Enhancing Information Extraction with Large Language Models: A Comparison with Human Annotation and Rule-Based Systems in a Real Estate Case Study

Renzo Alva Principe[1,2], Nicola Chiarini[2], Marco Viviani[1]

[1]Università degli Studi di Milano-Bicocca
[2]Datasinc
renzo.alvaprincipe@unimib.it, nicola.chiarini@datasinc.it, marco.viviani@unimib.it
Correspondence: renzo.alvaprincipe@unimib.it

Abstract

Information Extraction (IE) is a key task in Natural Language Processing (NLP) that transforms unstructured text into structured data. This study compares human annotation, rule-based systems, and Large Language Models (LLMs) for domain-specific IE, focusing on real estate auction documents. We assess each method in terms of accuracy, scalability, and cost-efficiency, highlighting the associated trade-offs. Our findings provide valuable insights into the effectiveness of using LLMs for the considered task and, more broadly, offer guidance on how organizations can balance automation, maintainability, and performance when selecting the most suitable IE solution.

1 Introduction

Information Extraction (IE) is a fundamental task in *Natural Language Processing* (NLP), enabling the transformation of unstructured text into structured data. IE involves identifying and extracting relevant information, such as entities, relationships, and events, and organizing it so that machines can process and analyze it effectively (Grishman, 2015; Piskorski and Yangarber, 2013). Many industries, such as finance, healthcare, and legal services, rely on IE to process large volumes of documents and extract critical information.

In this context, *domain-specific* IE poses further challenges compared to general-purpose IE (Hahn and Oleynik, 2020; Yamamoto et al., 2008; Yuan and Lipizzi, 2023; Trewartha et al., 2022; Zadgaonkar and Agrawal, 2021); documents often feature specialized terminology, structured content, considerable length, and various formats that require expert knowledge for accurate processing. *Human annotation*, while serving as the gold standard for accuracy, is costly, time-consuming, and prone to inconsistencies, especially with lengthy or complex documents. Traditional *rule-based systems* have been widely adopted in such settings for their precision, deterministic behavior, and interpretability. However, they require significant effort to develop and maintain, particularly when adapting to new document types. The adoption of *Large Language Models* (LLMs) and prompt engineering can offer a flexible alternative for reducing reliance on manually crafted rules. However, despite advancements in AI-driven methods, the transition to AI continues to present significant challenges. Widespread adoption remains hindered by several factors, including high implementation costs, integration complexities, data privacy concerns, and a lack of expertise (Alhosani and Alhashmi, 2024; de Bellefonds et al., 2024; Jiang et al., 2023; Mayer et al., 2025). In business environments, where performance, cost, and maintainability must be balanced, careful evaluation of IE approaches is key to selecting effective solutions.

To address this challenge and examine key trade-offs, we conduct an empirical comparison of human annotation, rule-based extraction, and LLM-based extraction for domain-specific IE tasks. We focus on the *real estate* domain in the Italian context, using auction documents to extract structured information—such as cadastral data and asset descriptions—crucial for business operations. We develop rule-based and LLM-driven models and evaluate their performance against a human-annotated ground truth. This study systematically compares the three approaches in terms of accuracy, scalability, and cost-efficiency, providing insights that can inform IE adoption across various industries. Our findings aim to assist organizations in balancing automation, maintainability, and performance when selecting the best IE solution for their needs.

2 Related Work

This section reviews the aforementioned IE strategies—human annotation, rule-based, and LLM-based methods—highlighting their applications, strengths, limitations, and key trade-offs across distinct domains.

Several studies have examined the behavior and performance of *human annotators*. Chau et al. (2020) examine the effects of self-review and peer-review processes among annotators in the real-estate domain, highlighting issues related to inter-annotator agreement and uncertainty. Hochheiser et al. (2016) focus on the pharmaceutical domain and find that crowdsourcing annotators can provide a reliable approximation of expert annotations. Similarly, Jin et al. (2023) introduce methods to enhance crowdsourced annotation—such as gamification—achieving expert-level accuracy in the medical domain. These studies underscore challenges such as subjectivity and the need for domain expertise, both of which are crucial for reliable annotation. They suggest that non-expert annotators can often approximate expert performance. However, they do not specifically evaluate annotator performance on tasks involving long documents, where the volume of text may significantly affect annotation quality.

Rule-based systems represent one of the earliest approaches to IE, relying on handcrafted patterns and domain-specific rules. These systems typically employ regular expressions, as in the seminal work by Hearst (1992), and in some cases support context-free constructs (Freitag et al., 2022b). They are also frequently implemented as frameworks (Cunningham et al., 2002; Valenzuela-Escárcega et al., 2020; Kluegl et al., 2016; Azimjonov and Alikhanov, 2018; Chiticariu et al., 2010; Manning et al., 2014). Rules are effective due to their transparency and the lack of need for training data. However, they struggle to generalize to minor input variations and are sensitive to noise and linguistic diversity (Waltl et al., 2018). While rule-based systems can initially boost precision and recall, capturing all linguistic nuances requires excessive manual effort as input complexity increases (Waltl et al., 2018). Although the rise of machine learning techniques has largely overshadowed rule-based approaches, they remained widely used in industry until recently (Chiticariu et al., 2013) and are still employed in academic research today—particularly in the early stages of rapid prototyping (Freitag et al., 2022a,b).

Since the introduction of *Large Language Models*, many studies on IE have emerged, generally falling into two categories: *training-based* and *training-free*. The former involves adapting LLMs to specific tasks by *fine-tuning* their parameters using domain-specific labeled datasets. This process enhances the model's ability to accurately identify and extract structured information from unstructured text. For example, *DeepStruct* introduced structural pre-training on task-agnostic corpora to improve LLMs' structural understanding (Wang et al., 2022). Similarly, GIELLM fine-tuned LLMs on mixed datasets to exploit mutual reinforcement effects, enhancing performance across multiple tasks (Gan et al., 2023).

In contrast, training-free approaches rely on *prompt engineering*—a technique that guides LLM behavior using task-specific prompts, without modifying model parameters. For instance, Zhang et al. (2023) highlight the gap between instruction-tuned LLMs and the structured output requirements of IE. QA-style prompting helps bridge this gap. Other methods, such as *PromptNER* (Ashok and Lipton, 2023), guide LLMs to generate explanations for entity extraction, while *ProgGen* (Heng et al., 2024) promotes self-reflection to improve output quality.

