Evaluation of Large Language Models in Contract Information Extraction

Weybson Alves da Silva[1] **and Tsang Ing Ren**[1]

[1]Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
Recife – PE – Brazil

{was5,tir}@cin.ufpe.br

Abstract. *Despite the rapid advancement of Large Language Models (LLMs), there is limited research focused on their effectiveness in extracting specific information from contracts. This study evaluates the effectiveness of state-of-the-art models—GPT-3.5-Turbo, Gemini-1.5-Pro, Claude-3.5-Sonnet, and Llama-3-70B-Instruct—in extracting key clauses from contracts using the Contract Understanding Atticus Dataset (CUAD). We explore the impact of prompting strategies and input context configurations across two scenarios: one covering all 41 clause categories and another focusing on a subset of three. Our findings reveal that LLMs can extract contract information efficiently, outperforming traditional human review in terms of time and cost. Performance, however, varies significantly depending on context size and task specificity, with reduced context approaches and focused extractions often improving recall at the expense of precision. Notably, Claude-3.5-Sonnet, with zero-shot with output example and reduced context, achieved a recall of 0.77 and precision of 0.66, surpassing prior benchmarks on full-category extraction. However, performance is inconsistent across clause types. Models like Llama-3-70B-Instruct, while less robust, demonstrated strong performance on simpler tasks, highlighting their potential in targeted use cases. Additionally, retrieval-augmented generation shows potential for improving extraction and efficiency in long documents, though its performance is constrained by retriever accuracy. Our experiments suggest that with further refinement, LLMs could be vital in automating complex legal tasks, particularly in efficiently handling dense legal texts such as contracts.*

1. Introduction

In the field of legal agreements, contracts serve as pivotal documents that outline the rights, responsibilities, and obligations of involved parties within a specific business context. Ensuring accurate interpretation of these documents is essential to uphold the agreed terms and prevent potential legal disputes. However, the contract review process is inherently complex, often requiring significant manual effort from legal professionals to extract critical information from extensive textual data.

The advent of large language models (LLMs), exemplified especially by tools such as OpenAI's ChatGPT, which are based on Generative Pre-Trained Transformers (GPT) architectures, has sparked considerable interest due to their versatility and potential to simplify various tasks, including contract review. Nonetheless, the specific application of

LLMs for extracting and categorizing information from legal contracts remains underexplored. [Martin et al. 2024] evaluated models such as GPT, PaLM, and Claude in identifying legal issues in procurement contracts, but their study was limited in scope, with a sample of only 10 contracts and analyzing only one approach with zero-shot prompting. [Savelka and Ashley 2023] expanded the scope to a broader range of legal documents but focused only on the classification of short excerpts using a zero-shot approach. Previous research by [Leivaditi et al. 2020] and [Hendrycks et al. 2021] evaluated the performance of older models, such as BERT and RoBERTa, on similar tasks.

To fill this research gap, we evaluated the effectiveness of generative LLMs in extracting information from various types of contracts using the Contract Understanding Atticus Dataset (CUAD) [Hendrycks et al. 2021]. CUAD is a comprehensive dataset for legal contract review, containing over 500 contracts annotated by legal experts to identify 41 types of clauses, with over 13,000 annotations.

In this paper, we evaluate the effectiveness of LLMs, specifically GPT-3.5-Turbo [Brown et al. 2020], Llama-3-70B-Instruct [Meta 2024], Gemini-1.5-Pro [Gemini Team et al. 2024], and Claude-3.5-Sonnet [Anthropic 2024a, Anthropic 2024b], in extracting key entities such as dates and involved parties, as well as specific clauses such as licensing and non-compete, from contracts in the CUAD dataset. To achieve this, we developed pipelines that combine various prompting and input context methods and analyzed the results across different levels of task complexity to provide a comprehensive and accurate assessment of these generative LLMs' capabilities in contract information extraction.

The remainder of the paper is structured as follows: Section 2 discusses the context and related work, Section 3 outlines our methodology, Section 4 describes our experiments, Section 5 presents our results and discussion, and Section 6 concludes with directions for future work.

2. Related Work

Recent research has underscored the remarkable generalization capabilities of large language models. [Zhou et al. 2023] demonstrated the effectiveness of fine-tuning LLMs on Information Extraction (IE) tasks, leveraging training data to enhance performance. Additionally, studies by [Wei et al. 2023] and [Wang et al. 2023] highlighted LLMs' ability to perform information extraction in few-shot and zero-shot scenarios, relying on in-context examples or instructions. These advancements illustrate the potential of LLMs across various practical applications, including legal document analysis.

Significant advancements in the legal domain, particularly in contract review, have been driven by the development of specialized datasets and models. A pivotal contribution in this area is the Contract Understanding Atticus Dataset (CUAD) [Hendrycks et al. 2021], introduced in 2021, which has set a new standard for legal contract review datasets. This dataset builds upon earlier efforts, such as those by [Chalkidis et al. 2017] and [Leivaditi et al. 2020], with the latter focusing on lease contracts and both being limited to a smaller set of label categories compared to CUAD. Thereby, with its broader range of expert annotations and contract types and its coverage of the inherent complexity of the legal domain, CUAD offers a rigorous benchmark for the NLP community. Their initial evaluations using transformer models such as BERT

and RoBERTa also demonstrated promising performance, further solidifying CUAD as a valuable resource for advancing contract analysis capabilities.

Addressing the challenge of processing long documents in the legal domain, [Li et al. 2023] proposed an efficient cascading pipeline. Their approach first employs a logistic regression model to eliminate irrelevant content, enabling a more efficient processing of inputs by a transformer-based model. Evaluated using a subset of label categories from the CUAD dataset, this method showed significant gains in training time and information extraction performance compared to a pure transformer-based approach, attributing the improvements to the model's ability to focus on more challenging examples.

The emergence of LLMs like ChatGPT has highlighted their potential in various legal text annotation tasks. [Savelka and Ashley 2023] examined the zero-shot semantic annotation capabilities of GPT-4 and GPT-3.5-turbo at the sentence level, comparing them to previous generations of GPT models across different legal text annotation tasks. The study found that GPT-4 exhibited promising results and outperformed its predecessors in two of the three tasks analyzed. Their findings underscore the evolving capabilities of LLMs in legal applications, particularly in contract review.

In comparing LLMs and traditional legal contract reviewers, [Martin et al. 2024] assessed LLMs' accuracy, speed, and cost-efficiency against junior lawyers and legal process outsourcers. A key finding from their analysis is the remarkable speed advantage of LLMs, which can complete contract reviews in seconds, vastly outpacing the hours required by human reviewers, while also costing significantly less. However, the study had limitations, such as a small sample size of ten procurement contracts and an analysis of only a zero-shot approach to carrying out the tasks. Additionally, improved versions of the models and new state-of-the-art models have been released since the study was conducted.

By situating our work within the broader context of IE, recent advancements in LLMs, and specialized legal datasets like CUAD, we contribute to the ongoing discourse on the application of machine learning in the legal domain. Our study not only builds on previous research but also explores new avenues for enhancing the accuracy and efficiency of contract review using advanced LLMs. Through a comprehensive approach and a representative sample of contracts, we seek to demonstrate the viability and usefulness of these models in simplifying and accelerating the contract review process.

3. Methodology

This section describes the methodology employed in our study. By detailing the dataset characteristics, preprocessing techniques, selected language models, prompt engineering methods, and input context strategies, we aim to provide a comprehensive overview of how each component contributes to optimizing performance and effectively assessing the models in extracting information from contracts (Figure 1).

3.1. Dataset

The CUAD stands out as a pivotal resource in the realm of contract review, addressing the critical need for high-quality, annotated datasets in the legal field. Developed by [Hendrycks et al. 2021], CUAD is specifically designed to facilitate the training and evaluation of machine learning models in identifying and extracting relevant clauses from

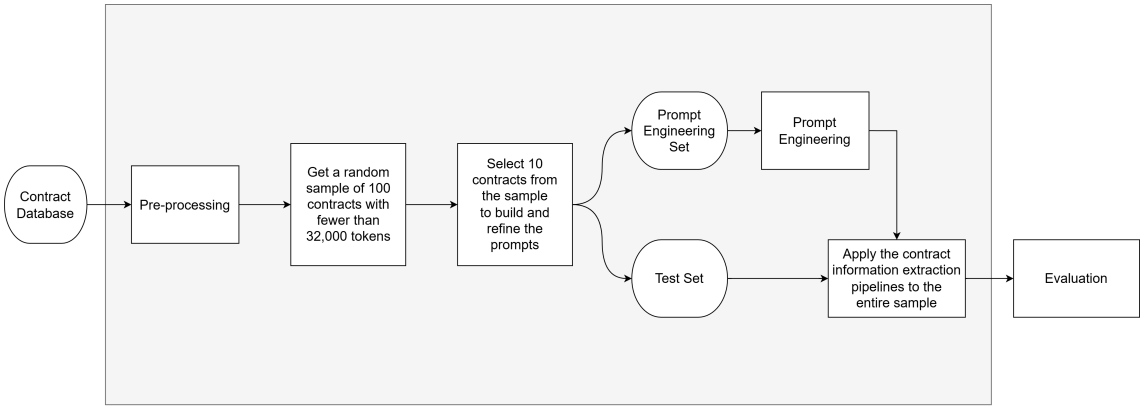

Figure 1. Overview of the methodology framework.

contracts—a task traditionally performed by junior law firm associates. This tedious and repetitive work involves manually sifting through extensive contract documents to identify and categorize clauses, making it a prime candidate for automation.

