Exploring Nested Named Entity Recognition with Large Language Models: Methods, Challenges, and Insights

Hongjin Kim
Konkuk University
jin3430@konkuk.ac.kr

Jai-Eun Kim
Saltlux
jaieun.kim@saltlux.com

Harksoo Kim[*]
Konkuk University
nlpdrkim@konkuk.ac.kr

Abstract

Nested Named Entity Recognition (NER) poses a significant challenge in Natural Language Processing (NLP), demanding sophisticated techniques to identify entities within entities. This research investigates the application of Large Language Models (LLMs) to nested NER, exploring methodologies from prior work and introducing specific reasoning techniques and instructions to improve LLM efficacy. Through experiments conducted on the ACE 2004, ACE 2005, and GENIA datasets, we evaluate the impact of these approaches on nested NER performance. Results indicate that output format critically influences nested NER performance, methodologies from previous works are less effective, and our nested NER-tailored instructions significantly enhance performance. Additionally, we find that label information and descriptions of nested cases are crucial in eliciting the capabilities of LLMs for nested NER, especially in specific domains (i.e., the GENIA dataset). However, these methods still do not outperform BERT-based models, highlighting the ongoing need for innovative approaches in nested NER with LLMs.

1 Introduction

Named entity recognition (NER) is a prominent task in information extraction (IE) that entails detecting entities within the text and classifying them into predefined categories, such as person, location, and organization. A notable challenge within NER is nested NER, where named entities are embedded within other entities. This complexity introduces an additional dimension of difficulty to the task, making it a fertile area for research.

Recent advancements in large language models (LLMs) have demonstrated their exceptional capabilities across a variety of natural language processing (NLP) tasks, showcasing notable zero-shot and few-shot generalization abilities (Xie et al., 2023). Among these models, ChatGPT (OpenAI, 2023) has emerged as a prominent example, significantly influencing the NLP research landscape. It has been observed that, in the context of generated text evaluation, ChatGPT may surpass human evaluators in terms of consistency and fairness, according to some studies. However, when it comes to IE tasks such as NER or relation extraction (RE), the performance of LLMs has been a subject of investigation. Despite their general ability, LLMs, including ChatGPT, have not consistently outperformed humans or even BERT-based fine-tuned models in NER tasks. For instance, empirical studies (Xie et al., 2023) involving zero-shot NER with ChatGPT have reported performance levels significantly below those of BERT-based models that have undergone fine-tuning. Although adopting a few-shot approach can enhance performance, results remain substantially lower compared to fully-supervised models. These findings suggest that the application of LLMs to NER, particularly nested NER, needs to be further explored.

Nested NER has yet to be extensively explored within the context of LLMs. Addressing this gap, our study aims to empirically evaluate the capabilities of various LLMs in handling nested NER, by using zero-shot, one-shot, and few-shot approaches, and instruction tuning. In this work, we explore methods to enhance the reasoning capabilities of LLMs on nested NER. Initially, we applied reasoning techniques from previous studies, observing that while these methods could effectively boost performance on flat NER, they significantly degraded performance on nested NER. Subsequently, we incorporated prompts that integrate concepts from prior nested NER research, specifically those designed for tuning BERT-based models. This approach demonstrated an improvement in nested NER performance. To further augment the capability of LLMs in handling nested NER, we investigated various prompting approaches, in-

[*]Corresponding author

cluding zero-shot, one-shot, and few-shot, with a particular focus on task-specific reasoning techniques. Notably, our task-specific reasoning techniques markedly enhanced the LLMs' performance on nested NER compared to previous methods. Among these techniques, providing descriptions of named entity (NE) categories and detailed explanations of the nested phenomenon proved especially effective in eliciting the in-context learning (ICL) capability of LLMs for nested NER. However, it is important to note that these results still fall short of those achieved by BERT-based fine-tuned models, i.e., fully-supervised models. We argue that despite LLMs possessing billions of parameters and being trained on trillions of tokens, they require specific tuning for nested NER tasks. To address this gap, we explored instruction tuning methods aimed at enhancing LLMs' performance on nested NER. We utilized instructions from our reasoning prompts to tune open LLMs. Our findings align with the initial observations, indicating that descriptions of NE categories and detailed explanations of the nested phenomenon are crucial, while techniques from prior work may impede nested NER performance. Through this research, we provide guidance for users seeking to employ open or closed LLMs for nested NER—a relatively unexplored application of LLMs. Our main contributions are summarized as follows:

1. We conduct a comprehensive analysis of LLMs' performance on nested NER, revealing their capabilities and limitations in handling complex entity structures.

2. Our introduction of novel task-specific reasoning techniques significantly improves LLMs' understanding and processing of nested entities, marking a departure from conventional methods.

3. We demonstrate the effectiveness of refining LLMs for nested NER, showcasing substantial improvements with instruction tuning.

2 Related Work

2.1 LLMs on Flat NER

Wang et al. (2023) proposed GPT-NER, which transforms sequence labeling into a generation task, enabling LLMs to identify entities through special token marking (e.g., "@@Columbus##"). Similarly, Xie et al. (2023) investigated the application of LLMs to flat NER, employing strategies such as decomposed question answering (QA), syntactic augmentation, tool augmentation, and self-consistency to improve zero-shot NER performance. However, the application of LLMs to nested NER remains less explored. To address this gap, our work investigates various reasoning methods for enhancing the performance of LLMs on nested NER tasks. Detailed differences between GPT-NER and our work are provided in Appendix A.

2.2 Nested NER

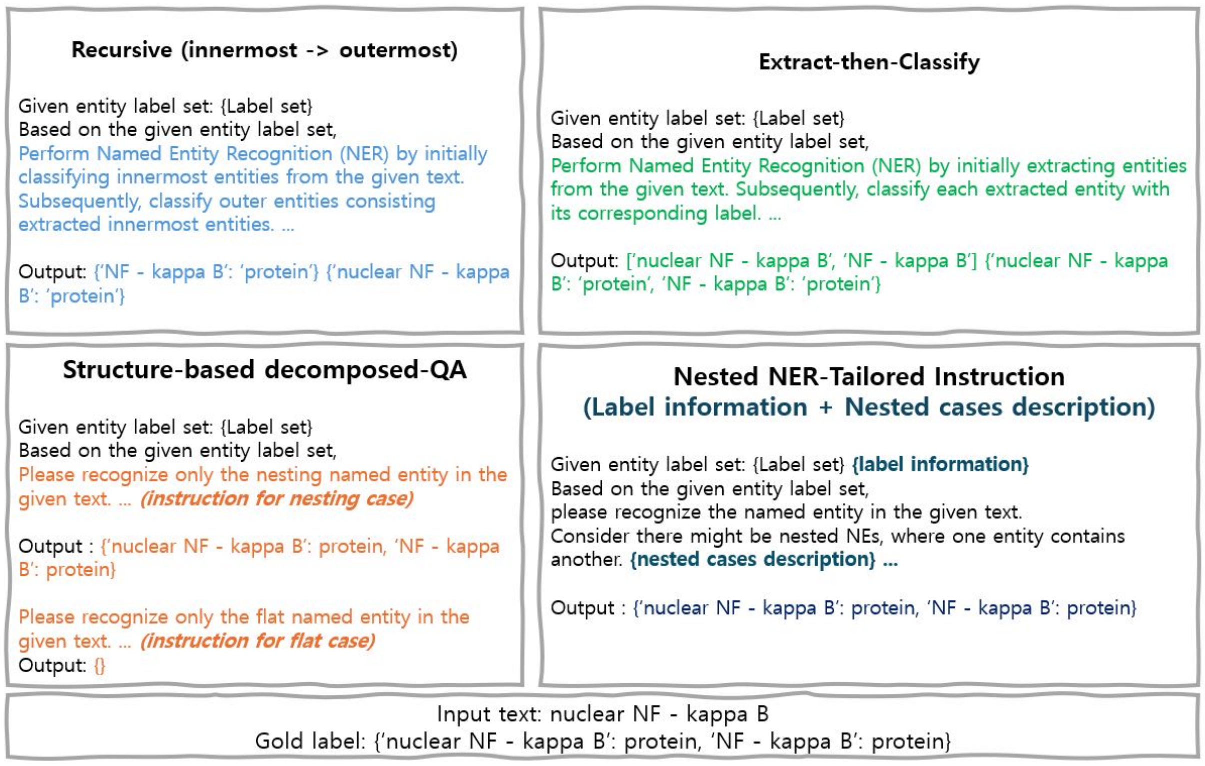
In this section, we explore prior nested NER research, focusing on sequence labeling-based and span-based approaches, with further methodologies detailed in Appendix B.

