Information Extraction in Legal Texts: Investigating LLMs' Performance on Traffic Accident Verdicts

Anonymous ACL submission

Abstract

This study investigates using Large Language Models (LLMs) for information extraction from Taiwanese traffic accident judgments, comparing large models (e.g., GPT, GEMINI) with smaller models (e.g., LLAMA-8B) under different prompts (Basic, Advanced, One-Shot).

Results indicate prompt choice significantly affects model performance, especially in handling long texts. GPT with One-Shot prompts achieved 89.2% accuracy, while longer prompts negatively impacted GEMINI. Fine-tuning boosted GPT's performance in specific fields, reaching 97.9% in "Depreciation Method" and 95.3% in "Repair Costs". chinese-llama showed limited improvement, while instruct-llama improved significantly, from 63.7% to 79.8%.

In summary, prompt design and fine-tuning are crucial for optimizing LLM performance, with further improvements expected through larger models and refined techniques.

1 Introduction

Traditional natural language understanding tasks, such as named entity recognition and relation extraction, typically require large amounts of manually labeled data for training. With the development of large language models (LLMs) like ChatGPT, many studies have explored using these models to replace manual annotation tasks (Li et al., 2023; He et al., 2024). While manual labeling is highly accurate, it is time-consuming and costly, whereas LLMs can effectively reduce these costs, especially in texts that do not require high levels of expertise.

Although ChatGPT has demonstrated impressive performance on tasks such as the U.S. Bar Exam (**?**), it still faces challenges when handling complex and highly specialized legal documents, such as traffic accident judgments. These judgments vary by case and require a high degree of accuracy and consistency, thus necessitating further evaluation and improvement of LLM performance in this domain.

This study evaluates the ability of Mega Large Language Models (MLLMs) and Small Large Language Models (SLLMs) to process complex traffic accident judgments. We selected traffic accident judgments as our dataset due to their inclusion of legal knowledge, fact descriptions, legal reasoning, and verdicts, which serve as a test bed for assessing the performance of LLMs in legal text processing. These judgments also support the determination of liability and legal reasoning in traffic accidents.

We focused on extracting 18 fields related to compensation (e.g., repair costs, total compensation) from the judgments, testing the native capabilities of various models and their performance with different prompts. This includes closed-source models such as GPT-4o-mini and Gemini-1.5-flash, as well as open-source models like Meta Llama-3-8B. Fine-tuning was applied to GPT and LLAMA models, limited to prompt adjustment, to test their performance improvements in specific tasks.

The results show that prompt design significantly impacts model performance. Specifically, the GPT model achieved the best performance in string-based tasks with the One-Shot prompt, reaching an average accuracy of 89.2% due to the inclusion of more context in the prompt. However, the GEMINI model's performance declined with longer prompts, particularly in handling lengthy texts, indicating its limitations in processing extended prompts. This highlights the importance of aligning prompt design with the model architecture to optimize performance.

Furthermore, fine-tuning results revealed that the GPT model showed significant improvements

in string and numeric data extraction, with accuracy rates of 97.9% in the "Depreciation Method" field and 95.3% in the "Repair Costs" field. In contrast, the `chinese-llama` model, despite its strong initial performance, exhibited limited improvements after fine-tuning, suggesting lower responsiveness to tuning. On the other hand, `instruct-llama` showed a substantial improvement in string accuracy, increasing from 63.7% to 79.8%.

The results indicate that even with successful fine-tuning, SLLMs (such as `LLAMA-8B`) do not outperform MLLMs (`GPT`, `GEMINI`). This emphasizes the advantages of MLLMs in more complex tasks or when training data is limited.

The contributions of this study are as follows:

1. Investigating whether large language models can extract necessary information from specialized traffic accident judgments, and evaluating their application in professional domains to improve the efficiency and accuracy of information extraction for judges.

2. Analyzing the performance of models such as `GPT` and `LLAMA` in processing legal texts, showing that fine-tuned SLLMs still underperform compared to MLLMs.

3. Proposing a method for calculating the consistency of numerical and string annotations to assess model performance in information extraction tasks.

