PaDeLLM-NER: Parallel Decoding in Large Language Models for Named Entity Recognition

Anonymous ACL submission

Abstract

In this study, we aim to reduce generation latency for Named Entity Recognition (NER) with Large Language Models (LLMs). The main cause of high latency in LLMs is the sequential decoding process, which autoregressively generates all labels and mentions for NER, significantly increase the sequence length. To this end, we introduce **Pa**rallel **De**coding in **LLM** for **NER** (PaDeLLM-NER), a approach that integrates seamlessly into existing generative model frameworks without necessitating additional modules or architectural modifications. PaDeLLM-NER accelerates decoding by simultaneously generating all mentions at once, i.e., a label-mention pair per sequence. This results in shorter sequences and faster inference. Experiments reveal that PaDeLLM-NER significantly increases inference speed that is 1.76 to 10.22 times faster than the autoregressive approach for both English and Chinese. Concurrently, it maintains the prediction quality as evidenced by the micro F-score that is on par with the state-of-the-art across various datasets.

1 Introduction

Named Entity Recognition (NER), a fundamental task in Natural Language Processing (NLP), aims to extract structured information from unstructured text data. This includes identifying and categorizing key elements such as Organization, Geopolitical Entity and so on (referred to as *"labels"*) in inputs, and pairing them with relevant text spans extracted from the text (termed *"mentions"*). Conventionally, NER tasks are carried out through an extractive paradigm that entails token-level classification and the subsequent extraction of identified tokens (Ma et al., 2020; Liu et al., 2021).

Recent advancements in Large Language Models (LLMs) (Raffel et al., 2020a; Muennighoff et al., 2022; Touvron et al., 2023a,b; Bai et al., 2023; Yang et al., 2023a) have revolutionized numerous foundational tasks in NLP, including NER tasks (Paolini et al., 2020; Lu et al., 2022; Das et al., 2023; Lu et al., 2023; Wang et al., 2023c), through the adoption of a generative paradigm. This paradigm involves instruction-tuning a sequence-to-sequence (seq2seq) model. The model takes a sequence of unstructured text as input and produces a sequence of structured label-mention pairs as output. Generally, the output structured string should be formatted to meet two criteria: (1) it should have a clear and straightforward structure that facilitates post-processing for label and mention extraction, and (2) it needs to be generated fluidly and efficiently from the perspective of language models (Wang et al., 2023b).

In Table 1, we list two typically used autoregressive output formats found in the literature : (1) accommodate original input text to contain label information, which is referred to as *"augmented language"* (Paolini et al., 2020; Das et al., 2023); (2) directly using a customized, easily-parsed structured format to output all labels and mentions, which is called *"structured annotation"* (Lu et al., 2022, 2023; Wang et al., 2023c). These formats present certain challenges. For example, *augmented language* necessitates duplicating all original input text, thereby increasing output length and resulting in inference inefficiency. While *structure annotation* avoids replicating the entire input, it produces all labels and mentions in an autoregressive manner. This implies that each subsequently generated pair depends on its preceding pairs, and when the number of label-mention pairs is large, it will lead to longer sequences. As demonstrated in Chen et al. (2023c); Ning et al. (2023), high latency in LLMs mainly stems from lengthy sequence generation, we believe that by reducing the length of sequence, a more efficient inference scheme can be provided for NER tasks.

In light of this, we propose **Pa**rallel **De**coding

Variant	Input Unstructured Text	Output Structured Label-mention String
Augmented Language (Paolini et al., 2020; Das et al., 2023)	Japan, co-hosts of the World Cup in 2002 and ranked 20th in the world by FIFA, are favourites to regain their title here.	[Japan | LOC], co-hosts of the [World Cup | MISC] in 2002 and ranked 20th in the world by [FIFA | ORG], are favourites to regain their title here.
Structured Annotation (Lu et al., 2022, 2023; Wang et al., 2023c)	Cuttitta announced his retirement after the 1995 World Cup , where he took issue with being dropped from the Italy side that faced England in the pool stages.	((PER): (Cuttitta), (MISC): (1995 World Cup), (LOC): (Italy), (LOC): (England), (ORG): (NULL))

Table 1: Structured output string format used in the literature. The examples come from *CoNLL2003* dataset.

in **LLM** for **NER** (PaDeLLM-NER), a novel approach to accelerate the inference of NER tasks for LLMs. PaDeLLM-NER empowers the model with the capability to predict a single label-mention pair within a single sequence, subsequently aggregating all sequences to generate the final NER outcome. Specifically, in the training phase, we reconstruct the instruction tuning tasks, enabling LLMs to predict the count of mentions for a specific label and to identify the n^{th} mention within the entire input for that label (Figure 1). In the inference phase, LLMs first predict the number of mentions for all labels, then predict all label-mention pairs in parallel (Figure 2). Finally, results from all sequences are aggregated and duplicate mentions across labels are eliminated based on prediction probability. This approach results in a more efficient inference method, producing shorter sequences and enabling parallel decoding label-mention pairs in batches.

Comprehensive experiments have been conducted, demonstrating that PaDeLLM-NER effectively reduces the number of tokens produced in each sequence, thereby decreasing inference latency. Additionally, it maintains or even enhances prediction quality in both flat and nested NER for English and Chinese languages, compared to existing methods in the literature. To conclude, our contributions are as follows:

- We present PaDeLLM-NER, a novel approach tailored for NER using LLMs. This approach can predict all label-mention pairs in parallel, effectively reducing inference latency.
- Extensive experiments have been conducted, revealing that PaDeLLM-NER significantly improves inference efficiency. By completely decoupling the generation of label-mention pairs, the average sequence length is reduced to around 13% of that produced by conventional autoregressive methods. Correspondingly, the inference speed is 1.76 to 10.22 times faster than these previous approaches.
- Comprehensive experiments demonstrate that, in addition to its enhanced prediction speed, PaDeLLM-NER also maintains or surpasses the prediction quality of conventional autoregressive methods, on par with state-of-the-art performance on many NER datasets.

