

Large Language Models for Reliable Information Extraction



Lukas Baliunas

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Churchill College

August 2023

Declaration

I, Lukas Baliunas, of Churchill College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

All of the software in this project was developed using Python, and can be found in a public GitHub repository¹. This project extended the ELICIT library² to make use of Large Language Models and perform speaker-centric information extraction. The Large Language Models, used in this project, were retrieved through the Huggingface transformers³ library. The model used in this project, Vicuna⁴ — a fine-tuned LLaMA model — is a publicly available model with its use subject to the model license of LLaMA. The use of LLaMA was approved by Meta AI after filling out a request form. Low-Rank Adaption fine-tuning was done by modifying the code in the Alpaca-LoRA repository⁵, which is available under an Apache-2.0 license, and uses the PyTorch⁶ and transformers libraries. The embedding model used in the project, 'all-mpnet-base-v2', came from the SentenceTransformers⁷ library. Vector index functionality was provided by the LangChain⁸ and ChromaDB⁹ Python libraries. OpenAI's GPT-4 API¹⁰ was used for generating synthetic data under a paid license.

This dissertation contains 14993 words, excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices.

Lukas Baliunas

August 2023

¹<https://github.com/lbaliunas/elicit/tree/lb956-project>

²<https://github.com/Bradley-Butcher/elicit>

³<https://huggingface.co/docs/transformers>

⁴<https://lmsys.org/blog/2023-03-30-vicuna/>

⁵<https://github.com/tloen/alpaca-lora>

⁶<https://pytorch.org>

⁷<https://www.sbert.net/>

⁸<https://python.langchain.com/>

⁹<https://www.trychroma.com>

¹⁰<https://openai.com/gpt-4>

Acknowledgements

I wish to express my sincere thanks to my supervisor Dr. Miri Zilka, and Dr. Jiri Hron for their guidance and invaluable lessons throughout this project.

I would also like to thank my girlfriend, Ieva, and my family for their support and continuous encouragement.

Abstract

This project addresses the ongoing challenge of achieving reliable Information Extraction (IE), particularly in domains which require near-perfect precision. ELICIT (Butcher et al., 2023) introduced a novel approach that combines the processing speed of automated IE tools with the precision of manual annotation through a unique approach of weak supervision labeling and human validation. The existing setup achieved impressive results when it came to precision, but the recall of the system can still be greatly improved. Recognizing the potential of Large Language Models (LLMs) in tasks that require language understanding, this project focuses on extending ELICIT's capabilities with LLMs and evaluating their effectiveness in the context of structured IE. The research is grounded in the legal domain, focusing on evaluations on UK Crown Court sentencing remarks, and court transcripts. The latter usually contains multiple speakers with different narratives, necessitating the extraction of reliable information along with speaker attribution. This project builds upon the existing system, improving it in two significant ways: firstly, by introducing LLMs as an extraction tool, and secondly, by extending the system's functionality to extract information attributed to specific speakers. The integration of LLMs into ELICIT resulted in improved recall compared to the previous version. Moreover, through the utilization of human-validated extractions for fine-tuning, the enhanced system showcased improved recall on both already extracted documents and unseen ones. The research demonstrates how fine-tuning with limited data leads to enhanced performance, while also requiring fewer extractions to be shown to the user, thereby reducing potential manual efforts.

Table of contents

1	Introduction	1
1.1	Context and Motivation	2
1.2	Contributions	3
1.3	Thesis Outline	4
2	Background	5
2.1	Information Extraction and Question Answering	5
2.2	Multi-Value Extraction and Source Attribution	7
2.3	Large Language Models	7
2.3.1	Auto-regressive Language Models	8
2.3.2	Transformer	8
2.3.3	LLaMA	10
2.3.4	Vicuna	11
2.3.5	Fine-tuning using Low-Rank Adaptation	12
2.4	ELICIT	13
2.4.1	Approach	13
2.4.2	Labelling Functions	14
3	Methodology	17
3.1	Datasets	17
3.1.1	Sentencing Remarks	17
3.1.2	Court Transcripts	18
3.2	Information Extraction	20
3.2.1	Approach	20
3.2.2	Passage Retrieval	20
3.2.3	Value Labelling	23
3.3	Source Attribution	24
3.3.1	Source Attribution as the Final Component	25

3.3.2	Source Attribution in Passage Retrieval	25
3.3.3	Source Attribution in Value Labelling	26
3.3.4	Implementation	26
3.4	Fine-tuning	28
3.5	Metrics	29
3.5.1	End-to-End Metrics	29
3.5.2	Supervised Dataset Metrics	30
4	Results and Discussion	35
4.1	Sentencing Remarks Dataset	35
4.1.1	End-to-end Performance	35
4.1.2	Fine-tuned Performance on Unseen Documents	37
4.1.3	Fine-tuning for Recalibration	42
4.2	Court Transcripts Dataset	43
4.2.1	Performance on Unseen Documents	43
4.2.2	Fine-tuning for Recalibration	47
4.2.3	Recall and Shown Extractions	49
5	Conclusions	53
5.1	Project Recap	53
5.2	Key Findings	53
5.3	Limitations	54
5.4	Future Directions	55
	References	57
	Appendix A Fine-tuning Details	63
A.1	Sentencing Remarks	63
A.1.1	Passage Retriever	63
A.1.2	Value Labellers	64
A.2	Court Transcripts	65
A.2.1	Passage Retriever	65
A.2.2	Value Labellers	66

Chapter 1

Introduction

In recent years, Large Language Models (LLMs) have demonstrated exceptional performance in tasks which require language understanding (Brown et al., 2020). Among the domains standing to benefit from this advancement is Information Extraction (IE), which concerns the extraction of structured information from unstructured textual data. IE finds applications in many fields like healthcare (Javaid et al., 2022; Qayyum et al., 2020), finance (Ghoddusi et al., 2019), and law (Bansal et al., 2019; Carnaz et al., 2020).

However, these domains often involve sensitive data, the use of which has serious consequences, highlighting the importance of IE with near-perfect precision. The legal sector for instance, is especially sensitive to inaccuracies in IE, which could lead to wasted resources, unfair legal outcomes, or the loss of public trust. These issues could be addressed by tasking a human with IE, but such a solution requires significant time and financial resources. Conversely, automated methods, while demonstrating rapid processing, compromise on precision of the extracted information (Yang et al., 2022).

Addressing this challenge, the ELICIT framework, introduced by Butcher et al. (2023), offers a novel approach by combining automated IE systems with human validation. This approach attains precision comparable to manual annotation, surpassing automated tools, and most importantly reduces annotation time relative to manual annotation.

Nevertheless, the limiting element of the system is its imperfect recall, primarily caused by the automated component. In light of the promising advancements by LLMs in natural language processing, this project extends ELICIT by incorporating LLMs as the automated IE element. Furthermore, the objective of this research is to evaluate the effectiveness of LLMs in IE.

Since it is not possible to fit the full documents into LLM prompts due to their context window limits, the decision was made to divide them into smaller passages, extracting the information from the most relevant ones. While more commonly embedding models are used

for passage retrieval (Kamalloo et al., 2023), this research also explores the use of prompting LLMs for relevant passage retrieval. Moreover, the impact of using validated extractions for fine-tuning are evaluated, with the objective of enhancing the system’s performance.

The research is grounded in the legal domain through evaluations on UK Crown Court sentencing remarks, the same dataset utilized by Butcher et al. (2023). Additionally, a novel task of IE involving source attribution is introduced, focusing on extraction of information provided by the speakers present in the text. Synthetic court transcript, generated using GPT-4 (OpenAI, 2023b) based on real sentencing remarks, serve as the basis for evaluation.

In the baseline model setting, the developed system exhibits superior recall to the current version of ELICIT. The baseline LLM performs worse than the embedding model in passage retrieval, showing the limitation of direct use of general LLMs. However, fine-tuning on a limited amount of human validated extractions results in significant improvement in performance, matching the embedding model’s performance for sentencing remarks, and even exceeding it for court transcripts. This improvement also transfers to the complete IE system, which includes the human validator. Notably, the fine-tuning results in significant increases in recall in both unseen documents, and already processed ones, requiring fewer extractions shown to the human validator.

1.1 Context and Motivation

The motivation for the project comes from the desire to accomplish structured information extraction on various types of documents, specifically those in the legal domain - sentencing remarks and transcripts of court proceedings. Sentencing remarks, which are the judge’s remarks during the announcement of the sentence to the defendant, follow a consistent format - they are given from the perspective of the judge and directed to the defendant. They encapsulate a lot of information about the case, including the description of the crime, substantial evidence, and discussions on the mitigating and aggravating factors. Thus, they present a valuable summary of information related to the case, whose extraction would benefit future researchers with more comprehensive datasets from the legal domain.

The project is also dictated by the types of variables aimed to be extracted from these documents. The extraction system is designed to have an open nature, accommodating a wide variety of categorical variables. The ability to extract information, regardless if it’s presented explicitly, or if it is subjective, is desired. For instance, users should be able to define simple variables, such as the victim’s sex, but also more complex and subjective one’s such as the presence of emotional abuse or whether remorse was shown by the defendant.

Transcripts of court proceedings represent another source of information. Criminal court proceedings consist of multiple stages, including counsel’s opening and closing remarks, witness evidence (examination), and others. Thus, these are extensive documents, usually running into hundreds of pages, presented in dialogue format with each speaker identified by a dialogue tag. Whereas sentencing remarks are from a single person’s perspective, court transcripts consist of the multiple parties involved in the case. This variety adds a layer of subjectivity to the information presented, since the defense and prosecution often present opposing narratives and provide different interpretations of the same facts. The reliability of witness testimonies can also vary, since the examinations are led by the prosecutor or defense and will usually consist of questions favouring the respective counsel’s narrative. Therefore, it’s evident that court transcripts pose a significantly more challenging setting for information extraction.

The inherent subjectivity and potential presence of contradictory facts in court transcripts necessitate the need for an alternative information extraction process. Unlike with sentencing remarks, where the goal is to extract a single value for a variable, the aim now is to extract all values presented by each speaker for a given variable. This changes the nature of the problem from variable-centric to speaker-centric.

This type of information could potentially reveal what facts or interpretations of those facts were presented by each counsel. Moreover, what evidence was provided by each witness and whether there were any contradictions could be identified. Extraction of data through this novel approach, can lead to a more comprehensive understanding of legal cases, as well as, potential practical applications for future cases and policy-making.

1.2 Contributions

The main contributions of this project are the following:

1. Development of an end-to-end system for structured information extraction employing LLMs, and its implementation in the ELICIT codebase. The developed system outperforms the original ELICIT system in terms of recall in sentencing remarks using Vicuna-13B (Chiang et al., 2023), and with a significant margin on human-validated data after fine-tuning the models.
2. Introduction of a novel task of information extraction with source attribution, which focuses on the extraction of information for distinct speakers within documents. The challenges inherent in this task are outlined, and a methodology utilizing LLM prompt-

ing is proposed and implemented. The evaluation of this methodology is carried out on a dataset of synthetic legal court transcripts.

3. A method of generating synthetic text documents, namely legal court transcripts, is proposed, which uses an iterative summarization approach to overcome context window limitations of GPT-4.
4. Analysis of the impact of human-validated data for LLM-finetuning. This fine-tuning enhances recall for already extracted and unseen documents, requiring fewer extractions to be shown to the human validator.
5. Comparison of passage retrieval using a vector index storing passage embeddings with an LLM-based approach. Particularly noteworthy is the demonstration of the performance of Vicuna-13B fine-tuned with limited data, which is comparable to an embedding model specifically trained for semantic search.

1.3 Thesis Outline

In Chapter 2, the background pertinent to understanding the concepts utilized in the project is introduced. This encompasses the concepts of IE and Question Answering (QA), LLMs, and the ELICIT framework.

In Chapter 3, the methodology employed to achieve the established research goals is described. This includes the selection of datasets, including the generation of synthetic court transcripts, as well as the evaluation metrics. Furthermore, the specific design choices made to accomplish information extraction using LLMs are outlined.

In Chapter 4, the results of evaluating the different components of the LLM-enhanced system are presented and discussed, both in a supervised dataset and end-to-end setting.

In Chapter 5, an overview of the project is provided, key findings are reiterated, and potential future research directions are proposed.

Chapter 2

Background

In this chapter, I present the background relevant to the project in order to better understand the context and motivation behind the research. I begin by introducing the concepts of Information Extraction and Question Answering, which set the stage for our project task. Next, I discuss the idea of extracting multiple variable values and assigning them to a specific speaker. The concepts of Large Language Models, the specific model that was chosen, and the selected framework of fine-tuning are covered. Finally, the ELICIT system (Butcher et al., 2023), which is the practical setting for our information extraction task, is introduced.

2.1 Information Extraction and Question Answering

Information Extraction (IE) is a crucial task in natural language processing, which aims to extract structured information from unstructured textual data. In this project, the goal of the system is to create tabular datasets from sentencing remarks and court transcripts with variables consisting of categorical values.

Many IE tasks are typically focused on explicit information, with a common approach being Named Entity Recognition (NER) methods. These include the identification of persons, locations, dates, and other information. Additionally, Relationship Extraction methods seek to identify the semantic connections between different entities in a document. Conversely, Question Answering (QA) presents a broader spectrum of tasks since it can accommodate a multitude of different queries.

QA tasks are usually categorized by the type of context they encompass: closed-book tasks, which provide no context, and open-book tasks, where context is given. Within IE, QA is considered an open-book task, more specifically a reading comprehension one, where the goal is to find evidence in a provided context. The output of QA tasks also varies, ranging

from span prediction, where part of the context serves as the answer, multiple choice QA, or free-form responses.

The SQuAD dataset (Rajpurkar et al., 2018, 2016) is one of the most common benchmarks for QA tasks, containing question-answer pairs generated from Wikipedia articles, and requiring that answers form part of the provided context. Currently the best performing methods are pre-trained language models fine-tuned on question answering datasets (Lan et al., 2020; Raffel et al., 2020). More recently, large language models in a few-shot setting have also demonstrated competitive performance in such tasks (Brown et al., 2020).

A recent topic of interest is the application of IE and QA in conversational settings. The FriendsQA (Yang and Choi, 2019) dataset was introduced to tackle questions asked about multi-party dialogue, utilizing transcripts from the TV show "Friends". Its tested outputs include answer utterance selection and answer span prediction. The paper highlights the increased complexity of QA tasks in a multi-party dialogue scenario compared to conventional QA tasks.

Another dataset, DREAM (Sun et al., 2019), introduced a multiple-choice QA problem for dialogue-based texts. This dataset presents unique challenges since most answers are not extractive and necessitate multi-sentence reasoning. The language in a conversational setting can also include interruptions or reasoning, which spans multiple turns. The Molweni dataset (Li et al., 2020) consists of multi-party chat conversations on the topic of Ubuntu. The authors showcase that strong QA models display a significant drop in their performance compared to the SQuAD 2.0 dataset. However, Molweni only consists of question answering over short spans of text. QAConv (Wu et al., 2022) introduces a dataset for question answering with long and complex conversations as the context. Furthermore it consists of two testing scenarios - where only the relevant chunk is provided, or the full text. This is closer to the setting of this project, where information needs to be extracted over long documents.

These recent research undertakings provide the context for the task at hand, which involves extracting information from sentencing remarks and court transcripts. Sentencing remarks align with standard IE tasks but introduce complexity due to the length of the documents and the complex variables being extracted, which are not explicit factual information. Court transcripts introduce a conversational setting with multiple parties, a scenario which recent research has identified as particularly complex. This helps in understand how the work of this project fits into the larger field of IE and QA, and shows the need for more advanced methods to solve these problems.

2.2 Multi-Value Extraction and Source Attribution

The unique context of court transcripts introduces additional challenges to standard Information Extraction (IE) and Question Answering (QA) tasks. One such complication arises from the fact that a single variable within a document can attain multiple values, dependent on the speaker.

Traditional IE methods, such as Named Entity Recognition (NER), are capable of extracting a varying number of values. However, as discussed in the previous section, it is not sufficient for the task, which demands extraction of complex, non-explicit information.

On the other hand, most QA tasks are set with a context that contains a single answer, or no answer at all. This structure is notably different from the setting, where multiple variable values may exist in the text, derived from multiple speakers.

The task of this project holds some resemblance to argument mining (Lawrence and Reed, 2019) or opinion mining (Sun et al., 2017), since the information about a variable could be thought of as an argument or opinion. While these methods focus on explicit arguments or opinions, the task of this project involves a wider range of textual nuances, such as implications, answers to questions, and other forms of language. Thus, it necessitates a broader approach to capture the richness and diversity of information present in the texts.

In addition to dealing with multiple possible values for a variable, the project also aims to attribute these values to their respective speakers. While quotation-to-speaker attribution in dialogue-based texts of known format can often be completed with pattern matching, which is a relatively straightforward approach, research which attempts to assign the extracted information to one of multiple speakers has not been identified.

2.3 Large Language Models

The core of this thesis is Large Language Models (LLMs), which have become very popular tools in natural language processing tasks, competing with most specialised models (Radford et al., 2020; Sun et al., 2020; Yang et al., 2020). LLMs are neural networks with many parameters (with state-of-the-art models in the trillions), which are trained on vast amounts of unlabeled texts from general sources, such as Wikipedia (Wikimedia Foundation, 2001), Common Crawl (Common Crawl, 2007) and others.