LLM-based techniques have been widely applied across domains such as legal (Breton et al., 2025; Ribeiro de Faria et al., 2025; Hussain and Thomas, 2024), medical (Yang et al., 2022; Xu et al., 2024), and fintech (Rajpoot and Parikh, 2023a,b). Recently, studies have compared LLM-based with rule-based systems (Wang et al., 2024; Thakkar et al., 2024) and human annotators (Gu et al., 2025; Pavlovic and Poesio, 2024). However, no study has so far comparatively evaluated the three strategies together. Additionally, these works focus on performance comparisons, overlooking cost analysis, development efforts, and the length of documents.

3 The Real Estate Domain

This section provides an overview of the domain we focus on in this study, namely the *real estate* sector. In particular, the term "real estate" refers to the ownership, management, and trade of properties, including land, buildings, and other structures. In this domain, data primarily consists of information about properties, such as their location, dimensions, ownership details, market value, and legal status. Within the Italian cadastral system, properties are uniquely identified through *cadastral coordinates* organized in a hierarchical structure. These include *foglio* (sheet), *particella* (parcel), and, where applicable, *subalterno* (subunit), which together provide a standardized reference for each property. In the real estate market, properties are sold privately, through agencies, or at auctions. This study focuses on *real estate auctions*, which play a key role in judicial sales and debt recovery, with the goal of extracting structured data from auction documents.

3.1 Real Estate Auctions

Auction notices generally commence with an *introductory section* outlining court details, procedural information, and the parties involved. Subsequently, the documents present a comprehensive *description of the assets* included in each lot. They specify pricing and sale conditions, including base prices and terms, as well as transfer requirements and buyer obligations.

However, despite their structured format, extracting cadastral coordinates from auction notices presents considerable challenges. These difficulties arise from the use of abbreviations, alternative nomenclature, simplifications, and typographical errors in key terms. Equally problematic is the length of the documents, which makes manual extraction both time-consuming and complex. Moreover, each component of the coordinate is meaningful only when accurately associated with the others; misalignment with coordinates from a different property can result in incorrect identification.

3.2 Task Description

In this study, we focus on monitoring auctioned properties by extracting structured and precise information from auction documents, including *cadastral coordinates* and *asset descriptions*. To accomplish this, the relevant information includes *lotti* (lots), with each property identified by its cadastral coordinates (*foglio*, *particella*, and *subalterno*). Properties are classified as either *terreno* (land) or *fabbricato* (building). In this process, we aim to uniquely identify each property by combining all metadata into a 5-tuple, which is then provided in a structured format, i.e., `property = <lotto, foglio, particella, sub, type>`, where the `sub` is an optional coordinate that is mandatory only for the `fabbricato` type. For each auction notice document, we anticipate an array of JSON objects containing only the cadastral coordinates of the properties available for sale.

4 Information Extraction Solutions

In this section, we detail the human annotation process, the rule-based methods, and the LLM-based approaches we implemented for the comparative evaluation of the three extraction techniques.

4.1 Human Annotation

The human annotation process was conducted by a fixed pool of 10 annotators recruited through our internal Datasinc network, selected via brief interviews or referrals to ensure stable annotation quality. While no formal domain expertise was required, basic reasoning ability and attention to detail were expected.

Annotators underwent a brief onboarding phase consisting of a short demo session without additional documentation. They then carried out the annotation task using a dedicated internal platform (*REcognition*), which guided them step-by-step and provided built-in quality controls at multiple levels. These included automated consistency checks within the platform, as well as external validation through the Italian land registry system (*Sister*) and heuristic cross-checks to reject implausible or inconsistent data entries.

Thanks to the platform's intuitive guidance and minimal training, annotators were able to complete the task efficiently. Consequently, each auction notice was assigned to a single annotator without overlap, so *Inter-Annotator Agreement* (IAA) was not measured. Compensation was tied to the number of extracted coordinates, regardless of their relevance—for example, mentions of neighboring properties outside the project scope were also counted.

4.2 Rule-Based Information Extraction

We developed the rule-based IE engine by leveraging *Parsing Expression Grammars* (PEGs) (Laurent and Mens, 2015), a formalism for defining language syntax. PEGs are conceptually similar to *Context-Free Grammars* (CFGs) but differ in key aspects that make them particularly well-suited for parsing tasks. Unlike CFGs, PEGs provide deterministic parsing through ordered choice: when multiple parsing options are available, only the first match is selected, eliminating ambiguity. This approach ensures that any input yields either a single valid parse tree or none at all, thereby enhancing efficiency. PEGs also surpass regular expressions in expressive power by supporting more complex constructs, including recursion and hierarchical structures, rather than being limited to flat, non-recursive patterns. To implement this approach, we utilize the open-source *Parsimonious* library.[1]

[1] https://github.com/erikrose/parsimonious

4.2.1 Core Rules

This rule set is designed to capture the *core elements* of the task when they appear in their most straightforward form, delegating any complexities to other rule sets. Examples (Table 1) include:

- **Simple elements**: *key–value* pairs that associate a keyword with a numeric value, such as *lotto*, *foglio*, *particella*, and *sub*;
- **Complex elements**: Structures like a *tupla*, which groups the coordinates of a property, or a *selling item*, which represents a lot and the properties it includes. These elements allow us to represent the entire document as a collection of selling items;
- **Alternative names**: Variations of keywords, including abbreviations and differences in text formatting, such as word breaks introduced by carriage returns;
- **Other details**: Enumerators and separators (e.g., commas, hyphens, the Italian conjunction 'e', i.e., 'and', and slashes) used when multiple properties share the same *foglio* and *particella* but differ in the *sub*.

Simple elements	**Lotto**: lotto 2 **Foglio**: foglio 46 **Particella**: particella 24 **Sub**: sub 9
Complex elements	**Tupla**: Foglio 46, particella 24, sub 9 **Selling item**: LOTTO 3: Terreno agricolo a Brescia, foglio 25, p.lla 71, appartamento al Fg 46, p.lla 2440, sub 9
Alternative names	**Foglio**: fog., fgl., fg, f.lio, f.io, f., fol, foglio, ... **Particella**: prt., part.lla, part, p.c., p/lla, ptc, mappale, mapp., mappale, m.n., p., ... **Sub**: subalterno, subb., sub., ...
Enumerators	nr, n.ro, n.ri, n., n°
Separators	',' / '-' / 'e' / '/'

Table 1: Examples of core elements.

