighteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 282–289. ISBN 1558607781.

LEIBON, G. et al. Bending the law: geometric tools for quantifying influence in the multinetwork of legal opinions. **Artificial Intelligence and Law**, Springer, v. 26, p. 145–167, 2018.

LEITNER, E.; REHM, G.; MORENO-SCHNEIDER, J. A dataset of German legal documents for named entity recognition. In: CALZOLARI, N. et al. (Ed.). **Proceedings of the Twelfth Language Resources and Evaluation Conference**. Marseille, France: European Language Resources Association, 2020. p. 4478–4485. ISBN 979-10-95546-34-4. Available from Internet: <https://aclanthology.org/2020.lrec-1.551>.

LI, B. et al. **Evaluating ChatGPT's Information Extraction Capabilities: An Assessment of Performance, Explainability, Calibration, and Faithfulness**. 2023. Available from Internet: <https://arxiv.org/abs/2304.11633>.

LI, J. et al. A survey on deep learning for named entity recognition : Extended abstract. In: **2023 IEEE 39th International Conference on Data Engineering (ICDE)**. [S.l.: s.n.], 2023. p. 3817–3818.

LI, M. et al. Coannotating: Uncertainty-guided work allocation between human and large language models for data annotation. **arXiv preprint arXiv:2310.15638**, 2023.

LIU, J. et al. What makes good in-context examples for gpt-3? **arXiv preprint arXiv:2101.06804**, 2021.

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 55, n. 9, jan 2023. ISSN 0360-0300. Available from Internet: <https://doi.org/10.1145/3560815>.

LOU, R.; ZHANG, K.; YIN, W. Large Language Model Instruction Following: A Survey of Progresses and Challenges. **Computational Linguistics**, p. 1–43, 08 2024. ISSN 0891-2017. Available from Internet: <https://doi.org/10.1162/coli_a_00523>.

MARRERO, M. et al. Named entity recognition: Fallacies, challenges and opportunities. **Computer Standards Interfaces**, v. 35, n. 5, p. 482–489, 2013. ISSN 0920-5489. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0920548912001080>.

META. **Meet Llama 3.1**. 2024. <https://llama.meta.com/>. Accessed: 2024-09-08.

NADEAU, D.; SEKINE, S. A survey of named entity recognition and classification. **Lingvisticae Investigationes**, John Benjamins, v. 30, n. 1, p. 3–26, 2007.

OPENAI. **Introducting ChatGPT**. 2023. <https://openai.com/index/chatgpt/>. Accessed: 2024-09-08.

OPENAI. **Prompt Engineering**. 2023. <https://platform.openai.com/docs/guides/prompt-engineering>. Accessed: 2024-09-08.

OPENAI et al. **GPT-4 Technical Report**. 2024. Available from Internet: <https://arxiv.org/abs/2303.08774>.

QUINLAN, J. R. Induction of decision trees. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 1, n. 1, p. 81–106, mar 1986. ISSN 0885-6125. Available from Internet: <https://doi.org/10.1023/A:1022643204877>.

RADFORD, A. et al. Language models are unsupervised multitask learners. **OpenAI blog**, v. 1, n. 8, p. 9, 2019.

RENZE, M.; GUVEN, E. The effect of sampling temperature on problem solving in large language models. **arXiv preprint arXiv:2402.05201**, 2024.

RINGLAND, N. et al. NNE: A dataset for nested named entity recognition in English newswire. In: KORHONEN, A.; TRAUM, D.; MÀRQUEZ, L. (Ed.). **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**. Florence, Italy: Association for Computational Linguistics, 2019. p. 5176–5181. Available from Internet: <https://aclanthology.org/P19-1510>.

TORISAWA, K. et al. Exploiting wikipedia as external knowledge for named entity recognition. In: **Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)**. [S.l.: s.n.], 2007. p. 698–707.

VASWANI, A. Attention is all you need. **Advances in Neural Information Processing Systems**, 2017.

WANG, S. et al. Gpt-ner: Named entity recognition via large language models. **arXiv preprint arXiv:2304.10428**, 2023.

WEI, J. et al. Chain-of-thought prompting elicits reasoning in large language models. **Advances in neural information processing systems**, v. 35, p. 24824–24837, 2022.

XIE, T. et al. Empirical study of zero-shot ner with chatgpt. **arXiv preprint arXiv:2310.10035**, 2023.

ZHANG, H. et al. Judicial nested named entity recognition method with mrc framework. **International Journal of Cognitive Computing in Engineering**, v. 4, p. 118–126, 2023. ISSN 2666-3074. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S2666307423000128>.