To the best of our knowledge, our technique stands as a pioneering approach in accelerating NER inference in LLMs by parallel decoding all label-mention pairs. This unique characteristic makes it complementary to other inference acceleration methods such as LLM.int8() (Dettmers et al., 2022) and speculative sampling (Chen et al., 2023a; Leviathan et al., 2023). Thus, it can be efficiently integrated with these methods.[1]

2 Related Work

2.1 Generative Models for NER

Before the era of LLMs, most research approached NER as a sequence labeling task, where each token is assigned a pre-defined tag (e.g., BIO scheme). In this line of work, usually pre-trained transformer-based language models (Ma et al., 2020; Liu et al., 2021) is combined with a tailored prediction head to perform a token-level classification, followed by the extraction of identified tokens.

Encouraged by the success of unifying multiple NLP tasks into a single seq2seq paradigm (Brown et al., 2020; Lester et al., 2021), especially with the evolution of LLMs (Raffel et al., 2020b; Achiam et al., 2023; Touvron et al., 2023a; Yang et al., 2023a), the trend of applying seq2seq models to NER tasks is gaining momentum (Xu et al., 2023), with both inputs and outputs being represented as sequences of text (Paolini et al., 2020; Lu et al., 2022; Das et al., 2023; Lu et al., 2023; Wang et al., 2023c). Recently, the focus of work on NER using LLMs has shifted towards zero-shot (Xie et al., 2023; Sainz et al., 2023) or few-shot learning (Ashok and Lipton, 2023; Chen et al., 2023b;

[1]Code is available at *URL masked for anonymous review*.

Das et al., 2023; Wang et al., 2023b), utilizing in-context learning (Chen et al., 2023b; Wang et al., 2023b), self-consistency (Wang et al., 2022b; Xie et al., 2023) or learning programming (Friedman et al., 2023; Sainz et al., 2023).

Unlike previous studies emphasizing few-shot performance with training-free prompt learning, our work focus on a fully supervised setting. More importantly, our primary objective is to speed up NER inference.

2.2 Inference Speedup in LLMs

Modern LLMs employ a sequential decoding strategy for token generation, which poses a significant challenge in terms of parallelization, especially as model size and sequence length increase (Ning et al., 2023). There is plenty of work in the literature to address this challenge (Wang et al., 2021; Frantar et al., 2023; Santilli et al., 2023; Xiao et al., 2023). One line of work falls into training-free category such as introducing extra modules for speculative sampling (Chen et al., 2023a; Leviathan et al., 2023). Another approaches explore modifying model architecture to accelerate inference, such as exiting at earlier layer (Elbayad et al., 2019; Schuster et al., 2022), or designing entirely different training and inference mechanisms (Lan et al., 2023; Yang et al., 2023b; Zhang et al., 2023). Different from previous works, we focus on exploring the inference speedup in LLMs with a focus on the NER task without the change of model architecture or introducing extra modules.

3 Method

In this section, we delve into the details of PaDeLLM-NER. First, we focus on reframing the instruction tuning tasks as outlined in Section 3.1. Second, we explore the two-step inference process, detailed in Section 3.2. Finally, we discuss the aggregation of results and the technique for eliminating duplicate mentions across labels, which is elaborated in Section 3.3. An illustration of PaDeLLM-NER is shown in Figure 1 and Figure 2.

3.1 Reframing of Instruction Tuning

Illustration of the reframing is presented in Figure 1. As an example, we use a case from the *CoNLL2003* dataset including four labels: person (PER), miscellaneous (MISC), location (LOC), and organization (ORG). The specifics of the input text and the corresponding ground truth are provided in the second row of Table 1.

During reformulation, a single unstructured text containing all label-mention pairs is split into several sequences. Each new sequence's output includes the count of mentions for a specified label (denoted as *"entity type"*), followed by the n^{th} mention of that label (denoted as *"<mention n>"*). Note that the count of mentions and their respective indices are represented using corresponding digit tokens from the LLM's vocabulary. Specifically, if there are no mentions, the model is trained to immediately predict the *"<eos>"* token, bypassing the need to predict mentions.

Therefore, in this example, one original training data is transformed into five new training data entries. These include two for predicting *"LOC"* (with 2 mentions), one for predicting *"MISC"* (with 1 mention), one for predicting *"PER"* (with 1 mention), and one for predicting *"ORG"* (with 0 mentions, directly predicting *"<eos>"*). Moreover, the number of mentions for each label and the text corresponding to each mention index can be easily obtained from the original ground truth, meaning that the number of new examples depends on the ground truth of that particular example.

With the newly reformulated training examples, we then apply the standard instruction tuning procedure. The model takes a sequence of text $t_1, t_2, \ldots, t_T$ consisting of input unstructured text and output structured label-mention pair. The optimization objective is cross-entropy loss $\mathcal{L}$ which can be defined as follows:

$$\mathcal{L} = -\frac{1}{T}\sum_{i=1}^{T} \log P\left(t_i \mid t_1, t_2, \ldots, t_{i-1}\right) \quad (1)$$

where $P\left(t_i \mid t_1, t_2, \ldots, t_{i-1}\right)$ represents the probability of i^{th} token t_i given the sequence of preceding tokens $t_1, t_2, \ldots, t_{i-1}$, as predicted by the model. Note that loss calculation begins from the number of mention tokens (i.e., texts enclosed by dashed-line frames). Theoretically, loss from text spans such as *"<mention n>"* could be ignored during this calculation, since they simply prompt the mention's order, which does not necessarily need to be generated by the model. However, our ablation studies show that ignoring these texts has negligible impact on model performance, a point further discussed in Section 4.3. Therefore, we adhere to the standard instruction tuning procedure.

This reformulation allows the model to focus one label-mention pair at a time, shortening the

