Rethinking Negative Instances for Generative Named Entity Recognition

Yuyang Ding[1], Juntao Li[1]*, **Pinzheng Wang[1], Zecheng Tang[1], Bowen Yan[2], Min Zhang[1]**

[1]Soochow University   [2]Tsinghua University

{yyding23,pzwang1,zctang}@stu.suda.edu.cn

{ljt,minzhang}@suda.edu.cn, yanbw@mail.tsinghua.edu.cn

Abstract

Large Language Models (LLMs) have demonstrated impressive capabilities for generalizing in unseen tasks. In the Named Entity Recognition (NER) task, recent advancements have seen the remarkable improvement of LLMs in a broad range of entity domains via instruction tuning, by adopting entity-centric schema. In this work, we explore the potential enhancement of the existing methods by incorporating negative instances into training. Our experiments reveal that negative instances contribute to remarkable improvements by (1) introducing contextual information, and (2) clearly delineating label boundaries. Furthermore, we introduce an efficient longest common subsequence (LCS) matching algorithm, which is tailored to transform unstructured predictions into structured entities. By integrating these components, we present GNER, a Generative NER system that shows improved zero-shot performance across unseen entity domains. Our comprehensive evaluation illustrates our system's superiority, surpassing state-of-the-art (SoTA) methods by 9 F_1 score in zero-shot evaluation.[1]

1 Introduction

Named Entity Recognition (NER) is a critical and challenging task in the field of Natural Language Processing (NLP). Previous NER models are constrained by a pre-defined label set and require extensive human annotations, which limits their flexibility and adaptability to unseen entity domains. Recent advantages in LLMs have enabled the models to be capable of generalizing to unseen tasks (Ouyang et al., 2022; Achiam et al., 2023) in an auto-regressive generation manner, making it possible to construct powerful NER systems. However, despite these advancements, recent studies (Wei et al., 2023; Li et al., 2023) show that

---

*Corresponding author

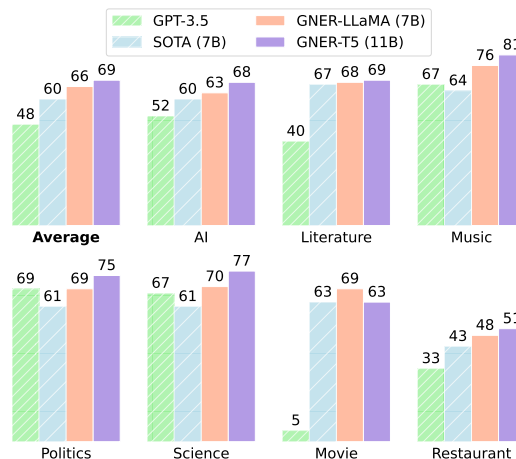[1]Code, datasets, and models are publicly available: https://github.com/yyDing1/GNER

Figure 1: Zero-shot performance of our models. Our models GNER-LLaMA and GNER-T5 both outperform the SoTA (Sainz et al., 2023) in zero-shot settings. GPT results are from Zhou et al. (2023).

the zero-shot performance of LLMs still falls behind the supervised training state-of-the-art (SoTA) methods, as LLMs train with limited NER data.

To bridge this gap, recent works have fine-tuned open-sourced LLMs on diverse NER datasets, enhancing their domain adaptability for NER tasks. They utilize varied task schemas to handle NER tasks across multiple domains. Specifically, InstructUIE (Wang et al., 2023) is fine-tuned on a wide range of IE datasets using a single-round conversation manner. Meanwhile, UniversalNER (Zhou et al., 2023) found that querying all entities at once is less effective than making multiple inquiries, with each inquiry focusing on one entity type at a time. Additionally, GoLLIE (Sainz et al., 2023) enhances zero-shot performance with well-crafted code-style guidelines. However, these approaches primarily adopt an entity-centric training strategy, focusing on recognizing entities while overlooking the non-entity text, which is crucial as negative instances. Actually, negative instances play an important role in traditional classification models like

BERT Tagging (Devlin et al., 2018). For generative models, the role of negative instances in the training process has not yet been fully explored.

To calibrate the potential enhancement of including negative instances in training, we first conduct a preliminary study. Through experiments, we show that negative instances can significantly boost the model's performance by (1) incorporating the contextual information, and (2) enhancing the label boundary between entities and non-entities. The possible drawback of introducing the negative instances is the increase of the prediction length, leading to inaccurate predictions, reflected by the word addition, omission and substitution. To tackle the inaccuracy drawbacks, we aim to design a more accurate and efficient algorithm to convert unstructured text into structured entities.

Inspired by the above observations, we design an effective and efficient **G**enerative **NER** framework named GNER. We first design a proper task schema integrating negative instances into the instruction tuning process. Additionally, we design an LCS Matching algorithm to tackle the issues in the structuring process efficiently. This innovation ensures accurate categorization and alignment of extracted entities. We also demonstrate that zero-shot performance can be enhanced with beam search through a self-correction mechanism. These strategic developments collectively advance the GNER framework, setting a new standard for accuracy and efficiency in the field of NER.

We conduct experiments on two representative generative models, Flan-T5 and LLaMA. The resulting models, GNER-T5 and GNER-LLaMA, outperform SoTA by a large margin. As stated in Fig. 1, GNER-LLaMA-7B outperforms the GoLLIE (Sainz et al., 2023) trained on Code-LLaMA-7B by 6 F_1 score. Furthermore, compared to the similarly configured model UniversalNER, GNER-LLaMA-7B shows an improvement of 12.7 F_1 score, with a 2.5× boost in inference speed. We also showcase the potential of smaller models with our 780M GNER-T5-large model, which outperforms all baseline models in both zero-shot and supervised scenarios.