ZHANG, S.; ELHADAD, N. Unsupervised biomedical named entity recognition: Experiments with clinical and biological texts. **Journal of Biomedical Informatics**, v. 46, n. 6, p. 1088–1098, 2013. ISSN 1532-0464. Special Section: Social Media Environments. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S1532046413001196>.

ZHAO, W. X. et al. A survey of large language models. **arXiv preprint arXiv:2303.18223**, 2023.

ZHONG, H. et al. How does NLP benefit legal system: A summary of legal artificial intelligence. In: JURAFSKY, D. et al. (Ed.). **Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics**. Online: Association for Computational Linguistics, 2020. p. 5218–5230. Available from Internet: <https://aclanthology.org/2020.acl-main.466>.

9
Appendix

9.1
Entity Ontology

In this section, we describe the ontology for each entity extracted from Correia et al. (2022).

Table 9.1: Academic Citations ontology extracted from Correia et al. (2022).

Fine-grained Entity	Type	Description
Title	Text	The work published title.
Collection Title	Text	The collection title, if the publication is part of a collection (e.g., journal title).
Author	Text	The publication first author.
Co-author	Text	The publication co-author(s).
Publisher	Text	The work's publisher.
Year of Publication	Number	Year of publication of the work.

Table 9.2: Legislative References ontology extracted from Correia et al. (2022).

Fine-grained Entity	Type	Description
Legal Act	Text	The legislative act that was cited (*e.g.*, Federal or State Constitution, Legal Statutes).
Institution	Text	When the act is not legislative, such as regulations or internal rules, which institution issued it (*e.g.*, STF internal rules or Federal Reserve resolution).
Origin	Text	The Federation entity that issued regulation, municipality, state, or the Union.
Section	Number	The legal act section.
Paragraph	Number	The legal act paragraph.
Subsection	Letter	The legal act subsection.
Clause	Letter	The legal act clause.

Table 9.3: Precedents ontology extracted from Correia et al. (2022).

Fine-grained Entity	Type	Description
Legal Procedure Number	Number	The number that identifies the legal procedure in court.
Legal Procedure Class	Text	Signals the kind of legal procedure. It is often used along the case number to uniquely identify a legal procedure within STF.
Legal Procedure Origin	Text	Indicates from what state (or federation unit) the legal procedure came, usually just the acronym (*e.g.*, RJ stands for Rio de Janeiro).
Decision type	Text	Indicates if the decision referred is related to an internal appeal or motion.
Reporting Justice	Text	Identifies the Justice responsible for the decision (if a monocratic decision, the Justice is also the origin).
Court	Text	The court which rendered the decision.
Judgment date	Date	When the decision was taken.
Publication date	Date	When the decision was introduced to the official record.

9.2 Annotations Percentage of Votes on Examples Database

In this section, we describe the percentage of votes for each annotation present on the examples database and assess the misassignments produced by the annotators for each entity.

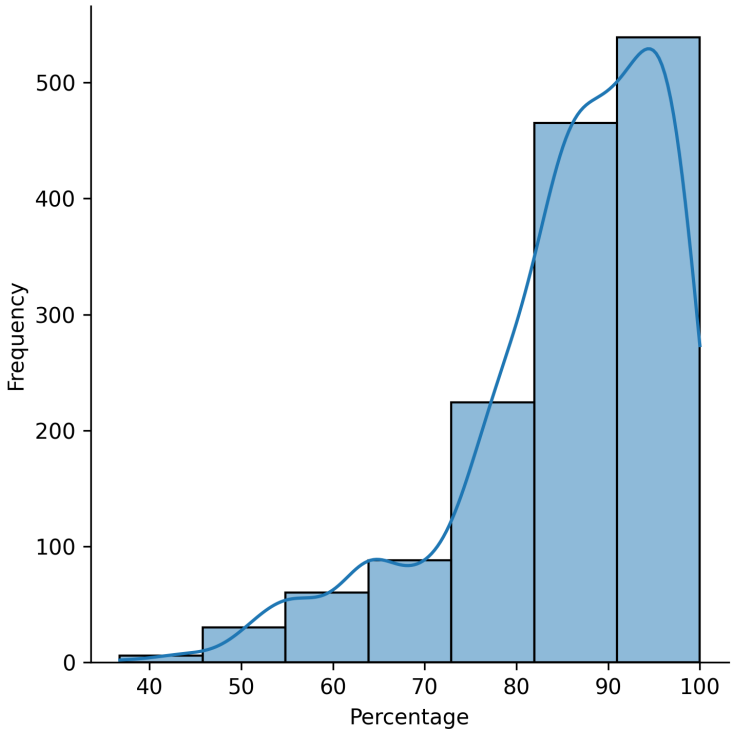


Figure 9.1: Percentage of votes for each annotation.