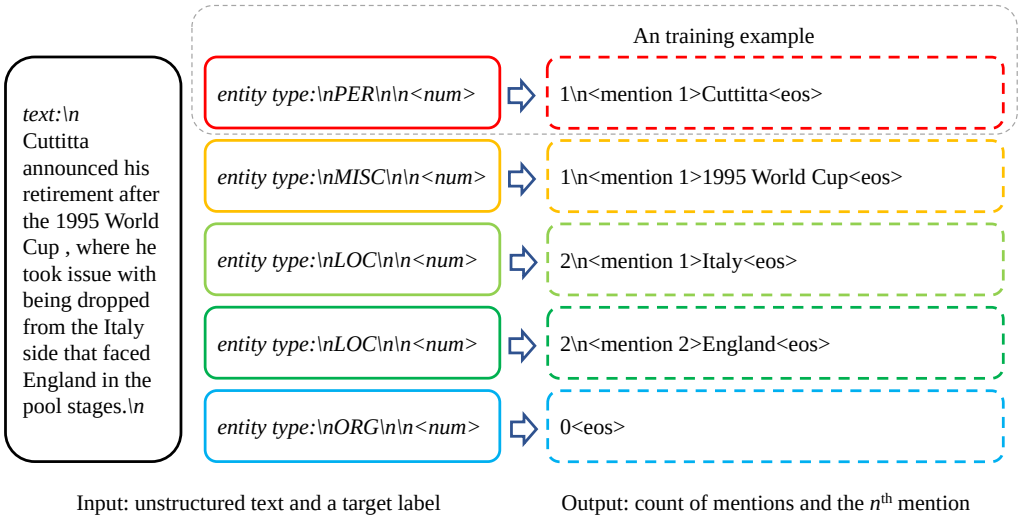


Figure 1: PaDeLLM-NER training paradigm: texts within frames of the same color represents one training example, where texts inside the solid-line frame are the input, and those inside the dashed-line frame are the output. *Italic* texts are prompt templates. The *"entity type"* signifies the label being predicted. The *"<num>"* indicates count of mentions for that label, and *"<mention n>"* refers to the n^{th} mention of a label in the input.

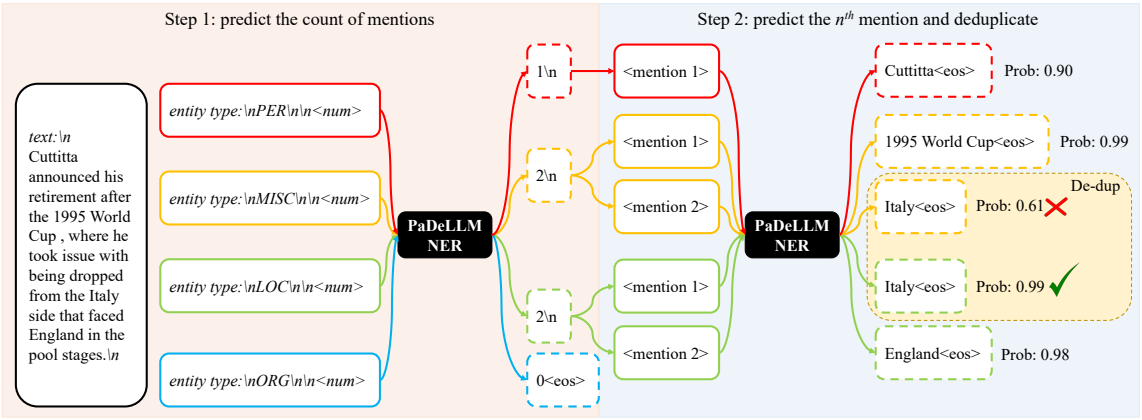


Figure 2: PaDeLLM-NER inference paradigm: texts enclosed in frames with identical colors indicate sequences of the same label. Specifically, the texts within solid-lined frames represent the added templates, while those within dashed-lined frames denote the prediction. In Step 1, the model predicts the number of mentions for all labels while in Step 2, it predicts the mentions. By aggregating mentions and labels from all sequences, the final NER results are obtained. Duplicate mentions appearing in different labels are resolved using prediction probabilities.

generated length per sequence. More details are shown in Appendix B.

3.2 Inference of Label-Mention Pairs

Given a trained LLM, we propose a two-step inference approach: firstly, to predict the number of mentions for a specific label based on the prompt; and secondly, given the label and provided index to precisely identify the corresponding mention.

Figure 2 shows the overview of PaDeLLM-NER inference. In Step 1, the model predicts the total number of mentions for each label in the input, based on the label prompt. A separate token *"\n"* signals the completion of this count prediction. If no mentions of the given label exist, the model generates an *"<eos>"* token, skipping Step 2 for that label. In Step 2, following adding the predicted mention count to the input, mention indexes templates are appended. Formally, if the predicted number of mention is m, then *"<mention n>"*, indicating the n^{th} mention of the specified label, is appended for each n within the set $\{1, 2, 3, \ldots, m\}$ and n is an integer. Subsequently, the corresponding mention is generated by the model conditioned on preceding tokens. Note that the decoding of all label-mention pairs occurs in parallel, allowing for

their simultaneous generation.

In practice, if there are sufficient GPU resources, the inference for the number of mentions for each label, as well as the subsequent inference for the mention text spans, can be allocating on separate GPUs. If GPU resources are limited, the inference can also be deployed on a single GPU using batch inference, facilitating parallel decoding. Using Figure 2 as an example, in Step 1, the batch size is four, as there are four labels in the dataset. In Step 2, the batch size is five, reflecting the five label-mention pairs determined in Step 1 (i.e., 1 in *"PER"*, 2 in *"MISC"*, 2 in *"LOC"*). This parallel decoding strategy is effective in reducing inference latency, especially in scenarios where inputs are received in a streaming manner.

3.3 Removal of Duplicate Mentions

Unlike autoregressive decoding, where subsequent label-mention pairs can attend preceding ones, PaDeLLM-NER generates each label-mention pair independently. This inference strategy means that the model might generate mentions erroneously repeated in multiple labels. As exemplified in Figure 2, the model correctly predicts the first mention of *"LOC"* as *"Italy"*, but it also incorrectly predicts the second mention of *"MISC"* as *"Italy"*.

To address the issue of duplicate mentions, we suggest employing prediction probability to remove repeated mentions. Specifically, we calculate the prediction probability for each instance of the mention. This is done using the formula: $P = \prod_{i=b}^{e} P(t_i|t_1, t_2, \ldots, t_{i-1})$ where b represents the starting token index of the mention text, and e denotes the ending token index. Then, for a mention that appears in multiple labels, the mention instance with the highest probability will be preserved. As illustrated in Figure2, *"Italy"* is categorized as *"MISC"* with only a 0.61 probability, which is lower than that for *"LOC"*, resulting in its removal. In practice, the probability of each token can be calculated concurrently with token generation. Consequently, this method enables an efficient and accurate identification of duplicate mentions without incurring additional costs. The effectiveness of this de-duplication approach is further explored in Section 4.3.

4 Experiments

In this section, we showcase the effectiveness of PaDeLLM-NER in terms of prediction quality and inference acceleration through experiments.