2 Related Work

Named Entity Recognition Early works format Named Entity Recognition (NER) as a sequence labeling problem (Chiu and Nichols, 2016; Huang et al., 2015; Akbik et al., 2018; Qin et al., 2019; De-

Figure 2: A simplified example of instructions in InstructUIE (Wang et al., 2023), UniversalNER (Zhou et al., 2023) and GoLLIE (Sainz et al., 2023).

vlin et al., 2018), utilizing the BIO-Tagging scheme. Then, different methods are proposed to address more complex scenarios, i.e., nested and discontinuous NER. These methods regard NER as question answering (Li et al., 2020a; Mengge et al., 2020), span classification (Fu et al., 2021; Li et al., 2020b), dependency parsing (Yu et al., 2020), word-level relation classification (Li et al., 2022a), and so on. In most of these approaches, negative instances have played a crucial role in the training process, either by integrating all negative instances or employing sampling methods to select part of them (Li et al., 2022b). However, the performance of the above-mentioned supervised models significantly decreases in zero-shot settings (Liu et al., 2021), especially when the data and domain distribution significantly diverge from those seen of training.

Zero-shot NER Instruction tuning (Wei et al., 2021; Chung et al., 2022), also known as multi-task fine-tuning, has emerged as a leading method to achieve generalization to unseen tasks by fine-tuning pre-trained LLMs on a diverse collection of tasks phrased as text-to-text problems (Longpre et al., 2023). In NER, numerous works have explored the potential of LLMs across diverse domains. For instance, InstructUIE (Wang et al.,

2023) is fine-tuned on a wide range of IE datasets and achieves impressive results in both zero-shot and supervised settings. UniversalNER (Zhou et al., 2023) explores the effectiveness of knowledge distillation and multi-round conversational training paradigms in enhancing model generalization, achieving superior results. GoLLIE (Sainz et al., 2023) introduces an innovative strategy by integrating well-crafted code-style guidelines into instructions, which has been found to further improve the model's zero-shot performance. A simplified version of the task schema of the mentioned methods above is shown in Fig. 2. It can be concluded that the task schema plays an important role in determining the learning paradigm of models, significantly influencing their performance. We observe that these methods are **entity-centric**, meaning only the entity portions are involved in training and used for backpropagation to train the model.

3 Incorporating Negative Instances

In this section, we start from the entity-centric schema and explore the possible improvements by incorporating negative instances. Through experiments, we demonstrate the impact of negative instances in mitigating Unlabeled Errors (UE), Noisy Errors (NE), and Boundary Errors (BE) by (1) introducing contextual information and (2) enhancing entity boundaries.

3.1 Definition & Settings

The Named Entity Recognition (NER) task can be formally defined as a function mapping of input tokens $X = \{x_1, x_2, \ldots, x_n\}$ and a pre-defined set of entity types $L = \{l_1, l_2, \ldots l_m\}$ to entity labels $Y = \{y_1, y_2, \ldots, y_n\}$. The positive and negative instances can be formulated as follows:

$$\begin{aligned} P &= \{(x_i, y_i) \mid i \in \{1, \ldots, n\}, y_i \in L\}, \\ N &= \{(x_i, y_i) \mid i \in \{1, \ldots, n\}, y_i = O\}, \end{aligned} \quad (1)$$

where O represents non-entity text.

For the experimental setup, we choose Flan-T5-large, a model with 780M parameters, as our backbone. Additionally, we sample 10K samples from the Pile-NER (Zhou et al., 2023) dataset as our training set and 200 samples for each subtask of CrossNER (Liu et al., 2021) validation set to evaluate the model's zero-shot performance. To conduct a more detailed evaluation, in addition to the F_1 score, we introduce the following three metrics to assess model performance:

An example of the constructed prompt

Token inputs (X)**:** John explored Tokyo , sampling its famed sushi , and flew back to New York .
Entity type (L)**:** [Person, Location]

Training prompt
w/o context (entity-centric):
[John](Person) [Tokyo](Location) [New York](Location)
w/ context length 1:
[John](Person) explored [Tokyo](Location) , ... to [New York](Location) .
w/ full context:
[John](Person) explored [Tokyo](Location) , sampling its famed sushi , and flew back to [New York](Location).
w/ full context and label boundary (BIO):
John(B-Person) explored(O) Tokyo(B-Location) ,(O) sampling(O) its(O) famed(O) sushi(O) ,(O) and(O) flew(O) back(O) to(O) New(B-Location) York(I-Location) .(O)
w/ full context and label boundary (BIOES):
John(S-Person) explored(O) Tokyo(S-Location) ,(O) sampling(O) its(O) famed(O) sushi(O) ,(O) and(O) flew(O) back(O) to(O) New(B-Location) York(I-Location) .(O)

Figure 3: Constructed prompts used for training.

	UE	NE	BE	F_1
w/o context (entity-centric)	7.8	16.7	3.8	59.0
w/ context length 1	7.6	15.7	3.8	60.3
w/ full context	7.7	14.9	3.7	61.0
w/ full context & BIO	7.5	14.4	3.3	**61.8**
w/ full context & BIOES	7.6	14.7	3.5	61.2

Table 1: Unlabeled Error (UE), Noisy Error (NE) and Boundary Error (BE) in our preliminary study.