4. Demonstrating that more detailed prompts do not necessarily yield better results, and that prompt design should be adapted based on the model's capabilities.

2 Related Work

Experiments with `GPT-3` for NLP dataset annotation suggest that, despite needing improvement, `GPT` provides effective, low-cost annotation, achieving near-human performance even on a budget Zhou et al. (2022). However, it still struggles with structured or domain-specific data. He et al. (2024) found `GPT-3.5` can match or exceed human annotators in various NLP tasks, highlighting its potential in resource-constrained settings.

Li et al. (2023) introduced the CoAnnotating framework, boosting annotation efficiency by over 40% while preserving accuracy. Models generate initial annotations, refined by humans, with a voting system to enhance diversity and accuracy. Shen et al. (2023) noted positional bias in language models from option placement; thus, varied evaluation methods or randomizing choices, along with human-reviewed datasets, are suggested to mitigate biases.

While many studies focus on classification, large language models don't always outperform traditional models in certain tasks. Wang et al. (2023) proposed GPT-NER, reframing NER as a generative task, but it is prone to errors and hallucinations. The study concludes that supervised learning still outperforms large language models in this area.

In resource-limited settings, the 8-billion parameter `LLAMA` model uses QLoRA Dettmers et al. (2023) to lower costs, allowing efficient performance and task flexibility. `LLAMA`, an open-source model by `Meta`, aims to improve NLP tasks. Touvron et al. (2023) note that its multi-layer training enhances language generation and understanding. With extensive context and more parameters, it excels in semantics, entity recognition, text classification, and translation, offering superior precision and resource efficiency compared to models like `GPT`.

In summary, while large language models show promise in NLP tasks, they struggle with information extraction and entity recognition, often producing irrelevant or incorrect outputs. Fine-tuning requires extensive annotated data, creating a bottleneck, and the high computational demands of training and inference limit practical use.

3 Method

While previous research has demonstrated that large language models can reduce the burden of manual annotation and facilitate human-machine collaboration, relying on language models to fully automate complex annotation tasks remains challenging. Therefore, this study aims to evaluate the performance of existing large language models on complex tasks.

We selected traffic accident judgments as the source of text data. These judgments contain detailed information about the circumstances of the accident, the timing, and compensation amounts. The richness of the content, along with the presence of numerous computational details, makes accurate data extraction a challenge, particularly for the 18 fields related to compensation. These

fields are categorized into numeric and string types based on the nature of the extracted information.

Given the constraints of hardware resources, a typical environment accessible to most users is a GPU with 24GB of memory. In consideration of practical application, we define models that can be fine-tuned in such hardware environments and have fewer than 8 billion parameters as Small Large Language Models (SLLMs). Models that cannot be fine-tuned in this environment are referred to as Mega Large Language Models (MLLMs).

3.1 Different Prompt Types

To increase experimental diversity and temperature, we designed prompts for field extraction tasks at three difficulty levels: Basic, Advanced, and One-Shot. These levels vary in rule descriptions and prompt lengths: Basic has 223 words, Advanced 553 words, and One-Shot 2772 words. The prompts vary in complexity, from Basic with minimal details to One-Shot, which provides extensive context and example answers.

The prompt design mimics the diversity and complexity of real-world task descriptions. By comparing model performance across different prompts, we evaluate the generalization ability of language models and identify factors affecting performance. Prompts are input into various models, generating annotation results for statistical analysis and comparison. To ensure accurate extraction, we provide clear and detailed extraction conditions, including examples or format requirements, allowing models to autonomously identify the content to extract. The goal is to offer precise guidance, enabling models to efficiently perform information extraction to meet real-world needs.

3.2 Fine-Tuning Models

Fine-tuning language models aims to improve their specialization and accuracy in specific domains. While large language models excel in various tasks, they struggle with entity recognition and specialized fields like law. To overcome these limitations, we used real legal verdicts as input data and manually reviewed standard answers as outputs, enabling models to learn the nuances of legal language, terminology, and reasoning, thus enhancing their legal text analysis capabilities.