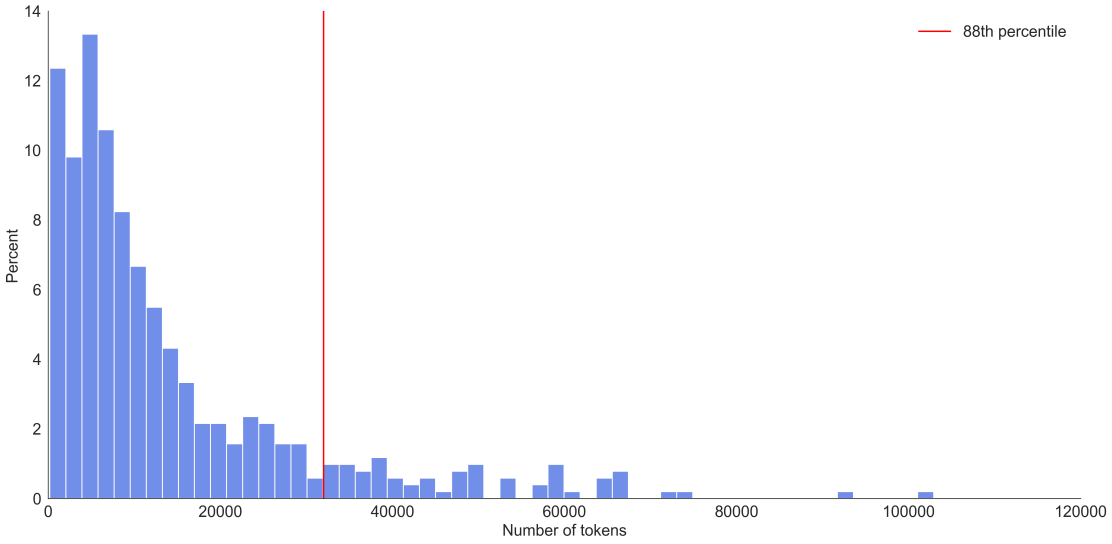

Figure 2. Distribution of the number of tokens in contracts from the full dataset. The line indicates the 88th percentile of the data, at 32,000 tokens.

CUAD comprises 510 contracts, featuring a total of 13,101 labeled clauses across 41 distinct categories. These contracts exhibit considerable variability, encompassing 25 different types and varying significantly from a few to over one hundred pages. Notably, only about 10% of each contract is highlighted on average, indicating the portions containing relevant clauses [Hendrycks et al. 2021]. This selective annotation underscores the efficiency and focus required in contract analysis.

For our study, we have selected a random sample of 100 contracts from the CUAD dataset, ensuring that only the smallest contracts, with 32,000 tokens or less, were included. We used the "cl100k_base" encoder, the encoder used by GPT-3.5-Turbo, to tokenize [Minaee et al. 2024] the contracts. We implemented this restriction to manage our limitations related to the LLMs APIs' costs while ensuring a representative distribu-

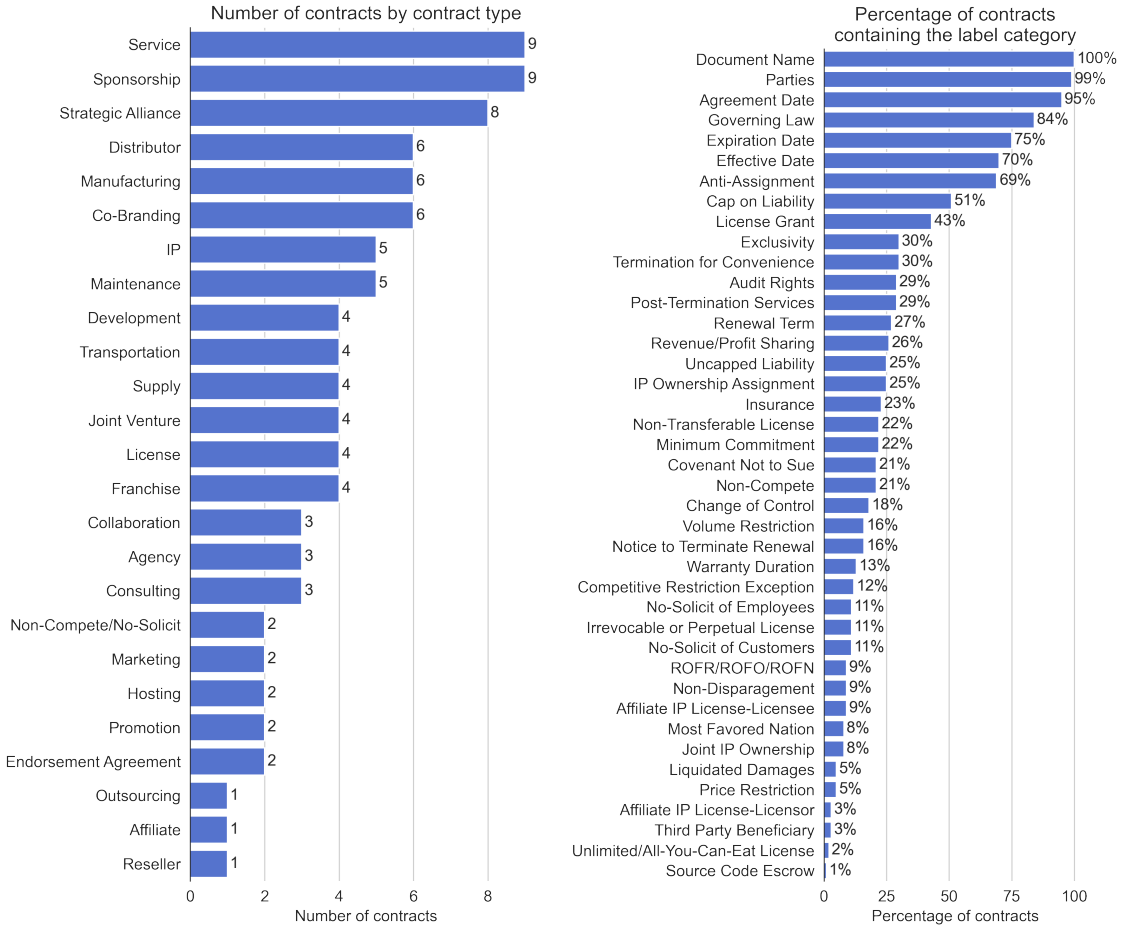

Figure 3. Number of contracts by type and presence of label categories in the evaluation sample.

tion of contract types for our experiments. This sampling strategy is supported by the observation that 88% of the CUAD database contracts contain fewer than 32,000 tokens, as shown in Figure 2. Figures 3a and 3b provide a detailed breakdown of our sample's contract types and label categories.

3.2. Preprocessing

Preprocessing is crucial for preparing datasets for efficient model evaluation. Poorly structured contexts can significantly affect model performance in extraction tasks, leading to ambiguous or incorrect interpretations and reduced accuracy and reliability. Thus, well-structured prompts and context are vital for optimizing model performance.

The text extractions from the contracts provided by [Hendrycks et al. 2021] were inconsistent, breaking the original formatting of the contracts, for example by condensing several paragraphs into a single line. To address this, we used Python's PyMuPDF library [PyMuPDF 2024] to re-extract the texts from the PDFs. PyMuPDF offers advanced functionalities for extracting, analyzing, and manipulating PDF contents, enabling more precise and uniform extraction of contractual texts. This step ensured that the structure of the extracted text did not bias our experimental results. A comparison between both

extractions is shown in Figure 4.

CUAD

```
1. RIGHTS GRANTED 1.1. Grant of Franchise. 1.1.1 We grant you the right, and you accept the obligation, to use the Proprietary Marks and the System to operate one Restaurant (the "Franchised Business") at the Premises, in accordance with the terms of this Agreement. The term "Premises" means the location shown on the cover page to this Agreement or a location that is determined under process set out in Section 4.1 [Site Selection and Approval] below. 1.1.2 Your rights under this Agreement have limits, such as the following: (a) You have no right to operate the Franchised Business at any location other than the Premises, as described in Section 1.2 [Activities of Franchised Business] below; (b) You have no right to sublicense either the Proprietary Marks or the System to anyone else; and (c) You have no right to use the Proprietary Marks or the System in any manner except as expressly authorized under this Agreement. [...] We may periodically establish rules and policies regarding such advertising. Page 2 of 39

Source: PF HOSPITALITY GROUP INC., 10-12G, 9/23/2015
```

PyMuPDF

```
1. RIGHTS GRANTED
1.1. Grant of Franchise.
1.1.1 We grant you the right, and you accept the obligation, to use the Proprietary Marks and the System to operate one
Restaurant (the "Franchised Business") at the Premises, in accordance with the terms of this Agreement. The term "Premises" means the location
shown on the cover page to this Agreement or a location that is determined under process set out in Section 4.1 [Site Selection and Approval]
below.
1.1.2 Your rights under this Agreement have limits, such as the following:
(a) You have no right to operate the Franchised Business at any location other than the Premises, as described in
Section 1.2 [Activities of Franchised Business] below;
(b) You have no right to sublicense either the Proprietary Marks or the System to anyone else; and
(c) You have no right to use the Proprietary Marks or the System in any manner except as expressly authorized under
this Agreement.
[...]
We may periodically establish rules and policies regarding such advertising.
Page 2 of 39
Source: PF HOSPITALITY GROUP INC., 10-12G, 9/23/2015
```

Figure 4. Comparison between CUAD's extraction and extraction using PyMuPDF.

3.3. Models

In this study, we compared the performance of commercial and open-source state-of-the-art large language models.

GPT-3.5-Turbo. Developed by OpenAI, GPT-3.5-Turbo [Brown et al. 2020] is an generative language model with a context window of 16,385 tokens. It handles both conversational and non-conversational tasks effectively. We primarily chose GPT-3.5-Turbo due to its lower cost compared to GPT-4 [OpenAI et al. 2023] and its strong performance in diverse NLP applications. Access was provided via the OpenAI API with the `gpt-3.5-turbo-0125` version.

Llama-3-70B-Instruct. Meta's Llama-3-70B-Instruct [Meta 2024] is a state-of-the-art instruction-fine-tuned language model with 70 billion parameters and an 8,192 token context window. We selected Llama-3-70B-Instruct for its availability as open-source, making it a strong choice for enterprises seeking complete control over their solutions and for its robust performance, outperforming closed-source models on specific benchmarks. The model was accessed via Groq API using the `llama3-70b-8192` version.