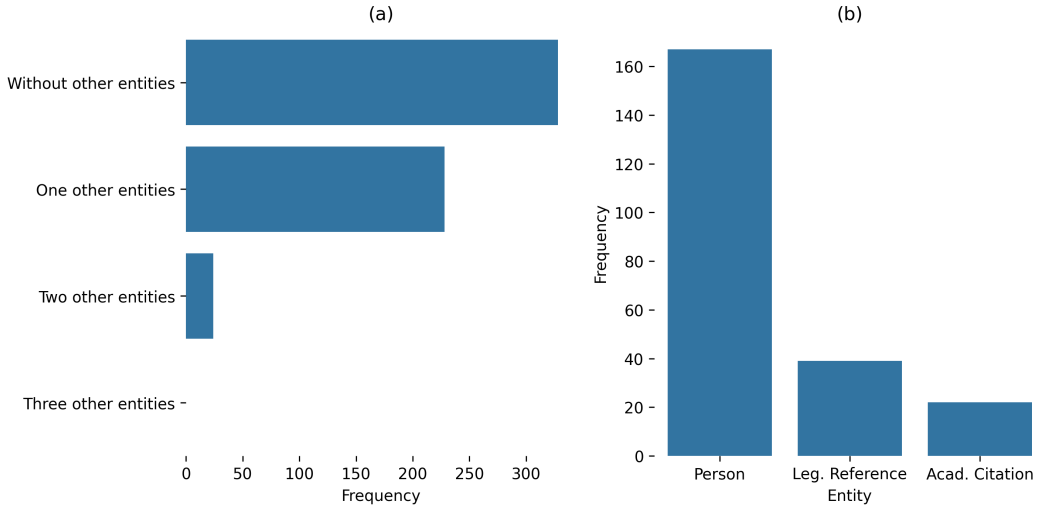


Figure 9.2: Number of precedent annotations assigned as other entities. (a) shows the number of precedent annotations which has some part mistakenly assigned as another one, two, or three entities. (b) expands the scenario where the precedent annotation was wrongly assigned with only one other entity.

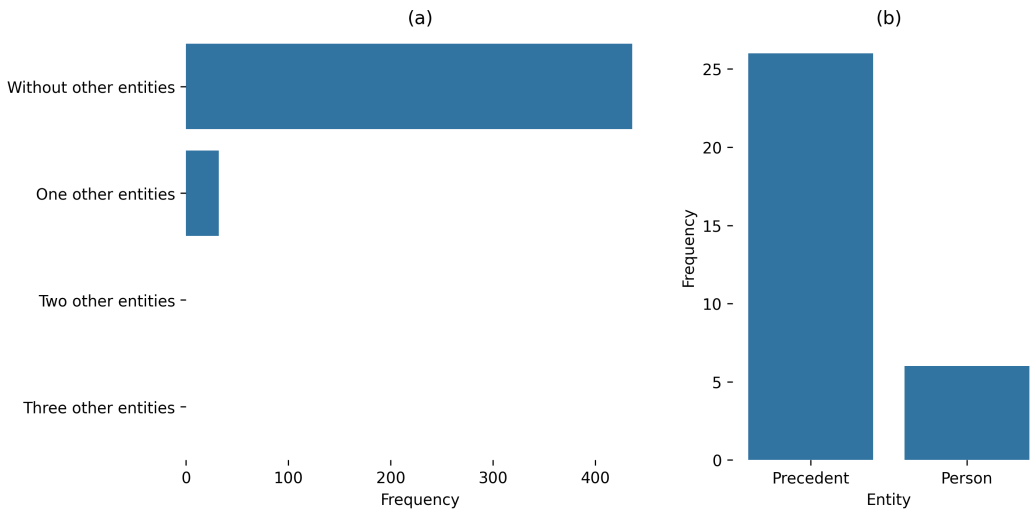


Figure 9.3: Number of legislative references annotations assigned as other entities. (a) shows the number of legislative reference annotations which has some part mistakenly assigned as another one, two, or three entities. (b) expands the scenario where the legislative reference annotation was wrongly assigned with only one other entity.