Unlabeled Error (UE) The model fails to recognize the entity and labels it as "O".

Noisy Error (NE) The model mistakenly label an entity with another incorrect entity tag.

Boundary Error (BE) The model correctly predicts the entity type but fails to identify its full extent, either capturing only a portion of the entity or resulting in overlaps.

3.2 Learning with Entity Context

We integrate the contextual information before and after an entity into our training process to explicitly enable the model to recognize entities based on their surrounding context. Specifically, we introduce negative instances that are closest to the entity, extending up to a length L, until encountering the boundary of the sentence, as part of our training instances. An example of our constructed training prompt is shown in Fig. 3, and the corresponding results are summarized in Table 1 and Fig. 4.

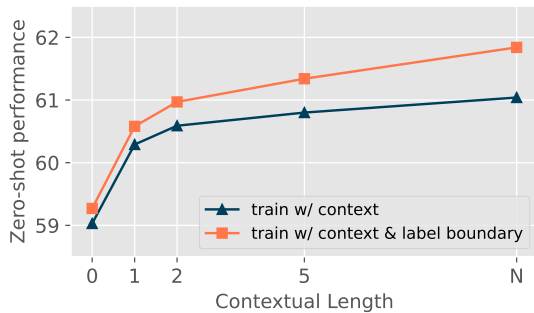The context surrounding entities plays a signif-

Figure 4: Zero-shot performance of training with entity context and enhanced boundary strategies. A contextual length of 0 indicates no context is included, while a length of N signifies that the entire sentence is included.

icant role in determining their categories, with a notable improvement observed when increasing the contextual length from 0 to 1. As the contextual length increases, performance progressively improves, showing that the model benefits from the context, and the ratio of noisy error (NE) significantly decreases, largely contributing to the improvement in the final F_1 score. Qualitatively, we further analyze through case studies why the closest negative instances contribute to improvement. We discover that the model tends to learn more from the context, such as "flew to" prompting the model to focus more on the following entity "New York" instead of merely memorizing "New York" as a location. We also experiment by placing terms with multiple meanings, such as Jordan, Amazon, and Mercury, after "flew to" and observe that the model consistently identified them correctly.

3.3 Entity Boundary of Generative Model

The above analysis has demonstrated the effectiveness of entity contexts, where labels are applied exclusively to entity parts. We then adopt BIO-tagging (Huang et al., 2015) to enrich all label information, with the training prompt as shown in Fig. 3. The results in Table 1 indicate that the introduction of BIO tagging effectively strengthens the boundaries around entities, leading to a significant reduction in boundary errors. As shown in Fig. 4, compared to training with contexts alone, incorporating the BIO-tagging strategies results in consistent improvements across various contextual lengths. In addition to BIO tagging, we also try the BIOES tagging scheme, based on the intuition that the BIOES tagging method provides a stronger delineation of entity boundaries. However, we find

that the performance of BIOES tagging is not as good as BIO tagging. The results are listed in Table 1. Upon further analysis, we discover that the BIOES tagging seems harder to learn under the auto-regressive generation manner, where each token is predicted sequentially. For instance, both "B-" and "S-" can serve as the beginning of an entity, but only "B-" can be followed by "I-", which may confuse the model for subsequent words.

3.4 The role of negative instances

The improvement in model performance (F_1 score) can be explained as follows:

Precision and Recall To put it more directly, the improvements over entity-centric approaches are primarily reflected in (1) Precision: The context surrounding an entity often leads to a more accurate determination of its type, and (2) Recall: The model is guided to make judgments on every token in a sentence (including those in non-entity texts), which helps recall more entities. We observe improvements in both recall and precision, which in turn lead to an increase in the F_1 score.

Less Unlabeled, Noisy and Boundary Error As indicated in Table 1, context helps mitigate Unlabeled and Noisy Errors, while BIO tagging strengthens entity boundaries, thereby reducing Boundary Errors. These reductions in errors directly contribute to improved performance.

4 Method

In this section, we present our GNER framework. We start by describing our task schema, which integrates negative instances into the training process for better usage of contextual information (section 3.2) and sensitivity to the entity boundaries (section 3.3), followed by the correlated tuning strategies. We also propose an effective longest common subsequence (LCS) matching algorithm, to convert the model's unstructured text outputs into structured data efficiently, thereby enhancing the accuracy of our system.

4.1 Task Schema

Integrating negative instances, specifically those parts of the sentence labeled as "O" to indicate non-entity text, enhances the generative process by including contextual information and the discrimination of entity boundaries, thereby boosting the model's performance, as detailed in section 3.

3464

Task Description:
Please analyze the sentence provided, identifying the entity type for each word on a token-by-token basis.
Output format is: word_1(label_1), word_2(label_2), ...
Guideline:
We'll use the BIO-format to label the entities, where:
1. B- (Begin) indicates the start of a named entity.
2. I- (Inside) is used for words within a named entity but are not the first word.
3. O (Outside) denotes words not part of a named entity.
Use the specific entity tags: $l_1, l_2, \ldots, l_m$ and O .

Input: $x_1 \quad x_2 \quad \ldots \quad x_n$

Output: $x_1(\hat{y}_1) \quad x_2(\hat{y}_2) \quad \ldots \quad x_n(\hat{y}_n)$

Figure 5: Prompt used for instruction tuning.