Claude-3.5-Sonnet. Anthropic's Claude-3.5-Sonnet [Anthropic 2024a, Anthropic 2024b] is a high-performance model known for its speed and cost-effectiveness. In certain benchmarks, it even surpasses GPT-4o [OpenAI 2024a], demonstrating its efficiency and output quality capability. The model features a 200K token context window, enabling it to process large volumes of text and perform complex reasoning tasks that require understanding intricate instructions and context, which makes it well-suited for applications such as detailed document analysis. The model was accessed via Google Cloud's Vertex AI API with the `claude-3-5-sonnet@20240620` version.

Gemini-1.5-Pro. Google's Gemini-1.5-Pro [Gemini Team et al. 2024] is a multimodal large language model that excels in retrieving information from long contexts, capable of handling up to 2 million tokens. We selected this model for its high recall and reasoning capabilities over extensive textual data, which is crucial for detailed information extraction from contracts. We utilize the `gemini-1.5-pro-001` version via Google Cloud's Vertex AI API.

Text-Embedding-3-Large. OpenAI's Text-Embedding-3-Large [OpenAI 2024b] model is designed for generating embeddings, which are numerical representations that capture the semantics of text. It outperforms other OpenAI embedding models in benchmarks for both multilingual retrieval and English tasks. By using embeddings, different text segments can be compared and their semantic similarity measured, enabling efficient and accurate information retrieval from contracts. We utilized the OpenAI API to access this model.

3.4. Prompt Engineering

Like other legal texts, contracts exhibit unique linguistic characteristics that pose significant challenges for effective information extraction systems. These documents often blend elements of standardized texts adaptable to various contexts, resulting in generalized and indeterminate content [Anesa 2007]. Additionally, it is common for contracts to intentionally incorporate ambiguous language, allowing for flexible interpretation of terms to accommodate unforeseen or evolving circumstances [Li 2017]. This deliberate vagueness, coupled with the inherent complexity of contracts—characterized by densely woven clauses and verbose legal terminology—further complicates the extraction process. As a result, these factors together create substantial obstacles for systems attempting to identify and extract relevant contract information accurately.

To address these challenges, effective prompt engineering becomes indispensable for optimizing large language models for contract information extraction. By meticulously designing prompts, we can guide the model's behavior to produce accurate, relevant, and consistent outputs despite the inherent complexity of contract language. This

process involves balancing specificity and flexibility to navigate the ambiguities and intricacies of legal texts, ensuring that the extracted data is accurate and comprehensive.

In our prompt development, we employe several components designed to address these challenges and enhance the model's performance in extracting information from contracts:

Persona and task definition. Assigning a "helpful AI legal assistant" persona and explicitly stating the task defines a specific context, helping to direct the model's perspective towards the task and the output's style.

Guidelines. Specific and direct instructions guide the model in performing the task. For instance, the model is instructed to transcribe all excerpts fully, without modifying content, and to handle redacted sections appropriately, ensuring alignment with the desired behavior and reasoning.

Delimiters. We use delimiters like "###" to clearly separate and highlight key sections of the prompt. This component helps the model distinguish between different parts of the instructions and maintain focus on each section's purpose.

Instruction reinforcement. Key instructions are emphasized along the prompt to ensure clarity and adherence to specific tasks or objectives. This repetition aims to prevent oversight and enhance the effectiveness of interactions with the LLMs.

Clear target and descriptions. The prompt explicitly describes each type of information to be extracted, such as contract dates, party names, and specific clauses. Clear definitions help the model precisely understand what to look for in the contract text. By providing detailed descriptions, we minimize ambiguity and ensure the model extracts the correct information.

Output format constraint. In order to ensure that the extracted information is presented in a consistent and structured manner, we provide detailed descriptions of the output format. This includes specifying the format in which the information should be returned, such as JSON arrays containing the extracted text snippets. Additionally, we stipulate that if a piece of information is not present in the contract, the model should return an empty array for that category. This measure is in place to prevent the model from generating irrelevant or speculative content.

Output example. Incorporating examples in the prompts further enhances performance. A relevant example from the training data provides a concrete illustration of the expected output, helping the model better understand the task.

Following this methodology, we created prompts for two approaches—zero-shot and zero-shot with output example—to evaluate the LLMs' performance. Each model was queried using identical prompts to ensure uniformity in evaluations.

For the zero-shot approach, we designed a detailed prompt that provides clear task instructions, expected output format, and guidelines to restrict the model's responses to the information explicitly contained within the contract. The prompt was manually refined through iterative testing on a subset of ten contracts to ensure optimal performance. The detailed zero-shot prompt is illustrated in Figure 5.

Zero-shot System Prompt Template

You're a helpful AI legal assistant. You will be given part of the text of a contract. Your task is to extract important information and clauses from the contract.

To carry out this task, ALWAYS stick to the following guidelines, delimited by ###:
###
- Extract only the information explicitly requested.
- Do not use prior knowledge. Use only the content of the contract to generate your answer.
- Fully transcribe all excerpts referring to each clause and information. Sentences must be extracted completely.
- Do not modify the content of the excerpts.
- If an excerpt pertains to more than one category of information, extract the excerpt for each relevant category.
- Extract any passages related to the specified information, even if the information is not explicitly stated. For example, if you are looking for the expiration date, also extract passages that imply this information. This could include sentences like, "This agreement shall become effective on the date stated above and shall continue in effect for a period of [X] years" even if a specific date is not mentioned.
- Some sections of the contract are redacted for confidentiality, denoted by asterisks (***) or underscores (___) or left as blank spaces. If a redacted section corresponds to one of the pieces of information to be extracted, extract the passage while keeping the redaction intact. For example, the passage "This Franchise Agreement ('Agreement') made this ____ day of _____, 19____" would have as Agreement Date "____ day of _____, 19____".

###

Please make sure you read and understand these instructions carefully. Please remember to thoroughly reread the contract and treat each check independently for accurate identification.

Read and analyze the contents of the contract carefully and identify ALL of the excerpts related to the information listed in the provided checklist. Submit your final response in JSON format. The keys should match the names of the information in the provided checklist, and the values should be arrays containing the corresponding excerpts from the contract. If a piece of information is not present in the contract, return an empty array. The JSON MUST include ONLY the information listed below, delimited by ```:

```
- <label_category_1>: <label_category_description_1>
- <label_category_2>: <label_category_description_2>
  [...]
```

Zero-shot User Prompt Template

Contract's content:
<contract_text>

Figure 5. Prompt for extracting information in the zero-shot approach.

We built upon the best-performing zero-shot prompt to construct the zero-shot with output example prompt. An example demonstrating the desired output was appended to the prompt. This example was randomly selected from the annotations of the contracts not used in the evaluation set, providing the model with a concrete illustration of the task. The zero-shot with output example, detailed in Figure 6, included the same comprehensive instructions as the zero-shot prompt, along with the example to facilitate better understanding and performance.

Zero-shot with Output Example System Prompt Template

You're a helpful AI legal assistant. You will be given part of the text of a contract. Your task is to extract important information and clauses from the contract.

To carry out this task, ALWAYS stick to the following guidelines, delimited by ###:
###
- Extract only the information explicitly requested.
- Do not use prior knowledge. Use only the content of the contract to generate your answer.
- Fully transcribe all excerpts referring to each clause and information. Sentences must be extracted completely.
- Do not modify the content of the excerpts.
- If an excerpt pertains to more than one category of information, extract the excerpt for each relevant category.
- Extract any passages related to the specified information, even if the information is not explicitly stated. For example, if you are looking for the expiration date, also extract passages that imply this information. This could include sentences like, "This agreement shall become effective on the date stated above and shall continue in effect for a period of [X] years" even if a specific date is not mentioned.
- Some sections of the contract are redacted for confidentiality, denoted by asterisks (***) or underscores (___) or left as blank spaces. If a redacted section corresponds to one of the pieces of information to be extracted, extract the passage while keeping the redaction intact. For example, the passage "This Franchise Agreement ('Agreement') made this ____ day of _____, 19____" would have as Agreement Date "____ day of _____, 19____".

###

Please make sure you read and understand these instructions carefully. Please remember to thoroughly reread the contract and treat each check independently for accurate identification.

Read and analyze the contents of the contract carefully and identify ALL of the excerpts related to the information listed in the provided checklist. Submit your final response in JSON format. The keys should match the names of the information in the provided checklist, and the values should be arrays containing the corresponding excerpts from the contract. If a piece of information is not present in the contract, return an empty array. The JSON MUST include ONLY the information listed below, delimited by ```:

```
- <label_category_1>: <label_category_description_1>
- <label_category_2>: <label_category_description_2>
  [...]
```

Example of output, delimited by ===:

```
===
{
"<label_category_1>":  [<label_category_extractions_1>],
"<label_category_2>":  [<label_category_extractions_2>],
[...]
}
===
```

Zero-shot with Output Example User Prompt Template

Contract's content:
<contract_text>

Figure 6. Prompt for extracting information in the zero-shot with output example approach.