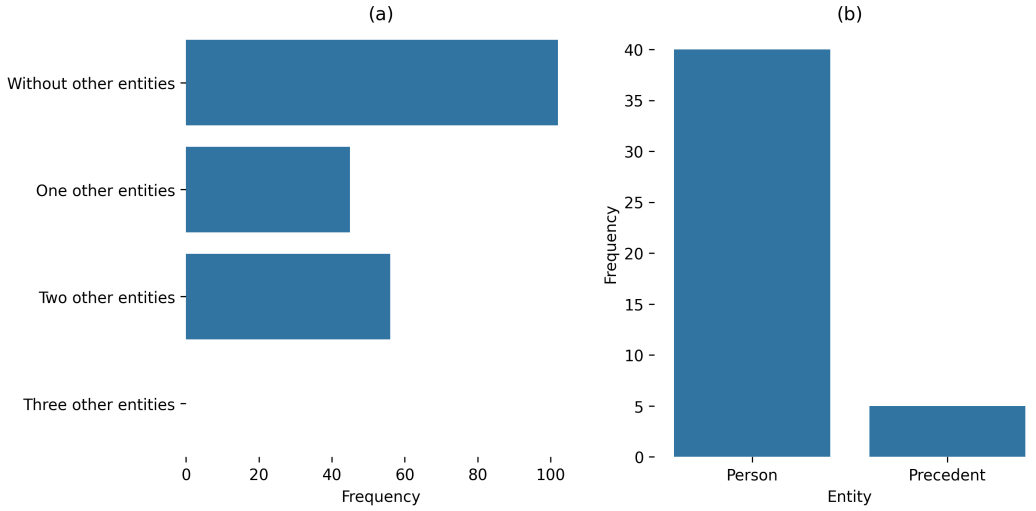


Figure 9.4: Number of academic citation annotations assigned as other entities. (a) shows the number of academic citation annotations which has some part mistakenly assigned as another one, two, or three entities. (b) expands the scenario where the academic citation annotation was wrongly assigned with only one other entity.

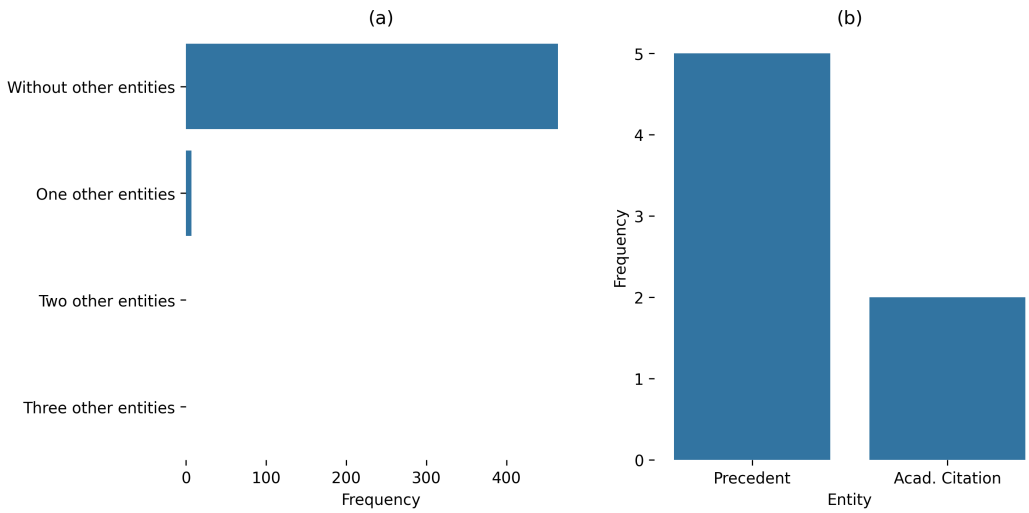
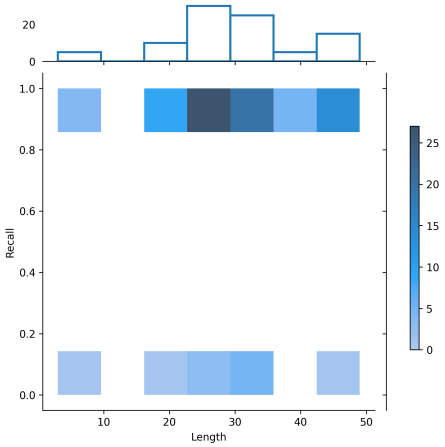


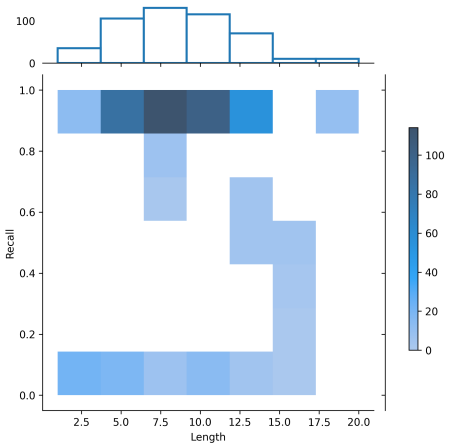
Figure 9.5: Number of person annotations assigned as other entities. (a) shows the number of person annotations which has some part mistakenly assigned as another one, two, or three entities. (b) expands the scenario where the person annotation was wrongly assigned with only one other entity.