Due to the token-by-token generation paradigm of generative models, we design a token-by-token prediction task schema, where the model predicts the category of each token as it generates them, either entities or non-entities. This schema offers a more direct and focused way, where each token is annotated individually and assigned a specific entity label based on its context within the sequence.

4.2 Instrucion Tuning

Instruction Format As shown in Fig. 5, our designed instruction prompt includes four parts: task description, guideline, input, and output. To enhance our model's ability to generalize across diverse labels and effectively handle real-world data, we implement some regularization strategies: (1) class order shuffling, where the order of entity classes is randomly shuffled, and (2) external entity sampling[2], involving the entity types that are absent in the given sentence in the training prompt.

Task Adaption & Supervised Fine-tuning Zero-shot capabilities of LLMs in NER are limited due to their exposure to relatively little NER data during training. To equip the model with capabilities specific to NER tasks, we first perform task adaptation on NER data spanning various domains. Subsequently, to assess the model's zero-shot capabilities, we evaluate it against unseen entity types. We proceed to extensively fine-tune our models on a wide range of publicly available NER data, aiming to enhance our model's effectiveness in supervised settings, followed by supervised evaluations.

---

[2]Zhou et al. (2023) refers to this as negative entity sampling, which is different from the negative instances discussed in this work. We term it "external" to differentiate it.

4.3 LCS Matching Algorithm

The prediction length of the model increases with the integration of entity contexts and BIO tagging strategies. A longer generation sequence might bring challenges to the popular generative LLMs. The model's output may include omissions, additions, and substitutions of words. We launch a detailed case study and find the potential causes of these issues: (1) noise in the original text, (2) missing words in the vocabulary, and (3) accumulative exposure bias. The representative examples, issue proportion, along with detailed analysis, are documented in Appendix C.

To handle these problems, we develop a LCS Matching algorithm that provides a straightforward and effective solution to these challenges. Formally, given a sentence $X = \{x_1, x_2, \ldots, x_n\}$, the generated outputs can be formatted as "$\tilde{x}_1(\tilde{y}_1) \quad \tilde{x}_2(\tilde{y}_2) \quad \ldots \quad \tilde{x}_m(\tilde{y}_m)$". Firstly, we utilize regular expression matching to obtain the predicted sequence $\tilde{X} = \{\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_m\}$ and the corresponding answers $\tilde{Y} = \{\tilde{y}_1, \tilde{y}_2, \ldots, \tilde{y}_m\}$. Due to the inherent uncertainties in generation, $\tilde{X}$ often differs from X. Next, we establish a one-to-one correspondence between the words in the original sequence X and the generated sequence $\tilde{X}$, then map the labels of the corresponding words $\tilde{Y}$ back to obtain the final prediction results $\hat{Y}$. A common method involves calculating the Longest Common Subsequence (LCS) between X and $\tilde{X}$ to identify the correspondence between words, using the classic dynamic programming algorithm with time complexity of $O(N^2)$. Combined with the actual NER task scenarios and our task schema, we make the optimizations in the matching algorithm and condition. The resulting algorithm can handle these issues effectively with a time complexity of $O(N \log N)$. The optimization in matching conditions (i.e., Back Tokenization) further enhances the robustness across various models in our system. More details concerning the optimization can be seen in Appendix D.

5 Experiments

5.1 Settings

Datasets The datasets used in our experiments include: (1) **Task Adaptation Datasets:** Following the setting of Zhou et al. (2023), we first train our model with Pile-NER, which consists of approximately 240K entities across 13K distinct entity categories. These passages are sampled from the

Model	Backbone	AI	Literature	Music	Politics	Science	Movie	Restaurant	Avg
ChatGPT	-	52.4	39.8	66.6	68.5	67.0	5.3	32.8	47.5
InstructUIE[†]	flan-t5-xxl (11B)	48.4	48.8	54.4	49.9	49.4	<u>63.0</u>	21.0	47.8
GoLLIE-7B[†]	Code-LLaMA-7B	60.3	67.1	64.5	60.8	60.5	<u>63.0</u>	43.4	59.9
GoLLIE-13B[†]	Code-LLaMA-13B	63.8	60.1	68.5	56.2	61.5	62.5	49.8	60.3
UniNER-7B[‡]	LLaMA-7B	53.5	59.4	65.0	60.8	61.1	42.4	31.7	53.4
UniNER-13B[‡]	LLaMA-13B	54.2	60.9	64.5	61.4	63.5	48.7	36.2	55.6
GLiNER-L[‡]	DeBERTa-v3-300M	60.6	<u>68.4</u>	69.5	<u>74.8</u>	69.4	57.2	42.8	63.2
GNER-T5[‡]	flan-t5-base (250M)	56.8	58.7	72.3	64.5	68.0	54.5	41.4	59.5
	flan-t5-large (780M)	62.6	58.2	76.7	67.0	<u>72.6</u>	58.6	48.6	63.5
	flan-t5-xl (3B)	62.1	64.9	<u>80.6</u>	73.7	68.7	<u>63.0</u>	<u>49.8</u>	66.1
	flan-t5-xxl (11B)	**68.2**	**68.7**	**81.2**	**75.1**	**76.7**	62.5	**51.0**	**69.1**
GNER-LLaMA[‡]	LLaMA-7B	<u>63.1</u>	68.2	75.7	69.4	69.9	**68.6**	47.5	<u>66.1</u>

Table 2: Zero-shot evaluation results, where † denotes IE Models and ‡ denotes NER Models. Results for ChatGPT and UniNER are from Zhou et al. (2023); InstructUIE are from Wang et al. (2023); GoLLIE are from Sainz et al. (2023); GLiNER-L are from Zaratiana et al. (2023). We bold the best results and underline the second-best results. More details about the performance including error bars are shown in Appendix F.