Figure 7 provides an illustrative example of a successful information extraction from a contract using these prompts. In this figure, the model accurately identifies and extracts key data such as contract dates, party names, and specific clauses. The structured JSON format of the output demonstrates both precision and consistency in alignment with the prompt's constraints. This structured output facilitates further processing and enhances interpretability and auditability, ensuring that the model's extraction aligns with the document's semantic structure.

```
{
    "Document Name": [
        "CONSULTING AGREEMENT"
    ],
    "Parties": [
        "Global Technologies, Ltd",
        "Timothy Cabrera",
        "(individually, a \"Party\"; collectively, the \"Parties\").",
        "Consultant",
        "Company"
    ],
    "Agreement Date": [
        "This Consulting Agreement (the \"Agreement\") is made and entered into as of this 2nd day of
        January 2020,"
    ],
    "Effective Date": [
        "2nd day of January 2020"
    ],
    "Expiration Date": [
        "This Agreement shall be in full force and effect commencing on January 2, 2020 and shall remain in
        effect for one (1) year or until Consultant completes the services requested"
    ],
    "Renewal Term": [],
    "Notice to Terminate Renewal": [],
    "Governing Law": [
        "This Agreement and the legal relations among the Parties hereto shall be governed by and construed
        in accordance with the laws of the State of Florida, without regard to its conflict of law
        doctrine."
    ],
    "Most Favored Nation": [],
    "Competitive Restriction Exception": [],
    "Non-Compete": [],
    "Exclusivity": [],
    "No-Solicit of Customers": [
        "he Company further agrees that neither it nor its employees, affiliates or assigns, shall enter
        into, or otherwise arrange (either for it/him/herself, or any other person or entity) any business
        relationship, contact any person regarding such Opportunity, either directly or indirectly, or any
        of its affiliates, or accept any compensation or advantage in relation to such Opportunity except
        as directly though Consultant, without the prior written approval of Consultant."
    ],
    "No-Solicit of Employees": [],
    "Non-Disparagement": [],
    "Termination for Convenience": [],
    "Right of First Refusal, Offer, or Negotiation (ROFR/ROFO/ROFN)": [],
    "Change of Control": [],
    "Anti-Assignment": [
        "Neither Party shall delegate the performance of its duties under this Agreement without the prior
        written consent of the other Party."
    ],
    "Revenue/Profit Sharing": [],
    "Price Restriction": [],
    "Minimum Commitment": [],
    "Volume Restriction": [],
    "IP Ownership Assignment": [],
    "Joint IP Ownership": [],
    "License Grant": [],
    "Non-Transferable License": [],
    "Affiliate IP License-Licensor": [],
    "Affiliate IP License-Licensee": [],
    "Unlimited/All-You-Can-Eat License": [],
    "Irrevocable or Perpetual License": [],
    "Source Code Escrow": [],
    "Post-Termination Services": [],
    "Audit Rights": [],
    "Uncapped Liability": [],
    "Cap on Liability": [],
    "Liquidated Damages": [],
    "Warranty Duration": [],
    "Insurance": [],
    "Covenant Not to Sue": [],
    "Third Party Beneficiary": []
}
```


Figure 7. Example of successful information extraction.

3.5. Input Context Methods

This section outlines the methodologies employed for handling input contexts for information extraction. We detail three approaches: total context, reduced context, and retrieval-based context, each designed to optimize the extraction process by effectively managing the model's context window.

3.5.1. Total Context

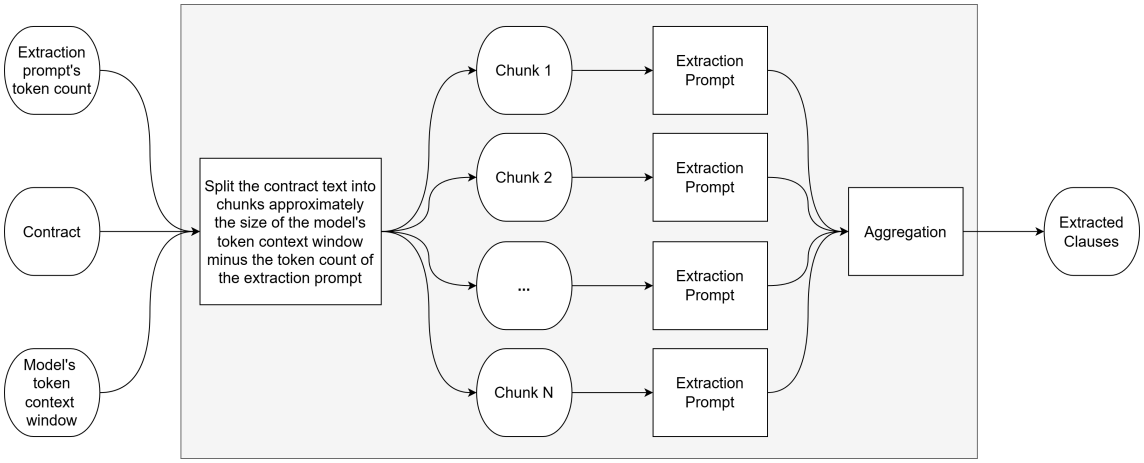

Figure 8. Extraction pipeline with total context approach.

In the total context approach, we divide the document into chunks, where each chunk, combined with the prompt, matches the maximum context window size of the model. This method ensures that the model utilizes its total capacity for context, potentially improving accuracy by providing comprehensive information for each extraction task (Figure 8). However, the primary limitation is that finding specific information becomes more complicated, as the information to be extracted may be surrounded by irrelevant details, making the extraction process less focused and more challenging.

3.5.2. Reduced Context

We employ a reduced context strategy to mitigate the complexity introduced by the total context approach. Here, documents are broken into chunks significantly smaller than the model's maximum context window, namely, chunks of 2,000 tokens in the case of GPT-3.5-Turbo and Llama-3-70B-Instruct and chunks of 5,000 tokens in the case of Claude-3.5-Sonnet and Gemini-1.5-Pro. By processing the contract little by little, this approach allows LLMs to focus more on the particularities of each section, enhancing the model's ability to extract relevant information accurately (Figure 9). This method balances the trade-off between context size and computational efficiency, helping manage resource usage while providing sufficient context for effective information extraction.

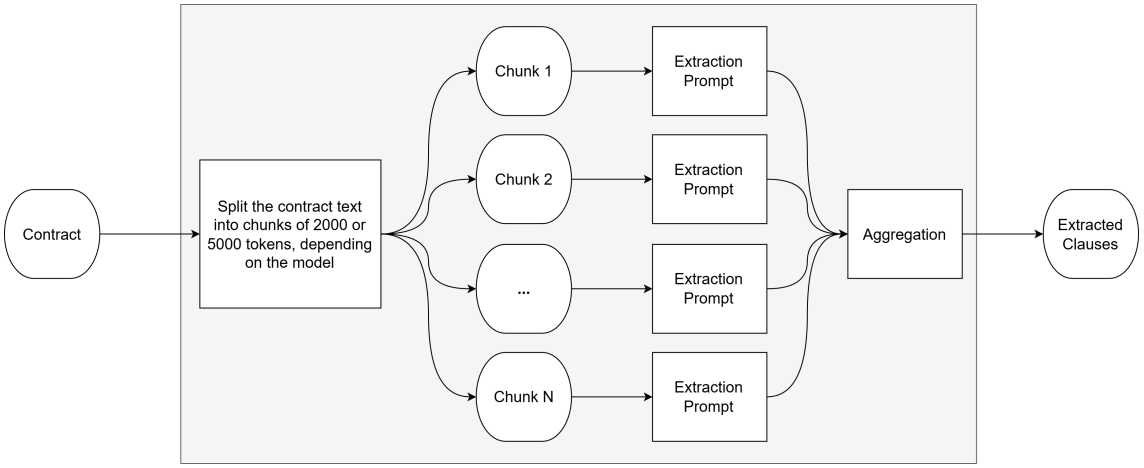

Figure 9. Extraction pipeline with reduced context approach.

3.5.3. Retrieval-based Context

The retrieval-based context method enhances extraction performance by focusing on the most relevant text segments. This approach involves breaking the document into smaller chunks and using Retrieval-Augmented Generation (RAG) [Lewis et al. 2020] to retrieve the most pertinent chunks for each extraction category. A crucial aspect of this method is the use of reference annotations to guide the retrieval process, ensuring that only the most relevant portions of the document are selected for extraction. These reference annotations consist of snippets that capture the essential semantics of each label category, effectively serving as examples for identifying relevant content.

In our approach, the reference annotations are selected from the CUAD database, specifically from contracts that are not part of our evaluation sample, by choosing those annotations in each category that exhibit the highest average cosine similarity [Gunawardana and Shani 2009] in their embeddings to others of the same category. This selection process ensures that the chosen annotations represent the category's semantics well, as they show a high degree of similarity to other category annotations. These reference annotations then guide the retrieval of relevant chunks for each category (Figure 11). For example, for the category "Audit Rights", we use an annotation such as the one shown in Figure 10.

> *"each party (the 'audited party') will, upon at least thirty (30) days' prior written request by the other party (the 'auditing party'), allow an independent certified public<omitted>accounting firm selected by the auditing party and reasonably acceptable to the audited party to audit such books and records at the audited party's premises to the extent necessary to verify the audited party's compliance or non-compliance with the provisions of this section 9 (or, in the case of company, section 5.4 [royalties]); provided, that: (a) any such audit is conducted during normal business hours and in a manner designed to not unreasonably interfere with the audited party's ordinary business operations; (b) audits may not occu r more frequently than once every twelve (12) months; and (c) each such audit may only cover the period commencing after the period covered by the last audit conducted pursuant to this section, if any."*

Figure 10. Reference annotation for retrieval of the "Audit Rights" category.

We use a retriever that employs a hybrid search combining BM25 [Robertson and Zaragoza 2009] and cosine similarity through embeddings, providing the retrieval of semantically relevant chunks. This method guarantees that the model is provided with contextually rich information that is conducive to accurate extraction. By utilizing both sparse and dense retrieval techniques, we capture different aspects of relevance, benefiting from the complementary strengths of each method [Gao et al. 2023].