2.2.1 Sequence labeling-based

Ju et al. (2018) introduced a novel neural architecture for identifying nested NEs, employing dynamically stacked and shared LSTM-CRF layers. This approach, which we classify as recursive, utilizes an inside-to-outside method that leverages the output from identifying inner NEs to extract outer NEs at a higher level. In contrast, Rojas et al. (2022) highlighted the limitations of traditional NER systems in recognizing nested entities and revisited the multiple LSTM-CRFs (MLC) model. This model, distinguished by its simplicity and effectiveness, involves training independent sequence labeling models for each entity type and comes with an open-source library for computing nested NER-specific metrics. Their findings offer fresh insights into the performance of existing models on nested NER. However, a critical limitation of the MLC approach is its requirement for constructing a separate model for each entity type, hindering its ability to recognize nested entities of the same type.

2.2.2 Span-based

Span-based models have significantly advanced nested NER. Sohrab and Miwa (2018) proposed treating all text regions as potential entity mentions for classification. Fei et al. (2020) introduced a model combining LSTM with an attention mechanism for nested mentions, further enhanced by contextualized embeddings like BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), and ELECTRA (Clark et al., 2019). Li et al. (2020b) transformed nested NER into a machine-reading comprehension task, while Yu et al. (2020) used a biaffine approach for precise span prediction. Yan et al. (2023) ad-



Recursive (innermost -> outermost)

Given entity label set: {Label set}
Based on the given entity label set,
Perform Named Entity Recognition (NER) by initially classifying innermost entities from the given text. Subsequently, classify outer entities consisting extracted innermost entities. ...

Output: {'NF - kappa B': 'protein'} {'nuclear NF - kappa B': 'protein'}

Extract-then-Classify

Given entity label set: {Label set}
Based on the given entity label set,
Perform Named Entity Recognition (NER) by initially extracting entities from the given text. Subsequently, classify each extracted entity with its corresponding label. ...

Output: ['nuclear NF - kappa B', 'NF - kappa B'] {'nuclear NF - kappa B': 'protein', 'NF - kappa B': 'protein'}

Structure-based decomposed-QA

Given entity label set: {Label set}
Based on the given entity label set,
Please recognize only the nesting named entity in the given text. ... ***(instruction for nesting case)***

Output : {'nuclear NF - kappa B': protein, 'NF - kappa B': protein}

Please recognize only the flat named entity in the given text. ... ***(instruction for flat case)***
Output: {}

Nested NER-Tailored Instruction
(Label information + Nested cases description)

Given entity label set: {Label set} **{label information}**
Based on the given entity label set,
please recognize the named entity in the given text.
Consider there might be nested NEs, where one entity contains another. **{nested cases description} ...**

Output : {'nuclear NF - kappa B': protein, 'NF - kappa B': protein}

Input text: nuclear NF - kappa B
Gold label: {'nuclear NF - kappa B': protein, 'NF - kappa B': protein}

Figure 1: Examples of prevalent reasoning techniques, our proposed structure-based decomposed-QA technique, and our proposed nested NER-tailored instruction.

dressed overlapping spans with a CNN for spatial relation modeling. However, these models often face high computational demands and struggle with partially matching spans. To overcome this, Shen et al. (2021) developed the Locate and Label model, focusing on identifying entity boundaries before classification, leveraging both boundary information and partially matched spans.

We apply concepts from sequence labeling-based and span-based methodologies independently to probe the capabilities of LLMs in nested NER. Additionally, we introduce reasoning techniques tailored specifically for nested NER, designed to enhance the LLMs' understanding of and performance on complex entity structures.

3 Method

Initially, we examine the impact of various output formats on the task's performance. After selecting the optimal output format, we independently apply concepts derived from prior nested NER studies and reasoning techniques previously utilized with LLMs on NER. Subsequently, we develop reasoning techniques specifically tailored for nested NER to enhance the LLMs' capability to understand complex entity hierarchies.

3.1 Designing Output Format

The performance of nested NER tasks can be notably affected by the output format, especially since LLMs such as ChatGPT and Llama operate as generative models. Motivated by existing research, we devise a variety of output formats for evaluation. These formats include span extraction, entity-category dictionaries, the insertion of distinguishable tokens, and a novel nested level BIO (i.e., Begin-Inner-Outer) tagging approach. Following the results of our various experiments, we chose the entity-category dictionary for the output format. The explanation of other formats and the results are in the Appendix C.

3.2 Entity-Category Dictionary

In this output format, we instruct LLMs to generate responses structured as {'Entity': 'Category'}. Given a sentence, "nuclear NF - kappa B", the anticipated output is {'nuclear NF - kappa B': 'protein', 'NF - kappa B': 'protein'}. Description of other formats, please see the Appendix C.

3.3 Concepts & Reasoning Techniques for Nested NER

Figure 1 shows the examples of various instructions for our study.

3.3.1 Recursive Technique

Drawing inspiration from the recursive approaches (Ju et al., 2018; Kim and Kim, 2024), we explore whether directing LLMs to sequentially recognize NEs, starting with the innermost NEs and progressing to the outermost (or vice versa), enhances nested NER performance. Initially, we instruct the LLMs to identify either the innermost or outermost NEs. Following this, a subsequent request is made to identify the NEs at the next level of nesting based on the initial results. This process is iterated until the LLMs no longer extract any new NEs, at which point the recursive extraction is deemed complete. For instance, consider the sentence S, "nuclear NF - kappa B". When we instruct LLMs to initially recognize the outermost NEs, the expected output is {'nuclear NF - kappa B': 'protein'}. In the subsequent step, targeting the next level of nesting, the anticipated output becomes {'NF - kappa B': 'protein'}, assuming the output format is structured as an entity-category dictionary.

3.3.2 Extract-then-Classify

Drawing inspiration from (Shen et al., 2021), we investigate a two-stage approach where LLMs are first instructed to identify all NEs, regardless of whether they are flat or nested. Subsequently, LLMs are directed to classify the categories of these extracted NEs. This method aims to elicit a chain of thought (COT) from LLMs by progressively refining the answer. For example, given the sentence, S, "nuclear NF - kappa B", the initial instruction for LLMs to identify NEs would yield the expected output ['nuclear NF - kappa B', 'NF - kappa B']. Following this extraction phase, the next step of classification is anticipated to produce {'nuclear NF - kappa B': 'protein', 'NF - kappa B': 'protein'}, with the output format assumed to be an entity-category dictionary.

3.3.3 Decomposed-QA

The extract-then-classify technique aims to simplify the answer derivation process, while the decomposed-QA technique targets the simplification of the problem itself. This latter approach has been effectively employed to address the challenges of nested NER even before the advent of LLMs. For example, Li et al. (2020b) introduced an MRC framework tailored for nested NER, which treats the identification of each NE category as a separate question-answering task. In a similar vein, Rojas et al. (2022) developed the MLC model, which is designed to extract each NE category independently through training multiple models. Building upon these methodologies, Xie et al. (2023) applied the decomposed-QA technique to NER tasks using LLMs, demonstrating its capability to improve zero-shot performance in flat NER. Inspired by these precedents, we explore the decomposed-QA technique's effectiveness and applicability specifically within the realm of nested NER.

Structure-based Decomposed-QA In an extension of this approach, we introduce a variant specifically for nested NER, which we term structure-based decomposed-QA. This technique strategically breaks down the task based on the structural complexity of NEs. We direct LLMs to differentiate between flat and nested NEs through separate inquiries: one set of queries prompts the identification of solely flat NEs, while another set focuses exclusively on nested NEs.