`Meta Llama-3-8B` AI@Meta (2024) served as the pre-trained model, with `QLoRA` used to compress parameters for efficient fine-tuning under limited computational resources. Training data included preprocessed legal verdict texts, annotated with keywords and sentences. Due to token count limits, data exceeding these constraints were filtered out, retaining only feasible data for training.

3.3 Similarity Calculation

To evaluate annotator consistency and model accuracy, we used different criteria based on the annotated field type.

For numerical fields like "Total Compensation Amount," "Repair Costs," and "Insurance Payout," values were normalized by converting to numerical form and removing units or extraneous information, forming a sequence of numerical data extracted from various verdicts.

Exact Match Ratio for Numerical Types: For numerical fields, we calculate the exact match ratio, which is the proportion of cases where annotated results exactly match the gold standard. Let A be the standard answers and B the extracted data, as defined in Equation (1):

$$\text{Exact Match Ratio} = \frac{\sum_{i=1}^{n} \mathbb{I}(A_i = B_i)}{n} \quad (1)$$

Here, $\mathbb{I}(A_i = B_i)$ is 1 if $A_i = B_i$, and 0 otherwise, measuring agreement between automated and manual annotations.

Cosine Similarity for String Types: For text fields like "Accident Details," "Injuries," and "Occupation," we use cosine similarity to assess annotation consistency, as shown in Equation (2).

$$\text{Cosine Similarity}(\mathbf{A}, \mathbf{B}) = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \times \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (2)$$

4 Data Construction

4.1 Evaluation Dataset

The evaluation dataset was drawn from civil cases on Taiwan's public judgment website (2012-2022). A random sample of 1,000 cases was used as the evaluation set for representativeness and manageability. Figure 1 shows the word count distribution of the selected judgments.

We compiled data from the judgments related to compensation in traffic accident cases, including information such as the accident date, the circumstances of the event, the victim's occupation, and injuries. In total, 18 parameters were extracted, as

exemplified in Table 1. A complete example of the extracted fields is shown in Table 2.

Example of a Verdict
Taiwan New Taipei District Court Sanchong Summary Civil Judgment No. 941 of 2018.... The plaintiff alleged that on August 25, 2017, at 9:30 a.m., the defendant drove a car with license plate number 000-0000, and while entering B1 parking lot at Lane 93, Ren'ai Street, Luzhou District, New Taipei City, failed to yield to a turning vehicle and collided with a vehicle insured by the plaintiff and driven by a third party OOO with license plate number 0000-00. The said vehicle was damaged and was repaired at a cost of... including sheet metal 11,065 NTD, painting 12,060 NTD, and parts 14,940 NTD)... The vehicle was manufactured in May 2010 (assumed on the 15th) and had a driving license on file. Using the declining balance method... Therefore, the repair cost that the plaintiff can claim from the defendant totals 24,619 NTD (... After evaluating both parties' negligence and related evidence, the court determined the negligence ratio to be 30% for the plaintiff and 70% for the defendant...

Table 1: Verdict example: Blue (Accident date), Red (Incident details), Orange (Repair cost), Violet (Manufacturing date), Brown (Depreciation method), Green (Final repair compensation), Pink (Liability ratio).

4.2 Manual Annotation

To evaluate the accuracy of various techniques for data annotation, we randomly selected 1,000 cases from the original dataset for manual labeling, covering the 18 fields listed in Table 2. Two annotators performed the labeling, and the consistency score between the techniques and manual annotations was calculated as a benchmark for assessing the methods. Higher similarity scores indicate greater labeling accuracy.

The initial annotation consistency was 0.68, indicating the complexity of this task and the difficulty for human annotators to achieve high consistency at the outset. After further discussion and review of the discrepancies, the final similarity score reached 0.93.

During the annotation process, we encountered higher complexity in certain fields, such as "Daily Home Care," "Home Care Days," and "Home Care Costs" from Table 2, which are often expressed in formulas. These fields did not clearly associate values with specific fields, requiring annotators to rely on common sense. Additionally, the "Depreciation Method" field often mentioned both the straight-line method and the declining balance method in the text, but the final formula used the residual value, which could mislead annotators, causing significant differences in their understanding and widening the gap in consistency. We retained these inconsistent cases for further analysis.