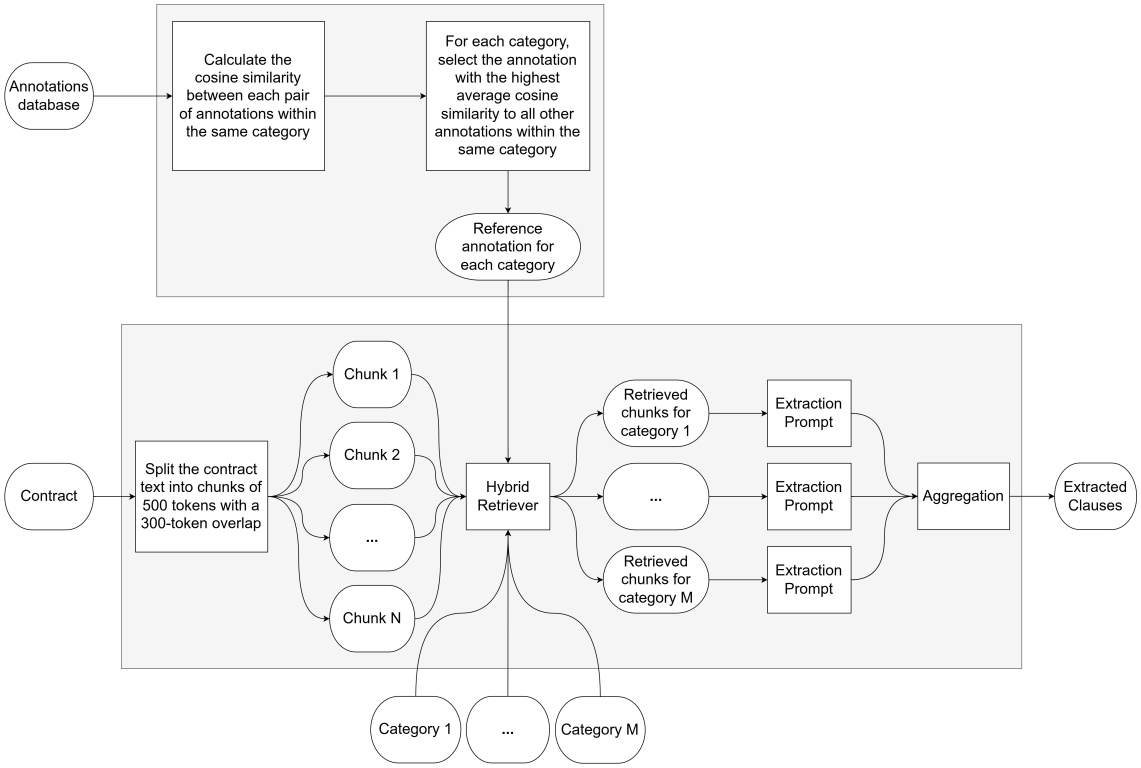

Figure 11. Extraction pipeline with retrieval-based context approach.

4. Experimental Setup

4.1. Task Structure

The purpose of this study is to evaluate the performance of large language models in the context of contract information extraction, focusing on both the breadth and depth of their capabilities.

To achieve this, we designed an experimental framework that tests the models under varying conditions of input contextualization and prompt engineering. The experiments were conducted across two levels of complexity: full-category extraction and subset-category extraction. These scenarios are further subdivided based on the prompt approach—zero-shot and zero-shot with output example—and the contextualization methods, namely total context, reduced context, and retrieval-based context.

In the full-category extraction scenario, the models are tasked with identifying and extracting all categories present in a given contract. This scenario leverages the comprehensive nature of the CUAD dataset, which includes a wide array of contract clauses, each requiring nuanced understanding and precise extraction. The complexity of this

task is amplified by the necessity for the models to parse through extensive and diverse contract language, distinguishing between relevant and irrelevant content to identify and extract various types of pertinent information.

For the subset-category extraction scenario, the focus narrows to three selected label categories: "Audit Rights", "License Grant" and "Cap on Liability". This setup allows us to closely examine the models' performance on a more targeted set of tasks, offering insights into their strengths and weaknesses when handling specific, high-priority clauses. Using the same categories as previous studies [Li et al. 2023] enables direct performance comparisons, thereby situating our findings within the broader research landscape.

The models are evaluated based on their performance in both scenarios, with specific attention to how the different input context methods and prompt engineering strategies influence the extraction accuracy. The Llama-3-70B-Instruct model, due to its smaller context window, is only evaluated under the reduced context approach. Moreover, it is also exclusively evaluated using the zero-shot prompt strategy in the scenario of extracting all categories.

On the other hand, the retrieval-based context approach is applied solely to the subset-focused extraction scenario and exclusively in conjunction with the zero-shot with output example prompting strategy, utilizing the GPT-3.5-Turbo and Claude-3.5-Sonnet models. Furthermore, the retrieval and extraction processes are conducted separately for each of the three label categories, ensuring that the retrieved context is specifically relevant to the category being extracted.

4.2. Parameters

In all experiments, we set the temperature parameter to 0 and the top_p parameter to 0.3 to minimize randomness in the outputs and maintain consistency across extractions. For GPT-3.5-Turbo and Llama-3-70B-Instruct, JSON outputs were enforced using the response_format parameter, while for Gemini-1.5-Pro, this was achieved through the response_mime_type parameter.

When utilizing the reduced context approach, input contracts were divided into chunks of 2,000 tokens for GPT-3.5-Turbo and Llama-3-70B-Instruct, while larger chunks of 5,000 tokens were used for Gemini-1.5-Pro and Claude-3.5-Sonnet due to their longer token context window.

For the retrieval-based context approach, texts were segmented into 500-token chunks with a 300-token overlap to maximize context retention between chunks. In this approach, we retrieved a maximum of six chunks from the contract: three with the highest BM25 scores and three with the highest cosine similarity scores. If a chunk appeared in the results of both retrieval techniques, only one copy was retained, potentially resulting in fewer than six chunks as input context for the models.

4.3. Evaluation Metrics

To evaluate the effectiveness of large language models in extracting information from contracts, we employed a comprehensive set of metrics addressing both qualitative and quantitative performance. Our primary focus was on precision, recall, and F1-score, which are particularly relevant due to the significant imbalance in our dataset, where relevant

clauses are vastly outnumbered by irrelevant ones. Precision measures the proportion of correctly extracted relevant clauses, recall assesses the proportion of all relevant clauses that were successfully extracted, and the F1-score, the harmonic mean of precision and recall, provides a balanced measure of performance. We calculated these metrics at the micro level, treating all extracted clauses as part of a single pool rather than categorizing them.

To evaluate the retriever component of our system, we used recall@k and precision@k. Recall@k reflects the percentage of relevant passages that appear among the top-k retrieved chunks, providing insight into the retriever's coverage and ability to capture the full range of relevant content. Precision@k, on the other hand, assesses the quality of the retrieved results by calculating the percentage of the top-k retrieved chunks that contain a relevant passage. Together, these metrics provide a comprehensive evaluation of the retriever's performance in both identifying and ranking relevant information.

Following the methodology inspired by [Han et al. 2024], we utilized the partial ratio similarity method from the RapidFuzz library [RapidFuzz 2024] to find matches between the extracted passages and the annotations and calculate true positives, false positives, and false negatives. This method, which evaluates the similarity between passages based on partial subsequences, offers a robust measure of match quality. The details of the method are shown in Algorithm 1. We set a predefined similarity threshold of 85 to determine if an extracted passage was correctly identified. This threshold ensures a high level of similarity between annotated and extracted passages, accounting for variations such as synonyms or minor rephrasing. It is important to note that this value was chosen empirically based on the dataset's specific characteristics and the task's nature, and the ideal threshold may vary depending on the use case. Before applying this method, we preprocessed the texts by normalizing case, removing duplicate spaces, and eliminating line breaks. Additionally, duplicated excerpts and annotations were excluded from the calculations to avoid inflating the metric results.

Algorithm 1 Partial-Ratio Matching

Input: The extracted passage e, the annotation a, the similarity threshold β.
Output: Return $True$ if the partial ratio similarity between the extracted passage e and the annotation a is greater than β, otherwise return $False$.

$score \leftarrow GetPartialRatioSimilarity(e, a)$
if $score > \beta$ **then**
    return $True$
else
    return $False$
end if

Efficiency metrics, such as average cost in dollars per contract and time in minutes, were also considered to evaluate the practical implications of deploying these models in real-world scenarios. The cost analysis focuses on the financial resources required to process each contract using the models, while time efficiency measures the duration of the information extraction process. To calculate these metrics, we accounted for the number of input and output tokens based on the tokenizers used by the LLMs, the cost per token associated with each model's API, and the response time of each API. These

metrics provide a holistic view of the trade-offs involved in employing LLMs for contract analysis, emphasizing both their accuracy and operational viability.

4.4. Baseline Methods

Evaluating large language models in the context of contract information extraction necessitates a robust comparison against established benchmarks. Our study utilizes three key baselines, each serving a distinct purpose in the overall assessment.

The first baseline is based on the work by [Hendrycks et al. 2021], which assesses models like BERT, RoBERTa, ALBERT, and DeBERTa across various contract types and label categories. This benchmark helps gauge our models' overall performance and accuracy in extracting relevant legal clauses across all categories.

The second baseline is informed by the study conducted by [Martin et al. 2024], which offers insights into the efficiency and cost-effectiveness of LLMs compared to human legal reviewers. Their findings serve as a reference point for evaluating our approaches' time efficiency and cost implications.

Lastly, we consider the performance results reported by [Li et al. 2023] as a baseline for the scenario involving the extraction of a reduced subset of label categories. Their results offer a critical benchmark for evaluating the efficiency of our proposed methods in handling specific, challenging categories within legal contracts.

5. Results and Discussion

In this section, we present the results of the experiments conducted with the LLMs at both levels of task complexity, using our sample of 100 contracts and combining the different approaches to prompts and input context. When comparing performance across different approaches and models, we prioritize configurations that achieve higher recall, as accurately identifying all relevant information is paramount in our task. In all tables, we boldly highlight the best results in each metric to facilitate comparison and evaluation.

5.1. Performance in Extracting All Categories with Total Context Approach

Table 1 shows the results of the models in the scenario of extracting all categories. GPT-3.5-Turbo achieved a recall of 0.39 and a precision of 0.83 with zero-shot with output example prompting, a performance inferior to Gemini-1.5-Pro, which achieved a recall of 0.67 and a precision of 0.70 with zero-shot with output example prompting, and Claude-3.5-Sonnet, which achieved a recall of 0.69 and a precision of 0.86 with zero-shot prompting. Across all models, there was a maximum difference of 0.03 in recall when using zero and zero-shot with output example prompting, indicating that the addition of the extraction example in the prompt had no relevant impact.