9.3 Annotation Capabilities per Entity Length

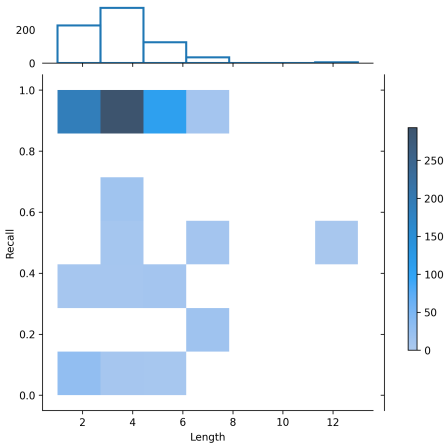
In this section, we present an analysis of the annotations recall per entity length to assess whether some model flaws are strictly due to entity length.



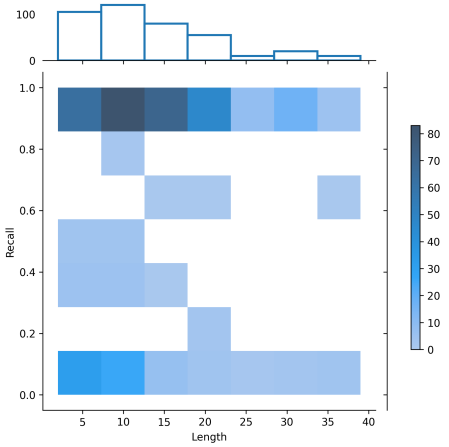
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length

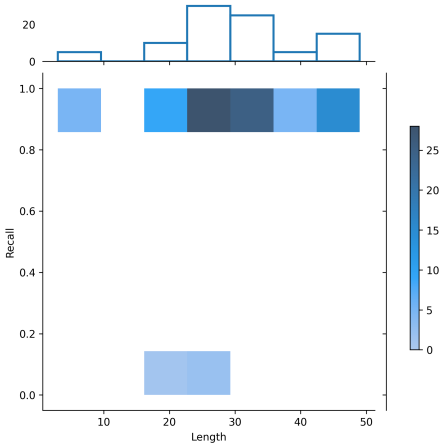


(c) Person Recall per Annotation Length

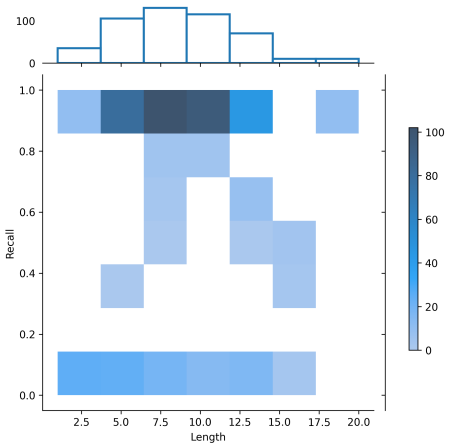


(d) Precedent Recall per Annotation Length

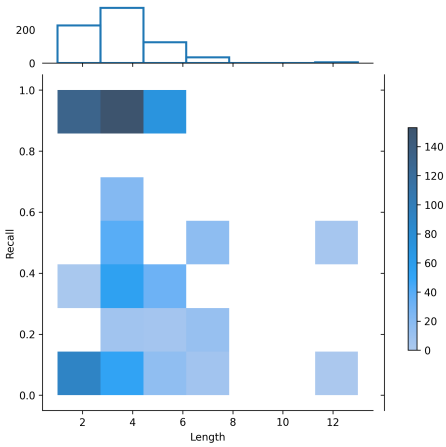
Figure 9.6: Gemini 1.5 Pro Random K=16 recognition capabilities per annotations' length.



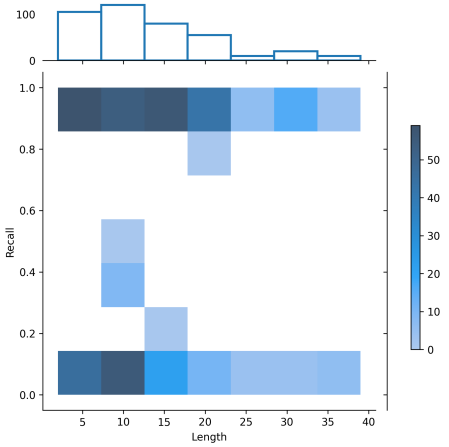
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length