LLMs are used in this project based on the fact that they have been shown to work well in different tasks which require language understanding (Brown et al., 2020; Chung et al., 2022), which is the main problem of our thesis - extracting information from unstructured texts.

In this section, the concept of auto-regressive language models, is described. This is followed by the introduction of the Transformer architecture (Vaswani et al., 2017), which is the most popular architecture of modern LLMs. This leads to the presentation of the LLaMA models (Touvron et al., 2023) and its extension, Vicuna (Chiang et al., 2023), which is the model used in this project. Finally, the concept of fine-tuning through Low-Rank Adaption (Hu et al., 2021) is introduced, and why it is a suitable fine-tuning approach for this project.

2.3.1 Auto-regressive Language Models

Auto-regressive models, such as (Chiang et al., 2023, introduced in Section 2.3.4), are a class of sequential models that generate each output by conditioning it on all the previously generated outputs.

This process can be formalized by the following formula:

$$p(\text{sequence}) = p(y_{0:N}) = p(y_0) \prod_{i=1}^N p(y_i | y_{0:i-1}) \quad (2.1)$$

This distribution can then be used to generate model responses $p(\text{response}|\text{prompt})$ using a decoding strategy (e.g. sampling token by token, or choosing the highest probability token at each step). In the context of language models, y_i represents a token at the i -th position in the sequence of length N .

A token is an integer that uniquely represents some text. Tokens are produced by a tokenizer, which is trained independently, before the language model itself. Various tokenization strategies exist, broadly categorized into word-level and subword tokenization. The Vicuna model, which is based on the LLaMA model (Touvron et al., 2023), employs a Byte-Pair Encoding tokenization strategy (Sennrich et al., 2016). This strategy splits words into subword units merging together frequently occurring pairs, resulting in an efficient tokenizer capable of handling words not present in the vocabulary.

The significant advancements in auto-regressive and other language models in recent years can be attributed to the introduction of the Transformer architecture by Vaswani et al. (2017). This architecture revolutionized the training of models with a large number of parameters on vast amounts of data.

2.3.2 Transformer

The Transformer addressed several limitations of previous state-of-the-art models, such as gated Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) models, and enabled effective handling of long-range dependencies.

The key concept behind the Transformer is the attention mechanism (Bahdanau et al., 2016), which allows the model to selectively focus or *attend* to different parts of the input context. Unlike sequential processing, the Transformer looks at multiple parts of the context simultaneously and focuses on parts it finds more important. This is achieved through *scaled dot-product attention* and *multi-head attention* mechanisms.

The **scaled dot-product attention**, shown in the left of Figure 2.1, is computed as:

$$\text{Attention} = \text{softmax} \left(\frac{QK^T}{\sqrt{d_k}} \right) V \quad (2.2)$$

where Q , K , and V represent the query, key, and value, respectively, and d_k is the dimension of each key. $Q = xW^Q$, $K = xW^K$, $V = xW^V$ are linear projections of the layer input x , itself a matrix with s (sequence length) rows and d (embedding dimension) columns. The weights are trained so that $Q_i^T K_j$ expresses how much the i -th input depends on the j -th input.

To enhance the attention mechanism, the Transformer employs **multi-head attention**, shown in the right of Figure 2.1, which uses multiple attention functions or "heads", allowing the model to focus on multiple segments of the context simultaneously. Each head computes the attention independently, and the results are concatenated and combined as follows:

$$\text{MultiHead} = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O \quad (2.3)$$

where $\text{head}_i = \text{Attention}(xW_i^Q, xW_i^K, xW_i^V)$ represents the attention computed by the i -th head. W_i^Q , W_i^K , and W_i^V are the weight matrices for each head, and W^O is the output weight matrix. Each head is applied to a subset of the embedding dimension. For h heads and an embedding dimension d_{model} , the weight matrices have dimensions $d_{\text{model}} \times d_w$, where $d_w = d_{\text{model}}/h$ (h is chosen such that d_w is an integer).

The Transformer architecture follows an encoder-decoder structure (Vaswani et al., 2017). Both the encoder and decoder consist of *self-attention* blocks. In the encoder self-attention mechanism, the queries, keys, and values are derived from the outputs of the previous encoder layer. On the other hand, in the decoder self-attention mechanism, the inputs are obtained from the outputs of the previous decoder layer, but with masked attention, as illustrated in the left of Figure 2.1, so that it only attends to preceding tokens. Equations 2.3.2 and 2.3.2 provide the formulas for self-attention. Additionally, the decoder includes a *cross-attention* block, however, it is not utilized in the models employed in this project; further details on cross-attention can be found in Vaswani et al. (2017).

The absence of recurrence in the Transformer architecture makes it highly suitable for parallelization on multiple GPUs, leading to significantly reduced training times. Addition-

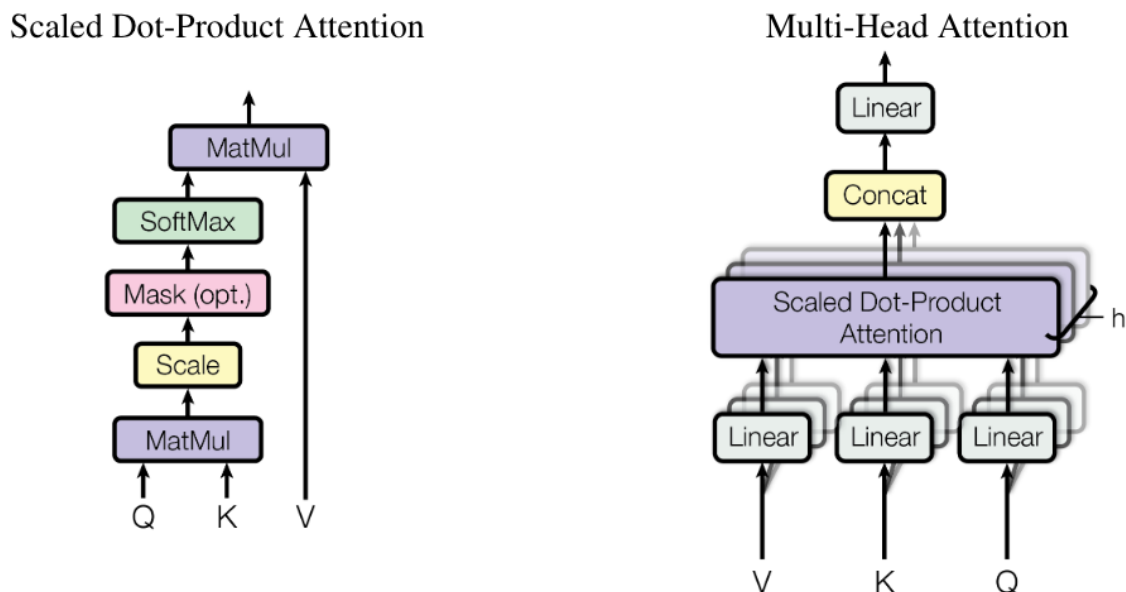


Fig. 2.1 Scaled dot-product attention (left) and Multi-head attention (right). Figures taken from Vaswani et al. (2017).

ally, Transformers do not suffer from vanishing and exploding gradient problems, which is an issue with RNNs and LSTMs when processing extremely long sequences (Pascanu et al., 2013). This advantage stems from the Transformer's ability to attend to all input positions simultaneously.

In the upcoming sections, the LLaMA model (Touvron et al., 2023) and its fine-tuned version Vicuna (Chiang et al., 2023), will be introduced, which are pre-trained auto-regressive language models based on the Transformer architecture. As mentioned before, these models only use the decoder part (without cross-attention), not the full architecture proposed by Vaswani et al. (2017).

2.3.3 LLaMA

LLaMA (Touvron et al., 2023) is a collection of auto-regressive language models developed by Meta AI, which have showcased competitive performance at a smaller parameter size. LLaMA was developed based on recent work in Hoffmann et al. (2022), which showed that better performance can be achieved not by making the models bigger, but by increasing the amount of training data. It was trained on 1 trillion tokens for the 7B and 13B models (the "B" indicates the number of parameters in billions), and 1.4 trillion for the 33B and 65B

models. LLaMA-13B has been shown to outperform GPT-3 (Brown et al., 2020), while being more than ten times smaller (Touvron et al., 2023).

LLaMA has become a popular option for fine-tuning for downstream tasks, since the smaller models can be run on a single GPU with lower floating point precision, but still result in competitive performance.

2.3.4 Vicuna

The open-source nature of LLaMA has resulted in many publicly available models fine-tuned on more specific datasets. Vicuna (Chiang et al., 2023) is an open-source language model, which was trained on 140k (for the 1.3 version, which is used in the project; the version in the article was trained on 70k) user-shared chatbot conversations from ShareGPT (Eccleston, 2023). Using an automated evaluation framework with GPT-4 (Zheng et al., 2023), the authors found that Vicuna-13B performs on-par with OpenAI's ChatGPT-3.5 (OpenAI, 2022) and Google's Bard (March 2023 version) (Manyika, 2023).

Vicuna was trained as an autoregressive chatbot model, meaning that it generates output in an autoregressive fashion and produces sequences in a chatbot conversation style - user messages are followed by assistant (chatbot) messages. The standard way of using Vicuna is to provide a prompt consisting of a series of user and assistant messages, with the final assistant message being left empty for the model to generate a response. Figure 2.2 showcases the prompting format of Vicuna.

A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user's questions.

USER:

How long would it take for a commercial airline to fly around the Earth?

ASSISTANT:

It would take a commercial airline approximately 40 hours to fly around the Earth, assuming a direct flight and no stops. However, most commercial flights do not take this route, as it would be both impractical and inefficient due to the need to refuel and the fact that most airports are not equipped to handle such long flights. Instead, commercial flights typically follow more direct routes that take them over oceans and between major hubs.

Fig. 2.2 An example of the prompt format in Vicuna. The black bold text represents the fixed template, which is required in every prompt for consistent performance. The red text highlights the user input, while the blue text is the completion from Vicuna-13B.

2.3.5 Fine-tuning using Low-Rank Adaptation

Low-Rank Adaptation (LoRA) (Hu et al., 2021) is a technique for parameter efficient fine-tuning of large pre-trained language models. Full fine-tuning large pre-trained models with billions of parameters can be computationally expensive. To address this, the authors propose the LoRA approach.

In LoRA, during the fine-tuning process, the update of a pre-trained weight matrix $W_0 \in \mathbb{R}^{d \times k}$ can be expressed as $W_0 + \Delta W$. Typically the weight matrices are full-rank, thus the authors propose expressing the update matrix as a low-rank decomposition $W_0 + \Delta W = W_0 + BA$, with matrices $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, where $\text{rank } r \ll \min(d, k)$. By freezing W_0 during fine-tuning, while B and A are updated, LoRA achieves parameter efficiency.

The low-rank assumption is based on the observation that the rank of the update matrix is often low when adapting a pre-trained model to a specific task. This is because the fine-tuning task is usually narrower in scope compared to the broad pre-training task. The authors refer to Aghajanyan et al. (2020), which shows that large pre-trained models have a low "intrinsic dimension" when adapting to a task, and thus can be reparameterised in more efficient forms.

LoRA offers several advantages over other existing fine-tuning methods:

- LoRA is parameter and memory efficient, resulting in reduced training times and lower GPU memory requirements. According to the authors, the number of trainable parameters can be reduced by up to 10,000 times, while the GPU memory requirements can be reduced by up to 3 times (Hu et al., 2021).
- A single pre-trained model can be used with multiple LoRA modules. This flexibility allows for task-specific modules to be easily swapped, enabling the use of a single pre-trained model for multiple downstream tasks. For instance, in this project, where the aim is to utilize user-validated data for fine-tuning for different subtasks, LoRA enables the use of just a single pre-trained model and fine-tune multiple modules to the specific subtasks, which eliminates the need of using multiple large models.
- LoRA introduces less additional inference latency compared to methods such as adapters (Houlsby et al., 2019). This is due to the linear nature of the method, where the existing weight matrices are modified by adding the update matrices, avoiding the need for additional modules. This property makes LoRA an efficient choice for real-time applications.

In summary, LoRA provides an efficient fine-tuning process that is suitable for this project. Its parameter and memory efficiency, along with the ability to use a single pre-trained model for multiple tasks, make it an appealing choice.

2.4 ELICIT

ELICIT is an information extraction system, introduced in Butcher et al. (2023), designed to enhance extraction efficiency and precision through a combination of weak supervision and human validation. The purpose of ELICIT was to address the imperfect accuracy of automated systems, and the slowness of manual annotation by combining the two. By incorporating modern language model capabilities and the ability to identify differing opinions and attribute them to specific speakers, we aim to further extend the system's functionality in this project.

2.4.1 Approach

In ELICIT, users begin by defining a set of variables they wish to extract, with each variable consisting of a range of possible values. This is called a categorical schema, an example of which is given in Listing 2.4.1. To handle cases where the desired information is not present in the document, an "Abstain" value is used in the system.

Listing 2.1 An example of a categorical schema. It defines the possible values that each variable can take.

```
relationship: # variable name
  # variable values:
  - family
  - friend
  - partner
  - colleague
  - none
```

To automate the extraction process, the users selects a set of labelling functions. These functions are automated tools that return a predetermined number of candidates, which are excerpts from the document labeled with a particular variable value. Each candidate is accompanied by a confidence score, indicating the system's level of certainty in its assignment.

ELICIT utilizes a ranking mechanism to prioritize the candidates and presents the Top-k candidates per variable to the user. This ranking enables the user to focus on the most relevant and potentially accurate extractions. The system also combines excerpts from different labelling functions with significant overlap.

Finally, the user engages in the validation process by reviewing and validating each candidate. This validation step contributes to the completion of the extraction process, ensuring that the extracted information is accurate and reliable.

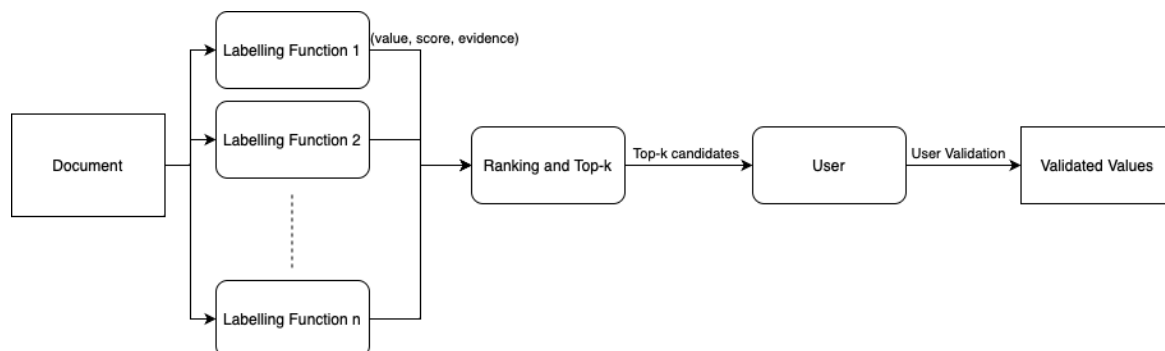


Fig. 2.3 A high-level overview of ELICIT for a single variable extraction.

2.4.2 Labelling Functions

The recall of ELICIT is mainly determined by the quality of the labelling functions. Each selected labelling function is applied to the complete set of documents under examination. ELICIT is comprised of four main labelling functions, each specifically designed to identify and categorize a singular variable within a document. These functions are explained in detail below.

1. **QA Model Followed by Zero-shot Classification:** This labeling function requires the user to define a question schema, which contains supporting questions for each variable. Subsequently, the document is split up into smaller contexts, and each question-context pair is supplied to a RoBERTa model that has been fine-tuned for question-answering tasks on the Squad2 (Liu et al., 2019; Rajpurkar et al., 2018). The output from the model is an extract from the context, where the model believes the answer to be. Along with extract, a relevance score is also computed, interpretable as the conditional probability $P_{QA}(\text{excerpt}|\text{question})$. This extracted portion is then fed into a RoBERTa Natural Language Inference zero-shot classification model (Yin et al., 2019), with the variable values operating as labels. The model generates a score $P_{NLI}(\text{label}|\text{excerpt})$ for every value, and the total confidence score is computed as the product of these two probabilities: $P_{NLI}(\text{label}|\text{excerpt}) \cdot P_{QA}(\text{excerpt}|\text{question})$.
2. **QA Model Followed by Cosine Similarity:** This labelling function has the same mechanism of the previous one, except for the final classification phase. Here, instead

of zero-shot classification, the function uses cosine similarity computation among RoBERTa embeddings for the final classification. Each text excerpt and variable value is embedded in the pattern “this is a {word}”, with “{word}” referring to the excerpt or variable value, which is being embedded. Subsequently, the cosine similarity between all excerpt embeddings and label (variable value) embeddings is calculated. The aggregate confidence score in this approach is the product of the cosine similarity and the QA probability: $P_{\text{cos}}(\text{label}, \text{excerpt}) \cdot P_{\text{QA}}(\text{excerpt}|\text{question})$.

3. **Semantic Search:** This method employs a transformer model to embed every sentence from the document. The dot product between each sentence embedding and the embeddings of questions from the question schema is calculated. The average score for all questions is taken as $P_{\text{questions}}(\text{sentence}, \text{questions})$. Then these sentence embeddings are compared to the embeddings of each label, yielding the score $P_{\text{label}}(\text{sentence}, \text{label})$. The total score for a sentence-label pairing is the product of these two scores, specifically, $P_{\text{label}}(\text{sentence}, \text{label}) \cdot P_{\text{questions}}(\text{sentence}, \text{questions})$.
4. **Keyword Search:** This labelling function requires the user to define a keyword schema, which contains a set of keywords for each variable. The function employs a phrase matcher from the SpaCy library (Honnibal et al., 2020) to detect all instances of these keywords in the document. Due to the absence of a scoring mechanism in the keyword matching, the extractions are attributed a default value of 0.1.