5.2. Performance in Extracting All Categories with Reduced Context Approach

In Table 2, we demonstrate how reducing the context provided to the models increases the information extraction performance. All models showed increased recall with both prompts, accompanied by a drop in precision. Claude-3.5-Sonnet with zero-shot with output example prompting stands out as the best-performing approach, with a recall of 0.77 and a precision of 0.66. Llama-3-70B-Instruct performed better than GPT-3.5-Turbo but did not surpass Gemini-1.5-Pro and Claude-3.5-Sonnet.

Table 1. Results of information extraction using zero-shot and zero-shot with output example prompts with total context approach for the all categories scenario.

Model	Prompt	Precision	Recall	F1-Score
GPT-3.5-Turbo	Without example	**0.89**	0.38	0.53
	With example	0.83	0.39	0.53
Gemini-1.5-Pro	Without example	0.79	0.64	0.71
	With example	0.70	0.67	0.69
Claude-3.5-Sonnet	Without example	0.86	**0.69**	**0.76**
	With example	0.75	0.68	0.71

Table 2. Results of information extraction using zero-shot and zero-shot with output example prompts with reduced context approach for the all categories scenario.

Model	Prompt	Precision	Recall	F1-Score
GPT-3.5-Turbo	Without example	**0.74**	0.50	0.59
	With example	0.66	0.47	0.55
Llama-3-70B-Instruct	Without example	0.67	0.59	0.63
Gemini-1.5-Pro	Without example	**0.74**	0.67	0.71
	With example	0.66	0.71	0.69
Claude-3.5-Sonnet	Without example	**0.74**	0.74	**0.74**
	With example	0.66	**0.77**	0.71

The results support our hypothesis that breaking the context into smaller parts improves the effectiveness of the models in the extracting the label categories, possibly due to decreased complexity and the ability to focus on more relevant information in a more individualized manner. However, the drop in accuracy suggests that the models may overestimate the relevance of certain passages, meaning that unwanted information may be extracted more frequently when the context is reduced. This phenomenon is in line with the findings of [Liu et al. 2024], which discusses the trade-off between model accuracy and the size of the input context, and shows that models perform worse at locating relevant information as the context increases.

It is interesting to note that the performance in all settings can be said to be superior to those presented by [Hendrycks et al. 2021]. Compared with the results in Figure 12, which shows the precision-recall curves of the main models used in their study, our models exhibit higher precision when analyzing scenarios with approximately the same recall. For example, Claude-3.5-Sonnet achieved a recall of 0.77 and a precision of 0.66, compared to DeBERTa-xlarge's recall of 0.80 and precision of 0.44 in CUAD, demonstrating a significant improvement in terms of precision. This increase in precision could indicate that the LLMs have a more remarkable ability to understand the content of contracts and filter out noisy information.

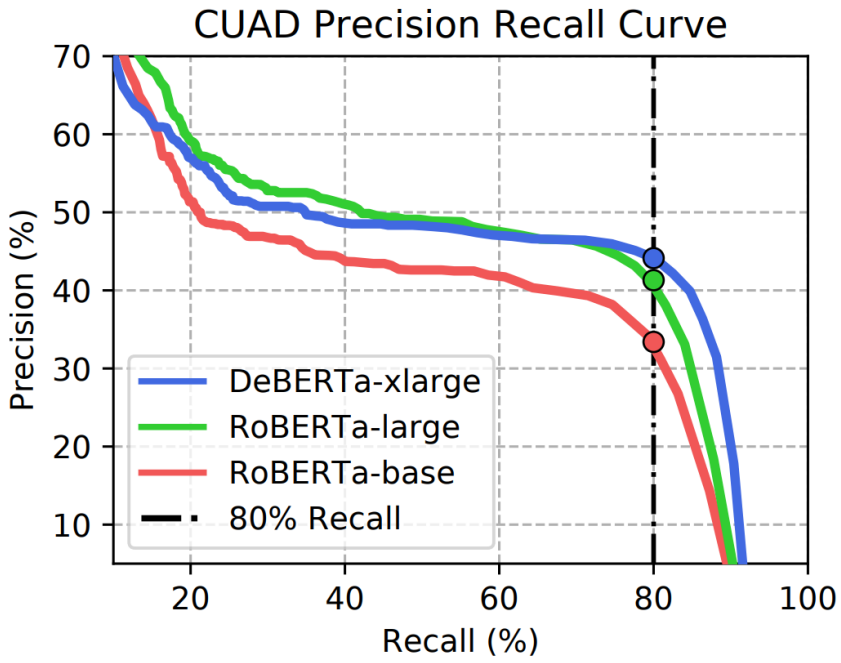

Figure 12. Precision-recall curves of the main models in the experiments conducted in the CUAD paper. Copyright by [Hendrycks et al. 2021].

5.3. Performance Analysis by Category

Figure 13 shows the recall of the best-performing approach, Claude-3.5-Sonnet with zero-shot with output example prompting and reduced context approach, for each of the 41 clause categories. The results reveal substantial variation in recall between different categories. Categories such as "Document Name" and "Governing Law" achieve recall values near 1.00, indicating that the model consistently extracts these clauses with high accuracy. On the other hand, categories like "Affiliate IP License-Licensor" and "Volume Restriction" exhibit significantly lower recall, with values close to 0.00.

Upon closer inspection, two of the categories with sharply reduced recall, "Volume Restriction" and "Uncapped Liability", were found to contain errors in labeling and description. For instance, the "Volume Restriction" clauses do not align with the definition provided by [Hendrycks et al. 2021], resulting in confusion in the prompt and hindering accurate extraction. Similarly, "Uncapped Liability" includes certain "Cap on Liability" clauses that were incorrectly labeled as "Uncapped Liability" in the dataset, contributing to the model's poor recall in this category.

In general, it can be seen that labels like "Document Name", which are short, clearly defined, and commonly structured, are easier for the model to identify. In contrast, categories such as "Affiliate IP License-Licensor" often involve more complex language and context, making accurate extraction more challenging. The relatively low recall in some these categories could also be attributed to the small representation of such clauses—only 3% of the contracts contain the "Affiliate IP License-Licensor" clause, for example. This limited presence may have resulted in these clauses being underrepresented in the training data of large language models, leading to challenges in their extraction.

Across all categories, the average recall is 0.64, with most categories achieving

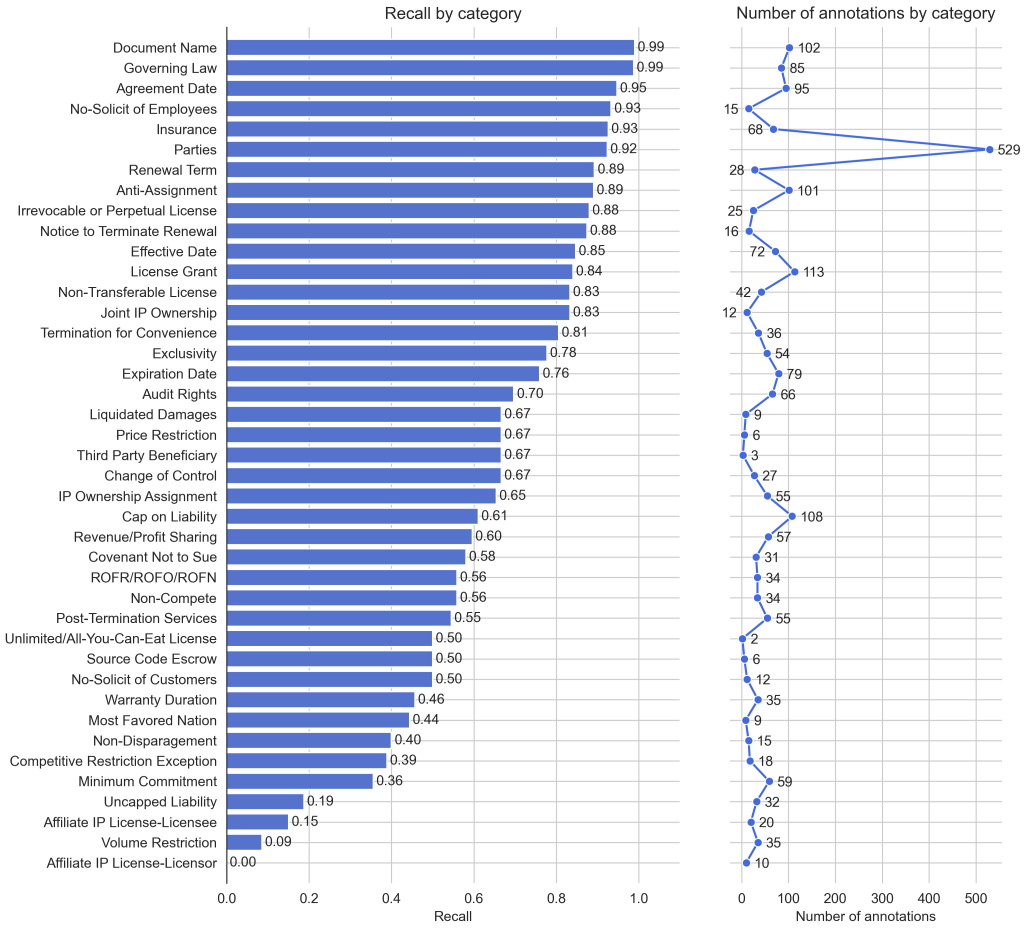

Figure 13. Recall of Claude-3.5-Sonnet with zero-shot with output example prompting and reduced context approach across all label categories. In addition to the recall by category, the number of annotations corresponding to each category is also shown.

values above 0.60. However, the variation in recall across categories—ranging from near-perfect scores to significantly lower values—indicates specific weaknessces in the model's ability to generalize across all clauses. This result suggests there is considerable room for improvement, particularly in handling more complex and ambiguous clauses.