4.1 Setup

Datasets We evaluate our method on English and Chinese NER datasets. English datasets include the general domain flat NER *CoNLL2003* (Tjong Kim Sang and De Meulder, 2003), the nested NER *ACE2005* (Kirkpatrick, 2010), and the biomedical nested NER *GENIA* (Ohta et al., 2002). Chinese datasets include four commonly used general domain flat NER benchmarks *Resume* (Zhang and Yang, 2018), *Weibo* (Peng and Dredze, 2015), *MSRA* (Levow, 2006) and *Ontonotes 4.0* (Pradhan et al., 2013) and two vertical industrial domain flat NER datasets *YouKu* (Jie et al., 2019) and *Ecommerce* (Ding et al., 2019). The statistics of all datasets are shown in Appendix A.

Training setup We employ pre-trained version of Llama2-7b (Touvron et al., 2023b)[2] and Baichuan2-7b (Yang et al., 2023a)[3] as base models for English and Chinese study respectively. Additional implementation details are in Appendix C.

Inference setup For all generative models, we use greedy search with a beam size of 1, a maximum of 512 new tokens, and a temperature of 1.0. As described in Section 3.2, for PaDeLLM-NER, we adopt two inference settings: (1) each example is inferred on multiple GPUs to implement parallel decoding (i.e., each sequence is assigned on one GPU), termed as **PaDeLLM$_{\text{Multi}}$**; and (2) each example is inferred on a single GPU, employing batch decoding for parallel decoding, termed as **PaDeLLM$_{\text{Batch}}$**. Note that for PaDeLLM$_{\text{Multi}}$, we sequentially predict each sequence of one example to simulate parallel decoding on multiple GPUs.

Baselines As the primary focus of this work is on reducing inference latency in NER tasks using LLMs, we compare our method, PaDeLLM-NER, with traditional autoregressive approaches. As mentioned in Section 1, the main points of comparison are autoregressive structured output formats used in Paolini et al. (2020); Das et al. (2023) and Lu et al. (2022, 2023); Wang et al. (2023c), referred to respectively as **AutoReg$_{\text{Aug}}$** and **AutoReg$_{\text{Struct}}$**, as these are the approaches very close to our system. We reimplemented these methods for both English and Chinese datasets, utilizing the same

[2] https://huggingface.co/meta-llama/Llama-2-7b

[3] https://huggingface.co/baichuan-inc/Baichuan2-7B-Base

pre-trained LLMs as in PaDeLLM-NER. More details on the re-implementation are provided in Appendix C. Besides, we compare our approach with other recent state-of-the-art methods, including **BINDER** (Zhang et al., 2022a), **Gollie** (Sainz et al., 2023), and **DeepStruct** (Wang et al., 2022a) for English benchmarks, as well as **W^{2}NER** (Li et al., 2022), **NEZHA-BC** (Zhang et al., 2022b), and **SSCNN** (Zhang and Lu, 2023) for Chinese benchmarks, to show PaDeLLM-NER's efficacy in prediction quality.

Evaluation Our evaluation encompasses two dimensions: prediction quality and acceleration of NER inference. For assessing prediction quality, in line with Lu et al. (2022); Wang et al. (2023c), we employ the micro F-score.

Regarding inference acceleration, as per Ning et al. (2023), we evaluate using latency (in milliseconds). We record the latency with the code: *start = time.time(); model.generate(); latency = time.time() - start*. In PaDeLLM-NER, we add the latency of mention counting and label-mention pair generation as the latency of each sequence. The final latency for the example is determined by the highest latency across sequences, as the user can only obtain the result of an example when the slowest sequence is generated. We conduct experiments three times and use the average result to alleviate the effect of randomness. We also report the average sequence length (tokenized) to clearly demonstrate the extent of sequence length reduction in Appendix D. Evaluations of all models were performed on the same NVIDIA A100 GPU.

4.2 Main Results

Latency evaluation We investigate how PaDeLLM-NER reduces the end-to-end latency compared to baseline methods. Table 2 presents the average latency for each method across all datasets. First, it's clear that both PaDeLLM$_{\text{Multi}}$ and PaDeLLM$_{\text{Batch}}$ significantly reduce inference latency when compared to baseline methods, as highlighted by the substantial reduction in mean latency. For example, the mean latency reduction achieved between PaDeLLM$_{\text{Multi}}$ and AutoReg$_{\text{Struct}}$ stands at an impressive 791.55 ms, underscoring the significant improvement.

To more intuitively quantify the latency reduction of PaDeLLM-NER, we break down its speedup across different datasets in comparison to baseline methods in Figure 3. The speedup is computed by dividing the latency of baselines by the latency of PaDeLLM-NER. We can observe that PaDeLLM-NER consistently show a speedup over baseline methods across all datasets. The highest speedup is observed in the *Weibo* dataset when comparing AutoReg$_{\text{Struct}}$ vs. PaDeLLM$_{\text{Multi}}$, with a speedup of 10.22x. When we narrow our focus to the comparison between PaDeLLM$_{\text{Batch}}$ and the baseline methods, considering these methods utilize a single GPU for inference, we can still observe substantial speedup ranging from 1.76x to 4.73x. The speedup factor varies across different datasets, suggesting that the efficiency gains of PaDeLLM-NER may be influenced by the characteristics of each dataset. Interestingly, we can observe that the PaDeLLM$_{\text{Batch}}$ is slower than PaDeLLM$_{\text{Multi}}$ (378.40 ms vs. 223.57 ms), more analysis about this is shown in Section 5.

Overall, the Table 2 and Figure 3 suggest that PaDeLLM-NER significantly reduces latency compared to autoregressive methods, though the extent of this reduction varies by dataset and the specific baseline method it's compared to.