In conclusion, the ELICIT system uses each of these labelling functions to produce candidate excerpts from the text along with the identified labels. The recall of the system is dependant on at least one labelling function being able to identify a relevant part of the document, as well as, to assign it to the correct label. However, the current version of ELICIT assumes that only a single variable value corresponds to a document, which is not the case for many documents, for example, court transcripts. In this project, the work is focused on extending ELICIT’s capabilities by introducing new labelling functions, which make use of LLMs, and which have the ability to attribute information to speakers in the text.

Chapter 3

Methodology

In this chapter, I describe the methodology of the project, which is used to achieve the research goals. The process of creating a suitable dataset for the project is detailed, and the specific approaches of using LLMs for information extraction in a single-speaker and multi-speaker setting. The technical modifications implemented in ELICIT to accommodate the approach are also discussed. Furthermore, the method of utilizing user-validated data for model fine-tuning is explained. Lastly, the metrics employed to assess and compare the impact of the decisions on the results are described.

3.1 Datasets

3.1.1 Sentencing Remarks

For a fair comparison against the extraction methods presented in Butcher et al. (2023), the same extracted sentencing remarks as in the paper are used for the single-speaker information extraction. The documents used are part of openly accessible cases published by the United Kingdom Judiciary for cases of public interest.

The authors have selected the sentencing remarks of nineteen murder cases. Each of the documents was manually labelled for eighteen categorical variables. To ensure privacy, each named person in the text was anonymized, and the names of the defendant and victim were replaced with generic identifiers "the defendant" and "the victim", respectively. The documents averaged 3357 tokens.

Out of the eighteen variables, thirteen, which occurred five or more times in the documents, were used for evaluation. These variables cover non-trivial information about the victim, the defendant, and the crime or case. The victim-focused variables include their sex, employment status, pregnancy status, and whether they were considered vulnerable. The

variables covering the defendant include prior convictions, remorse, and whether their age was considered a mitigating factor. The remaining variables, which do not strictly fall under the previous two categories, include evidence of physical abuse, domestic abuse, instance of emotional abuse, the premeditation of the crime, and whether the crime was sexually motivated.

In every document, the victim sex is noted, either through pronouns or gender-specific words. Some information might be explicitly stated, inferred through other facts, or not mentioned at all. Most variables require complex reasoning through implied statements by the judge or from the descriptions of the crime.

The sentencing remarks dataset comprises 19 documents with 13 distinct variables, resulting in a total of 189 non-abstain and 58 abstain data points. This dataset provides a good setting to test the ability of LLMs to comprehend non-trivial information.

3.1.2 Court Transcripts

For the analysis of information extraction in a multi-speaker setting, transcripts of UK court proceedings were used as the document of choice. Although transcripts of UK court proceedings are not available to the general public and cannot be used for public research, their format is known. To overcome this limitation, GPT-4 (OpenAI, 2023b) was used to produce synthetic court transcripts based on the information present in real sentencing remarks, as modern LLMs have shown to be capable of producing human-like text (Hu et al., 2023). Although synthetic text has limitations, it is expected that approaches which do not work here will also be ineffective with the more challenging real data.

I established the following criteria for the generated documents:

1. **Size:** The generated transcripts should be at least a few pages long, totaling at least a few thousand words and thus exceeding the context window (the number of tokens in the generated sequence, including the prompt) of most commonly available LLMs, thus, facilitating the need for splitting the text, mimicking the process of extracting from real, lengthy transcripts. For reference, the Vicuna models were trained on a context length of 2048 tokens (approximately 1500 words).
2. **Multi-turn dialogue:** The transcripts should consist of multi-turn dialogue, including speakers addressing someone, raising questions and answering them.
3. **Multiple variable values:** The documents should contain instances of multiple different values applying to the same variable, including some speakers indicating different values to the same variable, i.e. giving contradicting statements.

4. **Stage structure:** The transcript should follow a predefined stage structure, starting with opening remarks by the prosecutor and the defense (counsel), followed by a series of cross-examinations of different witnesses, and end with closing remarks by the counsel. This structure, with additional stages, is found in real court transcripts, and also facilitates the inclusion of competing narratives.

The chat completion model of GPT-4 was used to generate the transcripts, which consists of three types of messages: system, user, and assistant. While the user and assistant messages work the same way as described for Vicuna in Section 2.3.4, the system message allows us to preconfigure the model so that it adheres to a certain tone, style or task in its future responses (OpenAI, 2023a). In this case, the system message contained a description of the task, and outlined the structure that the transcript should follow. Specific variable values were included by providing a list of these variables as statements in the system message, such as "the victim was unemployed". Additionally, the real sentencing remarks are provided in the system message for better context.

Generating a transcript with the desired length using a single model response is challenging due to the model's tendency to produce compact responses. Instructing the model to produce longer responses did result in slightly longer generations, but not enough to meet the needs of the project. Similarly, instructing the model to generate text of a defined word count would result in generations significantly below the requested word length. To overcome this, an iterative approach is employed. The model is prompted to generate the transcript of a single stage, and the responses are concatenated to produce the full transcript. This iterative approach allows better control over the length of the transcript, and over the content of the transcript.

To ensure variation in variable values across speakers, a subset of variables that can logically take different values in the text is established, such as premeditation, remorse, vulnerability of the victim, emotional abuse, and defendant's age as a mitigating factor. For each witness, two of these variables were randomly sampled, and the model was instructed to include the information during the examination by the prosecutor and defense, with each counsel taking opposing values.

To ensure a logically consistent document, the model has to take into account its previous generations. However, the available model has a context length limit of 8000 tokens, which is not sufficient to include previous generated stages. To address this, an iterative summarization approach is implemented. The system message and the opening remarks are included in every generation, and after generating a pair of witness examinations, the model is requested to summarize them. These summarizations replace the longer generations in the message

history, circumventing the limited context length problem while ensuring that the model sees the essential information.

Using this approach, five transcripts were generated, with an average of 5592 tokens. A single data point is defined as the tuple (document, variable, value, speaker), representing a speaker indicating a distinct variable value in a document. The transcripts were manually annotated to identify the variable values each speaker produces, resulting in a total of 171 non-abstain data points. The annotation revealed occasional issues with the synthetic data, including repeating sentences in the opening and closing remarks, and instances of logical incoherence. Nevertheless, the synthetic dataset is a good starting point for analyzing the challenges that would be faced in information extraction from real court transcripts.

3.2 Information Extraction

3.2.1 Approach

In order to extend ELICIT with the capability of using LLMs for information extraction, a new labelling function had to be created, which would take the document text and variable name as input, and return a set of candidate extractions as the output. Each extraction should contain the identified variable value and a corresponding confidence score.

3.2.2 Passage Retrieval

The first challenge faced was the issue of large document length, which exceeded the context length of most modern LLMs. To address this, the approach of splitting the text into smaller passages was taken. Subsequently, extraction on the n most relevant ones identified by the passage retrieval part is performed. This technique, visualized in Figure 3.1, is a standard approach in IE and QA systems over long documents or multiple documents (Choi et al., 2017; Izacard and Grave, 2021; Xu et al., 2011).

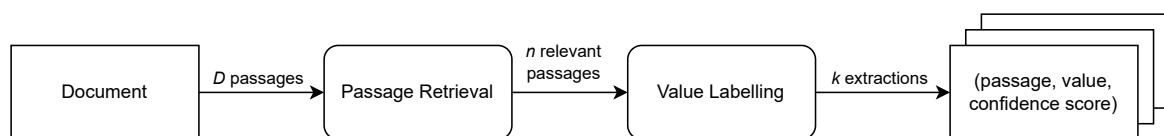


Fig. 3.1 High-level overview of the extraction process for the LLM-enhanced system. For each variable, the document is split into D passages, which are provided to retrieval system. The passage retrieval system returns n most relevant ones, which are fed to the value labelling part. It assigns a confidence score to each variable value for the passage, and top k extractions, based on confidence, are shown to the user.

In the project, experimentation was conducted with two passage retrieval methods, described in the following subsections. The first approach involved storing embedded passages in a vector index and querying them to identify the n most relevant ones. The second approach used an LLM in a *Yes/No* question format to identify whether the passage contains the relevant information.

Vector Index

In this approach to passage retrieval, semantic search is performed across all passages from a single document. The *all-mpnet-base-v2* pre-trained model from SentenceTransformers, which has shown strong performance in semantic search (Reimers and Gurevych, 2019), was used to generate vector embeddings of the individual passages.

The Chroma vector index (Huber, 2022) was utilized for storing these embeddings and performing the similarity search. This eliminates the need for recomputing the embeddings of a single document. Additionally, a vector index allows relevant passages to be retrieved using efficient approximate nearest neighbour search for high-dimensional vector data (Malkov and Yashunin, 2018).

The LangChain Python package (Chase, 2022) was employed, providing a simple interface for storing and querying passages, and directly supporting Chroma and other popular vector indices. All embedded passages from the documents are stored in a single index, with the document name assigned as metadata. During the search, filtering is applied to only include passages from the relevant document, removing the overhead of creating multiple indices for every new document and creating the possibility of performing extractions across a set of documents in the future. The queries are statements describing each variable, which the user defines in an *information schema*, for example, the statement "the relationship between the victim and the defendant" for the "relationship" variable.

The embedding vectors are stored as 768-dimensional L2-normalized vectors. The squared L2 (Euclidean) distance is used as the distance metric between the query and passage embeddings. The following formula was used to convert it into a cosine similarity score:

$$similarity = 1 - \frac{distance}{2} \quad (3.1)$$

LLM-Based Retrieval

In this method, for each passage, a binary classification approach is adopted. The LLM is presented with the passage and asked whether the context includes information about the

specified variable. The prompt template (the user message in Vicuna), as shown in the left of Figure 3.2, is used, with a sample prompt visualized in the right of the figure.

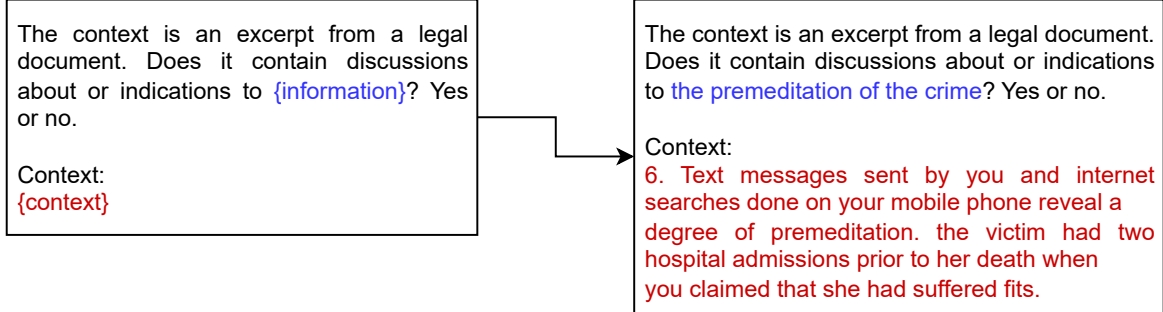


Fig. 3.2 Prompt template (left) and example prompt (right) for the LLM-based passage retriever. The `{information}` tag is replaced with the description of the variable from the information schema, as described in Section 3.2.2, while the `{context}` tag is replaced with the passage.

Preliminary experiments indicated that the majority of answers start with a "Yes" or "No", thus, decoding is unnecessary, and it is sufficient to check the conditional probability of the completion being "Yes" or "No", given the prompt.

This is accomplished by concatenating the answer to the prompt and completing a single forward pass, which yields the logits for each token in the vocabulary at each sequence position. The logits, representing the unnormalized score for each token in the vocabulary, are passed through a softmax function. This results in the (log) probability distribution over the entire vocabulary at that sequence position, given the previous tokens. The conditional probability of the answer given the prompt is obtained by summing the log probabilities of the answer tokens and applying an exponential:

$$P(\text{answer}|\text{prompt}) = P(y_{N+1:N+T+1}|y_{1:N}) = \exp\left(\sum_{i=N+1}^{N+T+1} \log P(y_i|y_{1:i-1})\right) \quad (3.2)$$

where N is the number of tokens in the prompt, T is the number of tokens in the answer, and y_i denotes the token at the i -th position in the sequence.

Using a baseline model in this manner leads to some probability mass being lost to other generations, which have the same meaning but do not start with a "Yes" or "No", for example, the sequence "The answer is Yes". The conditional probabilities of the next token being "Yes" or "No" are used and normalized as follows:

$$P(\text{Yes}) = \frac{P(\text{label} = \text{Yes}|\text{prompt})}{P(\text{label} = \text{Yes}|\text{prompt}) + P(\text{label} = \text{No}|\text{prompt})} \quad (3.3)$$

The normalized conditional probability of "Yes" is used as the confidence score of the passage retrieval component, and the top n highest scoring passages are retrieved.

3.2.3 Value Labelling

The retrieved passages are taken as input to the value labelling component with the objective of producing a set of extracted variable values with respective confidence scores. This is a multi-class classification problem, but with the aim of using a single model and with the ability to have an arbitrary number of possible variable values.

In the prompt to the LLM, given in Figure 3.3, the task is first described, which is to identify the most suitable variable value (referred to as 'label') based on the context (the candidate passage) and questions from the question schema. The latter are provided as additional context to the model, helping it align better with the specific task. This reduces potential confusion during classification. For example, in the classification of the 'victim sex' variable, the questions are "what sex was the victim?", "was the victim a male?". These questions guide the model to identify the sex of the victim specifically, thus reducing the chance of misclassification.

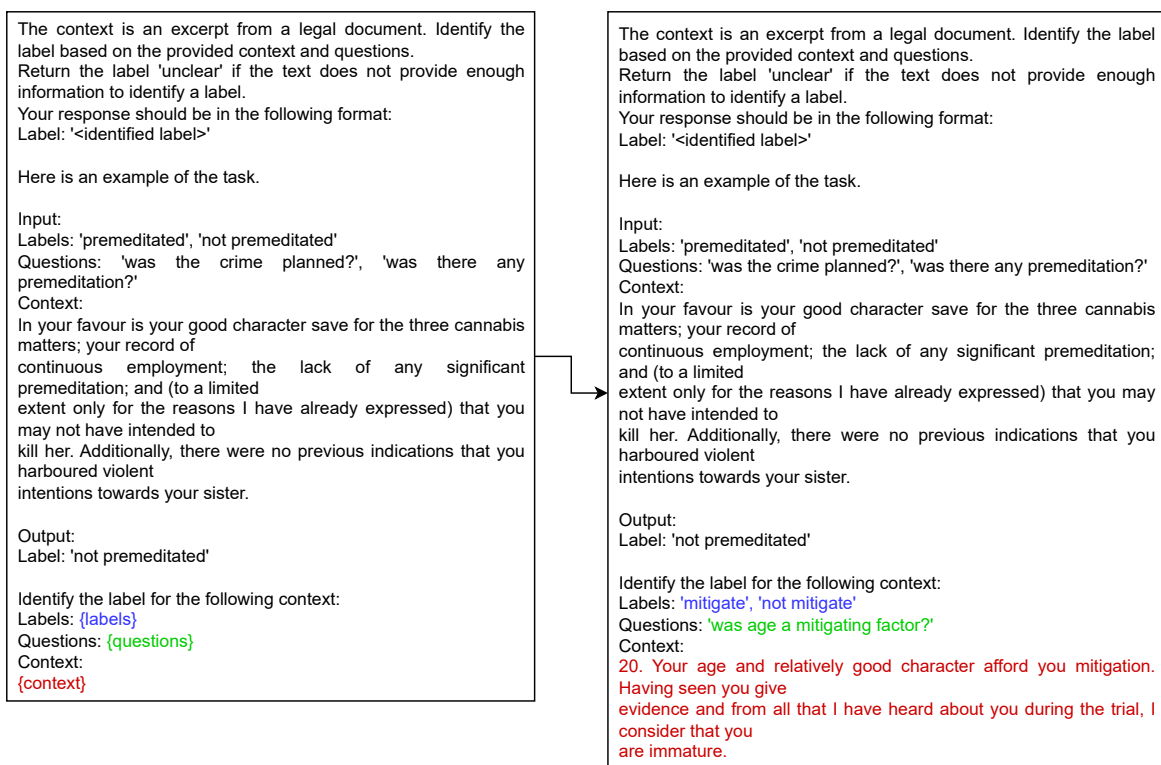


Fig. 3.3 Prompt template (left) and example prompt (right) for the value labelling model. A detailed explanation is given in Section 3.2.3.

Additionally, the model is instructed to return the label "unclear", if the context does not contain sufficient information to make a decision. This acts as a safeguard noting the fact that the passage retrieval system is not perfect and may return some irrelevant passages. Thus, the model does not need to make a decision between two incorrect values. The "unclear" label is also used to handle situations where the variable in question is mentioned, but a definitive value is not provided. For example, the sentence "We have yet to determine if the victim was pregnant." mentions the victim's pregnancy, but does not give a clear answer, and thus fits the "unclear" label.