One limitation of using generative models like Claude-3.5-Sonnet is the lack of granular control over the outputs. Unlike discriminative models, such as DeBERTa-xlarge used by [Hendrycks et al. 2021], which provide scores for each extraction, allowing thresholds to be adjusted for optimizing specific metrics like recall, generative models do not offer this flexibility. As a result, users cannot directly control the trade-off between precision and recall through threshold tuning, making it difficult to adapt the model's output to specific use cases where higher recall is prioritized. This limitation poses challenges for practical applications, especially in use cases where certain clause categories are critical and must be identified with high recall.

5.4. Performance in Extracting a Subset of Categories with Total and Reduced Context Approaches

We also analyze the performance of LLMs in extracting a subset of contract clauses—"Audit Rights", "License Grant", and "Cap on Liability"—using both zero-shot and zero-shot with output example prompts across total and reduced context approaches. The results are summarized in Table 3.

It is observed that, across all models, there is an overall increase in recall when comparing the reduced set of clauses to the broader clause extraction results discussed in Sections 5.1 and 5.2. However, this improvement in recall is generally accompanied by a decrease in precision, reflecting the well-established trade-off between recall and precision in information extraction tasks. This trend is better illustrated in Figure 14, which contrasts the recall for each category between the scenario with the entire clause set and the scenario with the reduced clause set for the best approaches of each model.

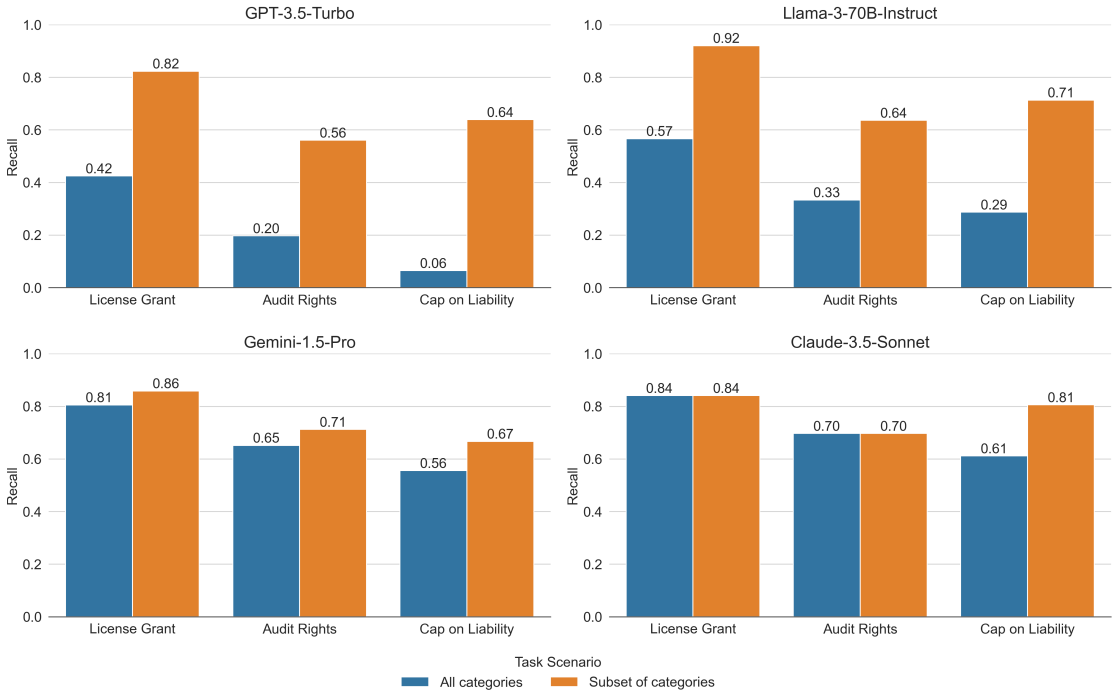

Figure 14. Comparison of the recall of each model for the clauses "License Grant", "Audit Rights" and "Cap on Liability" between the best approach in the scenario with all categories and the scenario with only these three categories.

GPT-3.5-Turbo exhibited the most significant improvement in recall when comparing both complexity scenarios. Notably, its recall for "Cap on Liability" increased markedly, from a low 0.06 in the full-category extraction scenario to 0.64 in the subset scenario. This result highlights the model's ability to extract more focused information when tasked with fewer, more specific clauses, particularly for those that are more challenging to detect in the full-category scenario.

In contrast, Claude-3.5-Sonnet maintained a more consistent recall across the full-category and subset extraction scenarios, showing stable performance for "License Grant"

Table 3. Results of information extraction using zero-shot and zero-shot with output example prompts with both total and reduced context approaches for the subset of categories scenario.

Model	Prompt	Context Approach	Audit Rights			License Grant			Cap on Liability			Overall		
			Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score	Precision	Recall	F1-Score
GPT-3.5-Turbo	Without example	Total context	**0.93**	0.30	0.46	0.61	0.46	0.53	0.86	0.53	0.66	0.73	0.45	0.56
		Reduced context	0.71	0.56	0.63	0.45	0.82	0.58	0.51	0.64	0.57	0.50	0.69	0.58
	With example	Total context	0.60	0.32	0.42	**0.89**	0.48	0.62	0.70	0.61	0.65	0.76	0.49	0.60
		Reduced context	0.65	0.58	0.61	0.61	0.75	0.67	0.43	0.66	0.52	0.54	0.68	0.60
Llama-3-70B-Instruct	Without example	Reduced context	0.54	0.68	0.60	0.33	**0.92**	0.49	0.49	0.69	0.57	0.40	0.78	0.53
	With example	Reduced context	0.51	0.64	0.57	0.50	**0.92**	0.65	0.43	0.71	0.54	0.48	0.78	0.59
Gemini-1.5-Pro	Without example	Total context	0.49	0.60	0.54	0.49	0.68	0.57	0.73	0.56	0.64	0.55	0.62	0.58
		Reduced context	0.39	**0.71**	0.51	0.40	0.86	0.54	0.64	0.67	0.65	0.45	0.75	0.56
	With example	Total context	0.53	0.59	0.56	0.55	0.73	0.63	0.73	0.56	0.63	0.59	0.63	0.61
		Reduced context	0.51	0.70	0.59	0.43	0.76	0.55	0.64	0.66	0.65	0.51	0.71	0.59
Claude-3.5-Sonnet	Without example	Total context	0.82	0.42	0.56	0.73	0.68	**0.71**	**0.88**	0.65	**0.75**	**0.79**	0.61	**0.69**
		Reduced context	0.63	0.67	0.65	0.53	0.87	0.65	0.63	0.77	0.69	0.57	0.78	0.66
	With example	Total context	0.89	0.33	0.49	0.79	0.65	**0.71**	0.63	0.63	0.63	0.73	0.57	0.64
		Reduced context	0.73	0.70	**0.71**	0.62	0.84	**0.71**	0.46	**0.81**	0.59	0.56	**0.79**	0.66

and “Audit Rights”. Although there was an improvement in recall for “Cap on Liability” of 0.20, the model’s overall performance remained relatively stable, demonstrating its ability to handle large and reduced sets of categories without significant loss of precision. Gemini-1.5-Pro similarly displayed stable performance across both scenarios, maintaining high recall in both cases.

This stability aligns with their high performance in the “Needle in a Haystack” [Kamradt 2023] task, where both Claude-3.5-Sonnet and Gemini-1.5-Pro achieved exceptional recall rates, as high as 99.7%, in scenarios with their maximum token context windows [Anthropic 2024b, Gemini Team et al. 2024]. The models’ ability to maintain high recall even in complex, contextually dense tasks reflects their proficiency in managing large amounts of information while extracting specific, relevant details.

Llama-3-70B-Instruct, while a smaller model compared to Claude and Gemini, performed competitively in the reduced context scenario, especially in the zero-shot with output example setting. Its overall recall was almost as high as that of Claude-3.5-Sonnet, even surpassing it in “License Grant” with a recall of 0.92. However, its precision was lower, indicating a higher rate of false positives.

Interestingly, the zero-shot with output example prompting approach did not uniformly outperform zero-shot prompting. For example, in Llama-3-70B-Instruct with the reduced context approach, zero-shot with output example prompting led to a marginal improvement in recall for “Cap on Liability”, but this gain was inconsistent across other categories. The same can be said about the precision in these cases. This result highlights our perception of the prompts in the previous scenario with all categories.

While these findings demonstrate the value of reduced complexity scenarios for improving recall, especially for more difficult clauses like “Cap on Liability”, it is important to note that none of the evaluated models surpassed the performance of the pipeline with RoBERTa-base detailed in [Li et al. 2023]. Despite improvements in recall, the evaluated LLMs still fall short of domain-specific models in overall F1 performance. Fine-tuning or higher-level prompting techniques, such as prompt chaining [Sun et al. 2024, Wu et al. 2022], could offer a path forward to bridge this performance gap.

5.5. Performance in Extracting a Subset of Categories with Retrieval-Based Context Approach

In investigating the effectiveness of Retrieval-Augmented Generation in extracting specific categories, we observed that RAG demonstrates superior performance compared to approaches using the models’ maximum context. Although it shows inferior performance to approaches using reduced context, it still achieves very close results, as shown in Table 4.

The 77% recall of the retrieved chunks indicates that there is significant room for improvement in this approach, as 23% of the relevant chunks were not retrieved, which directly impacts the precision of information extraction. Notably, the recall of the Claude-3.5-Sonnet model was very close to that of the retriever, achieving a recall rate of 75%. This result suggests that, when relevant excerpts were included in the provided context, Claude-3.5-Sonnet successfully extracted approximately 97% of them. These results highlight the model’s effectiveness in extracting information from the context provided

Table 4. Results of information extraction in the scenario with the subset of categories using the retrieval-based context approach.