4.2.2 Normalization Rules

In auction documents, when multiple properties are associated with the same *foglio* and *particella*, and extensive details are provided for each property, a list format is employed. This approach helps to organize information efficiently, avoiding information redundancy. To ensure compatibility with the core rules and preserve contiguous cadastral coordinates while excluding irrelevant tokens (*out-tokens*), a normalization step is applied. This process ensures that only relevant information is retained. Table 2 illustrates an example of the text before and after normalization.

Original Text	**Normalized Text**
Fg. 46, p.lla 24: - sub 9 - Piano 5-6 - Cat. A/2 (...) - sub 5 - Piano S2 - Cat. C/6 (...) - sub 37 - Piano SI - Cat. C/6 (...)	Fg 46, p.lla 24, sub 9 - Piano 5 - Cat. A/2 (...) Fg 46, p.lla 24, sub 5 - Piano S2 - Cat. C/6 (...) Fg 46, p.lla 24, sub 37 - Piano SI - Cat. C/6 (...)

Table 2: The effect of normalization rules.

4.2.3 Ambiguity Filtering Rules

Normalization helps exclude a specific type of out-tokens, although various cases exist. Out-tokens are not always easy to filter. The simplest cases involve tokens that appear before or after the relevant element, such as in "`Identificazione catastale: fg 16 p.lla 1268 sub. 3, rendita 140,73 Euro`", where leading and trailing irrelevant information can be easily ignored. However, more complex scenarios, like ambiguous cadastral coordinates, present greater challenges. For instance, in "`Foglio 60 particella 44, 45 sub 1, 2`", it is unclear which properties are being referenced, as it is not evident which *particella* each *sub* belongs to. A naive rule-based system might incorrectly extract `"foglio":60, "particella":44, "sub":1`. To address this, we developed specialized rules to identify and exclude ambiguous cases, preventing premature matches by the core rules.

```
tupla_wrong = wrong_1 / wrong_2 /
              wrong_3
wrong_1 = foglio jollies (map_list_nums/
          map_list_maps) jollies sub
wrong_2 = (foglio_list_nums/
          foglio_list_fogs) jollies map
wrong_3 = (foglio_list_nums/
          foglio_list_fogs) jollies map
          jollies sub
```

Listing 1: A simplified version of the ambiguity filtering rules.

Listing 1 illustrates the rules for handling ambiguous tuple matching. The primary rule, `tupla_wrong`, defines the possible ambiguous tuples, with specific rules for each case. In particular, `wrong_1` matches a *fabbricato* with multiple *particella*, `wrong_2` matches a *terreno* with multiple *foglio*, and `wrong_3` matches multiple *fabbricato* with multiple *foglio*.

4.2.4 Master Rules

This rule set defines the high-level document structure and acts as the backbone for the previously discussed rule sets. Listing 2 presents a simplified excerpt from the rule-based system entry point, where `avviso` (an auction document) is defined as a collection of multiple `lotto` instances. Each `lotto` is identified by its unique keyword and number, followed by either ambiguous or legitimate tuples, and continues capturing text until a new `lotto` is encountered. Finally, the set of tuples is described by its *foglio*, *particella*, and *sub* elements. Additionally, PEG rule consumption is greedy to ensure determinism and avoid ambiguity, as shown by placing `tupla_wrong` before regular tuples to prevent ambiguous extractions.

```
avviso      = (lotto/jolly) +
lotto       = (ord_lotto/lotto_num/
               lotto_unico/lotto_ord)?
               (tupla_wrong/tupla_mix/
               &ord_lotto/&lotto_num/
               &lotto_unico/&lotto_ord /
               jolly)+
ord_lotto   = ws ordinale ws lotto_tok
               comma?
lotto_ord   = lotto_tok ws ordinale
               comma?
lotto_num   = lotto_tok (ws ('nr.'/'nr'/
               'n.ro'/'n.ri'/'n.'/'n'))?
               ws numero
lotto_unico = ((lotto_tok ws unico)/
               (unico ws lotto_tok))
               comma?
tupla_mix   = foglio_single ((jollies
               map_single jollies sub) /
               (jollies map) )+
```

Listing 2: A simplified version of the master rules.

4.3 LLM-Based Information Extraction

In this approach, we utilize LLMs to extract property metadata through *prompt engineering*. Building on the insights from (Ashok and Lipton, 2023) and best practices outlined by Claude,[2] we iteratively design and refine a *series of prompts*, conducting one-shot extractions based on them.

Each prompt is structured into multiple sections, with instruction-related components enclosed within explicit opening and closing tags, except for the introductory section. Specifically, a prompt includes the following sections (as illustrated in Figure 1):

- **Introduction**: Defines the LLM's role as an IE engine and its main goal: following the instructions in the next sections;
- **Context**: Specifies that the domain of application is auctions;
- **Task**: Describes the objective, which is to extract real estate-related information from the input document and structure it into a JSON output;
- **Field definitions**: Provides brief descriptions of each field to be extracted (e.g., *foglio*, *particella*, *sub*, *lotto*, and *property_type*);
- **Example**: Includes a sample input document along with its expected output, structured according to the "field definitions" section;
- **Input**: Contains the document to be analyzed;
- **Response**: Initially left empty, serving as a placeholder where the LLM will generate the extracted information.

To improve the coverage and accuracy of the LLM's predictions, we introduced *three prompt versions*. The prompt in Figure 1 consolidates all three versions, with cyan-highlighted sections indicating additions from `V1` to `V2`, and red-highlighted sections marking modifications introduced in `V3`. Uncolored sections correspond to the original `V1` prompt. The `V1` prompt served as our initial attempt but exhibited significant errors, primarily due to confusion between *terreno* and *fabbricato*. To mitigate this issue, the `V2` prompt incorporates additional specifications to infer whether a property is a *terreno* or a *fabbricato* when not explicitly stated. Finally, the `V3` prompt addresses errors in the extraction of cadastral coordinates from example documents in the `Example` section. To resolve this, distinct labels are assigned to inputs: the example input is labeled "Input 1", while the input to be analyzed is labeled "Input 2". Additionally, separate references are used for responses. This approach ensures a clear mapping between inputs and outputs while explicitly instructing the LLM to analyze only "Input 2" within the *Introduction* and *Context* sections.