We examined the causes of poor consistency, including unclear field definitions, data ambiguity, and differences in annotator interpretation. For conflicts in understanding, we conducted re-annotation after reaching a consensus. This process highlighted the importance of maintaining consistency and accuracy in data annotation and provided insights for improving annotation strategies. We also found that annotating monetary values posed a challenge, as amounts could be divided into the plaintiff's claimed amount and the judge's actual ruling. This distinction easily confused annotators, adding to the difficulty of the task, which we identified as a major challenge in annotating monetary values.

The annotation interface is shown in Figure 2, where we selected court judgments related to traffic accidents for information extraction tests. Through manual annotation, we extracted the traffic accident-related fields, forming a dataset we named `TAVCD (Traffic Accident Verdict Compensation Dataset)` , which contains 1,000 labeled cases. This dataset not only includes the extracted results but also preserves the positions of the results within the original text and their contextual information, ensuring key data for training and future challenges are retained.

4.3 Preparation of Fine-Tuning Training Data

For fine-tuning, we used the `TAVCD` dataset, split into 80% training and 20% validation data, as shown in Table 3. Analyzing the annotations revealed that identical information often appears in different locations, highlighting text redundancy and its impact on model performance. This redundancy can improve model comprehension and generalization, enhancing response accuracy by recognizing identical information in varied contexts. This approach better aligns model annotations with human understanding, enhancing accuracy and practicality.

5 Results

We used regular expressions as a baseline for extraction. However, because the judgments were written by humans, the expressions of keywords and context often varied, which impacted extraction accuracy. For example, the "Occupation" field

Field	Description
Accident Date	The plaintiff alleges that on October 29, 2019, at 6:00 p.m., ...
Incident Details	On October 29, 2019, at 6:00 p.m., the defendant drove a private car with license plate number 000-0000 without a license and failed to notice the situation ahead, ...
Vehicle Manufacture Date	the vehicle in question was manufactured in May 2007...
Injuries	The collision caused the plaintiff to fall off the vehicle, sustaining blunt chest injuries combined with bilateral rib fractures ...,
Occupation	The plaintiff claimed ... average daily income for professional taxi drivers is 1,514 NTD...
Durable Years	The applicable durability period for the vehicle is 5 years...
Depreciation Method	The depreciation is calculated using the average method (i.e., subtracting the residual value from the fixed asset cost... "For fixed assets depreciated using the declining balance method, ...
Defendant Liability	Based on the negligence of both parties, the plaintiff is responsible for 40% contributory negligence, and the defendant is responsible for 60% negligence...
Painting, Labor Costs, Coating, Sheet Metal	The repair costs for the vehicle are as follows: painting 3,500 NTD, labor costs 6,000 NTD, coating costs 5,500 NTD, and sheet metal costs 4,300 NTD...
Repair Costs	The plaintiff is entitled to claim the necessary repair costs for the vehicle, totaling 12,083 NTD...
Total Compensation Amount	The plaintiff, based on tort compensation and insurance subrogation rights, requests the defendant to pay 63,734 NTD, ...
Insurance Payout	The plaintiff is entitled to claim compensation from the defendant for the accident, which should be deducted by 24,703 NTD already received.
Daily Home Care Cost, Home Care Days, Home Care Cost	the full-day home care cost is... (Calculation: 1,200 NTD × 30 days = 36,000 NTD).

Table 2: Definitions and examples of compensation-related fields: orange highlights show extracted data points. The dataset includes 18 fields in total.