The label "unclear" is not returned to the user as an extraction, but is employed to indicate uncertainty and remove some probability mass from the other variable values during model inference.

Finally, the prompt also contains the response format and a single example of the task. Both of these are used to focus more probability onto the exact possible labels, while the example also improves the model's understanding of the task, as prompts including in-context examples have shown to produce better results in LLMs Brown et al. (2020).

For the labelling, the conditional probability of each of the variable values, including "unclear", being the completion to the prompt is computed, as is done in Section 3.2.2. The probabilities are also normalized as some probability mass is distributed to alternative completions. This is formalized, for a label x , as:

$$P(\text{label} = x) = \frac{P(\text{label} = x | \text{prompt})}{\sum_{y \in \text{labels}} P(\text{label} = y | \text{prompt})} \quad (3.4)$$

This process results in a set of variable values with confidence scores for each retrieved passage. The overall confidence score of the end-to-end extraction system is the product of the passage retrieval confidence score and the value labelling confidence score. Finally, the Top-k highest-scoring passage-value pairs for each variable are presented to the user for validation.

3.3 Source Attribution

In this section, the process of extending ELICIT to assign source information to extracted variables when there are multiple speakers present, is described. While the primary focus of the project lies in dialogue, the methods and design choices described in this section hold the potential for future extension to other types of attributions, such as indirect quotations or

direct quotations in a non-dialogue context. The modified objective of the extraction process is to correctly identify the variable values given by a speaker.

The primary design choice lies in the placement of the source attribution component within the system. Three possible scenarios have been identified:

1. Integration of the source attribution as a final step after value labelling.
2. Incorporation of the source attribution within the passage retrieval process.
3. Embedding of the source attribution into the value labelling component.

3.3.1 Source Attribution as the Final Component

When source attribution is considered as the final step in the system, the component would receive the extracted passage-value pairs, and the approach taken would involve instructing the LLM to identify the speaker who provided evidence for a specific value.

The advantage of this approach is that the source attribution part would remain decoupled from the passage retrieval and value labelling parts, and thus could be improved independently through prompt engineering or fine-tuning. It would also be compatible with other labelling functions since it only requires an assigned variable value and a context.

However, this approach is limited in that not all possible speaker-value combinations can be extracted. If there are multiple speakers giving the same value in a passage, the model would have to choose between them, considering the task is formed as a multi-class classification one. If the LLM is used in a multi-label classification fashion, where multiple speakers can be identified for a single value, issues arise with getting the output in an expected format, as well, as getting a representative confidence score, which is speaker-specific. Furthermore, if there are multiples speakers giving different values in the same passage, the value labelling part will have distribute the probability to each of the values, or even give most of the probability mass to the "unclear" label. Hence, the score would not be reflective of the confidence of a single speaker giving a single value.

3.3.2 Source Attribution in Passage Retrieval

An alternative approach could be the incorporation of source attribution into the passage retrieval segment, allowing only those passages to be retrieved where the speaker gives some indication of the variable. Assuming that a list of speakers for the passage is available, it is necessary to determine the relevance of the passage for each speaker. This would involve modifying the LLM-based passage retriever's prompt to incorporate speaker information, for

instance, "Based on the context, are there discussions or indications given by the {speaker} about {variable}?".

With this methodology, only passages where the speaker discusses the specific variable are provided to the value labelling segment. However, this approach precludes the use of the designed strategy for value labelling without additional speaker context. For example, in a passage containing two speakers indicating differing variable values, the modified passage retrieval system would identify the passage as relevant for both speakers. However, providing the passage to the value labelling segment would pose a challenge for the model in identifying the correct value, given the absence of a distinct value for the passage. Therefore, this approach would also necessitate a modification to the value labelling segment, which would lead to two systems facing more complex tasks, and thus, likely a larger decrease in performance.

3.3.3 Source Attribution in Value Labelling

The final candidate approach is to include source attribution as part of the value labelling phase. Given a set of speakers present in the passage, the model would be prompted to identify the variable value, based on each specific speaker or, in other words, according to that speaker's point of view. This approach has several advantages over the previously outlined methods.

Firstly, it enables the capture of all values identified by a single speaker, since it performs labelling per-speaker, and not per-variable first. As a result, the confidence score is also reflective of assigning a distinct variable value to a distinct speaker. In contrast to the previous two methods, the case where multiple speakers give the same or different values in a single passage can be handled, and the speaker-value pairs extracted separately.

The only disadvantage of this method is that the model will be prompted for speakers that are irrelevant, i.e. who do not give any indication to any of the variable values. This is due to the fact that labelling is performed for every speaker appearing in the passage, and will have no information if that speaker discusses that variable.

3.3.4 Implementation

Taking into account all of the advantages and disadvantages of the proposed approaches, the decision was made to include the source attribution in the value labelling stage, as it best aligns with the objective of correctly identifying the variable values given by a speaker, and is expected to result in the smallest decrease in performance compared to the single-speaker approach.

The implementation of this involves two main modifications to the single-speaker scenario. The first modification is the extension of the passage retrieval component to store the speakers present in each passage.

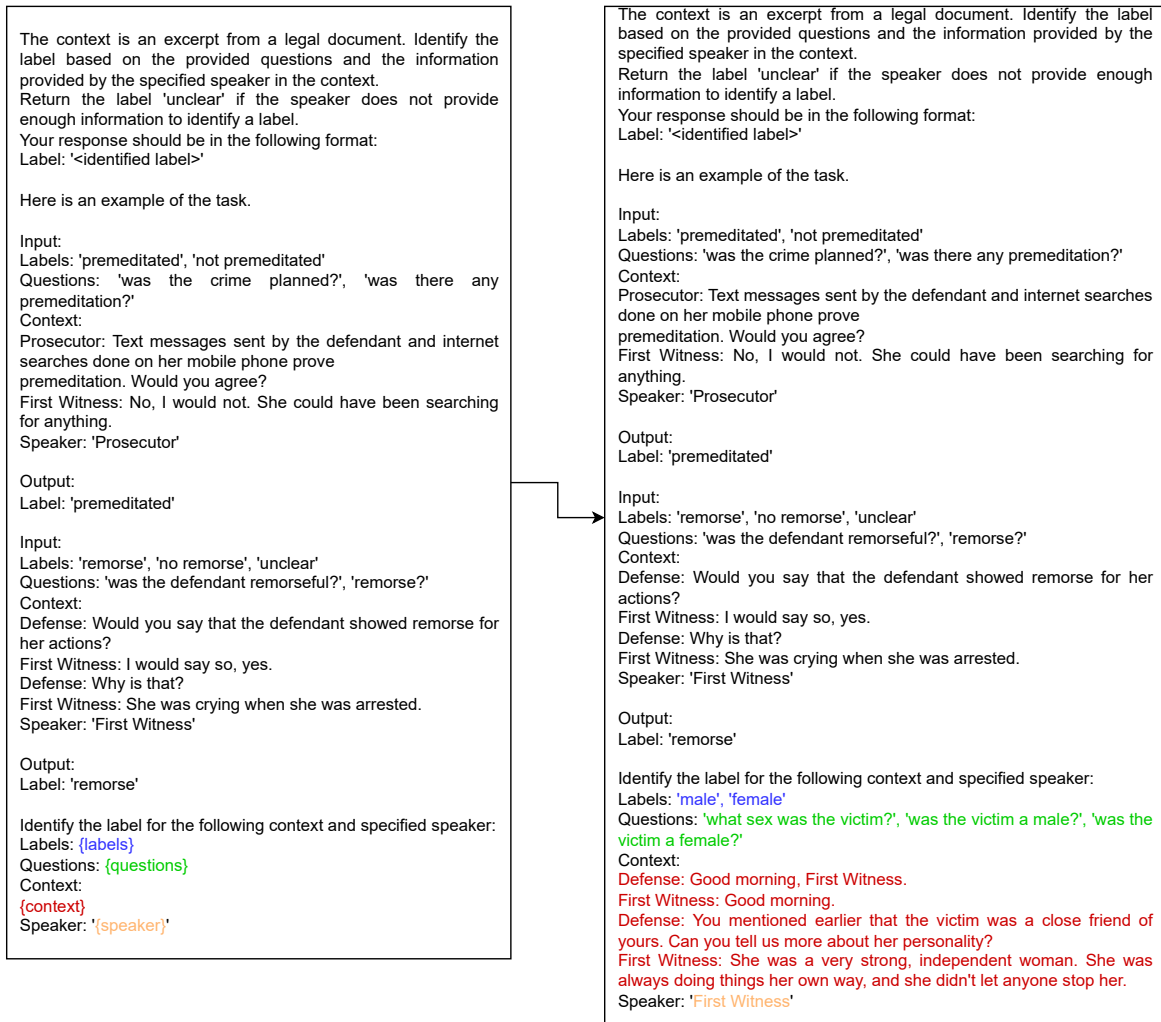


Fig. 3.4 Prompt template (left) and example prompt (right) for the value labelling based on the specified speaker. The prompt is a modified version of the prompt in Figure 3.3, but instructs the model to assign the value based on a specific speaker's point of view.

With every passage, the dialogue tag pattern matching is run to identify the speakers in it. If no dialogue tags are present, a check is performed to see if a speaker was assigned to the previous chunk. If it was, this suggests that a single speaker's speech was split into two passages, and thus the same speaker is assigned. If the previous chunk did not have a speaker assigned to it, then it means that the passage does not contain direct speech, and it is assigned the default speaker (narrator, or first person case).

In the passage retriever, all relevant passages for each speaker in the document are gathered. In both the vector index and the LLM-based approach this is done by only searching the passages which have been assigned the specific speaker. This results in a mapping of "speaker" to "relevant passages". This mapping is then used to execute the value labelling for each speaker's relevant passages.

In situations where only a single speaker is present in a passage, the prompt described in Section 3.2.3 is used. If there are multiple speakers in a passage, a modified prompt template, given in Figure 3.4, is used. The prompt is designed for the new task of identifying a variable value based on information from a specific speaker. The format of the prompt is similar to the prompt in Figure 3.3, with the addition of the speaker as part of the input and instructions to identify the label based on it.

The probabilities of each variable value are computed and normalized as described in Section 3.2.3. The Top-k passage-value pairs for each speaker and variable are presented to the user for validation.

3.4 Fine-tuning

The task of obtaining a complete supervised dataset for full fine-tuning and evaluation in this project's setup is practically infeasible. The documents are very long, containing many passages when split up, and each of these passages would need to be manually labelled for every single variable. However, the human validation component in ELICIT yields a smaller supervised dataset, which can be used for fine-tuning to achieve two goals: improving performance on unseen documents, and presenting new extractions in already analyzed documents.

A single human validation run gives two types of information for each passage shown to the user - whether its relevant for that variable, and if the assigned value is correct for the passage and speaker. This data can be used to further improve the LLM-based passage retriever and value labelling components through fine-tuning.

In order to evaluate the impact of fine-tuning on unseen documents, four documents are held for testing in the sentencing remarks dataset, and two in the court transcripts dataset. The remaining data samples are split into training and validation sets using an 80-20 ratio. Stratified sampling was applied to ensure the proportions of variables and variable values were consistent across the training and validation sets. Hyperparameter tuning was accomplished using a grid search to find the configuration resulting in the lowest validation loss.

After a single human validation run, there are some variables which had to be abstained, since the presented passages did not contain relevant or sufficient information. While some

of these are true abstain data points, the rest represent imperfect recall and the potential for further improvement. Therefore, the models are fine-tuned on the full set of extractions presented to the user, keeping 20% of the samples as validation data for early stopping, and the extraction is re-run again with the fine-tuned models. The human validator then validates the previously abstained variables with new extractions presented. This is expected to improve recall, and it also creates the opportunity to iteratively improve the model with newly validated data used as training examples.

The model fine-tuning is accomplished using Low-Rank Adaptation (LoRA), as introduced in Section 2.3.5. A result of this method is that only a small fraction of parameters are trainable, resulting in a small LoRA adapter containing the weight differences, which can be applied to the base model. The cross-entropy loss function was used with the input (prompt) tokens masked, which means that they were ignored when computing the loss and the model was only trained on the output tokens. This was done, since the model is used in a sequence-to-sequence fashion, where the objective is to maximise the probability of the output sequence given the input sequence:

$$\theta_{seq2seq}^* = \arg \max p(y_{N+1:N+T+1} | y_{0:N}; \theta) \quad (3.5)$$

where y_i is the i -th token, N is the length of the prompt, and T is the length of the completion. This is in contrast to the default auto-regressive objective of maximising the probability of the next token given the previous tokens:

$$\theta_{AR}^* = \arg \max p(y_0; \theta) \prod_{i=1}^N p(y_i | y_{0:i-1}; \theta) \quad (3.6)$$

3.5 Metrics

This section is dedicated to discussing the metrics used in the project to assess the performance of the different parts of the system. These metrics can be categorized into two distinct groups: end-to-end metrics and supervised dataset metrics.

3.5.1 End-to-End Metrics

End-to-end metrics are employed to evaluate the performance of the full information extraction system, which includes human validation. They are computed using a manually labelled gold standard for a set of documents. These metrics are used to compare the LLM-extended system against the earlier version of ELICIT, and also to compare between different configu-

rations of the components, such as the different passage retrievers, and the base model versus fine-tuned model.

Precision

Precision, as defined in Equation 3.7, measures how many non-abstain predictions were correct. The precision is largely determined by the human validator, hence the system is expected to have near-perfect precision. The only instance where losses in precision may appear are when the passages provided to the user are misleading, and do not contain the full context.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} = \frac{\text{Correct Non-Abstain Predictions}}{\text{Total Non-Abstain Predictions}} \quad (3.7)$$

Recall

Recall, as defined in Equation 3.8, measures how many of the true non-abstain samples were identified by the system. Assuming a near perfect ability by the human validator to identify the correct variable value given a relevant passage, the recall is largely determined by the capability of the automated component to retrieve a relevant passage and label it accurately. The Top-k parameter, that is, the number of extractions shown to the human validator, controls the trade-off between recall and time-efficiency. A large Top-k will require the user to scan through many extractions, increasing the likelihood of identifying the correct value, but consuming more time in the process. The objective of the project is to enhance the recall of system by improving the automated component.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} = \frac{\text{Correct Non-Abstain Predictions}}{\text{Total Non-Abstain Data Points}} \quad (3.8)$$

3.5.2 Supervised Dataset Metrics

The supervised metrics are a set of metrics used to evaluate the individual components of the system, specifically to compare the fine-tuned models against the base models. As explained in Section 3.4, a single extraction run with human validation results in labelled datasets of passages and values. Despite being imbalanced, containing significantly more irrelevant passages, these datasets represent a true use case of doing extraction with the system and improving it through human validation and fine-tuning.

Receiver Operating Curve and Area Under the Receiver Operating Curve

The Receiver Operating Characteristic (ROC) curve is used to represent the performance of a model in a binary classification task. It is a plot of the true positive rate (recall) in the y-axis against the false positive rate in the x-axis at different classification thresholds applied to the model's predicted probability for the positive class.

The ROC curve is employed in the passage retriever analysis, since it is a binary classification task - a passage is labeled as relevant or not. The ROC curve gives us an insight into the performance of the retriever at various thresholds.

The Area Under the ROC curve (ROC-AUC) gives an aggregate measure of the classifier's overall performance. It quantifies the probability that a randomly chosen true positive sample will be ranked higher, i.e. have a higher confidence score, than a randomly chosen true negative sample (Hanley and McNeil, 1982). This aligns with the passage retrieval task, where the top- n passages based on their confidence scores are retrieved, thus the aim is for the classifier to rank the true positives higher than the true negatives.

Precision-Recall Curve and Average Precision

The Precision-Recall (PR) curve illustrates the effectiveness of the binary classifier predictions. It is a plot of the precision (y-axis) against recall (x-axis) at varying classification thresholds. In the context of passage retrieval, the precision measures how many of the retrieved passages are truly relevant, while the recall measures how many of the true relevant passages are retrieved. The PR curve better reflects a classifier's performance in heavily imbalanced datasets than the ROC curve (Davis and Goadrich, 2006; Saito and Rehmsmeier, 2015), since the precision is not skewed by a large number of negative samples present in the dataset, whereas the ROC curve treats both false negative and false positives equally.

The Average Precision (AP) is a summary of the PR curve, which measures the average precision across all recall levels, and is commonly used for classification tasks with imbalanced datasets (Sofaer et al., 2019). In theory, it is computed as the integral of the PR curve, but in practice the following formula is used:

$$\text{AP} = \frac{1}{\sum_{i=1}^N \text{rel}(i)} \sum_{i=1}^N (P(i) \times \text{rel}(i)) \quad (3.9)$$

The term $P(i)$ is the precision at cut-off level i (only including the top- i results), while the term $\text{rel}(i)$ is the relevance score for a result at position i , with 1 if the result is relevant, and 0 otherwise.

AP is also employed to evaluate the value labelling system. The computation of AP is reflective of the use case of ELICIT. In the value labelling part, each passage receives a confidence score for each non-abstain (or non-"unclear") value. The Top-k of these are shown to the user for human validation. The objective is to ensure that most of these results (passage-value pairs) have the correct value assigned to them. Therefore, AP is used in the same way as in the binary case, by assigning $\text{rel}(i)$ to 1 if the result is assigned the correct value, and in $P(i)$ computing the amount of correct results present in the result set with cut-off i .