5 Experimental Evaluation

This section presents the experimental evaluations conducted to comparatively assess the three proposed IE solutions. First, we provide a detailed

[2] https://docs.anthropic.com/en/docs/build-with-claude/prompt-engineering/use-xml-tags

> *You are an Information Extraction engine. Analyze the document and extract the information according to the instructions provided below, following the format indicated in the example. In your response, skip the preamble and provide exclusively the "properties" JSON list that is being requested as present in Input 2.*
>
> *⟨Context⟩ The context pertains to auction notices where one or more properties are grouped into lots for sale. ⟨/Context⟩*
>
> *⟨Task⟩ You are asked to extract the lands and buildings from the document "Input 2" along with their respective fields, formatted as a JSON list. ⟨/Task⟩*
>
> *⟨Fields Definitions⟩*
> 1. *sheet: a positive integer.*
> 2. *parcel: a positive integer.*
> 3. *sub: a positive integer. This can only exist for a building.*
> 4. *lot: a string identifying the lot to which the property belongs.*
> 5. *property_type: (mandatory field) takes the value "land" or "building". If a property has the "sub" field, it is necessarily a "building"; otherwise, it could be either a "building" or "land".*
>
> *Lands and buildings are identified by coordinates in the following hierarchical order: sheet, parcel, and sub. ⟨/Fields Definitions⟩*
>
> *⟨Example⟩ Input 1: Here an example document is provided.*
>
> *Response for Input 1: Here the expected JSON is provided. ⟨/Example⟩*
>
> *Input 2: Here we provide the document to be analyzed.*
>
> *Response for Input 2:*

Figure 1: The English translation of the prompt (originally used in Italian) that consolidates the three versions, V1, V2, and V3, used in this work.

description of the construction of the *Ground Truth* (GT) and the dataset used for Information Extraction. Next, we outline the LLMs considered in this work, along with the *evaluation metrics* employed. Finally, we present the *results* and discuss their implications for the proposed solutions.

5.1 Ground Truth

To assess human extraction performance, we carefully reviewed and corrected annotation errors, which may result from fatigue, oversight, haste, superficiality, or incentives to maximize compensation. Both human annotators and models were evaluated against this corrected GT. Unlike typical GTs manually created by annotators, our approach also aims to evaluate annotator performance. To this end, we constructed a small, high-quality dataset by selecting a subset of human-annotated auction notices from our database and manually re-annotating them. As domain experts without financial incentives, we ensured high annotation quality by working on this limited subset over multiple days and resolving ambiguities through discussion.

5.2 Dataset

The dataset used to evaluate human annotators, rule-based, and LLM-based approaches consists of 96 auction documents, evenly split into a development set and a test set. The development set contains 132 estates, while the test set includes 148. Table 3 reports token statistics for the test set. Notably, the average token count per document is substantial across all LLMs. This count increases significantly when considering the full input prompt—including both the template and the document—exceeding 18k tokens for both Claude and Llama models. This is due to the inclusion of a sample document-output pair in the one-shot prompt. However, the output token count remains relatively low, as the extracted information is structured as an array of JSON documents.

LLM	**Template (#tok)**	**Document (AVG #tok)**	**Input Prompt (AVG #tok)**	**Output Prompt (AVG #tok)**
Llama	6.354	11.855	18.209	142
Claude	6.471	9.874	16.345	128

Table 3: Tokens counting calculated on the test set.

5.3 Models

Table 4 lists the LLMs evaluated in this work. Specifically, we tested Anthropic's Claude and Meta's Llama models, representing closed-source and open-source families, respectively. Both can process large input token volumes, though only Llama's parameter count is publicly known.

GLLM	**Source**	**#Params**	**Context size**
Claude 3 Haiku	closed	n.a.	200K
Claude 3 Sonnet	closed	n.a.	200K
Claude 3.5 Sonnet	closed	n.a.	200K
Claude 3 Opus	closed	n.a.	200K
LLama 3.1 8B Instruct	open	8B	128K
LLama 3.1 70B Instruct	open	70B	128K
LLama 3.1 405B Instruct	open	405B	128K

Table 4: LLM models used for comparative evaluation.

5.4 Metrics

To evaluate the effectiveness of the three IE strategies, we use *Precision*, *Recall*, and F1-*score*, calculated based on the counts of *True Positives* (TP), *False Positives* (FP), and *False Negatives* (FN). A property p_i predicted by the model is considered a TP if it matches the ground truth, and an FP if it does not. Conversely, a property p_j in the ground truth is classified as an FN if it is not predicted by the model. Due to the domain-specific nature of the task, metrics are computed based on the following definition of property equality. Given two properties:

$$\begin{aligned} p_i &= \langle \mathit{lotto}_i, \mathit{foglio}_i, \mathit{particella}_i, \mathit{sub}_i, \mathit{type}_i \rangle \\ p_j &= \langle \mathit{lotto}_j, \mathit{foglio}_j, \mathit{particella}_j, \mathit{sub}_j, \mathit{type}_j \rangle \end{aligned}$$

we define p_i and p_j as equal if the following condition hold:

$$\begin{aligned} p_i = p_j \iff \mathit{lotto}_i = \mathit{lotto}_j \land \mathit{foglio}_i = \mathit{foglio}_j \\ \land\, \mathit{particella}_i = \mathit{particella}_j \land \mathit{sub}_i = \mathit{sub}_j \\ \land\, \mathit{type}_i = \mathit{type}_j \end{aligned}$$