3.3.4 Nested NER-Tailored Instruction

Yin et al. (2023) investigated to determine the most critical component in instruction for model performance, finding a significant drop when task-specific label information was omitted. This underscores the importance of NE label information in facilitating LLMs' comprehension of the NER task. Drawing on this insight, we introduce a nested NER-tailored instruction technique that incorporates detailed NE label information into the instructions for LLMs. To provide comprehensive label information, we adopt a twofold approach: firstly, by retrieving label definitions from Wikipedia based on each label's title, and secondly, by generating label descriptions via ChatGPT to ensure the instructions are tailored to the NER task's specifics. While Wikipedia definitions offer a general understanding, they may not fully convey the intricacies of nested NER tasks due to their dictionary-like nature. Therefore, ChatGPT-generated descriptions are also utilized to bridge this gap. Furthermore, to enhance LLMs' grasp of nested NER tasks, we include descriptions of nested cases within our instructions, focusing on two scenarios: nested different type (NDT), where an entity encompasses a shorter entity of a different type, and nested same type (NST), involving entities represented by a hierarchical structure. By integrating these tailored instructions, we aim to significantly improve LLMs' performance on nested NER tasks. Full of nested NER-tailored instructions are in the Appendix G.

3.4 Instruction Tuning with PEFT

To assess the effectiveness of instruction-tuned LLMs on nested NER, we implement instruction tuning using parameter efficient fine-tuning (PEFT), especially QLoRa (Dettmers et al., 2024). In the interest of time and cost efficiency, we selectively employ the most effective reasoning techniques identified from few-shot approaches during the instruction tuning process. Our objective is to determine whether these reasoning techniques, proven effective in preliminary settings, also enhance LLM performance when applied through instruction tuning.

Let $P = \{p_0, p_1, ..., p_n\}$ be the instruction, where n denotes the number of tokens in the instruction. Let $X = \{x_0, x_1, ..., x_m\}$ and $Y = \{y_0, y_1, ..., y_l\}$ represent the input text and output answer, where m and l denote the number of tokens in the input text and output answer, respectively. We define our **Input** as the concatenation of the instruction and input text, $I = [P; X]$. Our goal is to generate the output answer Y, given a instruction, and input text I as follows:

$$Y = \prod_{i=1}^{|Y|} P_w(y_i | y_{i<}, [P; X]) \qquad (1)$$

Here $P(\cdot|\cdot)$ refers to the probability of generating the next token. W are the parameters of $P(\cdot|\cdot)$, and $y_{i<} = \{y_1, y_2, ..., y_{i-1}\}$.

The loss function we aim to minimize is defined as follows:

$$L = \frac{1}{D} \sum_{d \in D} L(d, \Phi_0, \Delta\Phi) \qquad (2)$$

where D is the dataset and d is each data instance, L is the cross entropy loss function employed to LLMs, Φ_0 denotes the original weights of the LLMs that are kept frozen, and $\Delta\Phi$ refers to the additional parameters used by the QLoRA (Dettmers et al., 2024). QLoRA employs low-rank matrices $B \in R^{d \times r}$ and $A \in R^{r \times k}$ with $r << min(d, k)$. The low rank matrices A and B are trainable:

$$h = W_0 x + BA_x = (W_0 + BA)x \qquad (3)$$

We fine-tune LLMs to learn the complex structure of nested NEs using different reasoning techniques. Unlike in-context learning, such as the few-shot approach, instruction tuning enables LLMs to align with nested NER consistently across all samples in the dataset, rather than being exposed to only a few examples. Through instruction tuning, we anticipate further performance improvements in nested NER tasks. Moreover, QLoRA prevents the LLMs from becoming over-fitted to the nested NER task, as their original weights are not extensively tuned.

4 Experiments

4.1 Experimental Setups

For our experiments, we utilized ChatGPT 3.5 and ChatGPT 4.0 APIs for closed LLMs and Llama2 13B[1], Llama3 8B[2], Mistral 7B[3] (Jiang et al., 2023), and Qwen2 7B [4] (Bai et al., 2023) for open LLMs. For detailed information on the implementation, please refer to Appendix D.

4.1.1 Datasets

In our experiments on nested NER, we evaluated LLMs using the widely used ACE 2004, ACE 2005, and GENIA datasets. In our investigation of zero-shot, one-shot, and few-shot approaches, we utilized the entire test portion of each dataset for assessment. Moreover, in our experiment of instruction tuning, we used the entire train and test portion of each dataset. For more details on the datasets, please see Appendix E.

4.2 Results

4.2.1 Metrics

Traditional metrics for NER are insufficient for accurately evaluating nested NER due to their inability to fully capture the complexity of nested entity relationships. To address this limitation, Rojas et al. (2022) introduced metrics specifically designed for nested NER, implementing three types of evaluation criteria. The *Flat* metric assesses the identification of non-nested entities, focusing on the simplest entity recognition tasks. The *Nested* metric is designed to evaluate the recognition of entities within nested structures, considering both inner and outer

[1]https://huggingface.co/meta-llama/Llama-2-13b-hf
[2]https://huggingface.co/meta-llama/Meta-Llama-3-8B
[3]https://huggingface.co/mistralai/Mistral-7B-v0.3
[4]https://huggingface.co/Qwen/Qwen2-7B

Method		ACE 2004			ACE 2005			GENIA		
		Flat	Nested	Nesting	Flat	Nested	Nesting	Flat	Nested	Nesting
PoS+SC	Zero	36.17	36.44	0.59	27.88	20.41	0.38	31.47	38.69	5.93
Recursive	Zero	26.29	26.08	1.68	32.43	24.23	1.46	43.85	23.60	3.13
	One	38.89	28.56	1.85	34.18	25.90	1.48	47.46	28.80	3.35
	Five	**39.11**	**30.48**	**1.96**	**37.32**	**26.57**	**1.59**	**51.69**	**29.11**	**3.77**
Extract-then-Classify	Zero	28.06	27.90	1.74	33.33	26.98	1.43	45.07	26.95	3.24
	One	37.32	28.10	1.81	37.81	27.12	1.60	47.35	**29.49**	**3.74**
	Five	**43.15**	**28.48**	**1.96**	**40.24**	**28.64**	**1.61**	**50.24**	24.21	3.02
Decomposed-QA	Zero	41.17	11.76	0.00	34.75	1.66	0.00	62.32	9.97	**0.57**
	One	47.24	14.08	0.23	39.54	**8.53**	0.00	63.46	**9.87**	0.55
	Five	**50.25**	**14.23**	**0.37**	$\underline{\mathbf{41.77}}$	5.76	0.00	$\underline{\mathbf{67.75}}$	6.70	0.15
Structure-based Decomposed-QA	Zero	41.61	36.02	2.22	34.72	31.32	1.83	37.17	17.15	3.39
	One	47.49	37.60	2.37	37.01	32.95	1.84	49.12	34.84	3.78
	Five	$\underline{\mathbf{51.97}}$	$\underline{\mathbf{39.96}}$	$\underline{\mathbf{2.43}}$	**39.80**	$\underline{\mathbf{34.01}}$	$\underline{\mathbf{2.88}}$	**53.22**	$\underline{\mathbf{35.82}}$	$\underline{\mathbf{9.99}}$
Label Information (Wikipedia)	Zero	35.19	35.82	0.00	27.69	31.07	0.00	28.10	33.78	0.34
	One	42,55	39.55	1.96	**35.60**	34.62	0.84	48.89	$\underline{\mathbf{49.19}}$	6.74
	Five	**45.95**	**42.11**	**5.46**	34.07	**38.26**	**5.89**	**51.69**	45.98	**11.45**
Label Information (Wikipedia) + Nested Case Description	Zero	35.70	36.39	0.00	27.84	31.03	0.00	36.74	40.36	0.34
	One	43.07	39.72	1.68	34.75	35.50	1.66	49.49	**47.53**	7.27
	Five	**46.17**	**43.03**	$\underline{\mathbf{6.58}}$	**35.09**	$\underline{\mathbf{39.46}}$	**6.06**	**52.75**	44.81	**12.05**
Label Information (ChatGPT)	Zero	36.80	36.84	0.00	27.82	31.65	0.00	25.69	32.10	0.33
	One	42.55	40.11	2.00	**35.45**	34.36	1.24	48.99	**47.91**	7.35
	Five	**45.88**	$\underline{\mathbf{43.21}}$	**5.82**	32.71	**36.93**	**6.36**	**50.62**	45.53	**11.21**
Label Information (ChatGPT) + Nested Case Description	Zero	35.35	36.63	0.00	27.89	30.65	0.00	34.32	39.19	0.97
	One	42.33	40.54	2.66	**34.88**	34.32	0.84	$\underline{\mathbf{54.07}}$	45.46	9.28
	Five	**46.75**	**42.53**	**5.94**	34.75	**38.29**	$\underline{\mathbf{6.63}}$	51.92	**45.52**	$\underline{\mathbf{12.66}}$