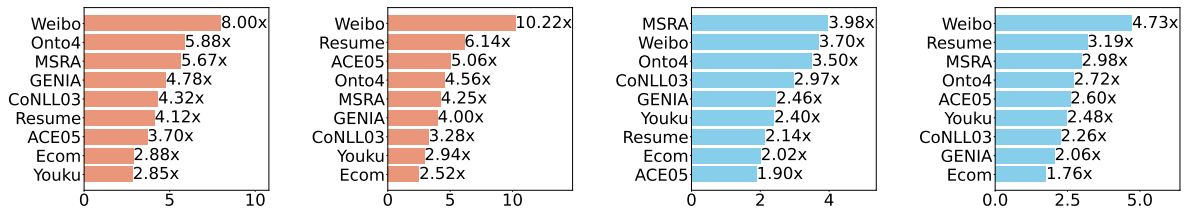
Prediction quality evaluation Table 3 shows the micro F-score of PaDeLLM-NER and other baseline methods. It's noteworthy that the micro F-scores for PaDeLLM$_{\text{Multi}}$ and PaDeLLM$_{\text{Batch}}$ are identical. Overall, PaDeLLM-NER outperforms baselines with the highest mean F-score of 84.79 and emerging as the best-performing method in 4 out of the 9 datasets. It excels particularly in the *Weibo*, *Youku* and *ACE2005* datasets and is competitive in others. We believe that the improved prediction could be attributed to shorter sequences, which lessen the challenge of long-range dependencies and enhance prediction quality. More analysis is discussed in Appendix E.

In summary, the results presented in Tables 2 and 3, demonstrate that our approach not only maintains superior prediction quality but also significantly reduces inference latency.

Comparison to state-of-the-art methods In Tables 4 and 5, we compare the micro F-scores of PaDeLLM-NER with other recent state-of-the-art methods. Our findings indicate that PaDeLLM-NER performs comparably to these methods across most datasets except for *Weibo*. Notably, in the *Youku* dataset, PaDeLLM-NER outperforms the previously best-performing method by an improvement of 1.81%. Note that the primary focus of our work is on accelerating prediction latency using

	English Dataset			Chinese Dataset						
AutoReg	**CoNLL03**	**ACE05**	**GENIA**	**Weibo**	**MSRA**	**Onto4**	**Resume**	**Youku**	**Ecom**	**Mean**
$\text{AutoReg}_{\text{Aug}}$	992.70	944.90	1,515.35	1,276.32	812.78	1,009.68	982.39	579.99	845.42	995.50
$\text{AutoReg}_{\text{Struct}}$	753.36	1,293.87	1,266.31	1,630.62	609.34	783.28	1,462.56	598.59	738.20	1,015.12
Ours										
$\text{PaDeLLM}_{\text{Multi}}$	**229.74**	**255.53**	**316.90**	**159.57**	**143.47**	**171.67**	**238.27**	**203.63**	**293.40**	**223.57**
$\text{PaDeLLM}_{\text{Batch}}$	333.89	498.50	616.01	344.75	204.24	288.43	459.20	241.25	419.40	378.40

Table 2: Comparison of inference latency (in milliseconds) between PaDeLLM-NER and baseline methods. Underscored font is the second-best method, while a bold font is the best method, also applied to subsequent tables.



(a) AR_{Aug} vs. $\text{PDLM}_{\text{Multi}}$ (b) $\text{AR}_{\text{Struct}}$ vs. $\text{PDLM}_{\text{Multi}}$ (c) AR_{Aug} vs. $\text{PDLM}_{\text{Batch}}$ (d) $\text{AR}_{\text{Struct}}$ vs. $\text{PDLM}_{\text{Batch}}$

Figure 3: Speedup of PaDeLLM-NER compared to Autoregressive methods.

	English Dataset			Chinese Dataset						
AutoReg	**CoNLL03**	**ACE05**	**GENIA**	**Weibo**	**MSRA**	**Onto4**	**Resume**	**Youku**	**Ecom**	**Mean**
$\text{AutoReg}_{\text{Aug}}$	**93.08**	83.04	70.16	59.04	**95.56**	79.20	**95.80**	86.07	76.02	81.99
$\text{AutoReg}_{\text{Struct}}$	91.87	82.99	77.90	56.07	90.92	**80.97**	95.74	86.85	81.57	82.76
Ours										
PaDeLLM-NER	92.52	**85.02**	77.66	**67.36**	95.03	80.81	94.98	**87.91**	**81.85**	**84.79**

Table 3: Comparison of prediction quality between PaDeLLM-NER and baseline methods.

State-of-the-art Method	**CoNLL03**	**ACE05**	**GENIA**
BINDER (Zhang et al., 2022a)	**93.33**	89.50	80.50
Gollie (Sainz et al., 2023)	93.10	**89.60**	-
DeepStruct (Wang et al., 2022a)	93.00	86.90	**80.80**
Our Method			
PaDeLLM-NER	92.52	85.02	77.66

Table 4: Comparing PaDeLLM-NER Performance with recent state-of-the-art methods on English datasets.

LLMs compared to baseline autoregressive methods, rather than achieving state-of-the-art results.

4.3 Ablation study

In this section, we set out to investigate the effects of the different aspects of PaDeLLM-NER.

Ignoring text spans in loss As discussed in Section 3.1, during training, it is permissible to overlook the loss of text span *"<mention n>"*, as the model does not need to generate this specific text, which is appended during inference. However, as shown in Table 6 illustrate, omitting these texts has minimal impact on prediction quality.

One possible explanation is that during training, the more significant challenge for LLMs lies in predicting the appropriate mention texts, rather than their format. As the model can readily learns to correctly position the format *"<mention n>"*, this aspect contributes minimally to the loss computation in training. In this case, computing the loss for all text is almost equivalent to "neglecting" the computation of loss for *"<mention n>"*.

De-duplication To demonstrate the effectiveness of the de-duplication technique, we established two configurations as detailed in Table 6. The *-De-duplication* denotes the pipeline operating without the de-duplication technique; $+\textit{De-duplication}_{\textit{Reverse}}$ indicates the pipeline that removes mentions with the highest probability, oppo-

State-of-the-art Method	Weibo	MSRA	Onto4	Resume	Youku	Ecom
NEZHA-BC (Zhang et al., 2022b)	-	-	-	-	-	**82.98**
SSCNN (Zhang and Lu, 2023)	71.81	-	82.99	96.40	86.10	81.80
W^{2}NER (Li et al., 2022)	**72.32**	**96.10**	83.08	**96.65**	-	-
Our Method						
PaDeLLM-NER	67.36	95.03*	80.81	94.98	**87.91**	81.85

Table 5: Comparing PaDeLLM-NER Performance with recent state-of-the-art methods on Chinese datasets. "*" indicates that results are not directly comparable.

Variant	CoNLL03	ACE05	GENIA	Mean
PaDeLLM-NER	**92.52**	**85.02**	**77.66**	**85.06**
+ Loss ignoring	92.01	**85.18**	73.47	83.55
- De-duplication	92.44	84.80	77.54	84.92
+ De-duplication$_{\text{Reverse}}$	92.38	84.44	77.38	84.73

Table 6: Ablations on ignoring loss and de-duplication.

site to the original de-duplication technique.