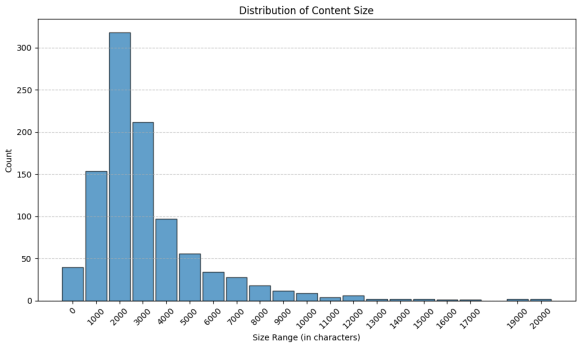

Figure 1: Word count distribution of selected verdicts

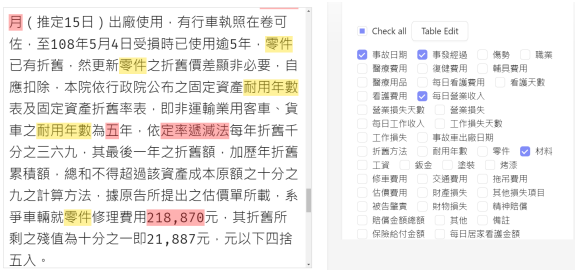

Figure 2: Annotation interface: The left side shows verdict data with yellow highlights for keywords and red for annotated data (e.g., five-year durable life, declining balance method, parts cost 218,870 NTD). The right side lists fields for annotators to mark.

Data Type	Count	Proportion
Training Data	800	80%
Validation Data	200	20%
Total	**1000**	**100%**

Table 3: Distribution of the TAVCD

lacks a fixed pattern, as it could be described in various ways, such as "this person is a full-time XXX" or "this person holds the position of XXX," increasing extraction difficulty and leading to poor performance of regular expressions. On the other hand, fields such as "Declining Balance Method" or "Straight-Line Method" were more easily identified by regular expressions.These observations indicate that contextual stability greatly influences regular expression performance.

5.1 Performance of Different Language Models

A comparison of performances is shown in Table 4, analyzing the performance of the GPT and GEMINI models in the information extraction task.

In string-type information extraction, the GPT model achieved the best performance using the One-Shot prompt, particularly in fields such as "Vehicle Manufacturing Date" and "Depreciation Method," reaching the highest accuracy. In con-

trast, the GEMINI model performed lower overall in string-type tasks but achieved results close to the GPT model in simpler fields such as "Defendant's Fault" and "Vehicle Manufacturing Date."

When analyzing the performance differences between prompts, the One-Shot prompt contained significantly more tokens (2,772 tokens) than the Advanced prompt (584 tokens). While providing more information can help the model, it may also increase the processing load. For GPT, the One-Shot prompt improved accuracy and stability in handling uncertain text; however, for GEMINI, the longer text became a burden, reducing its performance, with One-Shot prompt accuracy reaching 89% for GPT and 70% for GEMINI.

As language models improve in their ability to process prompts, the One-Shot prompt may become the preferred choice. Its strength lies in providing a complete example, offering more context and reference answers, which helps the model better handle complex strings. In contrast, the Advanced prompt outlines task rules but lacks specific examples, leading to lower performance in fields requiring high contextual understanding compared to the One-Shot prompt.

In comparing string and numeric information extraction, numeric data may require more flexible thinking, such as calculating amounts or explaining monetary figures. In some fields, the Advanced prompt performed as well as the One-Shot prompt, particularly with GEMINI, possibly due to the Advanced prompt's enhanced understanding of numeric patterns or its allowance for more flexible thinking. On the other hand, string-type data tends to have higher contextual variability and semantic complexity, such as event descriptions or occupational information, requiring stronger language understanding and contextual parsing. The GPT model excelled with the One-Shot prompt, effectively using detailed examples and rich context to parse complex string data, while the GEMINI model was relatively weaker in this regard.

Overall, the GPT model with the One-Shot prompt performed best in string data extraction, while numeric data extraction slightly favored the Advanced prompt. These results indicate that prompt selection has a significant impact on language model performance, particularly in numeric information extraction. The complexity of prompts (Basic, Advanced, One-Shot) may alter the model's understanding of context, influencing output quality. Future research could explore further optimization of prompts to improve the performance of different language models.