Normalized Discounted Cumulative Gain

The Normalized Discounted Cumulative Gain (nDCG), commonly used in information retrieval tasks, evaluates the quality of retrieved results by taking into account their relative scores or ranking. It is used according to the assumption that relevant results should appear higher in the results list. In the project, this metric is utilized to assess the effectiveness of the passage retrieval task, where the objective is to assign higher relevance scores to truly relevant passages. Similarly, in the value labelling task, the aim is to place the correct variable value assignments higher according to their confidence score. Both objectives aim to improve the recall of the end-to-end system for smaller Top-k settings, showing fewer but relevant extractions to the human validator. The nDCG score is computed using the same approach as AP, but with the following formula:

$$\text{nDCG}_p = \frac{\text{DCG}_p}{\text{IDCG}_p} \quad \text{DCG}_p = \sum_{i=1}^p \frac{\text{rel}(i)}{\log_2(i+1)} \quad (3.10)$$

where the parameter p defines the number of results under consideration, while $\text{rel}(i)$ is the relevance score as in Equation 3.5.2. The score is normalized by dividing the score with the ideal DCG score (IDCG), which, in this context, represents the scenario where all correct documents or values appear at the top of the result list. This normalization enables the comparison of different queries, which, in this task, corresponds to different variables.

The nDCG score is related to AP as both take into account the order of the presented results. However, the former emphasizes placing the correct results at the top of the list, whereas the latter focuses on maximising the number of correct results in the list. Thus, in the passage retrieval system, AP is a better indicator of performance, given the goal of providing the value labelling part with the maximum amount of truly relevant passages. Conversely, in the value labelling part, the focus is on displaying the correct extraction to the human validator as high as possible in the results list, making nDCG a better indicator of

its effectiveness. Both metrics are presented for the two tasks to provide a comprehensive performance evaluation.

Chapter 4

Results and Discussion

In this chapter, I present and discuss the outcomes of several experiments conducted using our LLM-enhanced information extraction system on the sentencing remarks and court transcripts datasets. Both sections follow a structured analysis of individual components - the passage retriever and value labeller. This is followed by the evaluation of end-to-end performance and recalibration. In Section 4.2, I also address limitations related to speaker-specific information extraction, and how fine-tuning impacts overall performance.

4.1 Sentencing Remarks Dataset

4.1.1 End-to-end Performance

In this section, the results from using ELICIT with all previous labelling functions, referred to as ELICIT-1.0, are compared to the results from LLM-enhanced ELICIT with the vector index passage retriever (ELICIT-VI), and the LLM-based passage retriever (ELICIT-LLM).

In the human validation step, the validator is presented with the Top-k extractions to validate if the assigned label is correct based on the passage presented. If none of the extractions are valid, then the variable is assigned the "abstain" value.

It is important to consider that the end-to-end results are validator specific, since the human validator determines if the extraction contains sufficient information to be assigned a specific label. Nevertheless, they are sufficient to highlight the differences in performance between the different configurations. In this project, I validated the extractions, however, in the future, a more comprehensive analysis should be accomplished with multiple different validators.

As detailed in Section 3.5.1, precision is primarily determined by the human validator, with losses potentially occurring if the presented passage is misleading. The difference in

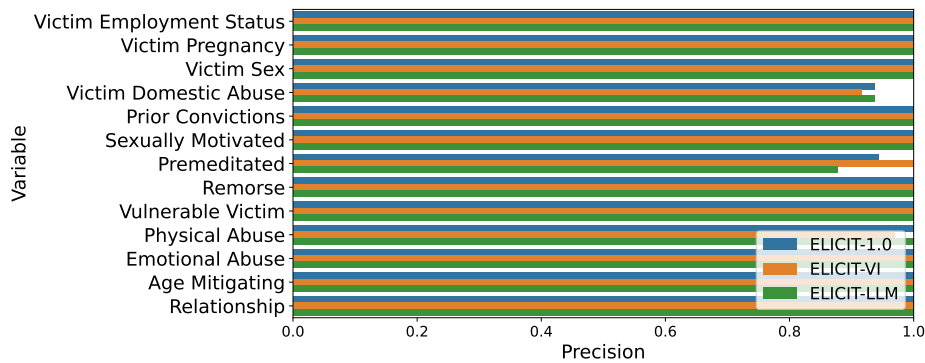


Fig. 4.1 Top-3 weighted precision on the complete sentencing remarks dataset.

observed precision between different systems is not substantial, as seen in Figure 4.1, nor does it provide a significant edge for any system. Typically, precision losses result from the validator indicating a negative value, for example "no physical abuse", when the true value was "abstain".

The recalls of the three configurations are compared in Figure 4.2 for Top-1 and Top-3. The recall is weighted by per-class support to avoid skewing results due to imbalanced class distributions. In Top-3, ELICIT-VI performs better or equally than ELICIT-1.0 in six variables, while ELICIT-LLM outperforms ELICIT-1.0 in seven out of thirteen variables. When we look at the recall averaged across all variables, both LLM-enhanced systems outperform ELICIT-1.0, by 0.06 and 0.11, respectively.

Using Top-1 reduces the recall for the LLM configurations by approximately 0.2, as seen in Table 4.1, suggesting a weaker automated performance. A performance drop is also observed with ELICIT-1.0, but it is less significant, only by 0.13. The difference in Top-1 between the LLM-enhanced systems and ELICIT-1.0 is less noticeable, by only 0.02.

Table 4.1 Top-1 and Top-3 mean recall across all variables for different configurations on the complete sentencing remarks dataset.

Configuration	Recall	
	Top-1	Top-3
ELICIT-1.0	0.3426	0.4769
ELICIT-VI	0.3685	0.5397
ELICIT-LLM	0.3687	0.5892

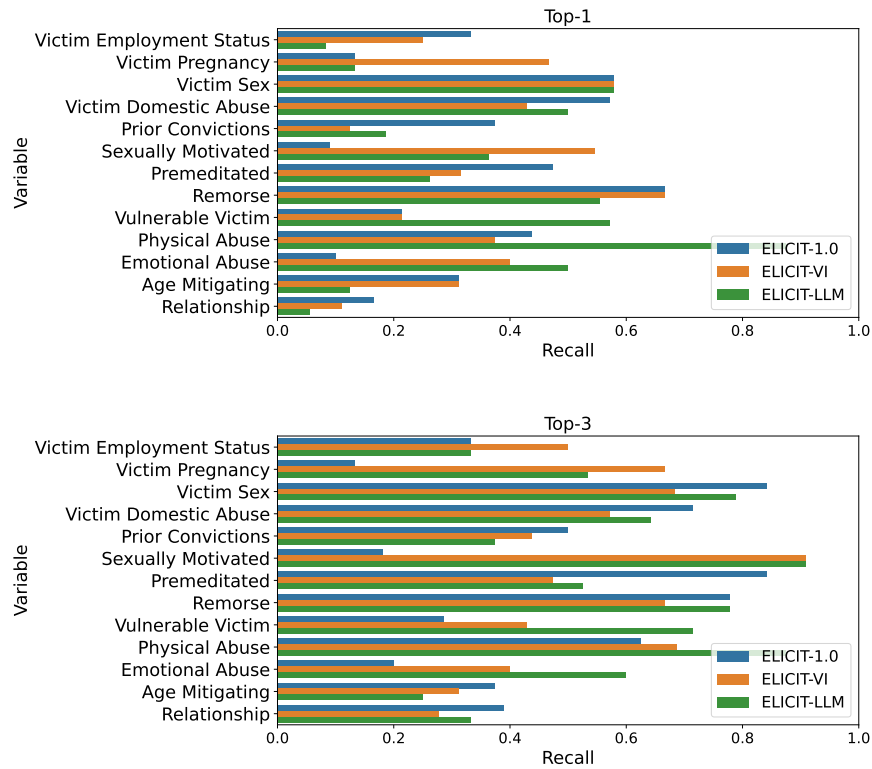


Fig. 4.2 Top-1 and Top-3 weighted recall on the complete sentencing remarks dataset.

4.1.2 Fine-tuned Performance on Unseen Documents

In this part, the performance of the passage retriever and value labeller with fine-tuning is analyzed. Both components are fine-tuned using a supervised dataset, which was created by validating the extractions of 15 out of the 19 sentencing remarks with a Top-3 setting, resulting in 585 extractions. The components are evaluated on the remaining 4 sentencing remarks (156 extractions). The fine-tuning details for this task are presented in Appendix A.1.

Passage Retriever

The different retrievers were tested in a binary classification manner to determine if the passage is relevant or not, as labeled by the human validator (the user of ELICIT). For the Vector Index retriever, the relevance scores between the query and passages were calculated as is described in Section 3.2.2. The ROC and PR curves, given in Figure 4.3, computed across all variables together, showcase the general classification capacity of our classifier. The ROC curve and ROC-AUC score demonstrate that the base model retriever struggles

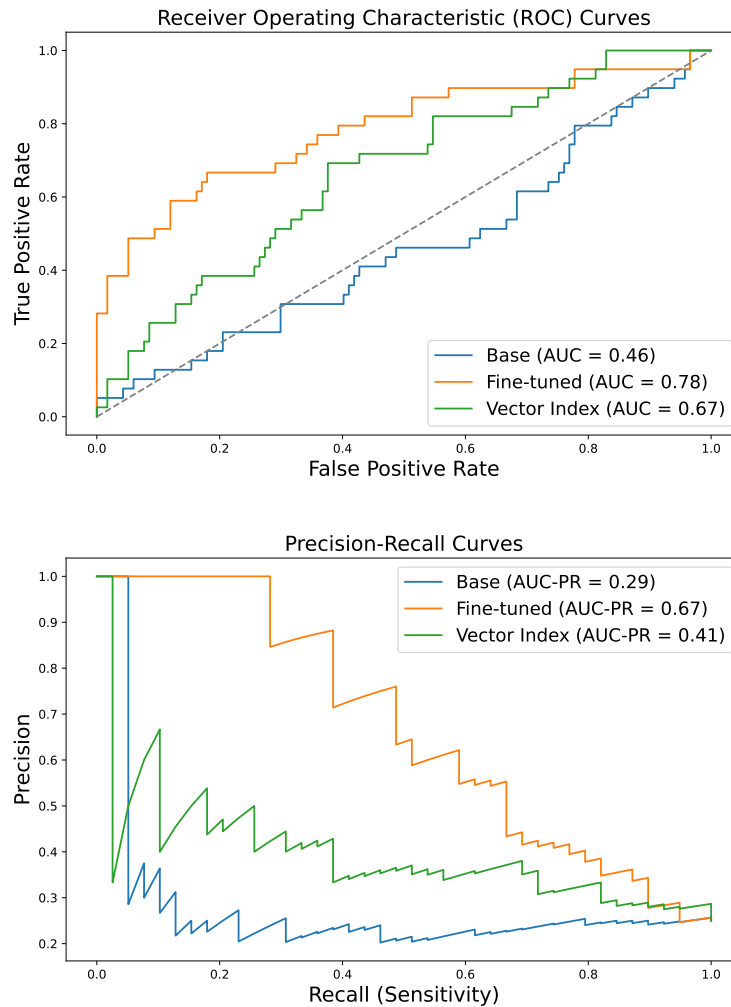


Fig. 4.3 ROC and Precision-Recall curves for different passage retrievers on supervised sentencing remarks data.

with this task, performing worse than random guessing¹. Despite this, the fine-tuned model excels, even surpassing the embedding model, in terms of ROC-AUC.

The PR curve displays a similar performance trend, with the fine-tuned model performing the best, resulting in higher precision over all recall levels. Both metrics indicate the improvement of the retrievers' ability to classify relevant from irrelevant passages when the model is fine-tuned on validated data, even when most of the samples are negative.

The nDCG score and Average Precision (AP) are utilized to evaluate how well the retrievers rank the passages by relevance for each variable. While ROC and PR are computed

¹The classification decision could be reversed in this case. This was tested and did not lead to better performance in other metrics due to the imbalanced dataset.

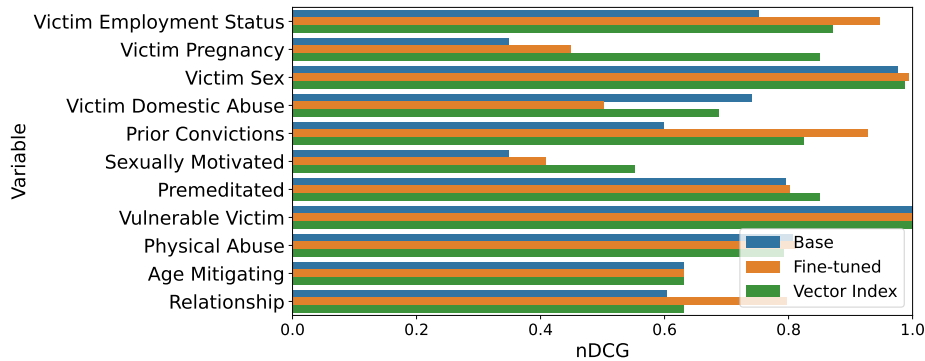


Fig. 4.4 Per-variable nDCG score for different passage retrievers on supervised sentencing remarks data.

for all predictions in the supervised dataset, these are computed per-variable by considering the predictions of each variable separately². In terms of these ranking scores, as illustrated in Figures 4.4 and 4.5, fine-tuning enhances or equals the performance in all variables except for "Victim Domestic Abuse".

The fine-tuned model has the highest score in 7 out of the 11 variables³, indicating strong overall performance. When the scores are averaged across variables, the fine-tuned retriever has lower nDCG and AP than the vector index retriever. This outcome is a result of the fine-tuned model marginally improving in most variables and surpassing the vector index, but it performs significantly worse in a few variables, most notably in "Victim Pregnancy". No notable correlation was found between the number of positive training samples for a variable and its corresponding scores. Overall, these results indicate that the passage retriever responds positively to fine-tuning from a relatively small amount of human validated data, and it is anticipated to perform even better with more validation. Increasing the Top-k setting during the validation of the first few documents could be a viable solution.

Value Labeller

In the value labelling task, the nDCG score increases in six variables, and slightly decreases in five for the fine-tuned model compared to the base model, as given in Figure 4.6. Overall, the nDCG score is higher by 0.07 on average across all variables⁴. Due to the model being more likely to label variables as "unclear", and the prevalence of it in our dataset, another model was trained with the non-"unclear" data points oversampled to match the number of

²The PR-AUC in Figure 4.3 is equivalent to AP computed for all predictions.

³Variables which did not have positive samples were removed when computing nDCG and AP.

⁴Uniform averaging is used when computing the average across all variables, i.e. each variable has equal weighting.

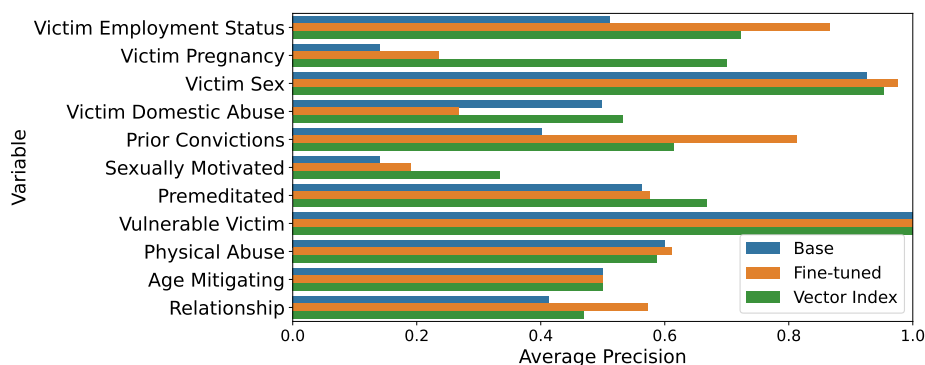


Fig. 4.5 Per-variable Average Precision for different passage retrievers on supervised test sentencing remarks data.

"unclear" samples for each variable. This led to a better overall performance in terms of nDCG, outperforming the base model in six variables, and the regularly fine-tuned model in seven. Averaged across all variables, the nDCG is higher by 0.124 than the base model. This shows that when dealing with an imbalanced dataset, the model benefits from seeing the positive samples more frequently, and thus results in better ranking capabilities.

The nDCG score evaluates the ability of the model to place the correct predictions higher. As the fine-tuned models result in higher nDCG, this is expected to transfer to stronger performance in smaller Top-k settings, as the quality of extractions returned will be better.

In terms of AP, visualized in Figure 4.7, both fine-tuned models displayed improvements in seven variables. On average, the standard fine-tuned model achieved an AP score higher by approximately 0.13, while the model trained on oversampled minority data showed a larger improvement of 0.18. AP summarizes the precision of the models at various recall levels, thus, this together with the nDCG scores signifies the strength of the fine-tuned automated

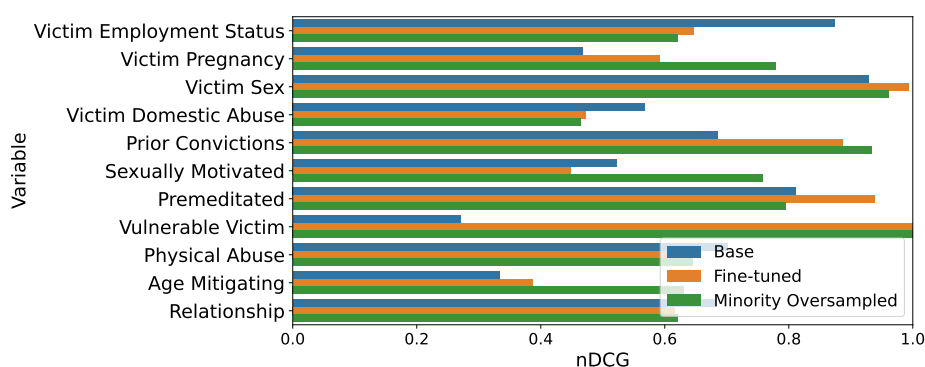


Fig. 4.6 Per-variable nDCG score for different value labellers on supervised test sentencing remarks data.