Theoretically, PaDeLLM-NER should be the top-performing method, as its de-duplication eliminates noisy mentions, enhancing precision. Following closely is the *-De-duplication*, allows duplicate mentions to persist. *+De-duplication$_{Reverse}$* ranks lowest since it removes correct mentions and retains incorrect ones, lowering recall and precision simultaneously. As shown in Table 6, the results consistently align with our expectations, thereby verifying the effectiveness of the de-duplication process. Moreover, the difference among these variants is subtle, which can be attributed to the rare cases where duplicate mentions exist. This further highlights the robustness of proposed method.

5 Speedup Analysis

One concern noted is that batch inference does not speed up as much as inference distributed across multiple GPUs. This observation is consistent with our expectations and supported by Chen et al. (2023c) who found that batch inference in LLMs tends to be slower than single sequence inference under identical conditions, likely due to limitations in GPU memory bandwidth (Cai et al., 2024).

Transitioning from these performance considerations, it's noteworthy that PaDeLLM-NER is self-contained and can be seamlessly integrated with various generative architectures, including well-established decoder-only models (Raffel et al., 2020a; Muennighoff et al., 2022; Touvron et al., 2023a,b; Bai et al., 2023; Yang et al., 2023a) and recent innovations like RWKV (Peng et al., 2023), as well as multi-modal LLMs (Liu et al., 2023b,a) for tasks like Key Information Extraction tasks (Huang et al., 2019), all without needing architectural changes or additional data/modules. Also, it could be incorporated with off-the-shelf LLMs such as ChatGPT (Achiam et al., 2023) and Claude-2[4] through prompt engineering without the need for further training, an aspect we plan to explore in future research.

6 Data Contamination Concerns

Since we are using LLMs as our foundational models, trained on extensive datasets from various online sources (Touvron et al., 2023b; Yang et al., 2023a), there is a chance that the models may have encountered parts of our evaluation sets during their pre-training phase, albeit unintentionally. This could potentially affect our experimental results. However, the primary focus of our experiments is the comparison of our proposed method with baseline methods. Given that these methods employ the same LLM as the base model, data contamination is unlikely to significantly impact the results.

7 Conclusion

In this work, we introduce **Pa**rallel **De**coding in **LLM** for **NER** (PaDeLLM-NER), a parallel decoding framework in LLMs for efficient NER. To achieve this, we recast autoregressive prediction of all label-mention pairs of traditional NER tasks into a two-step prediction: (1) predicting the number of mentions for a specific label, and (2) identifying the n^{th} mention for that label. This recast allows the model to parallel decode all label-mention pairs in batches. Extensive experimental results show that the proposed method can dramatically reduces inference time, achieving inference speedup ranging from 1.76 to 10.22 times, without compromising prediction quality. Lastly, extensive ablation studies are performed to clarify the design choices of PaDeLLM-NER.

[4] https://www.anthropic.com/news/claude-2

8 Limitations

One clear disadvantage of PaDeLLM-NER is the multiplication of training examples from one to $m * n$, where m is the label count and n the mention count. Despite this, given that low latency is a major bottleneck in LLMs, trading longer training for lower latency is justifiable. Also, given the impressive generalization ability of LLMs, we believe that this method can be smoothly adapted to few-shot scenarios requiring less computation resources, which will be explored in future work.

Additionally, accurately counting the number of mentions remains a challenge for LLMs as discussed in Appendix E. This issue could be alleviated by implementing a specialized counting model dedicated to this task (Liu and Low, 2023).

Finally, there are several instances of recomputation within the pipeline that can be optimized. Specifically, input texts are encoded multiple times throughout the process. During batch decoding, certain sequences may encounter the *"<eos>"* token earlier, but due to the nature of batch inference, these sequences continue to predict. We plan to improve this in the future by implementing enhancements like KV cache reuse and batch inference with an early quit mechanism, among other strategies.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, Jason D Lee, Deming Chen, and Tri Dao. 2024. Medusa: Simple llm inference acceleration framework with multiple decoding heads. *arXiv preprint arXiv:2401.10774*.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023a. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*.

Jiawei Chen, Yaojie Lu, Hongyu Lin, Jie Lou, Wei Jia, Dai Dai, Hua Wu, Boxi Cao, Xianpei Han, and Le Sun. 2023b. Learning in-context learning for named entity recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13661–13675, Toronto, Canada. Association for Computational Linguistics.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. 2023c. Punica: Multi-tenant lora serving. *arXiv preprint arXiv:2310.18547*.

Sarkar Snigdha Sarathi Das, Haoran Zhang, Peng Shi, Wenpeng Yin, and Rui Zhang. 2023. Unified low-resource sequence labeling by sample-aware dynamic sparse finetuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6998–7010, Singapore. Association for Computational Linguistics.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese ner with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1462–1467.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2019. Depth-adaptive transformer. In *International Conference on Learning Representations*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Dan Friedman, Alexander Wettig, and Danqi Chen. 2023. Learning transformer programs. *arXiv preprint arXiv:2306.01128*.

Zheng Huang, Kai Chen, Jianhua He, Xiang Bai, Dimosthenis Karatzas, Shijian Lu, and CV Jawahar. 2019. Icdar2019 competition on scanned receipt ocr and information extraction. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1516–1520. IEEE.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of NAACL*.

Andy Kirkpatrick. 2010. Researching english as a lingua franca in asia: The asian corpus of english (ace) project. *Asian Englishes*, 13(1):4–18.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. 2023. Copy is all you need. In *The Eleventh International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):10965–10973.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023a. Improved baselines with visual instruction tuning.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*.

Wei Liu, Xiyan Fu, Yue Zhang, and Wenming Xiao. 2021. Lexicon enhanced Chinese sequence labeling using BERT adapter. In *Proceedings of the ACL 2021 and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5847–5858, Online.

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023. Punifiedner: A prompting-based unified ner system for diverse datasets. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13327–13335.

Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5755–5772.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the ACL 2020*, pages 5951–5960, Online.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, et al. 2022. Crosslingual generalization through multitask finetuning. *arXiv preprint arXiv:2211.01786*.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Huazhong Yang, and Yu Wang. 2023. Skeleton-of-thought: Large language models can do parallel decoding. *arXiv preprint arXiv:2307.15337*.