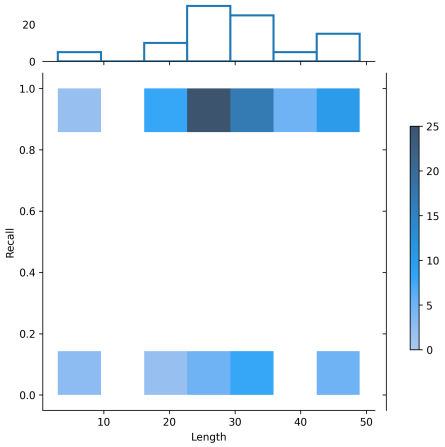


(c) Person Recall per Annotation Length

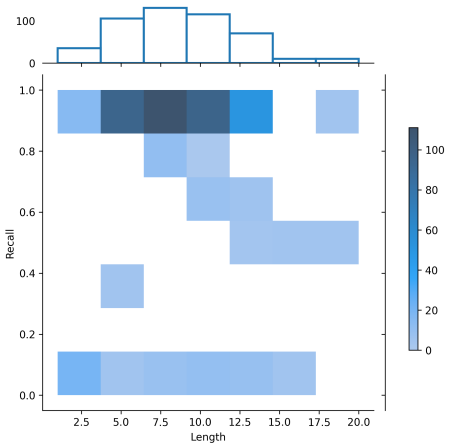


(d) Precedent Recall per Annotation Length

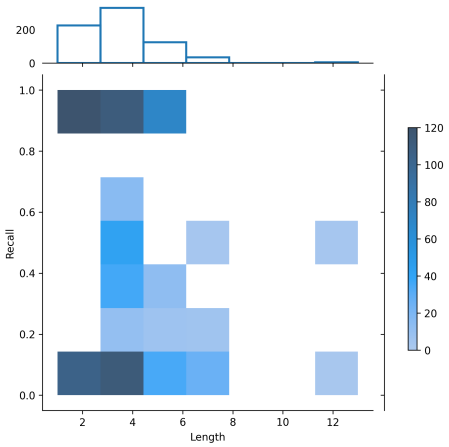
Figure 9.7: Gemini 1.5 Flash Random K=16 recognition capabilities per annotations' length.



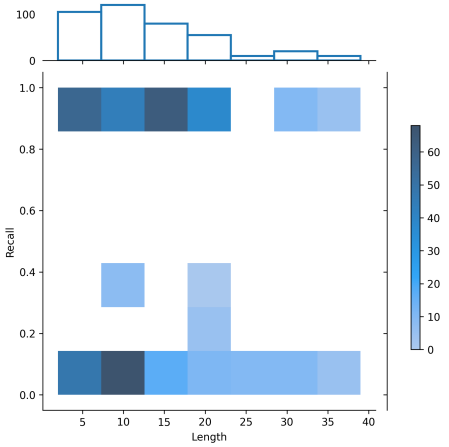
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length

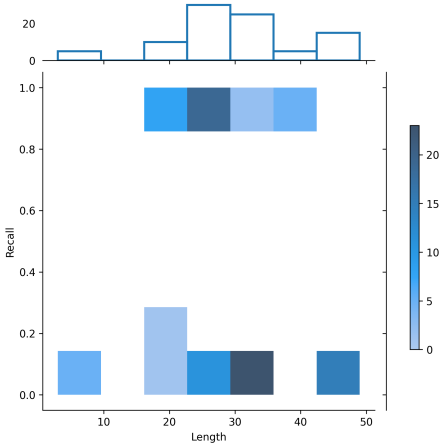


(c) Person Recall per Annotation Length

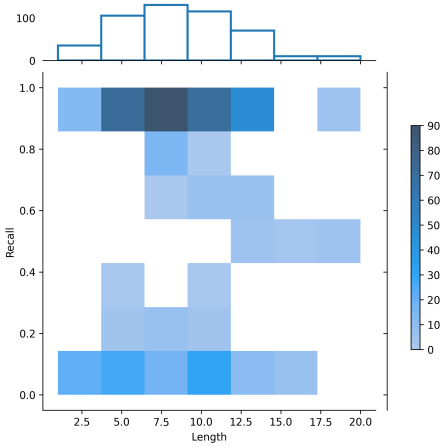


(d) Precedent Recall per Annotation Length

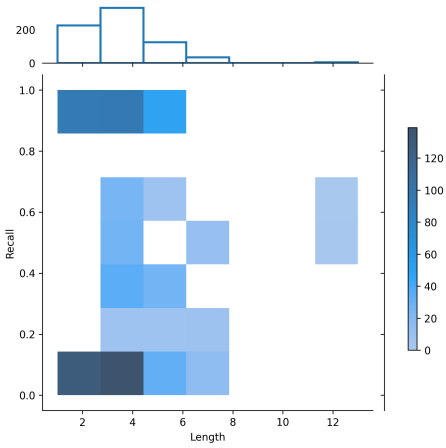
Figure 9.8: Llama 3.1 405B Random K=4 recognition capabilities per annotations' length.