This definition assumes an ideal scenario without ambiguity or instability. However, since both the ground truth and predictions—especially those from LLM extraction—are subject to language variability (e.g., inconsistent spacing such as `"lotto 4"` vs. `"lotto     4"`), differing data types (e.g., `"sub":4` vs. `"sub":"4"`), differing text formats (e.g., `"lotto 2"` vs. `"lotto due"`), or alternate naming conventions (e.g., `"lotto unico"` vs. `"unico lotto"`), a normalization step is required for both sources. Therefore, by applying the normalization function $\| \cdot \|$, we can address such differences:

$$\begin{aligned} ||p_i|| = ||p_j|| \iff \|\mathit{lotto}_i\| = \|\mathit{lotto}_j\| \\ \land \|\mathit{foglio}_i\| = \|\mathit{foglio}_j\| \\ \land \|\mathit{particella}_i\| = \|\mathit{particella}_j\| \\ \land \|\mathit{sub}_i\| = \|\mathit{sub}_j\| \land \mathit{type}_i = \mathit{type}_j \end{aligned}$$

For instance, $||\langle$`"due", "5", "4", "terreno"`$\rangle||$ $=$ $||\langle$`"2", 5, 4, "terreno"`$\rangle||$. Note that the *type* variable does not require normalization, as it only takes two possible values: `"terreno"` or `"fabbricato"`. In contrast, the other variables consist of free-text data within the documents and therefore require normalization.

5.5 Effectiveness Results

First, we present the results of the effectiveness of the prompts outlined in Section 4.3. This serves as a basis for the subsequent comparative evaluation of the best LLM-guided solution for IE against the other two strategies. Figure 2 illustrates, as an example, the improvements achieved through prompt engineering on Claude 3 Haiku.[3] The first update of the prompt (i.e., `V2`) shows significant gains in both Precision (+0.12) and Recall (+0.16), as `V1` struggled to distinguish between *fabbricato* and *terreno* properties. `V2` effectively addresses this issue, leading to substantial improvements. With `V3`, Precision increases further (+0.08) by resolving issues with coordinate extraction in the one-shot example. However, Recall experiences a slight decrease (−0.1), likely due to variations in LLM performance.

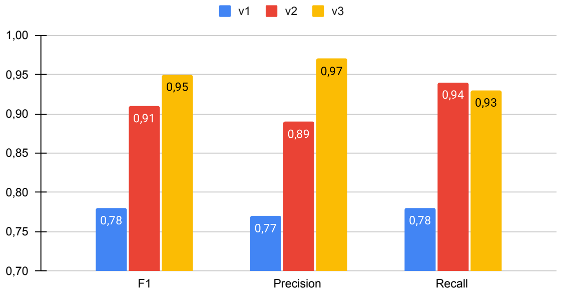

Figure 2: Performance improvements through prompt engineering, evaluated on Claude 3 Haiku.

Figure 3 compares the performance of human annotators, the rule-based system, and various LLMs using prompt `V3`. The top performers are the rule-based approach, Claude 3 Haiku, Claude 3.5 Sonnet, and Claude 3 Opus, with F1-scores between 0.93 and 0.95. However, there are notable differences in Precision and Recall. Claude 3 Haiku leads in Precision with 0.97, followed by the others scoring between 0.89 and 0.91. For Recall, Claude 3.5 Sonnet (0.98), Claude 3 Opus (0.97), the rule-based system (0.96), and Claude 3 Haiku (0.93) are the best performers. The open-source Llama 3.1 70B Instruct is also competitive, with an F1-score of 0.9 and a Recall of 0.94, surpassing Claude 3 Haiku. In contrast, the 8B and 405B Llama versions perform significantly worse.

Regarding human annotator performance, they are almost always outperformed by both the rule-based system and the Claude LLMs. However, a closer error analysis reveals that 66.7% of the properties extracted but not present in the ground truth (i.e., novel properties) are related to mistakes in the *lotto* field. Specifically, when only one lot exists, annotators sometimes label all properties under it

[3] Similar results were also obtained for the other LLMs.

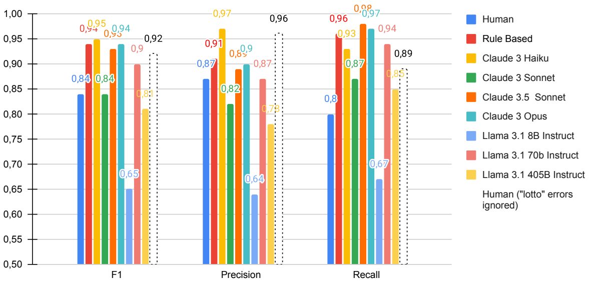

Figure 3: Performance comparison of human annotators, rule-based baseline, and LLMs.

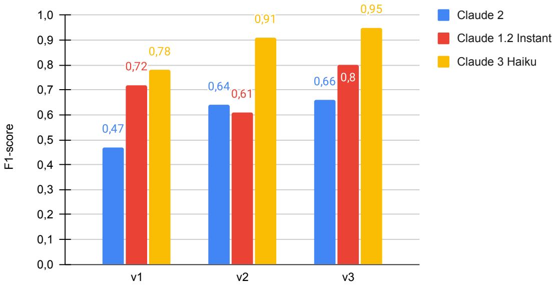

Figure 4: Prompt performance across three Claude models, illustrating improvements over time.

as "lotto unico" (single lot), even if the auction document uses a specific name (e.g., "lotto 2"). This inconsistency among annotators leads to mismatches and inflates error counts. While this might reflect a choice by some annotators, it is not consistent across all. To address this, we re-evaluated performance after excluding this specific error type. As shown by the dashed white bars, human performance improves significantly, becoming much more competitive with Claude LLMs and the rule-based system. Nevertheless, even with this adjustment, humans still lag behind in F1-score and Recall, though they achieve the second-best Precision across all models. That said, we consider the initial performance as the true measure of human ability, while the adjusted results serve only to highlight the strengths of the other models.

Figure 4 displays the F1-scores for prompts tested across three Claude models, listed chronologically: Claude 2 (July 2023), Claude 1.2 Instant (August 2023), and Claude 3 Haiku (March 2024). The results highlight that the same prompt can lead to varying performance levels, even across models within the same family. In general, newer models tend to achieve better performance with identical prompts. Furthermore, the trend observed in Figure 2 is consistent for both Claude 2 and Claude 1.2 instant, except for the prompt V2, which led to a performance decline in Claude 1.2 instant.