Pile Corpus (Gao et al., 2020) and subsequently processed using ChatGPT to generate the inherent entities openly. To evaluate the model's zero-shot performance in unseen entity types, we adopt two widely-used datasets, i.e., CrossNER (Liu et al., 2021) and MIT (Liu et al., 2013). (2) **Supervised Datasets:** Following the task adaptation phase, the performance of the model can be further enhanced by training across a wide range of well-annotated NER datasets (Zhou et al., 2023). To achieve this, we compile 18 public NER datasets in the BIO format for additional training, subsequently assessing performance on the test splits of these 18 datasets. From the 20 datasets used in Wang et al. (2023), we exclude two nested NER datasets, ACE2005 and GENIA, due to their incompatibility with the BIO format. Following the settings of Wang et al. (2023), we randomly select 10K data points from each dataset to create a mixed set. In cases where a dataset contains fewer than 10K samples, we incorporate its entire dataset. Additional information regarding the datasets is available in Appendix A.

Compared Baselines Our main point of comparison is **UniversalNER** (Zhou et al., 2023) as it is the approach closest to our system, with similar data and training procedures. Another baseline considered for comparison is **GLiNER** (Zaratiana et al., 2023), which utilizes bi-directional models to match entity types with textual spans in a latent space. We also include some strong Information Extraction (IE) systems like **InstructUIE** (Wang et al., 2023), which is based on Flan-T5-xxl (Chung et al., 2022) and fine-tuned on diverse informa-

tion extraction datasets, and **GoLLIE** (Sainz et al., 2023), which is based on Code-LLaMA (Roziere et al., 2023), and use guidelines to improve model's zero-shot performance. We use strict entity-level micro-F_1 as the evaluation metric for comparison. Previous work lack a uniform setting; UniNER removed the "else" entity type in the CrossNER dataset, while InstructUIE, GoLLIE, and GLiNER retained it. To standardize the settings, we reevaluated their methods using their publicly released checkpoints and code, ensuring that all test set configurations are consistent with UniNER.

Backbones & Implementation Generative models typically consist of two types of architectures, i.e., the encoder-decoder architecture and the decoder-only architecture. We conduct experiments on both of these architectures. Specifically, we select Flan-T5 (encoder-decoder) and LLaMA (decoder-only) as our backbone models. To ensure a fair comparison, our training settings for GNER-T5 align with those of InstructUIE (Wang et al., 2023), and those for GNER-LLaMA are consistent with UniversalNER (Zhou et al., 2023). Due to our model producing longer output sequences, we implement longer length limits for both input and output. More details can be found in Appendix B.

5.2 Zero-shot Evaluation

We evaluate the zero-shot performance of our models after the domain adaptation phrase. Table 2 summarizes the results. Our model demonstrates significant improvements compared to other models. Significantly, although our GNER-LLaMA

Method Backbone	ChatGPT -	InstructUIE GNER-T5 flan-t5-xxl (11B)		Δ	UniNER GNER-LLaMA LLaMA-7B		Δ
AnatEM	30.7	88.52	**90.30**	+1.78	88.65	90.24	+1.59
bc2gm	40.2	80.69	**84.29**	+3.60	82.42	83.18	+0.76
bc4chemd	35.5	87.62	**90.04**	+2.42	89.21	89.40	+0.19
bc5cdr	52.4	89.02	89.95	+0.93	89.34	**90.27**	+0.93
Broad Twitter	61.8	80.27	**84.56**	+4.29	81.25	83.74	+2.49
CoNLL2003	52.5	91.53	93.28	+1.75	93.30	**93.60**	+0.30
FabNER	15.3	78.38	83.20	+4.82	81.87	**85.39**	+3.52
FindVehicle	10.5	87.56	97.37	+9.81	98.30	**98.62**	+0.32
HarveyNER	11.6	74.69	**76.33**	+1.64	74.21	74.73	+0.52
Movie	5.3	89.58	89.28	-0.30	90.17	**90.23**	+0.06
Restaurant	32.8	82.59	**83.84**	+1.25	82.35	81.73	-0.62
MultiNERD	58.1	90.26	**94.35**	+4.09	93.73	94.30	+0.57
ncbi	42.1	86.21	87.27	+1.06	86.96	**89.27**	+2.31
Ontonotes	29.7	88.64	**91.83**	+3.19	89.91	90.69	+0.78
PolyglotNER	33.6	53.31	66.90	+13.59	65.67	**67.52**	+1.85
TweetNER7	40.1	65.95	**67.97**	+2.02	65.77	66.87	+1.10
WikiANN	52.0	64.47	85.19	+20.72	84.91	**86.87**	+1.96
wikiNeural	57.7	88.27	**93.71**	+5.44	93.28	**93.71**	+0.43
Avg	34.9	81.53	**86.15**	+4.62	85.07	86.09	+1.02

Table 3: Supervised evaluation results. Δ indicates the improvement over the corresponding baseline. Results for InstructUIE and UniNER are derived from Wang et al. (2023) and Zhou et al. (2023), respectively.