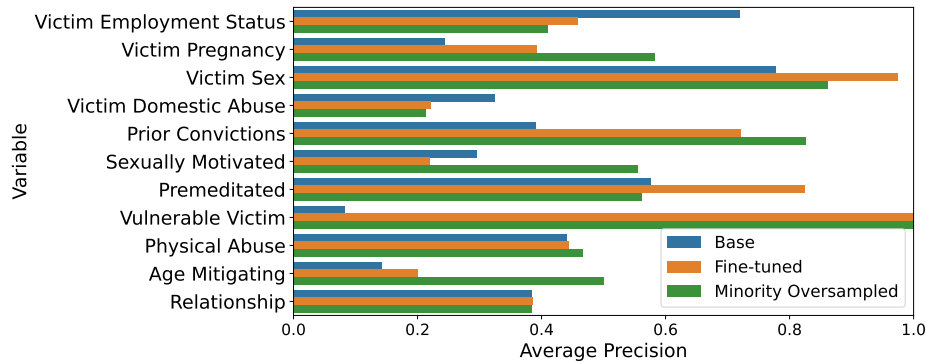


Fig. 4.7 Per-variable Average Precision for different value labellers on supervised test sentencing remarks data.

component of ELICIT, and suggests that using human validated data is expected to result in better value labelling on unseen documents.

End-to-end Performance

Since the supervised dataset was created from a single human-validation run, it means that the components were evaluated only on a fraction of the passages from the documents. By running a complete extraction process using fine-tuned models, we can see how the system operates when presented with full unseen documents. As given in Table 4.2, for Top-3, ELICIT-1.0 performs worse than ELICIT-VI, but slightly better than ELICIT-LLM. However, fine-tuning resulted in a significant improvement for the LLM-enhanced systems, with FT-ELICIT-VI improving by 0.09, while FT-ELICIT-LLM improved by 0.084, resulting in a higher recall than ELICIT-1.0. Overall, for Top-3, FT-ELICIT-VI has the highest recall, consistent with the stronger per-variable nDCG and AP scores of the vector index retriever.

An important observation is the enhanced performance in Top-1, which is the most restrictive setting, requiring the best performance from the automated components in a narrow recall window. FT-ELICIT-VI only improves by 0.04 from its base configuration, while FT-ELICIT-LLM improves by close to 0.15, even showing better performance than FT-ELICIT-VI. The difference in these two configurations indicates the differing performance of the passage retrievers for small Top-k settings. FT-ELICIT-LLM performs better in Top-1, which suggests that the LLM-based retriever is better at positioning the most relevant passage at the top, however, in Top-3, FT-ELICIT-VI achieves the highest recall, meaning that it places better passages at the second and third position. As more and better-quality data are used for training, it is expected that FT-ELICIT-LLM would surpass FT-ELICIT-VI across a wider Top-k range.

Table 4.2 Top-1 and Top-3 recall on unseen sentencing remarks for different configurations. "FT" refers to fine-tuned configurations.

Configuration	Recall	
	Top-1	Top-3
ELICIT-1.0	0.3141	0.4231
ELICIT-VI	0.3397	0.4744
ELICIT-LLM	0.2500	0.4038
FT-ELICIT-VI	0.3782	0.5641
FT-ELICIT-LLM	0.3974	0.4872

In conclusion, these results demonstrate the strength of using human validated data from extractions on other documents to fine-tune both the passage retriever and the value labeller. They both learn to position the most relevant passages and values at the top of the result list and can extract information better from unseen documents.

4.1.3 Fine-tuning for Recalibration

After the model is fine-tuned using labeled data from a single human validation run, the extraction process could be run again to identify new potential extractions from previously abstained variables. This potential improvement in the LLM-enhanced system primarily originates from two sources: the fine-tuned LLM passage retriever returning more relevant passages; and the fine-tuned value labeller assigning values with a greater accuracy and confidence. As illustrated in Figure 4.8, the performance of the fine-tuned ELICIT-LLM system showcases discernible improvements. Specifically, out of 13 variables, enhancements were observed in 10, increasing the average recall by 0.097 to 0.687. Conversely, ELICIT-VI exhibited a slightly smaller increase by 0.06, attributable mainly to the enhancement in the value labeller alone.

Overall, these results showcase the significance of human-validated data as an invaluable information source for the models. Using this data not only improves the performance on already extracted documents but also enhances the extraction of previously unseen ones through fine-tuning the models.

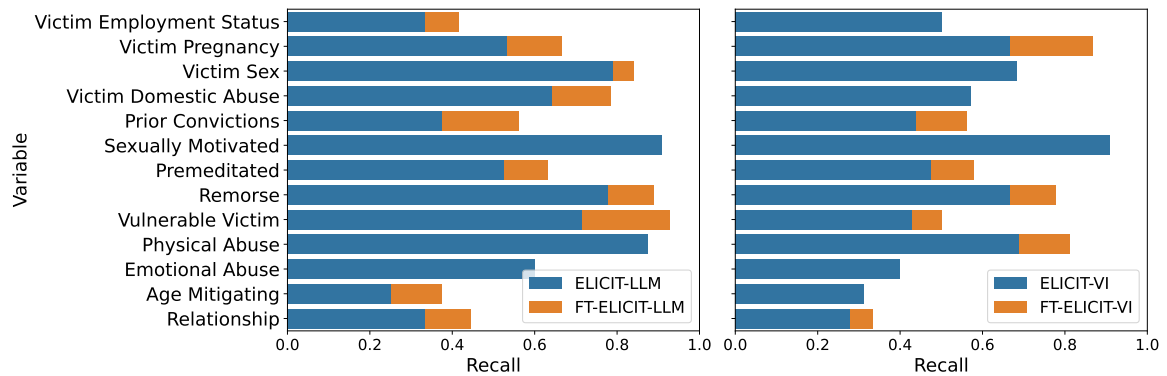


Fig. 4.8 Improvement in recall for ELICIT-LLM and ELICIT-VI from recalibration using fine-tuning on extracted sentencing remarks.

4.2 Court Transcripts Dataset

4.2.1 Performance on Unseen Documents

In this part, the passage retriever and value labeller were fine-tuned on validated Top-3 extractions of 3 out of 5 court transcripts, resulting in 663 extractions. The remaining transcripts form the test set containing 485 extractions. The fine-tuning details for this task are presented in Appendix A.2.

Passage Retriever

Transcripts of court proceedings have a known dialogue structure containing speaker tags, and featuring questions followed by answers. This inherent structure provides contextual cues that aid passage retrievers in assessing the passage relevance. In contrast, sentencing remarks lack these cues and contain more nuanced and indirect language.

These differences manifest in the performance of passage retrievers on unseen human validated court transcript passages. The ROC curve, presented in Figure 4.9, demonstrates better performance across all passage retrievers compared to the sentencing remarks dataset in Figure 4.3. The fine-tuned retriever emerges as the best with an ROC-AUC of 0.91. In comparison, the base retriever and vector index achieve scores of 0.65 and 0.70, respectively.

While ROC evaluates the general performance of the binary classifier, the Precision-Recall suggest how the model performs considering class imbalance. The retriever, fine-tuned on user-validated extractions from court transcripts, shows strong performance with an AUC-PR of 0.70, while the lower scores for the base and vector index retriever suggest them struggling with the minority class. The AUC-PR achieved by the retrievers on the court transcripts is similar as on the sentencing remarks, hence, the difference in AUC-ROC might

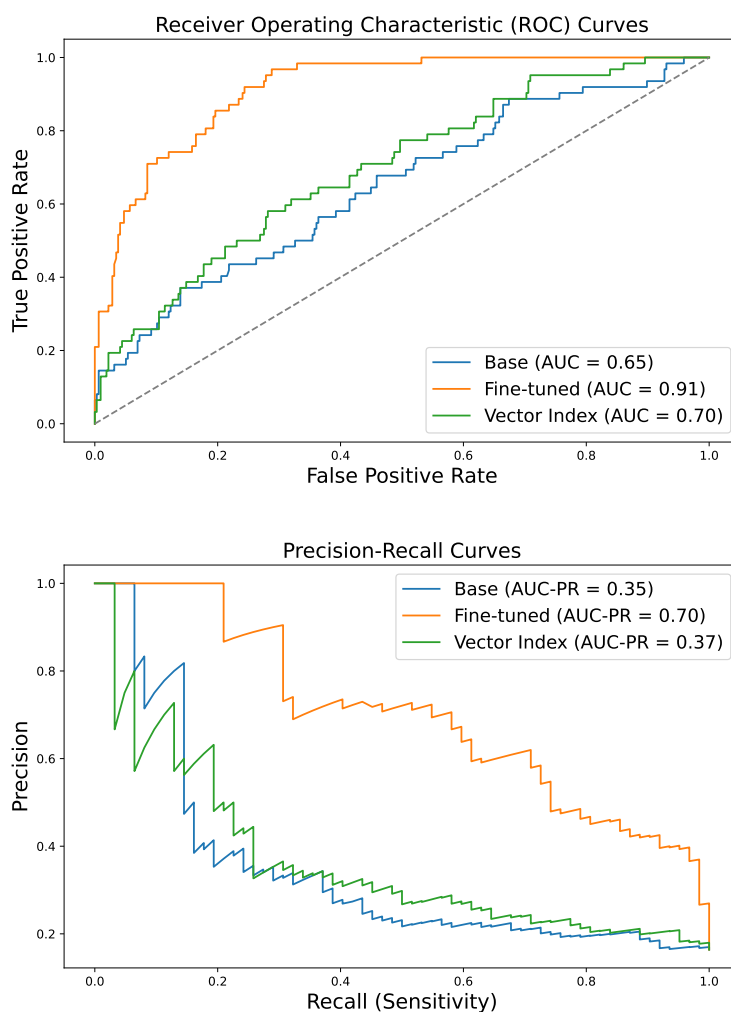


Fig. 4.9 ROC and Precision-Recall curves for different passage retrievers on supervised court transcript data.

be due to its sensitivity to class imbalance. The court transcripts dataset contained fewer positive samples, specifically 16%, compared to 25% in the sentencing remarks.

Evaluation using nDCG and AP reveals similar trends. In terms of nDCG, which focuses on how well the retriever positions relevant passages at the top of the results list, the fine-tuned retriever achieves the highest mean score of 0.85, improving from the base model's score of 0.66, and ahead of the vector index score of 0.79. In terms of AP, which measures precision across varying recall levels, the fine-tuned retriever leads again with a mean score of 0.74, while the vector index and base model score 0.63 and 0.46, respectively.

The improvements in nDCG and AP across all variables, as seen in Figures 4.10 and 4.11, highlight the fine-tuned retriever's effectiveness on court transcripts. Furthermore, it

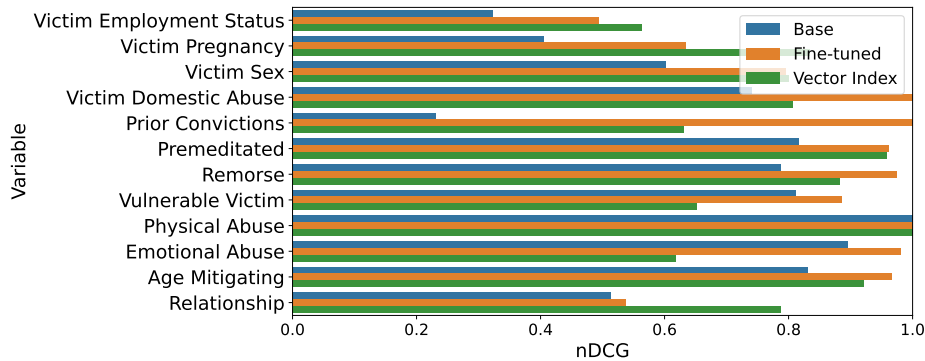


Fig. 4.10 Per-variable nDCG score for different passage retrievers on supervised test court transcript data.

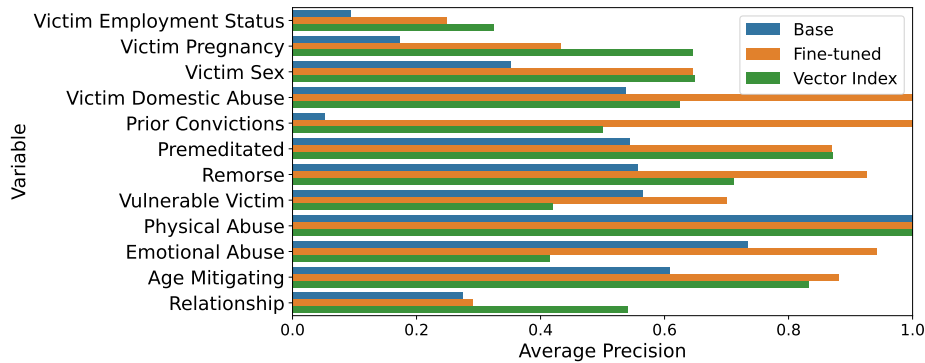


Fig. 4.11 Per-variable Average Precision for different passage retrievers on supervised test court transcript data.

showcases better performance across most variables compared to the vector index. These results indicate the model's successful learning from human-validated data, resulting in superior ranking and precision capabilities on unseen court transcript documents.

Value Labeller

The value labelling component, used for dialogue texts, is responsible for assigning variable values to passages based on specific speakers. If a speaker does not provide a value, it is instructed to assign the "unclear" value. The evaluation of nDCG and AP follows the same approach as in the sentencing remarks scenario, by comparing every non-zero probability non-"unclear" prediction with the correct label. As in Section 4.1.2, the model is fine-tuned with two approaches: using the complete imbalanced training set, and with oversampling the minority values for each variable.

In terms of nDCG, the labeller improves significantly from fine-tuning. On average, the regular fine-tuned model shows an improvement of 0.21, while the oversampled model

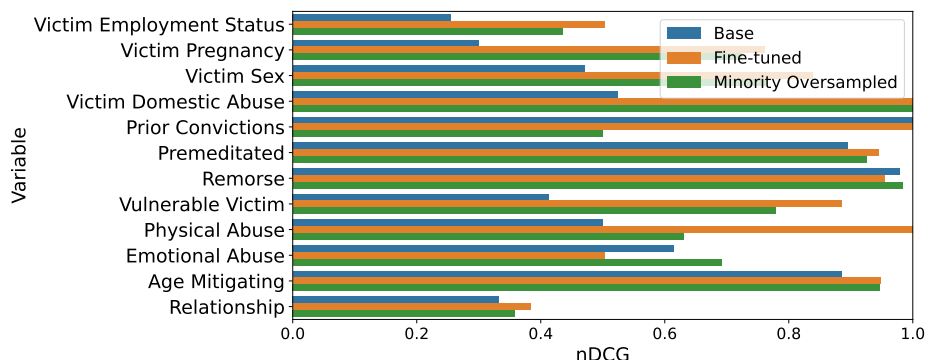


Fig. 4.12 Per-variable nDCG score for different value labellers on supervised test court transcript data.

improves by 0.13. Furthermore, as illustrated in Figure 4.12, the regular fine-tuned model improves over the base model in 10 out of 12 variables, while the oversampled model improves in 11. Notably, the increases in the regular fine-tuned model are more significant, surpassing the oversampled model's nDCG scores in all 10 of the improved variables.

With AP, as given in Figure 4.13, the same trends are apparent, with the regularly fine-tuned model resulting in higher mean AP, and improving in 9 variables over the base model.

The presented results outline the significant benefits of fine-tuning when dealing with the complexity of speaker-based labelling. This fine-tuning results in an improved value labeller, which more effectively places correct values at the top of the result list, and with greater confidence. Interestingly, it was also observed that contrary to the performance on sentencing remarks, oversampling the minority classes does not lead to better performance than using the full imbalanced dataset.

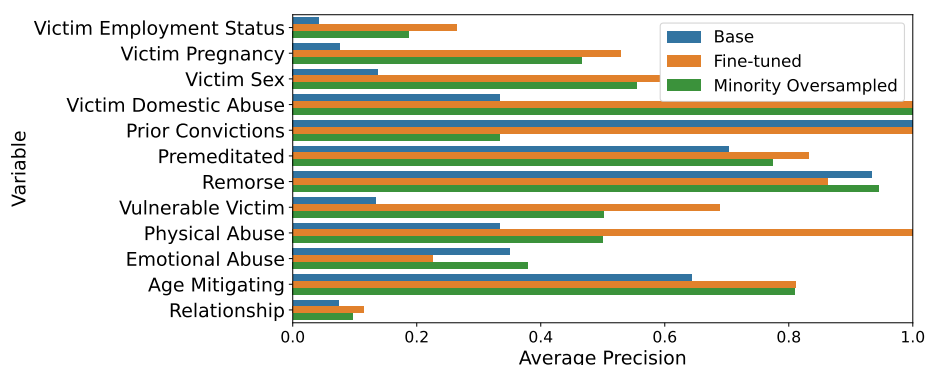


Fig. 4.13 Per-variable Average Precision for different value labellers on supervised test court transcript data.