Tomoko Ohta, Yuka Tateisi, Jin-Dong Kim, Hideki Mima, and Junichi Tsujii. 2002. The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the human language technology conference*, pages 73–77. Citeseer.

Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, RISHITA ANUBHAI, Cicero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2020. Structured prediction as translation between augmented natural languages. In *International Conference on Learning Representations*.

Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi

Zhou, Wei Li, and Peter J. Liu. 2020a. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020b. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, 21(1):5485–5551.

Oscar Sainz, Iker García-Ferrero, Rodrigo Agerri, Oier Lopez de Lacalle, German Rigau, and Eneko Agirre. 2023. Gollie: Annotation guidelines improve zero-shot information-extraction. *arXiv preprint arXiv:2310.03668*.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodola. 2023. Accelerating transformer inference for translation via parallel decoding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12336–12355, Toronto, Canada. Association for Computational Linguistics.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 -*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. 2023a. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2717–2739, Toronto, Canada. Association for Computational Linguistics.

Chenguang Wang, Xiao Liu, Zui Chen, Haoyun Hong, Jie Tang, and Dawn Song. 2022a. DeepStruct: Pretraining of language models for structure prediction. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 803–823, Dublin, Ireland. Association for Computational Linguistics.

Hanrui Wang, Zhekai Zhang, and Song Han. 2021. Spatten: Efficient sparse attention architecture with cascade token and head pruning. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 97–110. IEEE.

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

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, et al. 2023c. Instructuie: Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

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.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv preprint arXiv:2312.17617*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023a. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Nan Yang, Tao Ge, Liang Wang, Binxing Jiao, Daxin Jiang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Inference with reference: Lossless acceleration of large language models. *arXiv preprint arXiv:2304.04487*.

Miao Zhang and Ling Lu. 2023. A local information perception enhancement–based method for chinese ner. *Applied Sciences*, 13(17):9948.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2022a. Optimizing bi-encoder for named entity recognition via contrastive learning. In *The Eleventh International Conference on Learning Representations*.

Xin Zhang, Yong Jiang, Xiaobin Wang, Xuming Hu, Yueheng Sun, Pengjun Xie, and Meishan Zhang. 2022b. Domain-specific ner via retrieving correlated samples. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2398–2404.

Xinpeng Zhang, Ming Tan, Jingfan Zhang, and Wei Zhu. 2023. Nag-ner: a unified non-autoregressive generation framework for various ner tasks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 5: Industry Track)*, pages 676–686.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. 2023. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Variant	CoNLL03	ACE05	GENIA	Mean
PaDeLLM-NER	92.52	85.02	77.66	85.06
+ Model scale up to 13B	**93.02**	84.37	**78.84**	**85.45**

Table 7: Ablations on model scaling up.

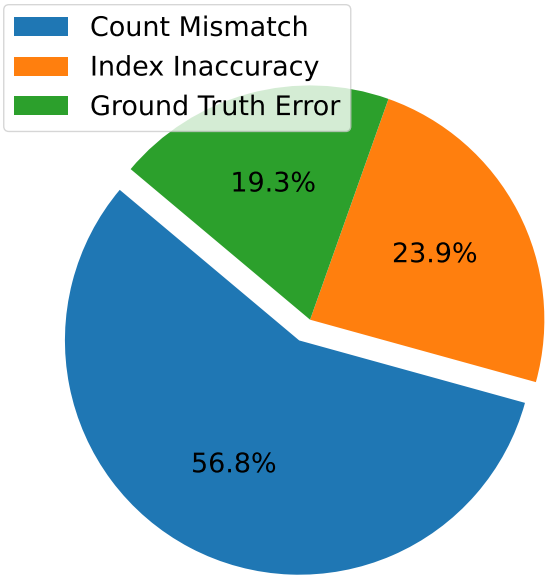


Figure 4: Percentage of different error types.

A Dataset Statistics

We evaluate our framework on 3 English and 6 Chinese flat/nested NER datasets. In Table 9, we present the detailed statistics. Note that while the statistics of the development set are reported, our training process does not involve the development set.

For the *MSRA* dataset, we excluded four outlier instances from the test set due to their excessively high number of names, significantly deviating from typical examples. These outliers not only posed challenges for model inference but also risked distorting the evaluation metrics, potentially leading to an inaccurate assessment of the model's performance on representative data.

Also, we perform label mapping to convert ground truth from special tokens to Chinese words following (Lu et al., 2023). Further details are provided in Table 10.

B Reformulation Examples

Two compete reformulated examples are presented in Table 11 for English and Chinese, respectively.

C Implementation Details

We train our model on all datasets for 4 epochs, using a batch size of 128 and a learning rate of $1e-5$, with the AdamW optimizer (Loshchilov and Hutter, 2018) and a cosine scheduler (Loshchilov and Hutter, 2017). The maximum input and output sequence lengths are set to 2048 and 512, respectively. Training is conducted on 8 NVIDIA A100 GPUs. This configuration is applied across all PaDeLLM-NER models, as well as two baseline models: $\text{AutoReg}_{\text{Aug}}$ and $\text{AutoReg}_{\text{Struct}}$.

D Sequence Length Reduction

Results of average sequence length produced by different approaches are presented in Table 8. Most notably, PaDeLLM-NER generates much shorter sequences than the other models across all datasets. The lengths range from 6.54 on *CoNLL20023* to 10.05 on *GENIA* for English datasets, and from 2.19 on *Weibo* to 4.87 on *Resume* for Chinese datasets. The mean length for PaDeLLM-NER is 4.86, which is significantly lower than the means of the other approaches: 35.54 for $\text{AutoReg}_{\text{Aug}}$ and 36.48 for $\text{AutoReg}_{\text{Struct}}$.

In summary, the result shows that PaDeLLM-NER produces much shorter generated sequences compared to the other methods, which is around 13.19% to 13.67% of the original length, respectively, indicating higher efficiency in its inference.