5.6 Execution Cost Results

Table 5 provides a comprehensive overview of the costs associated with each model evaluated on the test set, which consists of 48 auctions. For the LLMs, the breakdown includes the cost of the template, the input document for analysis, the entire input prompt (template + document), the output, the average execution cost per document, and the total execution cost for the dataset. Human annotators and the rule-based approach are also included for comparison. Human annotators incur the highest costs, set at €0.10 per estate identified and annotated, resulting in a total expense exceeding €16. In contrast, the rule-based approach is significantly more economical, with costs determined by the deployment infrastructure (AWS Lambda in this case), resulting in minimal execution expenses.

When comparing the models, human annotation and Claude 3 Opus emerge as the most expensive LLM, followed by Llama 3.1 with 405B parameters. Smaller LLMs incur lower costs, while the rule-based approach maintains an exceptionally low execution cost.

Model	$Temp.	$Doc. (AVG)	$Input (AVG)	$Output (AVG)	$Proc. (AVG)	$Proc. (TOT)
Human	-	-	-	-	0,34229	16,43
Rule-based	-	-	-	-	0,00065	0,03
Claude 3 Haiku	0,00162	0,00247	0,00409	0,00016	0,00425	0,20
Claude 3 Sonnet	0,01941	0,02962	0,04904	0,00192	0,05096	2,45
Claude 3.5 Sonnet	0,01941	0,02962	0,04904	0,00962	0,05096	2,45
Claude 3 Opus	0,09707	0,148125	0,24518	0,00192	0,25480	12,23
Llama 3.1 8B Inst.	0,00140	0,00260	0,00401	0,00003	0,00404	0,19
Llama 3.1 70B Inst.	0,00629	0,01174	0,01803	0,00014	0,01817	0,87
Llama 3.1 405B Inst.	0,03380	0,06307	0,09688	0,00228	0,09915	4,76

Table 5: Average and total extraction costs for each model and baseline based on the test set.

5.7 Cost-Performance Trade-off Analysis

Figure 5 presents both price and performance variables in a single plot. The visualization categorizes models into three distinct cost tiers: *high-cost models* (on the right), *mid-range models* (in the center), and *budget models* (on the left). Ideally, optimal models would occupy the upper-left quadrant (high performance, low cost), while underperforming models would cluster in the lower-right quadrant (low performance, high cost). Among the high-cost models, performance starts at a moderate level, with Claude 3 Opus standing out as the best performer. The mid-range models exhibit a similar performance spread, with Claude 3.5 Sonnet performing the best. On the left side, we find a low-performing model, Llama 3.1 8B Instruct, alongside two high-performing models: the rule-based model and Claude 3 Haiku.

Surprisingly, some models with the lowest computational costs also demonstrate the highest performance. In terms of human annotators, their high costs make them a less attractive option, even compared to the most expensive LLMs available. Additionally, we observe that nearly all Claude models perform exceptionally well on our task (except for Claude 3 Sonnet), while the performance of Llama models varies depending on the number of parameters. Interestingly, the 70B Instruct Llama model outperforms the 405B model, although the reason for this remains unclear.

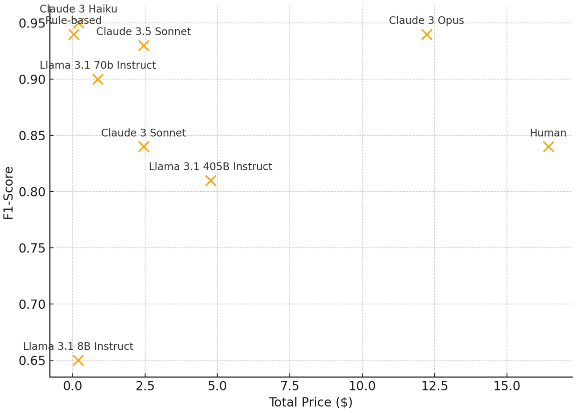

Figure 5: Summary plot of the evaluated approaches.

5.8 Implementation Cost and Requirements

In this section, we evaluate the development, maintenance costs, and skill requirements for each approach, crucial for enterprise applications and comprehensive analysis. While precise cost estimation is challenging, the following insights are based on our experience with these approaches:

- *Development costs*: Human annotation systems are straightforward to implement initially, offering flexibility for handling diverse input formats and extraction needs. However, the cost of development increases when the complexity of the task rises or if the domain knowledge required becomes more specialized. Rule-based systems have relatively low initial development costs when targeting medium performance, as rules for common patterns can be implemented quickly. However, addressing long-tail distributions significantly increases costs, requiring continuous updates with diminishing returns. Achieving higher performance often involves extensive trial and error, making the process resource-intensive. LLMs typically involve lower initial costs due to their ability to enable rapid prototyping and reduce technical demands. However, their behavior is not deterministic, and when performance stagnates, advanced prompt engineering or specialized techniques may be required, leading to increased costs over time;

- *Skills required*: Human annotation systems require moderate skill levels, but domain expertise is crucial for accurate annotation. Rule-based systems demand a high level of expertise, requiring both domain knowledge and technical skills to design and encode effective rules. LLMs generally require very low technical skills, as their operation is primarily driven by prompt engineering. However, optimizing their performance still benefits from proficiency in prompt design;

- *Maintenance costs*: Human annotation systems have the lowest maintenance costs, as annotators can easily adapt to changing requirements with minimal system reconfiguration. However, issues such as fatigue, bias, or subjectivity may arise, leading to potential rework or the need for additional quality control, which can incrementally increase costs. Rule-based systems have the highest maintenance costs due to the need for regular bug fixes and updates to handle new patterns. Any adjustments or new rules require rigorous testing to avoid regressions, which adds significant effort. LLMs have lower maintenance costs, mainly involving occasional adjustments to prompts. However, advanced prompt engineering may be necessary in some cases, particularly when adapting to evolving use cases.

6 Discussion

In this section, we summarize the key insights derived from the experiments and the comparative evaluations conducted on the various IE solutions.