Table 1: ChatGPT 4.0 results of reasoning concepts and nested NER-tailored instructions with reported F1 Scores. For the **Recursive** techniques, we instruct ChatGPT to identify the innermost and outermost first, respectively. We report the results of the innermost initial setting since it showed better performance. **Bold** numbers highlight the best scores within the respective categories, while Underlined numbers denote the highest scores across all reasoning techniques. Scores for each technique were averaged over five trials.

entities but not assessing their relationships simultaneously. To complement this, the *Nesting* metric is introduced to evaluate the accuracy of identifying nested entity relationships, requiring both inner and outer entities to be recognized correctly for an evaluation to be deemed accurate. We adopt these three metrics—*Flat*, *Nested*, and *Nesting*—to evaluate the performance of our methods using LLMs on nested NER tasks.

4.2.2 Results of prevalent reasoning concepts

To assess the impact of various reasoning techniques on nested NER, we carried out experiments as detailed in Section 3.3. The performance outcomes, presented in Table 1, highlight the efficacy of these techniques. Specifically, PoS+SC, a method previously proposed by Xie et al. (2023) that combines the Part-of-Speech (PoS) reasoning technique with self-consistency (SC), serves as a benchmark. The SC method, which verifies the responses of LLMs, has been shown by Xie et al. (2023) to consistently improve performance in an NER task. Despite its benefits, SC was not incorporated into our other experiments. This decision was due to the method's requirement to query LLMs multiple times with the same question, leading to increased costs. Firstly, we observed that the performance of nested NER improves as the number of provided examples increases. Secondly, while the decomposed-QA (D-QA) technique, which prompts LLMs to independently identify each NE category, markedly enhances flat NER performance, it conversely diminishes nested NER capabilities. This finding underscores the limited applicability of the D-QA technique, highly effective for flat NER as demonstrated by Xie et al. (2023), to nested NER scenarios. Furthermore, other well-regarded reasoning techniques, such as recursive and extract-then-classify, which have shown effectiveness in BERT-based fine-tuned models, also fall short in addressing nested NER challenges under the LLMs. Contrastingly, our proposed reasoning technique, the structure-based decomposed-QA (SD-QA), which directs LLMs to independently discern between flat and nested NER, surpasses all traditional techniques in both *Nested* and *Nesting* metrics. These findings validate the superiority of SD-QA in leveraging LLMs for nested NER, significantly outperforming existing reasoning methods. However, these results yet comparable with BERT-base fine-tuned models. To alleviate this gap, we investigate the nested NER-tailored instruction for eliciting the ability of LLMs.

4.2.3 Results of Nested NER-Tailored Instruction

To evaluate the effectiveness of our proposed instructions tailored for nested NER, we conducted experiments leveraging NE label information and descriptions of nesting cases. The results, presented in Table 1, align with observations from Section 4.2.2, indicating that nested NER performance improves with an increasing number of examples. In the zero-shot scenario, the label information, whether sourced from Wikipedia or generated by ChatGPT, did not enhance the LLMs' capability in nested NER. Intriguingly, under the five-shot setting, instructions incorporating our label information significantly surpassed the performance of the SD-QA technique—previously the most effective among the explored reasoning methods—across all datasets in both *Nested* and *Nesting* metrics. This trend underscores a notable performance leap associated with the incremental examples provided to our label information instructions, demonstrating their effective synergy. Particularly within the GENIA dataset, a domain-specific corpus, we observed that our label information significantly aids LLMs in adapting to specialized domains. In both the one-shot and five-shot settings, our nested NER-tailored instructions notably improved performance. This demonstrates that our label information effectively interacts with few-shot examples, enhancing their utility in specialized domains. Moreover, label information generated by ChatGPT proved to be more effective than that sourced from Wikipedia. This suggests that a general description, as provided by ChatGPT, is more conducive to LLMs' domain-specific adaptation than the more dictionary-like nature of Wikipedia definitions. Most importantly, the addition of both label information and nested case descriptions markedly improved performance on the *Nesting* metric. These findings confirm the significant efficacy of our approach for enhancing nested NER performance.

4.2.4 Results of LLMs Instruction Tuning

Table 2 shows the results of instruction tuning using various LLMs. We explore which method is effective for tuning LLMs on the GENIA dataset. The results of instruction tuning on ACE 2004 and ACE 2005 datasets are in Appendix F.

Decomposed-QA (D-QA) technique demonstrated poor performance on nested NER, indicating its suitability primarily for flat NER rather than nested NER. To assess the ability of structure-based decomposed-QA (SD-QA) to differentiate between *flat* and *nested* NEs, we evaluated models specialized for flat and nested NER independently. The results showed that the *Flat* SD-QA model, intended to identify flat NEs, inadvertently extracted more nested NEs compared to the *Nested* SD-QA model. This outcome suggests that distinguishing between flat and nested NEs remains a challenge for even instruction-tuned 13B LLMs. Our approach of using NE label information and nested case descriptions once again proved highly effective in the instruction tuning setting. Specifically, instructions incorporating NE label information generated by ChatGPT and nested case descriptions markedly enhanced the nested NER performance of all LLMs. This finding further confirms that a general description is more conducive to LLMs' performance in domain-specific scenarios. Additionally, Llama3 8B, Mistral-7B-v0.3, and Qwen2-7B are reported to outperform Llama2-13B on various LLM benchmarks, including mathematical reasoning and QA tasks. However, these newer models with 7B and 8B configurations did not surpass Llama2-13B in nested NER tasks. This may suggest that the capability to recognize complex named entities could improve with larger model sizes. Despite these advancements, none of these methods exceeded the performance of the existing state-of-the-art (SOTA) model, which is fully supervised and fine-tuned using BERT, even with the deployment of a 13B model. This highlights the ongoing need for further exploration within nested NER tasks, employing a variety of LLM sizes and methodologies.

4.3 Error Analysis

Recognizing the importance of analyzing LLM errors, we have conducted a detailed error analysis with instruction-tuned models. Figure 2 shows the analysis of error by best performance model (Label

Model	Method	GENIA								
		Flat			Nested			Nesting		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Llama2-13B	Decomposed-QA	67.02	40.82	50.74	43.05	28.88	34.57	5.53	3.66	4.40
	Structure-based Decomposed-QA	63.38	66.04	64.69	26.86	**52.23**	35.47	7.91	17.77	10.95
	Flat Case of Structure-based Decomposed-QA	59.39	62.12	60.72	**92.47**	36.47	52.31	27.50	1.92	3.58
	Nested Case of Structure-based Decomposed-QA	**76.27**	22.65	34.92	23.27	33.91	27.60	10.66	21.41	14.24
	Label Information (Wikipedia)	71.62	54.96	62.19	69.22	38.78	49.71	30.56	13.41	18.64
	Label Information (Wikipedia) + Nested Case Description	72.44	58.93	64.99	72.84	40.26	51.86	34.18	14.11	19.98
	Label Information (ChatGPT)	70.80	66.02	68.33	70.07	49.83	58.24	34.25	19.15	24.86
	Label Information (ChatGPT) + Nested Case Description	71.96	**68.37**	**70.12**	69.06	53.22	**60.11**	**34.93**	**22.82**	**27.61**
Llama3-8B	Label Information (Wikipedia)	69.63	53.87	60.74	82.92	33.25	47.47	31.00	5.40	9.20
	Label Information (Wikipedia) + Nested Case Description	**78.41**	60.19	68.98	71.08	42.26	59.96	22.70	14.79	17.91
	Label Information (ChatGPT)	72.39	65.88	68.98	73.23	46.95	57.21	**35.71**	17.42	23.42
	Label Information (ChatGPT) + Nested Case Description	73.77	**68.12**	**70.42**	**74.53**	**54.18**	**64.29**	32.70	**21.97**	**26.88**
Mistral-7B-v0.3	Label Information (Wikipedia)	76.70	68.00	72.09	74.81	49.01	59.22	37.12	17.07	23.39
	Label Information (Wikipedia) + Nested Case Description	**77.39**	69.49	73.23	77.88	48.51	59.79	39.08	16.20	22.91
	Label Information (ChatGPT)	76.07	64.68	69.91	**78.58**	47.52	59.23	**42.86**	17.77	25.12
	Label Information (ChatGPT) + Nested Case Description	76.09	**71.41**	**73.69**	68.29	**54.54**	**60.64**	33.07	**21.95**	**26.39**
Qwen2-7B	Label Information (Wikipedia)	76.10	60.91	67.66	**76.90**	44.22	56.16	32.70	12.02	17.58
	Label Information (Wikipedia) + Nested Case Description	74.09	65.15	69.34	73.37	47.28	57.50	**33.45**	16.03	21.67
	Label Information (ChatGPT)	73.29	64.00	68.22	72.36	47.01	57.08	37.40	15.19	24.01
	Label Information (ChatGPT) + Nested Case Description	**73.99**	**68.46**	**70.74**	70.82	**53.15**	**60.33**	34.81	**20.91**	**26.12**
BERT fine-tuned	SOTA	-	-	81.20	-	-	65.80	-	-	35.30