Model	#Params.	0-shot	Sup.	Instance/s
InstructUIE	11B	47.8	81.53	3.4
UniNER-7B	7B	53.4	85.07	1.6
GNER-T5-small	77M	48.2	77.43	32.5
GNER-T5-base	248M	59.5	83.21	20.2
GNER-T5-large	783M	63.5	85.45	11.5
GNER-T5-xl	3B	66.1	85.94	4.6
GNER-T5-xxl	11B	69.1	86.15	3.0
GNER-LLaMA	7B	66.1	86.09	4.0

Table 4: Model's performance and inference speed in zero-shot and supervised settings. The inference speed is tested in a single A100 node with batch size 4 per device. More details are outlined in Appendix F.

Sequence Length	0-60	60-100	100-200
LCS $O(N^2)$	1.0×	1.0×	1.0×
LCS $O(N \log N)$	2.9×	5.5×	6.1×
LCS (ours)	3.8×	12.6×	17.3×

Table 5: Acceleration effect of our optimized algorithm across different length ranges. All samples are selected from the generated results under supervised settings.

model shares the same backbone model (LLaMA-7B) and dataset (Pile-NER) with UniNER (Zhou et al., 2023), it demonstrates a notable improvement. Our results show that our 7B model outperforms the UniNER model of the same scale by approximately 12.7 F_1 score points on average, and exhibits improvements across every dataset. Remarkably, our 7B model surpasses the UniNER 13B model by 10.5 points. When considering smaller backbone models such as GNER-T5-base and GNER-T5-large, it's noteworthy that they also outperform all the aforementioned strong baselines.

5.3 Supervised Evaluation

To test our model's performance on supervised data, we conduct supervised multi-task fine-tuning based on the NER-specialized model. The results are summarized in Table 3 and 4. We first compare our approach with two closely related baselines, InstructUIE and UniNER, as we share the same backbone model and train with similar data. As a result, our method demonstrates significant improvements over these baselines: GNER-T5 achieves a 4.6-point increase in the F_1 score, while GNER-LLaMA shows a 1-point F_1 score improvement. Moreover, we observe consistent enhancements across almost all datasets. We also experiment with smaller models, considering both effectiveness and inference efficiency. As shown in Table 4, our GNER-T5-large model, with only **10%** the parameter size of UniNER, achieves superior performance and boasts **10×** the inference efficiency.

5.4 Ablation Results

We have demonstrated the effectiveness of negative instances in section 3. In this part, we conduct the ablation study to evaluate the performance of our LCS Matching Algorithm. Our focus lies in two aspects: (1) how optimization in the algorithm

Method	GNER-T5-large		GNER-LLaMA	
	0-shot	Sup.	0-shot	Sup.
Ours	63.47	85.45	66.07	86.09
w/o BT	63.16	85.09	66.07	86.09
w/o LCS+BT	62.31	84.91	65.77	85.99

Table 6: Ablation study of LCS Matching and Back Tokenization.

Beam size	1	2	3	4
UniNER-7B	53.46	52.87	-	-
GNER-T5-base	59.46	60.32	60.40	60.44
GNER-T5-large	63.47	64.13	64.27	64.31
GNER-T5-xl	66.12	66.81	66.86	66.88
GNER-T5-xxl	69.06	69.20	69.33	69.33
LLaMA-7B	66.07	66.87	67.00	67.08

Table 7: Zero-shot performance of UniNER and our model GNER via beam search.

A Self-correction Example with beam size 2

Token inputs: What was the fog rated ?
Ground Truth:
What(O) was(O) the(B-title) fog(I-title) rated(O) ?(O)

Medium prediction results
highest beam score:
What(O) was(O) the(O) fog(O)
second-highest beam score:
What(O) was(O) the(B-title) fog(I-title)

Final prediction results:
What(O) was(O) the(B-title) fog(I-title) rated(O) ?(O)

Figure 6: An example of the self-correction mechanism when using beam search.

increases the model's inference efficiency, and (2) how optimization in the matching condition improves the model's performance. For the former, we evaluate the algorithm's acceleration across various sentence length ranges, as shown in Table 5. As sentence length increases, the acceleration effect of our algorithm becomes more pronounced. For sentence lengths between 100 and 200, it achieves an average acceleration factor of 17.3. For the latter, we remove the Back Tokenization procedure from LCS and eliminate all LCS processes. The results, presented in Table 6, indicate that removing Back Tokenization and the whole LCS algorithm leads to a decrease in effectiveness, underscoring the efficacy of our LCS Matching algorithm.

6 Analysis

Scaling Law of Generative NER Models Our experiments show that even smaller models like
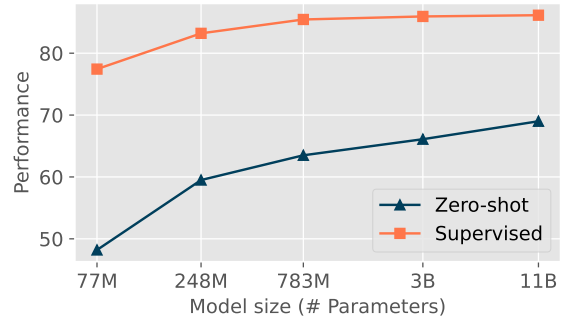

Figure 7: Scaling behavior of zero-shot and supervised performance with respect to model size (# parameters).

Flan-T5-large possess significant potential. We investigate the scaling law of Generative NER tasks in both zero-shot and supervised settings. The results are illustrated in Fig. 7. In the zero-shot setting, our methods scale well with model size. As the model size increases, the zero-shot capability of the model continues to rise, showing ample potential for further improvement with even larger models. In the supervised setting, our 783M model already demonstrates strong multi-task generalization abilities, and as the model size increases further, the improvements tend to converge.