- *How does prompt engineering affect performance?* The error analysis, followed by prompt updates in the prompt engineering phase, has resulted in a substantial improvement in both Precision and Recall. The only exception is the V3 update, which caused a minor decrease in Recall. However, this decline is negligible and likely due to fluctuations in LLM performance;

- *Are LLMs competitive with a rule-based approach?* Overall, Claude 3 Haiku provides the best balance of performance among the models, with most Claude LLMs performing strongly, except for Claude 3 Sonnet. The rule-based baseline remains a robust contender, making it difficult to surpass in both Precision and Recall. While open-source LLMs generally lag behind their closed-source counterparts, Llama 3.1 70B Instruct stands out as highly competitive, particularly in terms of Recall;

- *Can prompt effectiveness improve with model upgrades?* We observe that applying the same prompt across successive LLM generations leads to consistent performance gains, thanks to scaling laws. Larger models with more training data generally yield better results. Although specific details of Anthropic's LLMs are undisclosed, improvements in model parameters and datasets likely drive these gains. Future Claude iterations should continue to show similar improvements, supporting prompt stability and scalability;

- *Which approach has the best execution cost?* Our analysis clearly shows that the rule-based baseline is significantly cheaper than its LLM counterparts. This is mainly due to the nature of deep neural networks, which are expensive, even during inference, and the deployment choice. A serverless service like AWS Lambda is highly cost-effective, as it charges only for processing time, regardless of input length. In contrast, cloud-deployed LLMs are priced based on the number of tokens processed. Nonetheless, Claude 3 Haiku offers a very competitive price. Unsurprisingly, human annotation remains the most expensive option compared to all other solutions;

- *Which approach offers the best overall cost-effectiveness?* Considering the two key factors of performance and price, we conclude the following: (i) the rule-based model and Claude 3 Haiku offer the best trade-offs; (ii) the rule-based model is the cheapest overall, while Claude 3 Haiku delivers the highest performance; (iii) Llama 3.1 70B Instruct is notable for its strong performance and open-source nature, allowing for on-premise use and fine-tuning to potentially match the other models' performance while reducing costs; (iv) human annotation is neither competitive in terms of performance nor cost. However, when considering the development, maintenance costs, and required skills, LLMs emerge as the more cost-effective option compared to rule-based approaches. Additionally, the ease of performance improvements with newer LLM versions, coupled with the rapid advancements in generative models and decreasing costs, makes LLMs the optimal solution for this case study.

7 Conclusions and Future Work

In this study, we compared a rule-based system with LLM-based approaches for Information Extraction (IE) in the real estate domain, along with human annotation performance. Our findings show that the best-performing LLM outperformed both human annotators and the rule-based system in terms of overall performance, particularly in Precision and Recall. However, the rule-based approach remains a strong contender due to its reliable and consistent performance, largely stemming from the considerable time and effort invested in its development. LLMs, on the other hand, offer a faster and more scalable development process. With LLMs, the transition from error analysis to performance improvements is more efficient, and they do not require specialized skills such as knowledge of rules, grammars, or programming. This makes them a more accessible and cost-effective solution compared to rule-based systems. Furthermore, the continuous advancements in LLMs mean that their performance improves over time, often without the need for prompt modifications, making them a sustainable option for long-term applications. Human annotation, while flexible and adaptable, was found to be the least effective and most costly approach. Despite its high flexibility in handling diverse inputs, human annotation yielded unsatisfactory results compared to automated approaches and proved to be less cost-efficient. However, this may also be due to a suboptimal choice of evaluators or the human evaluation strategy adopted.

Hence, future work could benefit from a more granular error analysis to identify specific challenges each model faces when extracting particular fields, thereby guiding targeted improvements. To enhance the robustness of human annotations, future studies could incorporate overlapping document sets among annotators to enable the calculation of Inter-Annotator Agreement metrics. This should be complemented by a pilot phase on a small subset of data, followed by thorough error analysis and refinement of annotation guidelines to improve consistency and quality throughout the annotation process.

The methodology and findings presented here could be extended to a wider range of IE tasks and domains, including documents of varying lengths, to better understand the effects of document length and potential annotator fatigue. It is worth noting that while LLMs continue to advance, human input remains valuable—especially when models exhibit uncertainty (Trewartha et al., 2022). Additionally, fine-tuning open-source models like LLaMA 3.1 70B Instruct offers a promising and cost-efficient avenue for future research, with the potential to effectively balance performance and scalability.

References

Khalifa Alhosani and Saadat M Alhashmi. 2024. Opportunities, challenges, and benefits of ai innovation in government services: a review. *Discover Artificial Intelligence*, 4(1):18.

Dhananjay Ashok and Zachary C Lipton. 2023. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*.

Jahongir Azimjonov and Jumabek Alikhanov. 2018. Rule based metadata extraction framework from academic articles. *arXiv preprint arXiv:1807.09009*.

Julien Breton, Mokhtar Mokhtar Billami, Max Chevalier, Ha Thanh Nguyen, Ken Satoh, Cassia Trojahn, and May Myo Zin. 2025. Leveraging llms for legal terms extraction with limited annotated data. *Artificial Intelligence and Law*, pages 1–27.

Hung Chau, Saeid Balaneshin, Kai Liu, and Ondrej Linda. 2020. Understanding the tradeoff between cost and quality of expert annotations for keyphrase extraction. In *Proceedings of the 14th Linguistic Annotation Workshop*, pages 74–86.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Systemt: An algebraic approach to declarative information extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 128–137.

Laura Chiticariu, Yunyao Li, and Frederick Reiss. 2013. Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. Gate: A framework and graphical development environment for robust nlp tools and applications.

Nicolas de Bellefonds, Tauseef Charanya, Marc Roman Franke, Jessica Apotheker, Patrick Forth, Michael Grebe, Amanda Luther, Romain de Laubier, Vladimir Lukic, Mary Martin, Clemens Nopp, and Joe Sassine. 2024. Where's the value in ai?

Dayne Freitag, John Cadigan, John Niekrasz, and Robert Sasseen. 2022a. Accelerating human authorship of information extraction rules. In *Proceedings of the First Workshop on Pattern-based Approaches to NLP in the Age of Deep Learning*, pages 45–55.

Dayne Freitag, John Cadigan, Robert Sasseen, and Paul Kalmar. 2022b. Valet: Rule-based information extraction for rapid deployment. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 524–533.

Chengguang Gan, Qinghao Zhang, and Tatsunori Mori. 2023. Giellm: Japanese general information extraction large language model utilizing mutual reinforcement effect. *arXiv preprint arXiv:2311.06838*.

Ralph Grishman. 2015. Information extraction. *IEEE Intelligent Systems*, 30(5):8–15.