Table 2: Results of tuning utilizing various instructions. **Bold** numbers highlight the best scores within the respective method. Scores for each method were averaged over three trials.

Information from ChatGPT + Nested Case Description). Because nested NEs have complicated structures, we have analyzed whether the instruction-tuned models accurately recognized inner and outer NEs. The *Inner* metric specifically assesses the recognition of entities within nested structures, focusing on inner entities, while the *Outer* metric evaluates the identification of only the outer entities. The *Wrong type error* metric measures the error rate of incorrect entity type predictions. For the GENIA dataset, it is apparent that LLMs and BERT (49.30 for inner and 82.40 for outer) struggle more with recognizing inner entities as compared to outer entities. Nevertheless, LLMs demonstrate a notable ability in classifying entity types, as evidenced by the significantly low error rates for wrong types. However, there is still room for improvement in LLMs' capability to accurately identify nested entities.

Different from the GENIA dataset, LLMs exhibit similar capabilities in recognizing both inner and outer entities on the ACE2004 and ACE2005

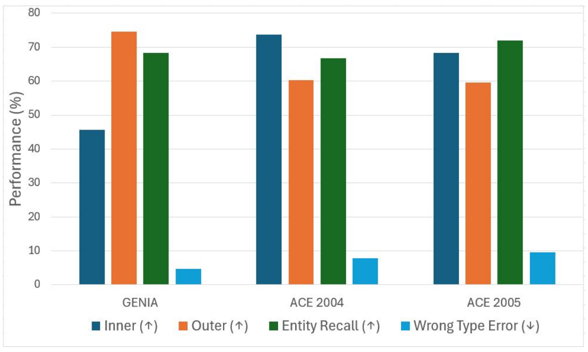


Figure 2: Results of detailed analysis

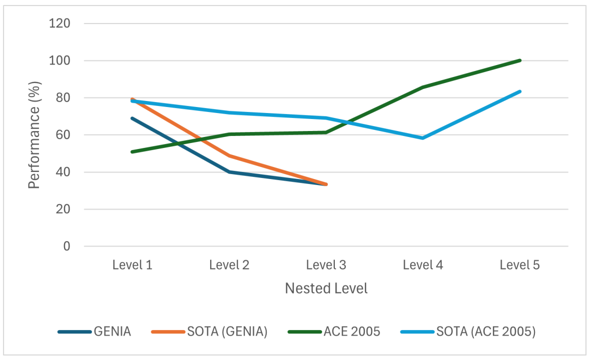


Figure 3: Analysis according to the nested level

datasets. A potential reason for this discrepancy could be the presence of more deeply nested entities (up to a maximum of 5-depth) in the ACE2004 and ACE2005 datasets, as opposed to the GENIA dataset (maximum 3-depth), posing additional challenges for LLMs. Furthermore, LLMs demonstrate a higher error rate in classifying wrong entity types compared to their performance on the GENIA dataset, while maintaining similar recall metrics. This may be attributed to the greater number of entity categories in the ACE datasets (seven categories) versus GENIA (five categories), underscoring the need for a more detailed analysis of error types in LLMs' performance on nested NER tasks.

Figure 3 shows the performance changes according to the nested level. In this figure, the "Nested Level" metric evaluates performance based on the depth of nesting. "Level 1" corresponds to the innermost entity, while "Levels 2" and "Level 3" represent entities nested within two and three depths, respectively. As illustrated, LLMs and BERT exhibit comparable performance on entities nested within three depths on the GENIA dataset. However, LLMs demonstrate a notably lower performance on innermost and 2-depth nested entities. This indicates that while LLMs show promise in identifying deeply nested entities, their effectiveness in extracting innermost and moderately nested entities (i.e., Recall) requires further improvement. Contrasting with the GENIA dataset, where LLMs show comparable performance to BERT, on the ACE2005 datasets, LLMs exhibit significantly poor performance on lower nested levels. This discrepancy is likely attributed to the more complex nested entity structures within the ACE datasets. Such complexity highlights the need for methodologies specifically tailored for training LLMs on nested NER tasks. This underscores the ongoing requirement to develop and refine approaches that more accurately capture the nuances of nested entity recognition with LLMs.

5 Conclusions

We explored the nested NER using various concepts, methodologies, and techniques from the previous nested NER works. We also proposed nested NER-tailored reasoning techniques and instructions, which demonstrated significant effectiveness on nested NER. We observe the output format that LLMs would generate is significantly important in the NER task. We also confirmed the previous reasoning technique, especially decomposed-QA, which is highly effective on flat NER, is not robust on the nested NER. Our proposed instruction that incorporating the NE label information and nested case description using LLMs is significantly effective on nested NER. Unfortunately, the performance of all of our methods yet outperformed the previous BERT-based fine-tuned models. It reveals nested NER still needs to be further explored.

Limitations

For cost efficiency, our experiments on instruction tuning were confined to using open LLMs, specifically models up to 13B in size. This limitation restricts our ability to compare performance across larger LLMs, which might yield different insights into the effectiveness of the proposed techniques for nested NER. The decision to employ 7B, 8B, and 13B models was also influenced by computational resource constraints. Access to more diverse or powerful computational resources would allow for a broader exploration of models and techniques, potentially leading to different outcomes. As the field of LLMs is rapidly evolving with frequent updates and new model releases, our findings, while relevant to the current technological landscape, may require re-evaluation as newer models and techniques become available.

Handling of Repeated Named Entities In the nested NER environment, our analysis indicates that there are no instances where the exact same entity is annotated with multiple labels. If the same entity appears multiple times within the same sentence and its entity type changing depending on the context, this indeed could be problematic. Through our analysis, we identified such samples in the ACE 2004 test set.

Context: "that requires even greater efforts on the part of the palestinian authority to try to control the crowds , to keep them away from israeli fixed positions and on the israeli side that would enable them to restrain the use of live fire ."

Answer of span extraction output format: [[GPE, 9, 11], [GPE, 24, 24], [GPE, 29, 31], [GPE, 30, 30], [GPE, 35, 35], [PER, 16, 17], [PER, 21, 21]]

Answer of entity-category output format: "the palestinian authority": [GPE], "israeli": [GPE, GPE], "the israeli side": [GPE], "them": [GPE, PER], "the crowds": [PER]

The entity-category output format has the disadvantage of not maintaining order, which complicates determining the placement of types like GPE and PER for "them." Currently, there are output formats that favor nested NER performance and those that are advantageous for post-processing, which we consider a trade-off. However, upon analyzing the ACE 2004, ACE 2005, and GENIA test datasets, we found that such cases were minimal, with only 4, 2, and 5 instances respectively. Moreover, the entity-category output format has shown significantly higher performance in generative models, making it the preferred option at this time.