Self-Correction Mechanism via Beam Search Beam search can enhance the performance of generative models by expanding the search space to include multiple hypotheses at each generation step. Previous research (Yan et al., 2021) has demonstrated that applying beam search in an entity-centric generation does not improve the model's performance or even degrade it. We conduct experiments on UniNER and our model, the results of which are shown in Table 7. We discover that as the beam size increases, the performance of the UniNER model decreases. In contrast, we observe a consistent improvement with beam search under our task schema. Upon a detailed case study of the model's generated results, we found that our task schema possesses a self-correction mechanism. The model retains some other hypotheses while generating subsequent results. In the decoding process that follows, the model can correct earlier mistakes. As demonstrated in Fig. 6, the model revises its previous incorrect prediction of "O" for "the fog" upon encountering the subsequent token "rated". This token is crucial for identifying the entity type associated with "the fog". A detailed analysis is provided in Appendix E.

7 Conclusion

This paper explores the potential of a strong Generative Named Entity Recognition (NER) system based on pre-trained LLMs by integrating the negative instances into training. Through experiments, we have demonstrated significant advancements. Our approach, which combines the inclusion of contextual information and a clear definition of entity boundaries through negative instances, has proven to be highly effective in improving the model's performance, especially in zero-shot scenarios where prediction uncertainty is high. The introduction of an LCS Matching algorithm further addresses the challenges of converting unstructured text into structured entities, ensuring accurate categorization and alignment. These findings highlight the crucial role of negative instances in NER tasks and the potential of generative models to revolutionize the field.

8 Limitation

Despite our system achieving impressive results, there remain limitations and space for improvement. In task settings, our approach focuses on the main-stream Flat-NER settings, where entities appear as continuous text segments, without addressing the discontinuous forms, i.e., discontinuous NER. Actually, it has always been challenging for generative models to adopt a unified paradigm to resolve all the complex settings. Previous entity-centric methods can address the discontinuous settings but fail to manage polysemy, where a phrase corresponds to different entity types in different sentence parts. The primary focus of this paper is to explore the impact of negative instances in the training process, and we will explore a unified framework for generative models in future work.

Ethics Statement

In this paper, we utilize the pre-trained large language models, i.e., Flan-T5 and LLaMA, as the foundational models. It's important to acknowledge that these models may contain inherent biases resulting from their pre-training processes. However, this issue is mitigated through our fine-tuning process, which refines the models to specifically concentrate on the Named Entity Recognition (NER) task, thereby reducing potential biases. Moreover, we strictly use all datasets and corpora in our study for scientific research purposes only.

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*.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th international conference on computational linguistics*, pages 1638–1649.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Pei Chen, Haotian Xu, Cheng Zhang, and Ruihong Huang. 2022. Crossroads, buildings and neighborhoods: A dataset for fine-grained location recognition. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3329–3339, Seattle, United States. Association for Computational Linguistics.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Jinlan Fu, Xuan-Jing Huang, and Pengfei Liu. 2021. Spanner: Named entity re-/recognition as span prediction. 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 7183–7195.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Runwei Guan, Ka Lok Man, Feifan Chen, Shanliang Yao, Rongsheng Hu, Xiaohui Zhu, Jeremy Smith, Eng Gee Lim, and Yutao Yue. 2023. Findvehicle and vehiclefinder: A ner dataset for natural language-based vehicle retrieval and a keyword-based cross-modal vehicle retrieval system. *arXiv preprint arXiv:2304.10893*.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

James W Hunt and Thomas G Szymanski. 1977. A fast algorithm for computing longest common subsequences. *Communications of the ACM*, 20(5):350–353.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17.

Aman Kumar and Binil Starly. 2022. "fabner": information extraction from manufacturing process science domain literature using named entity recognition. *Journal of Intelligent Manufacturing*, 33(8):2393–2407.

Bo Li, Gexiang Fang, Yang Yang, Quansen Wang, Wei Ye, Wen Zhao, and Shikun Zhang. 2023. Evaluating chatgpt's information extraction capabilities: An assessment of performance, explainability, calibration, and faithfulness. *arXiv preprint arXiv:2304.11633*.

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022a.

Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 10965–10973.

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

Yangming Li, Lemao Liu, and Shuming Shi. 2022b. Rethinking negative sampling for handling missing entity annotations. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7188–7197.

Yangming Li, Shuming Shi, et al. 2020b. Empirical analysis of unlabeled entity problem in named entity recognition. In *International Conference on Learning Representations*.

Jingjing Liu, Panupong Pasupat, Scott Cyphers, and Jim Glass. 2013. Asgard: A portable architecture for multilingual dialogue systems. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 8386–8390. IEEE.

Zihan Liu, Yan Xu, Tiezheng Yu, Wenliang Dai, Ziwei Ji, Samuel Cahyawijaya, Andrea Madotto, and Pascale Fung. 2021. Crossner: Evaluating cross-domain named entity recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13452–13460.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning. *arXiv preprint arXiv:2301.13688*.

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

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-fine pre-training for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. 2017. Cross-lingual name tagging and linking for 282 languages. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1946–1958, Vancouver, Canada. Association for Computational Linguistics.

Sampo Pyysalo and Sophia Ananiadou. 2014. Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Libo Qin, Wanxiang Che, Yangming Li, Haoyang Wen, and Ting Liu. 2019. A stack-propagation framework with token-level intent detection for spoken language understanding. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2078–2087.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

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*.