Feng Gu, Zongxia Li, Carlos Rafael Colon, Benjamin Evans, Ishani Mondal, and Jordan Lee Boyd-Graber. 2025. Large language models are effective human annotation assistants, but not good independent annotators. *arXiv preprint arXiv:2503.06778*.

Udo Hahn and Michel Oleynik. 2020. Medical information extraction in the age of deep learning. *Yearbook of medical informatics*, 29(01):208–220.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *COLING 1992 volume 2: The 14th international conference on computational linguistics*.

Yuzhao Heng, Chunyuan Deng, Yitong Li, Yue Yu, Yinghao Li, Rongzhi Zhang, and Chao Zhang. 2024. Proggen: Generating named entity recognition datasets step-by-step with self-reflexive large language models. *arXiv preprint arXiv:2403.11103*.

Harry Hochheiser, Yifan Ning, Andres Hernandez, John R. Horn, Rebecca Crowley Jacobson, and Richard David Boyce. 2016. Using nonexperts for annotating pharmacokinetic drug-drug interaction mentions in product labeling: A feasibility study. *JMIR Research Protocols*, 5.

Atin Sakkeer Hussain and Anu Thomas. 2024. Large language models for judicial entity extraction: A comparative study. *arXiv preprint arXiv:2407.05786*.

Yunqing Jiang, Patrick Cheong-Iao Pang, Dennis Wong, and Ho Yin Kan. 2023. Natural language processing adoption in governments and future research directions: A systematic review. *Applied Sciences*, 13(22):12346.

Mike Jin, Nicole M. Duggan, Varoon Bashyakarla, Maria Alejandra Duran Mendicuti, Stephen Hallisey, Denie Bernier, Joseph Stegeman, Erik Duhaime, Tina Kapur, and Andrew J. Goldsmith. 2023. Expert-level annotation quality achieved by gamified crowdsourcing for b-line segmentation in lung ultrasound. Cornell University Library, arXiv.org.

Peter Kluegl, Martin Toepfer, Philip-Daniel Beck, Georg Fette, and Frank Puppe. 2016. Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering*, 22(1):1–40.

Nicolas Laurent and Kim Mens. 2015. Parsing expression grammars made practical. In *Proceedings of the 2015 ACM SIGPLAN International Conference on Software Language Engineering*, pages 167–172.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Hannah Mayer, Lareina Yee, Michael Chui, and Roger Roberts. 2025. Superagency in the workplace: Empowering people to unlock ai's full potential.

Maja Pavlovic and Massimo Poesio. 2024. The effectiveness of llms as annotators: A comparative overview and empirical analysis of direct representation. *arXiv preprint arXiv:2405.01299*.

Jakub Piskorski and Roman Yangarber. 2013. Information extraction: Past, present and future. *Multi-source, multilingual information extraction and summarization*, pages 23–49.

Pawan Rajpoot and Ankur Parikh. 2023a. GPT-FinRE: In-context learning for financial relation extraction using large language models. In *Proceedings of the Sixth Workshop on Financial Technology and Natural Language Processing*, pages 42–45, Bali, Indonesia. Association for Computational Linguistics.

Pawan Kumar Rajpoot and Ankur Parikh. 2023b. Gpt-finre: in-context learning for financial relation extraction using large language models. *arXiv preprint arXiv:2306.17519*.

Joana Ribeiro de Faria, Huiyuan Xie, and Felix Steffek. 2025. Information extraction from employment tribunal judgments using a large language model. *Artificial Intelligence and Law*, pages 1–22.

Vedansh Thakkar, Greg Silverman, Abhinab Kc, Nicholas Ingraham, Emma Jones, Samantha King, and Christopher Tignanelli. 2024. Comparison of large language models versus traditional information extraction methods for real world evidence of patient symptomatology in acute and post-acute sequelae of sars-cov-2.

Amalie Trewartha, Nicholas Walker, Haoyan Huo, Sanghoon Lee, Kevin Cruse, John Dagdelen, Alexander Dunn, Kristin A Persson, Gerbrand Ceder, and Anubhav Jain. 2022. Quantifying the advantage of domain-specific pre-training on named entity recognition tasks in materials science. *Patterns*, 3(4).

Marco A. Valenzuela-Escárcega, Gus Hahn-Powell, and Dane Bell. 2020. Odinson: A fast rule-based information extraction framework. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2183–2191, Marseille, France. European Language Resources Association.

Bernhard Waltl, Georg Bonczek, and Florian Matthes. 2018. Rule-based information extraction: Advantages, limitations, and perspectives. *Jusletter IT (02 2018)*, 4.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022. Deepstruct: Pre-training of language models for structure prediction. *arXiv preprint arXiv:2205.10475*.

Xin Wang, Liangliang Huang, Shuozhi Xu, and Kun Lu. 2024. How does a generative large language model perform on domain-specific information extraction? a comparison between gpt-4 and a rule-based method on band gap extraction. *Journal of Chemical Information and Modeling*, 64(20):7895–7904.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. 2024. Large language models for generative information extraction: A survey. *Frontiers of Computer Science*, 18(6):186357.

Eiko Yamamoto, Hitoshi Isahara, Akira Terada, and Yasunori Abe. 2008. Extraction of informative expressions from domain-specific documents. In *LREC*. Citeseer.

Xi Yang, Aokun Chen, Nima PourNejatian, Hoo Chang Shin, Kaleb E Smith, Christopher Parisien, Colin Compas, Cheryl Martin, Mona G Flores, Ying Zhang, et al. 2022. Gatortron: A large clinical language model to unlock patient information from unstructured electronic health records. *arXiv preprint arXiv:2203.03540*.

Shiyu Yuan and Carlo Lipizzi. 2023. Information extraction in domain and generic documents: Findings from heuristic-based and data-driven approaches. *arXiv preprint arXiv:2307.00130*.

Ashwini V Zadgaonkar and Avinash J Agrawal. 2021. An overview of information extraction techniques for legal document analysis and processing. *International Journal of Electrical & Computer Engineering (2088-8708)*, 11(6).

Kai Zhang, Bernal Jiménez Gutiérrez, and Yu Su. 2023. Aligning instruction tasks unlocks large language models as zero-shot relation extractors. *arXiv preprint arXiv:2305.11159*.