Table 4.3 Top-1 and Top-3 recall results on unseen court transcripts for different configurations. "Per-variable" is obtained by taking the mean of the weighted recalls across all variables. "Total" is obtained by computing the recall considering all data points equally.

Configuration	Per-variable		Total	
	Top-1	Top-3	Top-1	Top-3
ELICIT-VI	0.4815	0.5858	0.5172	0.6724
ELICIT-LLM	0.3321	0.4595	0.4310	0.6207
FT-ELICIT-VI	0.5545	0.6494	0.6207	0.7759
FT-ELICIT-LLM	0.6390	0.7386	0.7586	0.8621

End-to-end Performance

The evaluation of the passage retriever and value labeller on the human validated supervised dataset serves as a proxy for the true end-to-end performance. This subsection focuses on the evaluation of the different configurations with base and fine-tuned models to determine if the performance improvements in the supervised dataset transfer to improvements on complete unseen transcripts with the human validation element.

As discussed in Section 4.1.2, precision is predominantly determined by the human validator rather than the automated component. Consequently, no significant differences in precision between various configurations were observed.

The mean recall results, given in Table 4.3, align with the trends identified in the evaluations on the supervised dataset. Among the base model configurations, the system using the vector index passage retriever (ELICIT-VI) performs better due to the better retrieving capabilities using the embedding model compared to the LLM (ELICIT-LLM).

With fine-tuning, the model with both LLM components (FT-ELICIT-LLM) achieves the highest recall. Notably, it improves from its base model by a more significant margin (0.31 in Top-1 and 0.28 in Top-3, per-variable) than FT-ELICIT-VI (0.07 in Top-1 and 0.06 in Top-3, per-variable), indicating that the majority of the recall gain comes from the fine-tuned passage retriever.

4.2.2 Fine-tuning for Recalibration

Fine-tuning the models with a complete human validation dataset (consisting of all five transcripts) and running extraction again to find new potential extractions results in similar trends as in previous tasks. As expected, the full LLM-enhanced system benefits significantly from fine-tuning due to the improvement of the value labeller and passage retriever, resulting in a mean improvement in recall of 0.12 per-variable and in total, as given in Table 4.4.

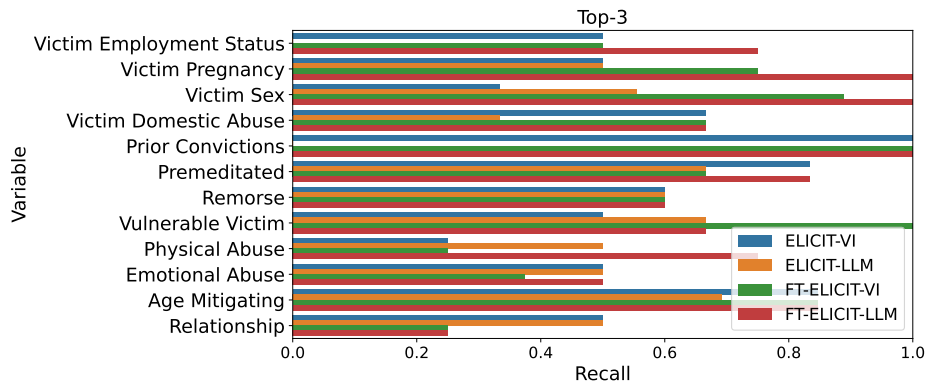


Fig. 4.14 Per-variable recall for different configurations on the unseen court transcripts with Top-3. FT-ELICIT-LLM outperforms its base configuration in 11 out of 12 variables, and achieves the highest overall recall in 10 variables.

Table 4.4 Improvement in mean total and per-variable recall on unseen court transcripts from recalibration through fine-tuning.

Configuration	Per-variable		Total	
	Original	Improved	Original	Improved
ELICIT-VI	0.5245	0.5828	0.5945	0.6667
ELICIT-LLM	0.5088	0.6290	0.5882	0.7059

Conversely, the configuration using the vector index passage retriever improves by 0.06 per-variable and 0.07 across all data points. The recalibrated ELICIT-LLM results in the highest overall recall, and it also improves in more variables (eight versus ELICIT-VI’s five), as visualized in Figure 4.15.

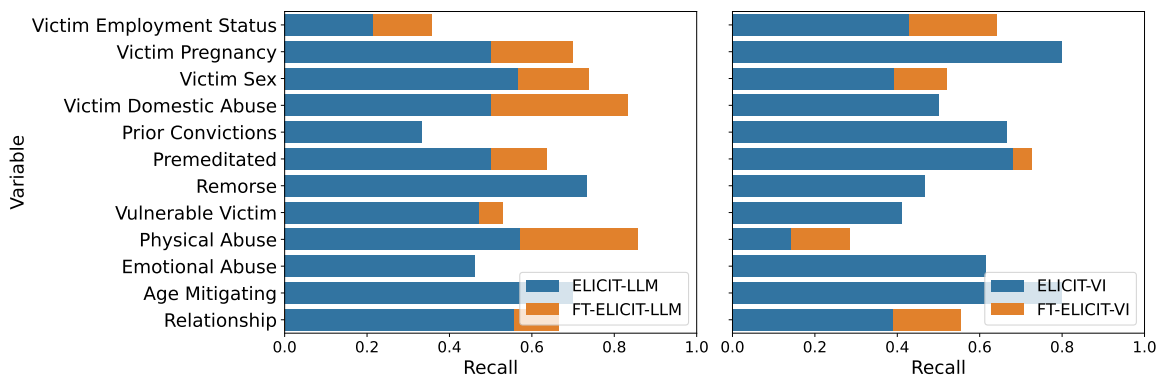


Fig. 4.15 Improvement in recall for ELICIT-LLM and ELICIT-VI from recalibration using fine-tuning on extracted court transcripts.

4.2.3 Recall and Shown Extractions

With the objective of finding every variable value indicated by a speaker, the number of extractions shown to the user can quickly increase and become cumbersome. Specifically, for a single document, the number of extractions shown the user is the product of the number of speakers, the number of variables, and k (in Top- k). While the user defines the number of variables and k , the speaker count is inherent to the document. A limitation of this is that any speaker which appears in the transcript will have extractions associated with it, even though in reality, they might not have given a relevant value. To avoid showing the user unnecessary extractions, a good automated component is essential.

This part details how for the same recall, fine-tuned configurations require fewer extractions be shown to the user, and how the fine-tuned labeller better separates relevant from irrelevant speakers for given passages.

Figure 4.16 illustrates the mean recall against the number of extractions shown to the user from a Top-3 validation run on the unseen court transcripts. The maximum number of extractions shown to the user is 448, when all top 3 extractions are shown for each speaker and variable combination. As fewer examples are shown to the user, based on their confidence scores, a logarithmic decrease in recall is observed across all configurations. This logarithmic trend signifies that, for equivalent recall levels, more efficient systems need a significantly smaller number of displayed extractions.

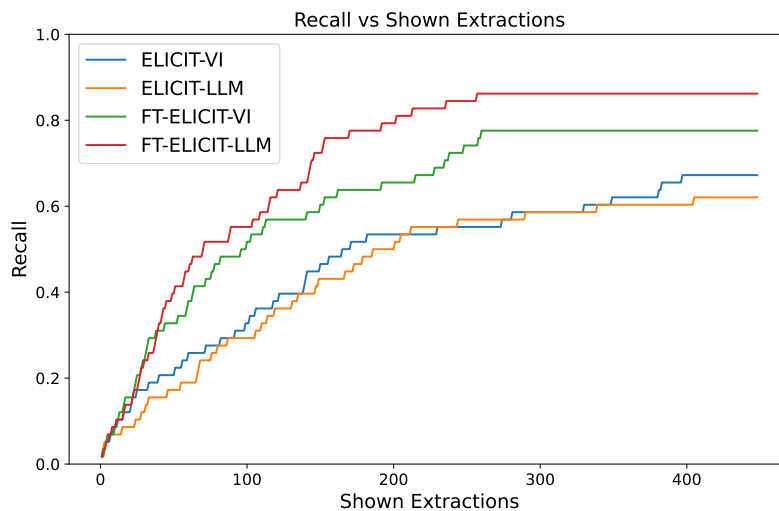


Fig. 4.16 Recall against number of shown extractions in a Top-3 run on unseen court transcripts. FT-ELICIT-LLM reaches the top recall level of ELICIT-LLM (achieved at 404 extractions) with just 120 extractions; of ELICIT-VI (achieved at 396) with 142; and of FT-ELICIT-VI (achieved at 259) with 191 extractions.

This outlined setup can be conceptualized as a global Top-k setting, which is applied across the entire extraction set rather than per variable. Thus, users can save manual extraction time while only marginally compromising on recall. This also indicates the benefits of fine-tuning on human-validated data, and as the models are trained on more data, they will require fewer extractions to be shown to the user.

In the analysis of unseen court transcripts, certain passages are repeatedly presented as part of extractions attributed to different speakers, however, not all of the speakers are relevant for the specific variable. An important part of our system is for the automated component to assign a lower confidence or rank for extractions associated with irrelevant speakers compared to those with relevant speakers.

To quantify this disparity in ranking, a comparative approach is employed by measuring the difference in rankings⁵ between relevant and irrelevant speaker extractions for identical passage-variable pairs, which were presented to the user for validation. This evaluation focuses on the configuration using the vector index passage retriever, enabling the isolation of the value labeller's performance.

It was found that FT-ELICIT-VI ranked the extractions of the relevant speaker on average 31.23 positions higher than the irrelevant one's, while ELICIT-VI only separated them by 9.18 positions. This suggests that fine-tuning improves the model's ability in differentiating relevant from irrelevant speakers for an identical passage and variable.

A specific example of this is the "Justice" speaker, which in the unseen transcripts only has procedural speech, not giving any valuable information. Thus, any extraction associated with it should be deemed irrelevant. However, the "Justice" dialogue tag appears in essential passages, such as the counsel's opening or closing remarks, which are full of important information. The value labeller is required to distinguish that the information in these passages is not associated with the "Justice" speaker, resulting in low confidence extractions. The distribution of per-variable rankings for each "Justice" extraction — where each is ranked compared to other extractions of the same variable — is depicted in Figure 4.17 for ELICIT-VI and FT-ELICIT-VI. Notably, both configurations effectively rank these extractions with low priority. However, the fine-tuned system displays a more favorable outcome, with a higher mean rank and reduced variance compared to the base configuration.

⁵The difference in rankings is used, since the confidence scores among various configurations are not calibrated, making them unreliable for comparison.

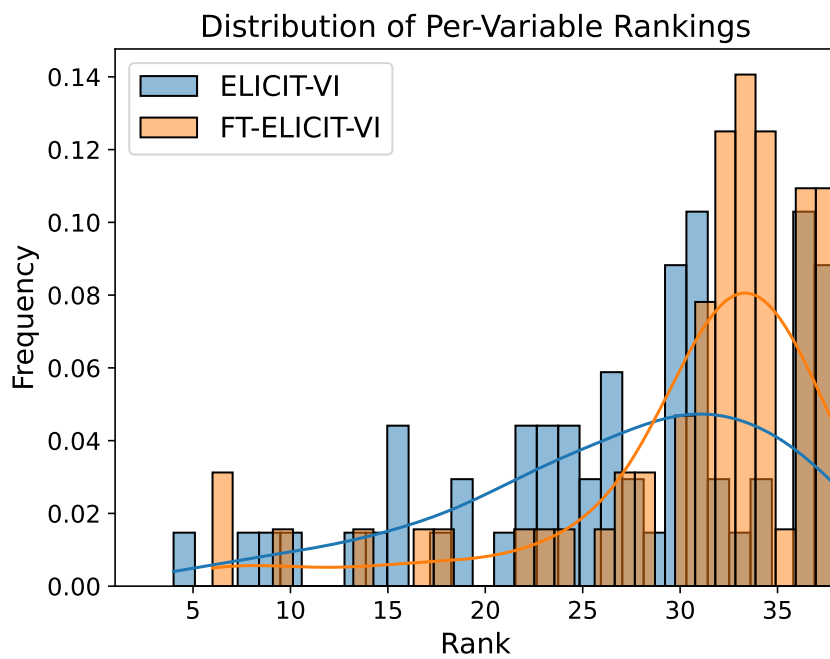


Fig. 4.17 Per-variable ranking distribution of extractions attributed to the "Justice" speaker.

Chapter 5

Conclusions

In this project I delved into the analysis and evaluation of utilizing LLMs for the extraction of structured information from long unstructured documents, specifically in the legal domain, focusing on sentencing remarks and court transcripts. This chapter presents a recap of the project, highlights the key findings, and proposes promising future research directions.

5.1 Project Recap

The primary objective of this project was to extend the capabilities of ELICIT by integrating the use of LLMs. The study focused on two particular tasks with distinct datasets: the extraction of single corresponding variable values from sentencing remarks, and the novel task of extracting speaker-specific variable values from court transcripts.

To overcome the challenges associated with the length of the documents, the approach of splitting the documents into smaller passages was taken. This resulted in the development and analysis of two separate components: (a) the passage retriever, which identifies relevant passages, and (b) the value labeller, which assigns variable values to these passages. The subsequent evaluation of these components led to a comprehensive investigation of how LLMs perform in the task of IE, the impact of the individual components to the overall performance, and how human validated data gathered through the use of ELICIT could be utilized for fine-tuning.

5.2 Key Findings

The evaluation on sentencing remarks and court transcripts, in supervised data and end-to-end settings, led to several key findings.

A comparative analysis of two *passage retrievers* — a vector index using the 'all-mpnet-base-v2' embedding model, and the Vicuna-13B LLM, which was prompted to identify if the passage is relevant or not — revealed that the LLM-based retriever initially underperformed but significantly improved from fine-tuning with human-validated data. In the case of court transcripts, the LLM-based retriever even outperformed the embedding model in both the supervised and end-to-end setting. A possible explanation for this is that the fine-tuned model aligns with the particular task and user, whereas the embedding model relies on semantic information and might result in misalignment.

The *value labeller* was designed as an LLM prompted to classify passages into the variable values based on the specific speaker. Fine-tuning this component led to improved ranking capabilities, which in turn resulted in better overall recall of the system. In the case of sentencing remarks, addressing data imbalance in human-validated data through oversampling the minority classes resulted in significant improvements. In court transcripts, fine-tuning notably improved the separation of values attributed in the same passages to relevant and irrelevant speakers.

The observations from the analysis of the separate components translated into the full end-to-end setting. Fine-tuned configurations, especially ones using the LLM-based retriever, resulted in better overall recall compared to base model configurations and the current ELICIT system. Additionally, the passage retriever was identified as the main factor in the overall performance of the LLM-enhanced ELICIT, with fine-tuning leading to substantial improvements in recall.

The final observation of the project was that the fine-tuned LLM systems resulted in increased efficiency in the human-in-the-loop format. Due to their enhanced automated performance, they required significantly fewer extractions shown to the user to achieve the same recall, thereby reducing potential human validation efforts.

5.3 Limitations

The main limitations of the project are the following:

1. The approach of splitting the document and doing extraction on smaller passages means that the system cannot extract information which spans multiple passages, and is limited to information which is contained in a single passage.
2. The project did not consider the processing time of the LLM-based passage retriever, which results in longer processing times, as the model is prompted for every passage. However, this can be done off-line before presenting the results to the human validator.

3. The court transcripts used are semi-synthetic; thus it is unclear how the shift to real data would affect the results.
4. The embedding model used in the vector index was not fine-tuned on human-validated data.
5. The project did not have access to long-context models, which could have been used as a baseline.

5.4 Future Directions

This study into the application of LLMs for legal information extraction in the legal domains opens up potential avenues for future research. The new task of speaker-centric information extraction is of great promise. It was demonstrated that the base model struggles to distinguish information based on the speaker, with the fine-tuned model performing marginally better. Thus, the burden of validating if the specified speaker provided the information still falls on the human validator. Alternative approaches through prompt engineering or fine-tuning could be explored for potential improvement.

Drawing inspiration from the success of using human-validated data for supervised fine-tuning, the application of Reinforcement Learning from Human Feedback (RLHF) (Bai et al., 2022; Ouyang et al., 2022) for fine-tuning presents an interesting path for improving passage retrieval and value labelling based on human preferences. Particularly in passage retrieval, the imperfect recall indicates that there are passages, which are relevant but the user does not get to validate. If these passages differ significantly from the ones being retrieved and validated, standard fine-tuning might not solve this. However, RLHF could help address this issue through the exploration of the RL agent.

While this project predominantly focused on the broader aspects of the IE process using LLMs, namely passage retrieval, value labelling, and fine-tuning, future investigations could be done into improving the prompts used, for example, by exploring the use of LLMs for designing prompts (Zhou et al., 2023).

Another area of future exploration lies in alternative strategies for dealing with long documents. The use of memory transformers (Bulatov et al., 2023) could be explored to overcome the limited context length of the LLMs and enable the extraction system to be more context-aware.

Lastly, given the sensitivity of legal data, additional effort should be given into the examination of potential biases inherent in the system, as well as potential mitigation strategies (Nozza et al., 2021, 2022).