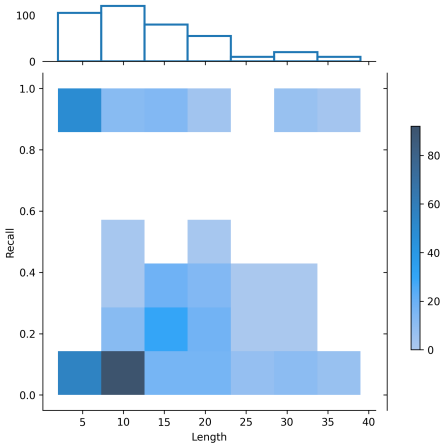
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length

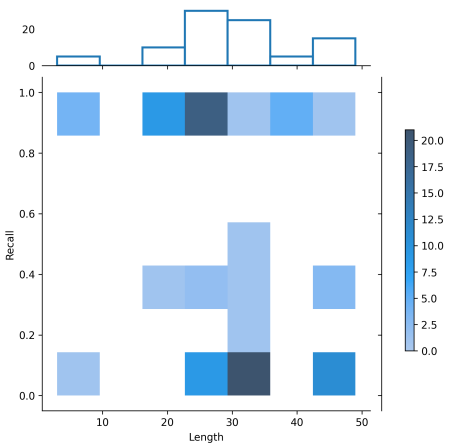


(c) Person Recall per Annotation Length

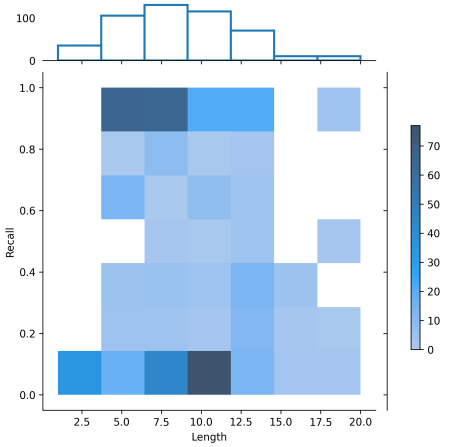


(d) Precedent Recall per Annotation Length

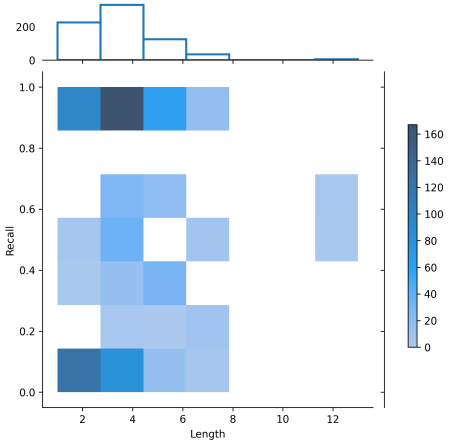
Figure 9.9: Llama 3.1 70B Random K=4 recognition capabilities per annotations' length.



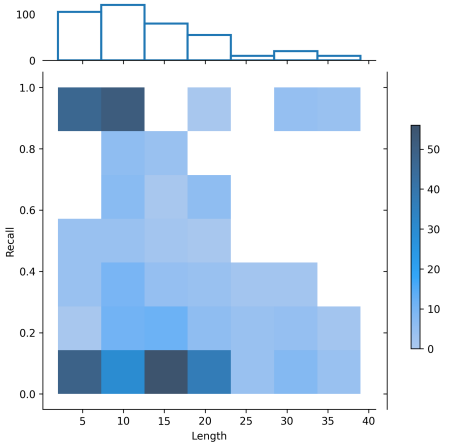
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length