Entities Identical and Post-processing The error rate for NEs in sentences not matching their generated counterparts was extremely low (0.53% of all errors) in the study (Xie et al., 2023), and similarly negligible in our study. Nevertheless, we acknowledge that these could potentially occur in generative models. We consider handling such errors to be improvements due to post-processing rather than inherent model capabilities; hence, we did not apply incorrect post-processing. Real-world applications will require careful consideration of these handling methods.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (RS-2024-00343989, Enhancing the Ethics of Data Characteristics and Generation AI Models for Social and Ethical Learning) and (RS-2024-00398115, Research on the reliability and coherence of outcomes produced by Generative AI).

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, Jianxin Ma, Rui Men, Xingzhang Ren, Xuancheng Ren, Chuanqi Tan, Sinan Tan, Jianhong Tu, Peng Wang, Shijie Wang, Wei Wang, Shengguang Wu, Benfeng Xu, Jin Xu, An Yang, Hao Yang, Jian Yang, Shusheng Yang, Yang Yao, Bowen Yu, Hongyi Yuan, Zheng Yuan, Jianwei Zhang, Xingxuan Zhang, Yichang Zhang, Zhenru Zhang, Chang Zhou, Jingren Zhou, Xiaohuan Zhou, and Tianhang Zhu. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Kevin Clark, Minh-Thang Luong, Quoc V Le, and Christopher D Manning. 2019. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Hao Fei, Yafeng Ren, and Donghong Ji. 2020. Dispatched attention with multi-task learning for nested mention recognition. *Information Sciences*, 513:241–251.

Yao Fu, Chuanqi Tan, Mosha Chen, Songfang Huang, and Fei Huang. 2021. Nested named entity recognition with partially-observed treecrfs. In *Proceedings* *of the AAAI Conference on Artificial Intelligence, Online*, pages 2–9.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 861–871.

Hongjin Kim and Harksoo Kim. 2024. Recursive label attention network for nested named entity recognition. *Expert Systems with Applications*, 249:123657.

Bing Li, Shifeng Liu, Yifang Sun, Wei Wang, and Xiang Zhao. 2020a. Recursively binary modification model for nested named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8164–8171.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Matías Rojas, Felipe Bravo-Marquez, and Jocelyn Dunstan. 2022. Simple yet powerful: An overlooked architecture for nested named entity recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2108–2117.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849.

Juncheng Wan, Dongyu Ru, Weinan Zhang, and Yong Yu. 2022. Nested named entity recognition with span-level graphs. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 892–903.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 204–214.

Jue Wang, Lidan Shou, Ke Chen, and Gang Chen. 2020. Pyramid: A layered model for nested named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5918–5928.

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*.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. 2023. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956.

Hang Yan, Yu Sun, Xiaonan Li, and Xipeng Qiu. 2023. An embarrassingly easy but strong baseline for nested named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1442–1452.

Fan Yin, Jesse Vig, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2023. Did you read the instructions? rethinking the effectiveness of task definitions in instruction learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3063–3079.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476.

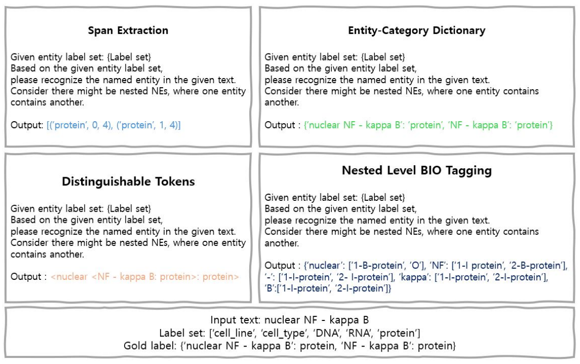
A Difference between GPT-NER and Our work

GPT-NER does not explore prompts/instructions specialized for nested NER: While GPT-NER shares the commonality of investigating prompts/instructions for utilizing LLMs in NER tasks, it does not conduct an in-depth study for nested NER. The authors of GPT-NER propose an effective few-shot sampling technique for flat NER prompts and merely apply these to datasets for nested NER tasks to conduct experiments. We illustrate the difference between GPT-NER's instruction and our nested NER-tailored instruction using the GENIA dataset as an example. Note that the label set for the GENIA dataset, which includes [cell_line, cell_type, DNA, RNA, protein], is common to both GPT-NER and our study.

GPT-NER's instruction (for DNA entity type):
I am an excellent linguist. The task is to label DNA entities in the given sentence.
Input: {Text of first example}
Output: {DNA Answer of the first example}
Input: {Text of the second example}
Output: {DNA Answer of the second example}
Input: {Text of the third example}
Output: {DNA Answer of the third example}
Input: Text of target input
Output: Generated output by LLM

Ours nested NER-tailored instruction:
Given entity label set: [cell_line, cell_type, DNA, RNA, protein].
cell_line: Describes a specific line of cells used in experiments or research.
cell_type: Describes the type or category of a cell.
DNA: Denotes the genetic material present in a cell.
RNA: Represents ribonucleic acid, involved in gene expression.
protein: Refers to large molecules essential for various biological functions.
Based on the given entity label set, please recognize the named entity in the given text. Consider there might be a nested case, where one entity contains another.
NDT: It consists of an entity containing a shorter entity tagged with a different type.
NST: This case usually occurs when entities are originally represented by a hierarchy.
Here are some examples.
Text 1: Text of the first example
Answer 1: Answer of the first example
Text 2: Text of the second example
Answer 2: Answer of the second example
Text 3: Text of the third example
Answer 3: Answer of the third example
Text: Text Input
Answer: Generated output by LLM

GPT-NER's instruction requires repetition for each entity type present in the dataset. In contrast, our proposed instruction allows the identification and recognition of all entity types in a single execution. Unlike GPT-NER, our approach provides entity type information and descriptions of nested cases to fully elicit the nested NER capabilities of LLMs. To reiterate, GPT-NER's instruction can extract only one entity type at a time, necessitating repeated prompts/instructions for each entity type. Moreover, GPT-NER's approach resembles the decomposed-QA methodology, a baseline method experimented with in our study. Decomposed-QA, as used with generative models like ChatGPT-3.5 and Llama2 in (Xie et al., 2023), demonstrates effectiveness in recognizing one entity type at a time in flat NER. However, its effectiveness in nested NER remains unclear. Additionally, GPT-NER proposes a method to enhance few-shot examples in prompts by searching for few-shot examples related to the input text. This search process uses a supervised fine-tuned NER model, leveraging similarity searches between entities extracted from the input text (i.e., text entered for NER purposes) and the entity representations of the fine-tuned model to provide few-shot examples with entities most similar to those in the input text. This reliance on a supervised fine-tuned NER model marks a significant divergence from our study, which does not depend on such models. Furthermore, since GPT-NER requires running a prompt for each entity type, it necessitates a few-shot example search with each



Span Extraction

Given entity label set: {Label set}
Based on the given entity label set,
please recognize the named entity in the given text.
Consider there might be nested NEs, where one entity contains another.

Output: [('protein', 0, 4), ('protein', 1, 4)]

Entity-Category Dictionary

Given entity label set: {Label set}
Based on the given entity label set,
please recognize the named entity in the given text.
Consider there might be nested NEs, where one entity contains another.

Output : {'nuclear NF - kappa B': 'protein', 'NF - kappa B': 'protein'}

Distinguishable Tokens

Given entity label set: {Label set}
Based on the given entity label set,
please recognize the named entity in the given text.
Consider there might be nested NEs, where one entity contains another.

Output : <nuclear <NF - kappa B: protein>: protein>

Nested Level BIO Tagging

Given entity label set: {Label set}
Based on the given entity label set,
please recognize the named entity in the given text.
Consider there might be nested NEs, where one entity contains another.

Output : {'nuclear': ['1-B-protein', 'O'], 'NF': ['1-I protein', '2-B-protein'], '-': ['1-I-protein', '2- I-protein'], 'kappa': ['1-I-protein', '2-I-protein'], 'B':['1-I-protein', '2-I-protein']}

Input text: nuclear NF - kappa B
Label set: ['cell_line', 'cell_type', 'DNA', 'RNA', 'protein']
Gold label: {'nuclear NF - kappa B': protein, 'NF - kappa B': protein}

Figure 4: Examples of output formats

prompt execution, resulting in considerable time complexity.