5.2 Fine-Tuned Models

We fine-tuned closed-source models like `GPT-4o-mini`, as well as open-source models such as `meta-llama/Meta-Llama-3-8B-Instruct`[1] and `shenzhi-wang/Llama3-8B-Chinese-Chat`[2], a Chinese fine-tuned model, to explore whether fine-tuning on limited hardware could optimize their performance in traffic accident judgment information extraction tasks and meet the demands of complex tasks.

We fine-tuned these models to more accurately associate text with relevant fields, enhancing information extraction performance. Based on the results in Table 4, fine-tuning data used the `Advanced` prompt. We experimented with various parameter combinations, ultimately finding that the optimal configuration was 600 training steps, a learning rate of 0.02, and a dropout rate of 0.2. If the training steps were too high or the dropout rate too low, the model tended to overfit, repeating the same text due to its limited parameter count, preventing it from fully grasping the complex task. This configuration significantly improved the model's accuracy and stability, as shown in Table 5.

In string-type extraction tasks, `chinese-llama` demonstrated high initial performance without fine-tuning, likely due to preliminary fine-tuning for the Chinese language. Its performance improved from 0.652 to only 0.748 after fine-tuning, leaving little room for further improvement. In contrast, `instruct-llama`, with an initial performance of 0.637, improved significantly to 0.798 after fine-tuning, indicating a high responsiveness to tuning. This suggests that the `instruct-llama` model, while weaker initially, significantly improved in understanding and extraction capabilities post-tuning.

These results highlight the importance of the model's initial state and training data: models that are pre-adjusted for specific languages or tasks benefit less from fine-tuning, while general-purpose models with poor initial performance can show substantial improvement after fine-tuning, offering valuable insights for future fine-tuning

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct/tree/main

[2] https://huggingface.co/shenzhi-wang/Llama3-8B-Chinese-Chat

strategies.

For numeric information extraction, chinese-llama showed limited improvement after fine-tuning, likely due to its strong initial ability to recognize numeric data. Meanwhile, instruct-llama exhibited larger improvements in numeric extraction, indicating that fine-tuning enhanced its adaptability. Overall, GPT showed the most significant improvement after fine-tuning, demonstrating excellent performance, particularly in fields like "Painting," "Wages," and "Repair Costs." In contrast, while chinese-llama and instruct-llama showed improvements, they did not match the performance of GPT, likely due to GPT's more flexible and accurate architecture or the influence of its larger parameter count. GPT's superior performance likely stems from its flexible architecture and large number of parameters, allowing it to quickly adapt and improve through fine-tuning, effectively utilizing contextual examples in training data to enhance its accuracy in handling complex contexts.

The training data used for fine-tuning consisted of 800 judgments, and the model was gpt-4o-mini-2024-07-18. Training parameters included 2 epochs, a total of 6,949,546 tokens, a batch size of 1, and a learning rate multiplier set to 1.8. The total training cost was approximately $15.57.

In summary, fine-tuning significantly improved language model performance, particularly for MLLMs like GPT. While SLLMs may not yet match the performance of MLLMs, future research should explore different models and fine-tuning methods to enhance SLLMs' practicality and effectiveness in specific applications, making them more accessible to a wider audience.

6 Conclusion

This study explored the application of LLMs in the task of information extraction from traffic accident judgments in Taiwan. The challenge lies in the interplay between natural language and numerical data in the text, much of which is unstructured. We analyzed the performance of models in extracting 18 fields, such as accident dates and compensation amounts, and designed various prompts to evaluate the impact of prompt design and fine-tuning on model performance.

The experimental results show that prompt design has a significant impact on model performance, but more detailed prompts do not always yield better results. Overly complex prompts may reduce model effectiveness, particularly when handling long texts. For instance, the GPT model, when using the One-Shot prompt, effectively extracted complex string-type data due to the inclusion of more contextual information, whereas the GEMINI model performed poorly with long prompts, suggesting that prompt design should be adapted to the characteristics of each model.

For more stable fields, such as "Depreciation Method" and "Repair Costs," model performance was higher, especially after fine-tuning GPT, achieving accuracies of 97.9% and 95.3%, respectively. However, for less consistent fields like "Occupation" and "Accident Details," model performance was weaker, as these fields require a greater ability to parse semantic meaning.