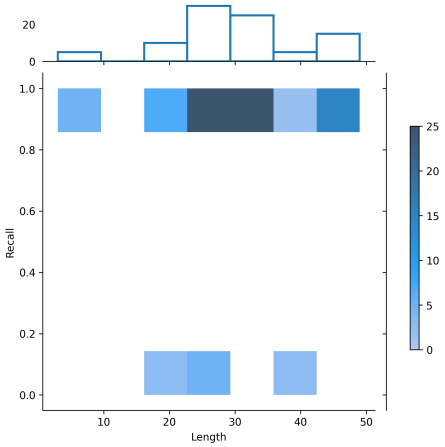


(c) Person Recall per Annotation Length

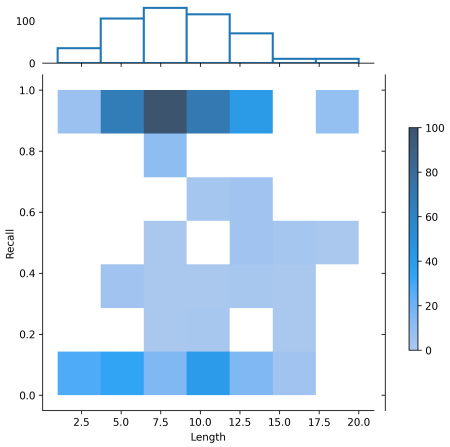


(d) Precedent Recall per Annotation Length

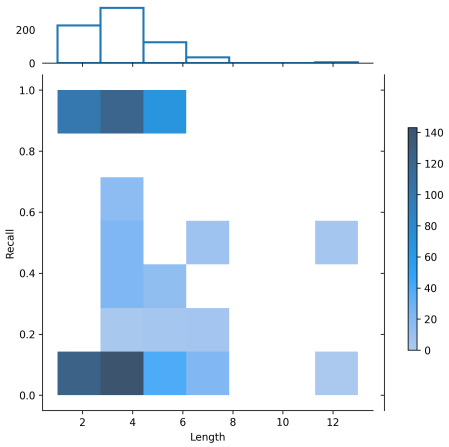
Figure 9.10: GPT-4o mini Random K=16 recognition capabilities per annotations' length.



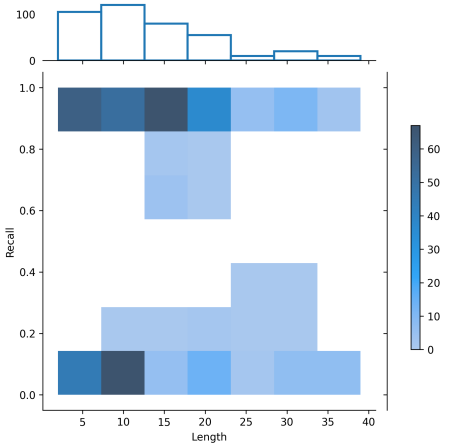
(a) Acad. Citation Recall per Annotation Length



(b) Leg. Reference Recall per Annotation Length



(c) Person Recall per Annotation Length



(d) Precedent Recall per Annotation Length

Figure 9.11: DeepSeek V2 Random K=16 recognition capabilities per annotations' length.

9.4 Statistical Tests

In this section, we present the statistical test results on the validation set for each LLM to determine whether there are significant differences between the examples selection strategy and the impact of the number of examples.

9.4.1 Statistical Test for Gemini 1.5 Pro

Table 9.4: Gemini 1.5 Pro Games-Howell Post-hoc test for the number of examples on the validation set

		4	8	16	32
4	p-value	-	0.286	<.001	<.001
8	p-value		-	<.001	<.001
16	p-value			-	0.250
32	p-value				-

Table 9.5: Gemini 1.5 Pro Kruskal-Wallis test results by examples selection strategy on the validation set

	X^2	df	p-value
F1-Score	0.764	2	0.682

9.4.2 Statistical Test for Gemini 1.5 Flash

Table 9.6: Gemini 1.5 Flash Tukey Post-hoc test for the number of examples on the validation set

		4	8	16	32
4	p-value	-	0.229	<.001	<.001
8	p-value		-	<.001	<.001
16	p-value			-	0.754
32	p-value				-

Table 9.7: Gemini 1.5 Flash Kruskal-Wallis test results by examples selection strategy on the validation set

	X^2	df	p-value
F1-Score	0.917	2	0.632

9.4.3 Statistical Test for Llama 3.1 405B

Table 9.8: Llama 3.1 405B Kruskal-Wallis test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	X^2	df	p-value
F1-Score	3.08	3	0.379

Table 9.9: Llama 3.1 405B ANOVA test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	F	df	df2	p-value
F1-Score	4.01	2	57	0.023

9.4.4 Statistical Test for Llama 3.1 70B

Table 9.10: Llama 3.1 70B Kruskal-Wallis test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	X^2	df	p-value
F1-Score	1.60	3	0.660

Table 9.11: Llama 3.1 70B ANOVA test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	F	df	df2	p-value
F1-Score	2.09	2	57	0.133

9.4.5 Statistical Test for GPT-4o mini

Table 9.12: GPT-4o mini ANOVA test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	F	df	df2	p-value
F1-Score	2.97	3	56	0.039

Table 9.13: GPT-4o mini ANOVA test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	F	df	df2	p-value
F1-Score	0.890	2	57	0.416

9.4.6 Statistical Test for DeepSeek V2

Table 9.14: DeepSeek V2 Tukey test results for the number of examples on the validation set ($\alpha = 0.01$)

		4	8	16	32
4	p-value	-	0.404	0.140	0.687
8	p-value		-	0.927	0.047
16	p-value			-	0.009
32	p-value				-

Table 9.15: DeepSeek V2 ANOVA test results by examples selection strategy on the validation set ($\alpha = 0.01$)

	F	df	df2	p-value
F1-Score	0.0102	2	57	0.990