B Structure-based nested NER Work

The exploration of nested Named Entity Recognition (NER) has evolved through various innovative approaches. Lu and Roth (2015) laid foundational work with a mention hypergraph representation for entity mention extraction, which was further enhanced by Muis and Lu (2017) through the use of mention separators and handcrafted features. Building on these concepts, Katiyar and Cardie (2018) introduced a directed hypergraph utilizing LSTM features to comprehend nesting structures. Li et al. (2020a) developed a recursively binary modification (RBM) model, leveraging modification relations among sub-entity types to define entity spans. Our model, RLAN, while recursive, diverges from RBM by processing the output from one layer into the next to delineate nested entity levels, operating at the sentence rather than region level, as RBM does. Further advancements include the Pyramid model by Wang et al. (2020), utilizing stacked NER layers in a pyramid shape for the token or text region embeddings, and an inverse pyramid for layer interaction. Similarly, Fu et al. (2021) employed TreeCRFs and a biaffine scoring module in a tree-shaped stack for entity boundary prediction and decoding. Wan et al. (2022) enhanced span representations with retrieval-based span-level graphs, linking spans and entities via n-gram feature similarity to integrate information through entity-entity and span-entity graphs. However, as noted by Wang and Lu (2018), these models often grapple with spurious structures and structural ambiguities, indicating ongoing challenges in nested NER research.

C Details for the output format

Figure 4 shows the examples of output formats.

C.1 Span Extraction

For this output format, we instruct LLMs to produce responses in the format of (Entity Category, start_index, end_index). For instance, given the input sentence S, "nuclear NF - kappa B" (from the GENIA dataset), the expected output is [('protein', 0, 4), ('protein', 1, 4)]. This approach allows us to leverage the mathematical capabilities of LLMs for precise entity span identification.

C.2 Insertion of Distinguishable Tokens

For this output format, we instruct ChatGPT and Llama2 to reproduce the input sentence S, embedding distinguishable tokens to mark entities within the text. For the given sentence S, "nuclear NF - kappa B", we expect the output to be "<nuclear <NF - kappa B: protein>: protein>", which allows us to clearly identify the entities and their categories.

C.3 Nested Level BIO Tagging

In this output format, we instruct ChatGPT and Llama2 to annotate all input words in sentence S with BIO tags, accounting for nested NER. Traditional flat NER BIO tagging fails to adequately represent nested entities due to their inability to assign multiple tags to a single entity. To address this limitation, we introduce a nested level BIO schema that combines both the nested level of entities and BIO tagging. For example, given the sentence S, "nuclear NF - kappa B", the expected output is {'nuclear': ['1-B-protein', 'O'], 'NF': ['1-I-protein', '2-B-protein'], '-': ['1-I-protein', '2-I-protein'], 'kappa': ['1-I-protein', '2-I-protein'], 'B':['1-I-protein', '2-I-protein']}. Here, 'nuclear' appears in the outermost nested entity level but not within any inner nested entity, thus receiving the tags '1-B-protein' and 'O'. This format requires LLMs to figure out the maximum nesting level within sentence S.

Given that the cost associated with the ChatGPT API increases with longer input and output lengths, we evaluate these output formatting methods to identify the most robust approach. This selection process is crucial for optimizing resource utilization in subsequent analyses, thereby achieving cost savings without compromising the effectiveness of our nested NER exploration.

Output Format	Zero-Shot			One-Shot			Five-Shot		
	Flat	Nested	Nesting	Flat	Nested	Nesting	Flat	Nested	Nesting
Span Extraction	2.08	2.86	0.00	2.17	3.14	0.00	2.87	3.88	0.04
Entity-Category Dictionary	**28.56**	**35.52**	**0.65**	**43.57**	**47.27**	**5.44**	**50.24**	**44.21**	**11.47**
Distinguishable Token	25.16	29.08	0.52	38.24	42.20	4.51	42.19	38.81	8.97
Nested Level BIO	16.28	22.08	0.10	24.17	28.65	2.65	30.40	34.15	3.99

Table 3: Results for different output formats. Each output format underwent evaluation in three separate trials, with the resulting scores averaged for consistency. The experiments were conducted using the ChatGPT 3.5 and 4.0 API.

Method	GENIA								
	Flat			Nested			Nesting		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Span Extraction	22.45	11.79	15.88	21.83	9.08	13.78	6.79	2.61	3.14
Entity-Category Dictionary	71.73	68.57	70.12	70.98	48.02	57.28	34.24	17.60	13.25

Table 4: Results of instruction tuning across different output formats.

C.4 Performance According to Output Format

To assess the impact of different output formatting methods on nested NER performance, we conducted experiments across various formats, employing the vanilla method as described by Xie et al. (2023) for cost efficiency. These experiments were carried out on the GENIA dataset, utilizing zero-shot, one-shot, and few-shot approaches to evaluate the effectiveness of each format. As shown in Table 3, the performance of nested NER is significantly influenced by the design of the output format. Interestingly, the span extraction format exhibited notably poor performance, despite ChatGPT's recognized strength in mathematical reasoning. Similarly, the Nested Level BIO format, which requires mathematical reasoning to interpret nested levels, also underperformed. In contrast, the entity-category dictionary (E-C dictionary) format demonstrated superior performance across all tested formats. Based on these observations, we conclude that the E-C dictionary format is the most effective and efficient for nested NER, offering optimal performance with minimal output length. Therefore, we have chosen the E-C dictionary format for subsequent experiments.

As shown in Table 4, we further confirmed that the output format significantly influences the performance of nested NER, affecting both in-context learning and instruction tuning settings. Notably, the entity-category dictionary output format substantially outperformed the span extraction format under identical instruction settings.

D Implementation Details

We used the ChatGPT 3.5 (gpt3.5-turbo-1106) and ChatGPT 4 (gpt-4-1106-preview) API to explore our target task, nested NER. Following the precedent set by an empirical study on flat NER using LLMs (Xie et al., 2023), we set the temperature to 0.0. Both top-p and top-n were configured to 1. For instruction tuning, we utilized the Llama2 13B, Llama3 8B, Mistral 7B, and Qwen2 7B models with QLoRa. We set the QLoRa alpha to 16 and dropout to 0.1. We set r, which is the dimension size of Lora, to 128. We set the batch size to 8 and accumulation steps to 2. We also set the learning rate to 2e-4.[5]

E Statistics of Datasets

The ACE 2004 and ACE 2005 datasets feature seven entity types, with 24% and 22% of their named entities being nested, respectively. We utilized the data preprocessing method as detailed by Katiyar and Cardie (2018) and Lin et al. (2019), dividing these datasets into training, development, and testing sets in an 8:1:1 ratio for instruction tuning purposes. For the GENIA dataset, specifically GENIAcorpus3.02p which includes five entity types (DNA, RNA, protein, cell line, and cell type), we adhered to the preprocessing approach outlined by Katiyar and Cardie (2018), assigning the first 90% of the dataset for training and the latter 10% for evaluation. Table 5 shows the statistics of the ACE2005 and GENIA datasets.

[5]Our code and all instructions will be available on Github

GENIA		
Nested Level	Train	Test
Level 1	42,826	4,942
Level 2	3,910	569
Level 3	91	15
Level 4	1	0
ACE 2005		
Nested Level	Train	Test
Level 1	19,676	2,724
Level 2	3,934	505
Level 3	731	102
Level 4	90	10
Level 5	7	1
Level 6	2	0

Table 5: The number of data in each level.

F Results of Instruction Tuning on ACE 2004 and ACE 2005

Table 6 presents the performance of instruction-tuned LLMs on the ACE 2004 and ACE 2005 datasets. Unlike the results observed with the GENIA dataset, Mistral demonstrated significantly better performance than Llama2 on both datasets, despite being much smaller in size. This suggests that recognizing nested NEs in domain-specific scenarios is more challenging with relatively smaller LLMs. Additionally, for general domains, as represented by the ACE 2004 and ACE 2005 datasets, the results indicate that newer models may have a superior ability to accurately follow the provided instructions.

G Details for Instructions

Table 7 and 8 show the examples of our nested NER-tailored instructions.