Fine-tuning results further revealed that GPT performed best in both string and numeric data extraction after tuning. In contrast, chinese-llama, despite its initially strong performance, showed limited improvement post-tuning, likely due to prior adjustments for the Chinese language. On the other hand, instruct-llama demonstrated high responsiveness to tuning, with string accuracy increasing from 63.7% to 79.8%, indicating substantial room for improvement in a fine-tuned state.

We found that proper fine-tuning can significantly improve accuracy in string data extraction, but its impact on numeric data is more limited, possibly due to the greater variability in numeric data formats. The results also indicate that repeatedly fine-tuning pre-tuned models may not yield additional benefits, as model performance may reach saturation for certain tasks.

In summary, this study highlights the critical impact of prompt design and fine-tuning strategies on model performance. It suggests that fine-tuning strategies should be tailored to the data characteristics and task requirements to avoid overfitting. Even with limited computational resources, Mega Large Language Models (MLLMs) still demonstrate significant advantages over Small Large Language Models (SLLMs), but further performance improvements may require larger models and more refined fine-tuning techniques.

Future research should focus on refining fine-tuning methods, optimizing prompt design, and improving model architectures and training strategies for tasks like legal text processing to enhance LLM performance in specialized domains.

	Method	RE	GPT			GEMINI		
	Prompt	-	basic	advanced	oneshot	basic	advanced	oneshot
String	Accident Date	0.810	**1.000**	**1.000**	0.994	0.993	**1.000**	**1.000**
	Incident Details	0.608	0.684	0.739	0.789	**0.825**	0.823	0.760
	Vehicle Manufacture Date	0.456	0.897	0.869	0.935	**0.946**	0.938	0.935
	Injuries	**0.887**	0.810	0.820	0.803	0.333	0.603	0.285
	Occupation	0.166	0.891	**0.912**	0.876	0.378	0.632	0.332
	Depreciation Method	**0.920**	0.854	0.865	0.863	0.560	0.642	0.653
	Defendant Liability	0.900	0.979	**0.984**	**0.984**	0.953	**0.984**	**0.984**
	Average (String)	0.678	0.874	0.884	**0.892**	0.713	0.803	0.707
Numerical	Painting	**0.938**	0.736	0.731	0.839	0.793	0.782	0.767
	Labor Costs	0.528	**0.860**	0.663	0.855	0.788	0.850	0.819
	Coating	0.803	0.870	0.865	0.933	0.938	**0.953**	0.834
	Sheet Metal	0.912	0.720	0.902	0.953	0.953	0.907	**0.969**
	Durable Years	0.477	0.969	0.922	0.974	0.912	0.907	**0.979**
	Repair Costs	0.347	0.503	**0.834**	0.487	0.409	0.808	0.420
	Total Compensation Amount	0.181	0.756	0.731	0.756	0.715	0.689	**0.803**
	Insurance Payout	**0.938**	0.653	0.544	0.601	0.679	0.601	0.788
	Home Care Days	0.959	0.964	0.959	0.959	0.959	**0.974**	**0.974**
	Home Care Costs	0.948	0.948	0.959	0.948	0.943	**0.969**	0.948
	Daily Home Care Amount	0.959	**0.979**	0.974	0.964	0.959	**0.979**	**0.979**
	Average (Numerical)	0.726	0.814	0.826	0.843	0.823	**0.856**	0.844

Table 4: Performance of different prompts with various language models. Bold indicates the best result for each field.