Model	Prompt	Retriever's precision@6	Retriever's recall@6	Precision	Recall	F1-Score
GPT-3.5-Turbo	With example	0.19	0.77	**0.69**	0.62	**0.65**
Claude-3.5-Sonnet	With example	0.19	0.77	0.53	**0.75**	0.62

while emphasizing the need to enhance the retriever's capability to capture all pertinent chunks.

This suggests that RAG could be a viable option for extracting specific categories in contracts. This approach allows the system to retrieve and process only the most relevant chunks of text without significant performance losses. It is particularly useful when processing the entire contract is not feasible, offering a balance between efficiency and accuracy in contract analysis.

5.6. Comparison of Time and Cost

Table 5. Average cost and time per contract for the best approaches of each model in the extraction scenario with all categories.

Model	Prompt	Context Approach	Cost (USD)	Time (Min)
GPT-3.5-Turbo	Without example	Reduced context	0.0138	0.614
Llama-3-70B-Instruct	Without example	Reduced context	0.0133	0.205
Gemini-1.5-Pro	With example	Reduced context	0.0872	0.738
Claude-3.5-Sonnet	With example	Reduced context	0.1084	0.507

Table 5 compares the time and cost of the best approaches of each model in the scenario of extracting all categories. The results reveal a significant difference in cost and efficiency between the models. Compared to human reviewers, LLMs also present significantly lower costs and faster processing times. For example, the average cost per document for a junior lawyer was $74.26 and took an average of 56.17 minutes per document [Martin et al. 2024]. In contrast, Llama-3-70B-Instruct completed the task with an average cost of $0.0133 and in 0.205 minutes. Similarly, the best-performing model, Claude-3.5-Sonnet, had an average time of 0.507 minutes and an average cost of $0.1084 per document, far surpassing the efficiency of human reviewers.

In comparison with the second scenario (Table 6), where the models focus on a subset of categories, the differences in cost and time remained evident. Llama-3-70B-Instruct continued to show cost efficiency with an average cost of $0.0084 per document, and the introduction of RAG approaches for GPT-3.5-Turbo and Claude-3.5-Sonnet further reduced their costs to $0.0058 and $0.0334 per document, respectively. These findings highlight the economic advantages of LLMs, especially when considering the potential for improvement in information extraction and the use of more refined contextualization approaches.

Table 6. Average cost and time per contract for the best approaches of each model in the extraction scenario with the subset of categories.

Model	Prompt	Context Approach	Cost (USD)	Time (Min)
GPT-3.5-Turbo	Without example	Reduced context	0.0069	0.202
	With example	Retrieval-based	0.0058	0.077
Llama-3-70B-Instruct	With example	Reduced context	0.0084	0.118
Gemini-1.5-Pro	Without example	Reduced context	0.0421	0.257
Claude-3.5-Sonnet	With example	Reduced context	0.0398	0.141
		Retrieval-based	0.0334	0.134

6. Conclusion

By comparing multiple models, prompts, and input context methods, we have provided a comprehensive analysis of how LLMs can be optimized for contract review tasks. This work fills a critical gap by offering insights into the relative effectiveness of different strategies, particularly in the context of legal text, which is often dense and complex.

Our findings reveal that the choice of input context and prompt engineering substantially affects the models' performance. Notably, the use of a reduced context showed an increase in recall for information extraction tasks, suggesting that simplifying the input context can help models focus more on relevant details. However, this came at the cost of reduced precision, highlighting the trade-offs in managing input size versus extraction accuracy. The results also showed that models like Claude-3.5-Sonnet and Gemini-1.5-Pro outperformed others in scenarios involving a more complex or reduced set of categories, indicating their robustness in handling nuanced and specific information extraction tasks.

Another key finding of our analysis is the variation in model performance across different clause categories. Clauses that are simpler and more structured tended to achieve strong recall, while those involving more complex or ambiguous language presented greater challenges, leading to significantly lower recall. This variation highlights the difficulties LLMs face with more complex or nuanced contract language. Interestingly, smaller models, such as GPT-3.5-Turbo Llama-3-70B-Instruct, performed competitively, especially when the task was simplified or narrowed down to a smaller set of clauses. This suggests that even less resource-intensive models can achieve strong results when the task is properly framed, emphasizing the importance of prompt engineering and task-specific context in optimizing performance.

A key challenge we faced was financial limitations, which restricted the inclusion of state-of-the-art models like GPT-4o in our experiments and made using the complete CUAD base unattainable. This limitation underscores the need for cost-efficient approaches in deploying large language models for practical applications. Open-source alternatives, such as Llama-3-70B-Instruct, emerge as promising candidates, offering a solid balance between cost and performance. However, additional investigation is essential to fully explore strategies that can achieve optimal performance without incurring high expenses.

Future research could explore more sophisticated prompting techniques, building on our proposed idea of breaking down the task into smaller, more manageable components that collectively provide the final answer. Techniques such as prompt chaining, where a sequence of prompts is used to refine the extraction process progressively, hold considerable potential. This approach could enable more targeted information retrieval, improving the extracted information's precision and recall.

Moreover, advancements in retrieval-based techniques could also enhance the accuracy and relevance of the extracted information. Exploring more advanced retrieval methods or employing neural retrievers capable of better understanding legal semantics could provide a more refined filtering process. These future directions could lead to more accurate and efficient systems for contract analysis, further bridging the gap between current LLM capabilities and the specific demands of the legal domain.

In conclusion, this study represents a significant step in understanding and optimizing LLMs for contract information extraction. The insights gained from our experiments provide valuable guidance for future research and practical applications, suggesting that with continued refinement, LLMs can play a crucial role in automating complex legal tasks.

References

Anesa, P. (2007). Vagueness and precision in contracts: a close relationship. *Linguistica e Filologia*, 24:7–38.

Anthropic (2024a). The claude 3 model family: Opus, sonnet, haiku. Technical report, Anthropic.

Anthropic (2024b). Claude 3.5 sonnet model card addendum. Technical report, Anthropic.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Chalkidis, I., Androutsopoulos, I., and Michos, A. (2017). Extracting contract elements. In *Proceedings of the 16th Edition of the International Conference on Articial Intelligence and Law*, ICAIL '17, page 19–28, New York, NY, USA. Association for Computing Machinery.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Guo, Q., Wang, M., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. *arXiv*, abs/2312.10997.

Gemini Team et al. (2024). Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*, abs/2403.05530.

Gunawardana, A. and Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *J. Mach. Learn. Res.*, 10:2935–2962.

Han, R., Yang, C., Peng, T., Tiwari, P., Wan, X., Liu, L., and Wang, B. (2024). An empirical study on information extraction using large language models. *arXiv*, abs/2305.14450.

Hendrycks, D., Burns, C., Chen, A., and Ball, S. (2021). Cuad: An expert-annotated nlp dataset for legal contract review. In Vanschoren, J. and Yeung, S., editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Kamradt, G. (2023). Needle in a haystack - pressure testing llms. `https://github.com/gkamradt/LLMTest_NeedleInAHaystack`. Accessed: 2024-09-24.

Leivaditi, S., Rossi, J., and Kanoulas, E. (2020). A benchmark for lease contract review. *arXiv*, abs/2010.10386.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Li, S. (2017). A corpus-based study of vague language in legislative texts: Strategic use of vague terms. *English for Specific Purposes*, 45:98–109.

Li, Z., Guha, N., and Nyarko, J. (2023). Don't use a cannon to kill a fly: An efficient cascading pipeline for long documents. In *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, ICAIL '23, page 141–147, New York, NY, USA. Association for Computing Machinery.

Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Martin, L., Whitehouse, N., Yiu, S., Catterson, L., and Perera, R. (2024). Better call gpt, comparing large language models against lawyers. *arXiv*, abs/2401.16212.

Meta (2024). Introducing meta llama 3: The most capable openly available llm to date. `https://ai.meta.com/blog/meta-llama-3/`. Accessed: 2024-07-10.

Minaee, S., Mikolov, T., Nikzad, N., Chenaghlu, M., Socher, R., Amatriain, X., and Gao, J. (2024). Large language models: A survey. *arXiv*, abs/2402.06196.

OpenAI (2024a). Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`. Accessed: 2024-07-14.

OpenAI (2024b). New embedding models and api updates. `https://openai.com/index/new-embedding-models-and-api-updates/`. Accessed: 2024-07-14.

OpenAI et al. (2023). Gpt-4 technical report. *arXiv*, abs/2303.08774.

PyMuPDF (2024). Pymupdf 1.24.10 documentation. `https://pymupdf.readthedocs.io/en/latest/`. Accessed: 2024-09-02.

RapidFuzz (2024). Rapidfuzz 3.10.0 documentation: partial_ratio. https://rapidfuzz.github.io/RapidFuzz/Usage/fuzz.html#rapidfuzz.fuzz.partial_ratio. Accessed: 2024-09-02.

Robertson, S. and Zaragoza, H. (2009). The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Savelka, J. and Ashley, K. D. (2023). The unreasonable effectiveness of large language models in zero-shot semantic annotation of legal texts. *Frontiers in Artificial Intelligence*, 6.

Sun, S., Yuan, R., Cao, Z., Li, W., and Liu, P. (2024). Prompt chaining or stepwise prompt? refinement in text summarization. *arXiv*, abs/2406.00507.

Wang, S., Sun, X., Li, X., Ouyang, R., Wu, F., Zhang, T., Li, J., and Wang, G. (2023). Gpt-ner: Named entity recognition via large language models. *arXiv*, abs/2304.10428.

Wei, X., Cui, X., Cheng, N., Wang, X., Zhang, X., Huang, S., Xie, P., Xu, J., Chen, Y., Zhang, M., Jiang, Y., and Han, W. (2023). Zero-shot information extraction via chatting with chatgpt. *arXiv*, abs/2302.10205.

Wu, T., Jiang, E., Donsbach, A., Gray, J., Molina, A., Terry, M., and Cai, C. J. (2022). Promptchainer: Chaining large language model prompts through visual programming. *arXiv*, abs/2203.06566.

Zhou, W., Zhang, S., Gu, Y., Chen, M., and Poon, H. (2023). Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv*, abs/2308.03279.