Model	Method	ACE 2004								
		Flat			Nested			Nesting		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Llama2-13B	Label Information (Wikipedia)	69.63	53.87	60.74	82.92	33.25	47.47	31.00	5.40	9.20
	Label Information (Wikipedia) + Nested Case Description	62.36	**63.40**	**62.87**	47.46	**52.86**	50.02	18.29	20.79	19.46
	Label Information (ChatGPT)	69.63	53.87	60.74	82.92	33.25	47.47	31.00	5.40	9.20
	Label Information (ChatGPT) + Nested Case Description	**63.52**	59.88	61.65	**52.01**	52.08	**52.05**	**20.86**	**20.97**	**20.92**
Mistral-7B-v0.3	Label Information (Wikipedia)	77.43	71.17	74.17	70.22	66.64	68.38	39.71	35.39	37.43
	Label Information (Wikipedia) + Nested Case Description	79.13	63.21	70.28	**87.85**	53.64	66.61	53.33	25.28	33.81
	Label Information (ChatGPT)	78.03	63.15	69.81	76.97	62.12	68.75	46.88	36.52	41.05
	Label Information (ChatGPT) + Nested Case Description	**80.68**	**65.49**	**72.30**	87.22	**56.89**	**68.86**	**55.29**	**30.34**	**39.18**
Qwen2-7B	Label Information (Wikipedia)	65.73	66.42	66.07	69.28	58.16	63.23	35.53	28.28	31.49
	Label Information (Wikipedia) + Nested Case Description	71.14	57.22	63.43	**77.71**	48.55	59.77	41.41	23.03	29.60
	Label Information (ChatGPT)	64.92	61.79	63.31	64.80	51.52	57.40	32.39	25.66	28.63
	Label Information (ChatGPT) + Nested Case Description	**67.78**	**53.64**	**59.89**	69.11	**45.23**	**54.68**	**33.22**	**18.91**	**24.11**

Model	Method	ACE 2005								
		Flat			Nested			Nesting		
		Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Llama2-13B	Label Information (Wikipedia)	69.63	53.87	60.74	82.92	33.25	47.47	31.00	5.40	9.20
	Label Information (Wikipedia) + Nested Case Description	**62.20**	59.48	**60.81**	59.32	**50.21**	54.39	22.70	18.10	20.14
	Label Information (Wikipedia)	69.63	53.87	60.74	82.92	33.25	47.47	31.00	5.40	9.20
	Label Information (ChatGPT) + Nested Case Description	60.75	**60.45**	60.60	**62.83**	49.79	**55.56**	**25.43**	**18.97**	**21.73**
Mistral-7B-v0.3	Label Information (Wikipedia)	76.45	71.04	73.65	71.83	64.14	67.77	42.82	37.93	40.23
	Label Information (Wikipedia) + Nested Case Description	73.77	62.45	67.64	73.60	53.42	61.91	40.25	28.02	33.04
	Label Information (ChatGPT)	77.65	69.26	73.22	74.90	60.42	66.88	45.68	36.42	40.53
	Label Information (ChatGPT) + Nested Case Description	**74.48**	**70.02**	**72.18**	**75.31**	**57.13**	**64.97**	**42.64**	**29.96**	**35.19**
Qwen2-7B	Label Information (Wikipedia)	64.90	59.43	62.04	74.21	47.59	57.99	34.55	20.47	25.71
	Label Information (Wikipedia) + Nested Case Description	67.94	62.29	64.99	**80.82**	54.77	65.29	43.96	25.86	32.56
	Label Information (ChatGPT)	67.70	48.24	56.34	78.53	41.35	54.17	38.92	17.03	23.69
	Label Information (ChatGPT) + Nested Case Description	**71.73**	**57.86**	**64.06**	71.06	**49.11**	**58.08**	**39.59**	**25.00**	**30.65**

Table 6: Results of tuning utilizing various instructions on ACE 2004 and ACE 2005 datasets. **Bold** numbers highlight the best scores within the respective method. Scores for each method were averaged over three trials.

Instruction Example for the GENIA dataset
Vanilla
Given entity label set: [cell_line, cell_type, DNA, RNA, protein]. Based on the given entity label set, please recognize the named entity in the given text. Consider there might be nested NEs, where one entity contains another. Text: {Text Input} Answer:
Label Information from Wikipedia
Given entity label set: [cell_line, cell_type, DNA, RNA, protein]. cell_line: A general term that applies to a defined population of cells that can be maintained in culture for an extended period of time, retaining stability of certain phenotypes and functions. cell_type: A classification used to identify cells that share morphological or phenotypical features. DNA: A polymer composed of two polynucleotide chains that coil around each other to form a double helix. RNA: A polymeric molecule that is essential for most biological functions, either by performing the function itself (non-coding RNA) or by forming a template for the production of proteins. protein: The large biomolecules and macromolecules that comprise one or more long chains of amino acid residues. Based on the given entity label set, please recognize the named entity in the given text. Consider there might be a nested case, where one entity contains another. Text: {Text Input} Answer:
Label Information Generated by ChatGPT
Given entity label set: [cell_line, cell_type, DNA, RNA, protein]. cell_line: Describes a specific line of cells used in experiments or research. cell_type: Describes the type or category of a cell. DNA: Denotes the genetic material present in a cell. RNA: Represents ribonucleic acid, involved in gene expression. protein: Refers to large molecules essential for various biological functions. Based on the given entity label set, please recognize the named entity in the given text. Consider there might be a nested case, where one entity contains another. Text: {Text Input} Answer:
Label Information Generated by ChatGPT + Nested Case Descriptions
Given entity label set: [cell_line, cell_type, DNA, RNA, protein]. cell_line: Describes a specific line of cells used in experiments or research. cell_type: Describes the type or category of a cell. DNA: Denotes the genetic material present in a cell. RNA: Represents ribonucleic acid, involved in gene expression. protein: Refers to large molecules essential for various biological functions. Based on the given entity label set, please recognize the named entity in the given text. Consider there might be a nested case, where one entity contains another. NDT: It consists of an entity containing a shorter entity tagged with a different type. NST: This case usually occurs when entities are originally represented by a hierarchy. Text: {Text Input} Answer:

Table 7: Instructions Examples

Instruction Example for the GENIA dataset

Label Information Generated by ChatGPT + Nested Case Descriptions

Given entity label set: [cell_line, cell_type, DNA, RNA, protein].
cell_line: Describes a specific line of cells used in experiments or research.
cell_type: Describes the type or category of a cell.
DNA: Denotes the genetic material present in a cell.
RNA: Represents ribonucleic acid, involved in gene expression.
protein: Refers to large molecules essential for various biological functions.
Based on the given entity label set, please recognize the named entity in the given text.
Consider there might be a nested case, where one entity contains another.
NDT: It consists of an entity containing a shorter entity tagged with a different type.
NST: This case usually occurs when entities are originally represented by a hierarchy.
Here are some examples.

Text 1: Alpha B2 proteins bound the PEBP2 site within the mouse GM-CSF promoter.
Answer 1: {"Alpha": "protein", "PEBP2": "protein", "PEBP2 site": "DNA",
"GM-CSF": "protein", "mouse GM-CSF promoter": "DNA"}

Text 2: Employing the EBV - transformed human B ell line SKW6.4.
Answer 2: {"EBV - transformed human B ell line": "cell_line", "human B ell line": "cell_line"}

Text 3: An IL - 2 promoter bearing a defective NF - chi B site was completely inactive in EBV - transformed B cells,
while it still had activity in Jurhat T cells.
Answer 3: {"IL - 2": "protein", "IL - 2 promoter": "protein", "NF - chi B": "protein", "NF - chi B site": "DNA", "chi B": "DNA",
"EBV - transformed B cells": "cell_line", "B cells": "cell_type", "T cells": "cell_type"}

Text 4: IL - 1 induces a rapid , protein synthesis - independent appearance of IL - 2R alpha mRNA
that is blocked by inhibitors of NF - kappa B activation.
Answer 4: {"IL - 1": "protein", "IL - 2R alpha mRNA": "RNA", "NF - kappa B": "protein"}

Text 5: In functional studies , PML expression was inhibited by addition of antisense oligomers targeting PML mRNA (alpha - PML).
Answer 5: {"PML": "protein", "PML mRNA": "RNA", "alpha - PML": "protein"}

Text: {Text Input}
Answer:

Table 8: Our Instructions Examples with Few-shot Samples.