	Method	chinese-llama		instruct-llama		GPT	
	Status	pre-trained	fine-tuned	pre-trained	fine-tuned	pre-trained	fine-tuned
String	Accident Date	0.238	0.668	0.166	0.725	**1.000**	0.933
	Incident Details	0.386	0.523	0.377	0.533	0.739	**0.897**
	Vehicle Manufacture Date	0.736	0.855	0.560	0.834	0.869	**0.964**
	Injuries	0.807	0.841	0.818	0.843	0.820	**0.917**
	Occupation	0.870	0.881	0.891	0.902	0.912	**0.938**
	Depreciation Method	0.544	0.487	0.663	0.772	0.865	**0.979**
	Defendant Liability	**0.984**	**0.984**	**0.984**	0.974	**0.984**	**0.984**
	Average (String)	0.652	0.748	0.637	0.798	0.884	**0.945**
Numerical	Painting	0.896	0.756	0.902	0.798	0.731	**0.995**
	Labor Costs	0.513	0.575	0.539	0.720	0.663	**0.953**
	Coating	0.886	0.793	0.876	0.865	0.865	**0.995**
	Sheet Metal	0.969	0.979	0.964	0.943	0.902	**0.995**
	Durable Years	0.777	0.922	0.725	0.943	0.922	**0.979**
	Repair Costs	0.497	0.663	0.472	0.642	0.834	**0.953**
	Total Compensation Amount	0.570	0.736	0.513	0.788	0.731	**0.922**
	Insurance Payout	0.762	0.705	0.720	0.741	0.544	**0.990**
	Home Care Days	0.969	0.959	0.959	0.964	0.959	**0.974**
	Home Care Costs	0.959	0.959	0.948	0.959	0.959	**0.979**
	Daily Home Care Amount	0.959	0.959	0.959	0.953	0.974	**0.979**
	Average (Numerical)	0.796	0.819	0.780	0.847	0.826	**0.974**

Table 5: Comparison of fine-tuning effects on `LLAMA` and `GPT` models, including `chinese-llama` and `instruct-llama`. Underlined values compare fine-tuned and pre-trained states of the `LLAMA` models, while bold values indicate the best performance overall. Prompts used are the `Advanced` prompts .

7 Limitations

- The current ground truth relies on manual annotation, conducted independently by two annotators and compared for consistency. However, cases where both experts agree but the annotation does not match the actual answer may occur. This could result from human oversight or misjudgment when interpreting context, which may introduce bias in the annotated data and affect the true performance evaluation.

- While experiments were conducted to test the model's performance under different temperature levels, temperature parameters were still used to enhance the diversity of the model's outputs. This approach may lead to reduced output stability, particularly when the model is required to handle data with complex structures or subtle semantic differences.

- This study primarily focuses on information extraction from Chinese traffic accident judgments, meaning the findings may not fully generalize to annotation tasks in other languages or text types. This limits the generalizability and applicability of the research.

- Due to hardware limitations, the models and experiments used may not be comprehensive, and future work could involve experiments using larger models from the same family to provide more thorough analysis.

- The dataset used in this study is relatively small (only 1,000 cases), which may impact the depth of model learning and generalization capabilities. Future research could expand the dataset size to provide more robust results and insights.

- Plan to explore techniques like DSPy or other methods for prompt optimization to improve extraction accuracy.

References

AI@Meta. 2024. Llama 3 model card.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:2305.14314.

Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen. 2024. AnnoLLM: Making large language models to be better crowdsourced annotators. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track), pages 165–190, Mexico City, Mexico. Association for Computational Linguistics.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023. CoAnnotating: Uncertainty-guided work allocation between human and large language models for data annotation. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, pages 1487–1505, Singapore. Association for Computational Linguistics.

Chenhui Shen, Liying Cheng, Xuan-Phi Nguyen, Yang You, and Lidong Bing. 2023. Large language models are not yet human-level evaluators for abstractive summarization. In Findings of the Association for Computational Linguistics: EMNLP 2023, pages 4215–4233, Singapore. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Raphaël Couprie, Edouard Grave, Guillaume Lample, and Alexis Conneau. 2023. Llama: Open and efficient foundation language models. Preprint, arXiv:2302.13971.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. 2023. Gpt-ner: Named entity recognition via large language models. arXiv preprint arXiv:2304.10428.

Yilun Zhou, Chunting Zhang, Shuohang Wang, Zhengyuan Liu, and Diyi Yang. 2022. Is gpt-3 a good data annotator? In Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 1051–1061. Association for Computational Linguistics.