	English Dataset			Chinese Dataset						
AutoReg	**CoNLL03**	**ACE05**	**GENIA**	**Weibo**	**MSRA**	**Onto4**	**Resume**	**Youku**	**Ecom**	**Mean**
$\text{AutoReg}_{\text{Aug}}$	33.85	37.10	60.50	45.02	27.42	35.90	30.39	18.21	31.50	35.54
$\text{AutoReg}_{\text{Struct}}$	28.36	49.95	49.03	62.45	18.97	25.53	53.02	18.56	22.51	36.48
Ours										
PaDeLLM-NER	**6.54**	**8.29**	**10.05**	**2.19**	**2.23**	**2.68**	**4.87**	**3.66**	**3.27**	**4.86**

Table 8: Comparison of the number of generated tokens per sequence by PaDeLLM-NER with baseline methods.

Dataset	**Sentence**				**Mention**			
	#All	**#Train**	**#Dev**	**#Test**	**#All**	**#Train**	**#Dev**	**#Test**
CoNLL2003	20,744	14,041	3,250	3,453	35,089	23,499	5,942	5,648
ACE2005	9,210	7,194	969	1,047	30,634	24,441	3,200	2,993
GENIA	18,546	15,023	1,669	1,854	56,015	46,142	4,367	5,506
Weibo	1,890	1,350	270	270	2,701	1,894	389	418
MSRA*	50,725	44,364	-	4,361	80,214	74,703	-	5,511
OntoNotes 4.0	24,371	15,724	4,301	4,346	28,006	13,372	6,950	7,684
Resume	4,759	3,819	463	477	16,565	13,438	1,497	1,630
Youku	10,002	8,001	1,000	1,001	15,905	12,754	1,581	1,570
Ecommerce	4,987	3,989	500	498	15,216	12,109	1,540	1,567

Table 9: Dataset Statistics. "#" denotes the amount. For *MSRA*, we remove four outlier examples in test set.

Dataset	**#Entity**	**Entity**
Weibo	8	{"PER.NAM(Specific Name)":"名称特指", "PER.NOM(Generic Name)":"名称代称", "GPE.NAM(Specific Geo-Political Entity)":"行政区特指", "GPE.NOM(Generic Geo-Political Entity)":"行政区代称", "LOC.NAM(Specific Location)":"地点特指", "LOC.NOM(Generic Location)":"地点代称", "ORG.NAM(Specific Organization)":"组织特指", "ORG.NOM(Generic Organization)":"组织代称" }
MSRA	3	{"LOC":"地点', "PER":"名称", "ORG":"组织"}
OntoNotes 4.0	4	{"GPE":"地缘", "LOC":"地点", "PER":"名称", "ORG":"组织"}
Resume	8	{"NAME":"名称", "CONT(Nationality)":"国籍", "RACE":"民族", "TITLE":"职位", "EDU":"学历", "ORG":"公司", "PRO(Profession)":"专业", "LOC(Place of Birth)":"籍贯"}
Youku	3	{"TELEVISION":"电视剧", "PER(Celebrity)":"明星", "MISC":"其他"}
Ecommerce	2	{"HP(brand)":"品牌", "HC(commodity)":"商品"}

Table 10: Entity tag of each dataset and the conversion from tag used in dataset to corresponding Chinese natural language. For some tags that are hard to understand, we provide their meaning in brackets. "#" denotes the amount of entity types.

E Error analysis

PaDeLLM-NER error analysis For our error analysis, we utilize the *ACE2005* dataset. We sample and manually examine 50 erroneous examples for analysis. We seek to identify the root causes of errors, which we have categorized into three types: (1) incorrect mention count, referred to as *Count Mismatch*; (2) inaccuracies in the mention corresponding to a specific index, termed *Index Inaccuracy*; and (3) errors in the ground truth data, known as *Ground Truth Errors*.

The distribution of each error type is illustrated in Figure 4. It is important to note that a significant portion of the errors stem from inaccuracies in mention counts (i.e., Count Mismatch, about 56.8%), underscoring the necessity for enhancements in the model's counting capabilities. Accurate mention counts are pivotal for the quality of predictions. Overestimating the mention count often leads the model to either repeat the last entity or, more problematically, fabricate an entity, thereby escalating the rate of false positives. Conversely, underestimating the mention count results in the model's

Language	Input	Output
English	text: But Fischler agreed to review his proposal after the EU 's standing veterinary committee , mational animal health officials , questioned if such action was justified as there was only a slight risk to human health . entity type: PER <num>	1 <mention 1>Fischler
Chinese	文本(text): 公报最后说，墨西哥政府认为，贩毒以及洗钱等与毒品有关的活动是威胁到国家主权和安全的一个全球性问题。(The communique concluded by stating that the Mexican government considers drug trafficking and related activities such as money laundering to be a global issue that threatens national sovereignty and security.) 指定NER标签(entity type): 地点(LOC) <数量>(<num>)	1 <第1文段>(<mention 1>) 墨西哥(Mexican)

Table 11: Reformulated examples for English and Chinese dataset, respectively. We provide translations to facilitate understanding. The examples come from *CoNLL2003* and *MSRA* dataset.

inability to identify some entities, thus increasing the incidence of false negatives. Following closely is the Index Inaccuracy error, indicating that the model sometimes struggles to accurately pinpoint the correct mention for a given index, further emphasizing areas for improvement.

Interestingly, our analysis reveal that a significant portion of the model's predictions, specifically 19.3%, are actually correct, challenging the accuracy of the ground truth data. This observation suggests the presence of inaccuracies within the ground truth, contributing to an elevated rate of false positives. Prior research, as noted in studies by Min et al. (2022); Wang et al. (2023a); Zhou et al. (2023), has demonstrated that LLMs predominantly acquire their knowledge during the pretraining phase. These models develop certain "core beliefs" that tend to align more closely with human judgment. In this context, it appears that the models possess an inherent capability to rectify errors in the ground truth data, demonstrating their potential to improve data accuracy beyond initial human annotation.

F Model Scaling Up

As we increase the model size to 13B, Table 7 presents a mix of results. In datasets like *CoNLL2003* and *GENIA*, the model shows a significant improvement in predictions. In contrast, the results on *ACE2005* are slightly worse. Note that the improvement in *GENIA* is substantial, at approximately 1.18%. Based on these findings, it seems reasonable to suggest that continuously scaling up the model size has the potential to maintain the performance that is at least on par, or even superior, especially in specific industrial domains. However, this hypothesis warrants further investigation, involving more families of models (Raffel et al., 2020a; Muennighoff et al., 2022; Touvron et al., 2023a,b; Bai et al., 2023; Yang et al., 2023a) and a broader range of datasets. We leave this exploration for future work.