References

- Aghajanyan, A., Zettlemoyer, L., and Gupta, S. (2020). Intrinsic Dimensionality Explains the Effectiveness of Language Model Fine-Tuning.
- Bahdanau, D., Cho, K., and Bengio, Y. (2016). Neural Machine Translation by Jointly Learning to Align and Translate.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., Joseph, N., Kadavath, S., Kernion, J., Conerly, T., El-Showk, S., Elhage, N., Hatfield-Dodds, Z., Hernandez, D., Hume, T., Johnston, S., Kravec, S., Lovitt, L., Nanda, N., Olsson, C., Amodei, D., Brown, T., Clark, J., McCandlish, S., Olah, C., Mann, B., and Kaplan, J. (2022). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback.
- Bansal, N., Sharma, A., and Singh, R. K. (2019). A Review on the Application of Deep Learning in Legal Domain. In MacIntyre, J., Maglogiannis, I., Iliadis, L., and Pimenidis, E., editors, *Artificial Intelligence Applications and Innovations*, IFIP Advances in Information and Communication Technology, pages 374–381, Cham. Springer International Publishing.
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., 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. M., 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.
- Bulatov, A., Kuratov, Y., and Burtsev, M. S. (2023). Scaling Transformer to 1M tokens and beyond with RMT.
- Butcher, B., Zilka, M., Cook, D., Hron, J., and Weller, A. (2023). Optimising Human-Machine Collaboration for Efficient High-Precision Information Extraction from Text Documents.
- Carnaz, G., Nogueira, V., Antunes, M., and Ferreira, N. (2020). An Automated System for Criminal Police Reports Analysis.
- Chase, H. (2022). LangChain. <https://github.com/hwchase17/langchain>.
- Chiang, W.-L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J. E., Stoica, I., and Xing, E. P. (2023). Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

- Choi, E., Hewlett, D., Uszkoreit, J., Polosukhin, I., Lacoste, A., and Berant, J. (2017). Coarse-to-Fine Question Answering for Long Documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 209–220, Vancouver, Canada. Association for Computational Linguistics.
- Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, Y., Wang, X., Dehghani, M., Brahma, S., Webson, A., Gu, S. S., Dai, Z., Suzgun, M., Chen, X., Chowdhery, A., Castro-Ros, A., Pellat, M., Robinson, K., Valter, D., Narang, S., Mishra, G., Yu, A., Zhao, V., Huang, Y., Dai, A., Yu, H., Petrov, S., Chi, E. H., Dean, J., Devlin, J., Roberts, A., Zhou, D., Le, Q. V., and Wei, J. (2022). Scaling Instruction-Finetuned Language Models.
- Common Crawl (2007). Common Crawl. <http://commoncrawl.org/>.
- Davis, J. and Goadrich, M. (2006). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd International Conference on Machine Learning - ICML '06*, pages 233–240, Pittsburgh, Pennsylvania. ACM Press.
- Eccleston, D. (2023). ShareGPT. <https://github.com/domeccleston/sharegpt>.
- Ghoddusi, H., Creamer, G. G., and Rafizadeh, N. (2019). Machine learning in energy economics and finance: A review. *Energy Economics*, 81:709–727.
- Hanley, J. A. and McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*, 143(1):29–36.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. (2022). Training Compute-Optimal Large Language Models.
- Honnibal, M., Montani, I., Van Landeghem, S., and Boyd, A. (2020). spaCy: Industrial-strength Natural Language Processing in Python.
- Houlsby, N., Giurgiu, A., Jastrzebski, S., Morrone, B., de Laroussilhe, Q., Gesmundo, A., Attariyan, M., and Gelly, S. (2019). Parameter-Efficient Transfer Learning for NLP.
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2021). LoRA: Low-Rank Adaptation of Large Language Models.
- Hu, X., Chen, P.-Y., and Ho, T.-Y. (2023). RADAR: Robust AI-Text Detection via Adversarial Learning.
- Huber, J. (2022). Chroma: The AI-native open-source embedding database. <https://github.com/chroma-core/chroma>.
- Izacard, G. and Grave, E. (2021). Leveraging Passage Retrieval with Generative Models for Open Domain Question Answering.
- Javaid, M., Haleem, A., Pratap Singh, R., Suman, R., and Rab, S. (2022). Significance of machine learning in healthcare: Features, pillars and applications. *International Journal of Intelligent Networks*, 3:58–73.

- Kamalloo, E., Zhang, X., Ogundepo, O., Thakur, N., Alfonso-Hermelo, D., Rezagholizadeh, M., and Lin, J. (2023). Evaluating Embedding APIs for Information Retrieval.
- Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2020). ALBERT: A Lite BERT for Self-supervised Learning of Language Representations.
- Lawrence, J. and Reed, C. (2019). Argument Mining: A Survey. *Computational Linguistics*, 45(4):765–818.
- Li, J., Liu, M., Kan, M.-Y., Zheng, Z., Wang, Z., Lei, W., Liu, T., and Qin, B. (2020). Molweni: A Challenge Multiparty Dialogues-based Machine Reading Comprehension Dataset with Discourse Structure.
- Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs].
- Malkov, Y. A. and Yashunin, D. A. (2018). Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs.
- Manyika, J. (2023). An overview of Bard: an early experiment with generative AI. *Google AI*.
- Nozza, D., Bianchi, F., and Hovy, D. (2021). HONEST: Measuring Hurtful Sentence Completion in Language Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2398–2406, Online. Association for Computational Linguistics.
- Nozza, D., Bianchi, F., and Hovy, D. (2022). Pipelines for Social Bias Testing of Large Language Models. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 68–74, virtual+Dublin. Association for Computational Linguistics.
- OpenAI (2022). ChatGPT. <https://chat.openai.com>.
- OpenAI (2023a). GPT-4. <https://openai.com/research/gpt-4>.
- OpenAI (2023b). GPT-4 Technical Report.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. (2022). Training language models to follow instructions with human feedback.
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training Recurrent Neural Networks.
- Qayyum, A., Qadir, J., Bilal, M., and Al-Fuqaha, A. (2020). Secure and Robust Machine Learning for Healthcare: A Survey.
- Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. (2020). Improving Language Understanding by Generative Pre-Training.

- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.
- Rajpurkar, P., Jia, R., and Liang, P. (2018). Know What You Don't Know: Unanswerable Questions for SQuAD. arXiv:1806.03822 [cs].
- Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ Questions for Machine Comprehension of Text.
- Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks.
- Saito, T. and Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, 10(3):e0118432.
- Sennrich, R., Haddow, B., and Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units.
- Sofaer, H. R., Hoeting, J. A., and Jarnevich, C. S. (2019). The area under the precision-recall curve as a performance metric for rare binary events. *Methods in Ecology and Evolution*, 10(4):565–577.
- Sun, C., Qiu, X., Xu, Y., and Huang, X. (2020). How to Fine-Tune BERT for Text Classification?
- Sun, K., Yu, D., Chen, J., Yu, D., Choi, Y., and Cardie, C. (2019). DREAM: A Challenge Dataset and Models for Dialogue-Based Reading Comprehension.
- Sun, S., Luo, C., and Chen, J. (2017). A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36:10–25.
- Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., and Lample, G. (2023). LLaMA: Open and Efficient Foundation Language Models.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention Is All You Need.
- Wikimedia Foundation (2001). Wikipedia. <http://wikipedia.org/>.
- Wu, C.-S., Madotto, A., Liu, W., Fung, P., and Xiong, C. (2022). QAConv: Question Answering on Informative Conversations.
- Xu, W., Grishman, R., and Zhao, L. (2011). Passage Retrieval for Information Extraction using Distant Supervision. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 1046–1054, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.
- Yang, Y., Wu, Z., Yang, Y., Lian, S., Guo, F., and Wang, Z. (2022). A Survey of Information Extraction Based on Deep Learning. *Applied Sciences*, 12:9691.

- Yang, Z. and Choi, J. D. (2019). FriendsQA: Open-Domain Question Answering on TV Show Transcripts. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 188–197, Stockholm, Sweden. Association for Computational Linguistics.
- Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2020). XLNet: Generalized Autoregressive Pretraining for Language Understanding.
- Yin, W., Hay, J., and Roth, D. (2019). Benchmarking Zero-shot Text Classification: Datasets, Evaluation and Entailment Approach. arXiv:1909.00161 [cs].
- Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. (2023). Judging LLM-as-a-judge with MT-Bench and Chatbot Arena.
- Zhou, Y., Muresanu, A. I., Han, Z., Paster, K., Pitis, S., Chan, H., and Ba, J. (2023). Large Language Models Are Human-Level Prompt Engineers.

Appendix A

Fine-tuning Details

A.1 Sentencing Remarks

A.1.1 Passage Retriever

Table A.1 Optimal hyperparameters of the Vicuna-13B passage retriever fine-tuned on user-validated sentencing remarks extractions.

Type	No. of Training Samples	No. of Validation Samples	Batch Size	Learning Rate
Regular	467	117	32	0.0012

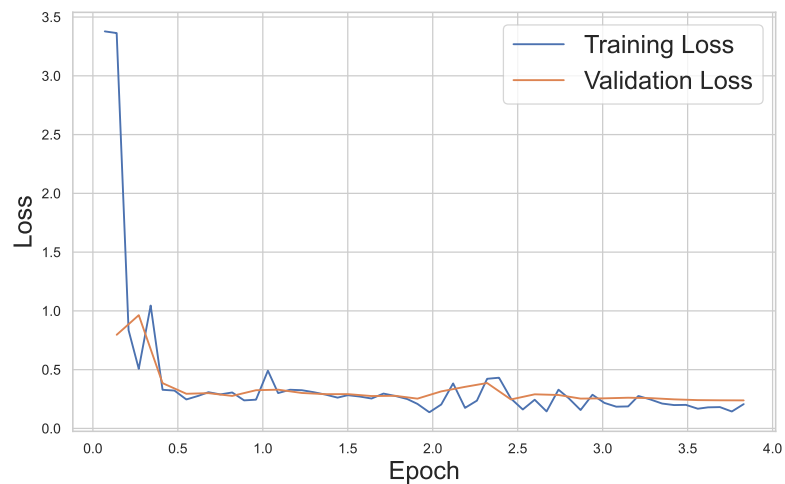


Fig. A.1 Training and validation loss of the Vicuna-13B passage retriever fine-tuned on user-validated sentencing remarks extractions.

A.1.2 Value Labellers

Table A.2 Optimal hyperparameters of the Vicuna-13B value labellers fine-tuned on user-validated sentencing remarks extractions.

Parameter	Regular	Minority Oversampled
No. of Training Samples	467	926
No. of Validation Samples	117	117
Batch Size	8	16
Learning Rate	0.0006	0.00008

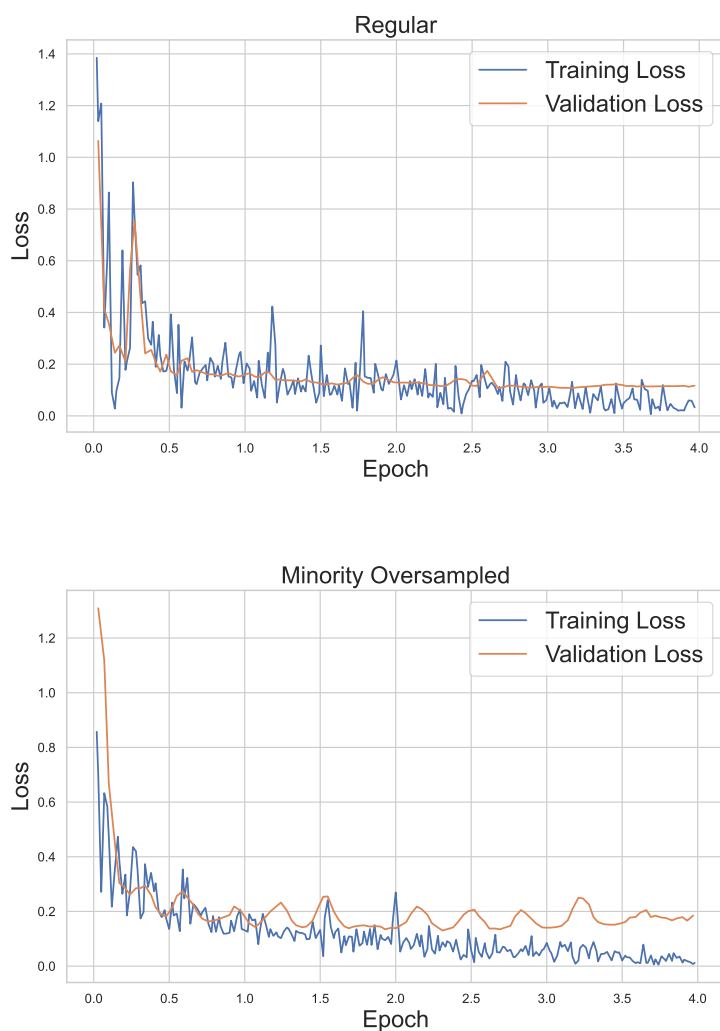


Fig. A.2 Training and validation losses of Vicuna-13B value labellers fine-tuned on regular user-validated sentencing remarks extractions, and with minority classes oversampled.

A.2 Court Transcripts

A.2.1 Passage Retriever

Table A.3 Optimal hyperparameters of the Vicuna-13B passage retriever fine-tuned on user-validated court transcript extractions.

Type	No. of Training Samples	No. of Validation Samples	Batch Size	Learning Rate
Regular	424	113	32	0.0012

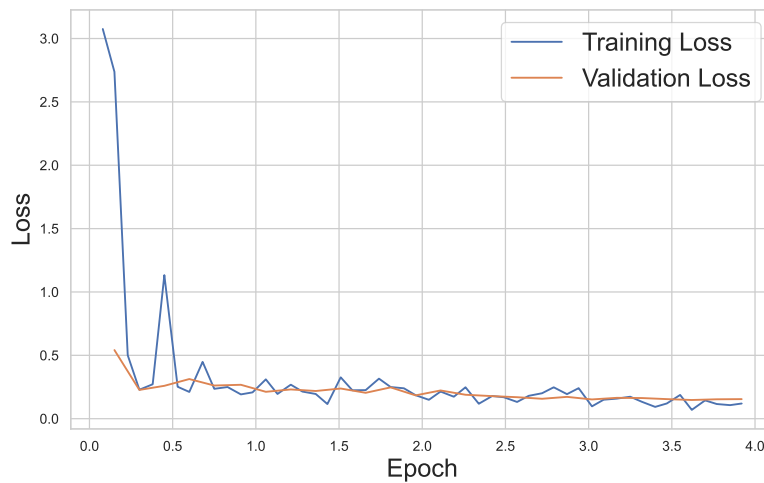


Fig. A.3 Training and validation loss of the Vicuna-13B passage retriever fine-tuned on user-validated court transcript extractions.

A.2.2 Value Labellers

Table A.4 Optimal hyperparameters of the Vicuna-13B value labellers fine-tuned on user-validated court transcript extractions.

Parameter	Regular	Minority Oversampled
No. of Training Samples	520	878
No. of Validation Samples	138	138
Batch Size	32	16
Learning Rate	0.0006	0.00008

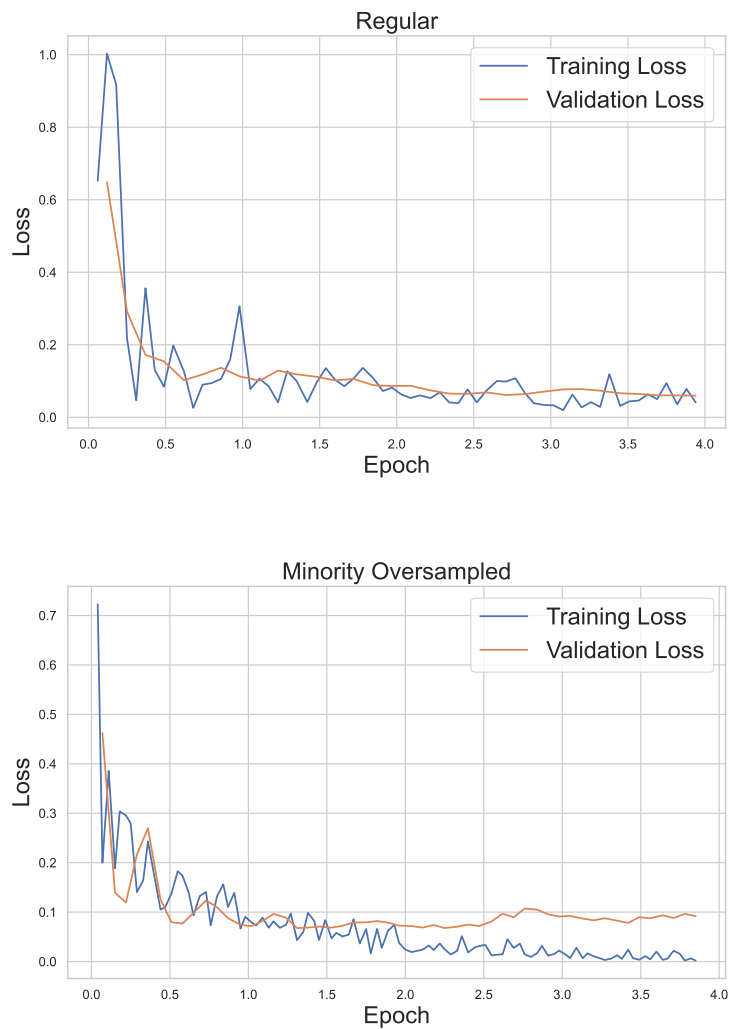


Fig. A.4 Training and validation losses of Vicuna-13B value labellers fine-tuned on regular user-validated court transcript extractions, and with minority classes oversampled.