Erik Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Larry Smith, Lorraine K Tanabe, Cheng-Ju Kuo, I Chung, Chun-Nan Hsu, Yu-Shi Lin, Roman Klinger, Christoph M Friedrich, Kuzman Ganchev, Manabu Torii, et al. 2008. Overview of biocreative ii gene mention recognition. *Genome biology*, 9(2):1–19.

Simone Tedeschi, Valentino Maiorca, Niccolò Campolungo, Francesco Cecconi, and Roberto Navigli. 2021. WikiNEuRal: Combined neural and knowledge-based silver data creation for multilingual NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2521–2533, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. MultiN-ERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812, Seattle, United States. Association for Computational Linguistics.

Asahi Ushio, Leonardo Neves, Vitor Silva, Francesco. Barbieri, and Jose Camacho-Collados. 2022. Named Entity Recognition in Twitter: A Dataset and Analysis on Short-Term Temporal Shifts. In *The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing*, Online. Association for Computational Linguistics.

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

Zihan Wang, Jingbo Shang, Liyuan Liu, Lihao Lu, Jiacheng Liu, and Jiawei Han. 2019. Crossweigh: Training named entity tagger from imperfect annotations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5157–5166.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23:170.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, et al. 2022. Sustainable ai: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems*, 4:795–813.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. 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 5808–5822.

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.

Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. 2023. Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526*.

Wenxuan Zhou, Sheng Zhang, Yu Gu, Muhao Chen, and Hoifung Poon. 2023. Universalner: Targeted distillation from large language models for open named entity recognition. *arXiv preprint arXiv:2308.03279*.

A Data Statistics and Pre-processing

We show the full dataset statistics in Table 8, including the domain of datasets, the number of instances in train/valid/test data, and their download address. In particular, we have to pre-process the Pile-NER (Zhou et al., 2023) dataset to fit in our task schema. We observe nuances between our compiled datasets and those referenced by Wang et al. (2023) and Zhou et al. (2023). Specifically, their MultiNERD and PolyglotNER datasets omit the last 10,000 training samples. Furthermore, they miss the last sample in some datasets for the validation and test sets, such as CrossNER politics, MIT Movie, and MIT Restaurant. We have included these omitted instances in our dataset, adhering to the original dataset compositions. The modifications have a negligible impact on our results. This is because our sampling approach aligns with those used in the referenced studies, ensuring that the number of data instances sampled from each training set, up to 10,000 samples, is consistent. Moreover, adding a single extra sample in the test sets hardly affects the final results.

B Hyper-parameters settings

In our experiments, we train all models using a batch size of 256, employing the AdamW optimizer (Loshchilov and Hutter, 2018) for optimization. For the T5 model, we set a constant learning rate of 5×10^{-5} and impose a length limitation of 640 tokens for both the encoder and decoder. For the LLaMA model, we adopt a cosine learning rate schedule, initiating with a warm-up phase that covers 4% of the training steps, ramping up to a learning rate of 2×10^{-5}, followed by a decay phase for the remainder of the training steps. The length limitation is set to 1280. Due to our prediction sequences being longer, more training steps are required. The number of training epochs for our models varies by size: 20 epochs for both the small and base models, 10 epochs for the large and xl models, and 6 epochs for the xxl model. For the LLaMA model, we set the number of epochs to 3. We observe an interesting phenomenon that the T5 model often requires more training steps to converge. A possible explanation is that the backbone model, Flan-T5, an instruction-tuned model without any Named Entity Recognition (NER) related data in the instruction-tuning process, requires more training steps to adapt to the NER task.

C Problems in long sequence

In response to the issues in long predictions mentioned in section 4.3, we conduct a detailed case study. The representative examples are presented in Table 9. The problems can primarily be categorized into word omission, addition, and substitution, with omission and substitution accounting for the majority. We can conclude the following causes:

Noise in the original text Some of the issues can be attributed to noise in the original text. For example, in case 4, the model corrects "manattan" to "manhattan", and in case 7, it corrects the misuse of "the". However, we also observe that the model can introduce errors, as seen in cases 6 and 8, where the entities with repeated words, "norz norz norz" and "wet wet wet", confuse the model.

Missing words in the vocabulary Furthermore, we find that a certain proportion of issues can be derived from missing words in the model's vocabulary. As a result, these words naturally do not appear in the model's output. For instance, in case 10, "brontë" was replaced with "bront" because "brontë" does not exist in the vocabulary. We also discovered that several special characters do not exist in the T5 vocabulary, leading to more occurrences of omission and substitution.

Accumulative exposure bias The issue of repetitive generation of words and phrases is common in long text generation (LTG) due to the accumulative exposure bias as the prediction length increases. As illustrated by cases 3 and 9, the model produces meaningless and repetitive information.

D Optimization in LCS Matching

Optimization in Matching Algorithm We optimize the complexity of the LCS algorithm using a hierarchical divide-and-conquer approach through the following steps: (1) If the sequence does not have the above problem, i.e., $\tilde{X} = X$, it is obvious that $\hat{Y} = \tilde{Y}$. The time complexity is $O(N)$, (2) For the omission case, where $\tilde{X}$ is a subsequence of X, the matching process can be accomplished in $O(N)$ through greedy matching. (3) In other cases, we have implemented a fast version of the LCS algorithm (Hunt and Szymanski, 1977) within $O(N \log N)$, based on the nature of the small number of duplicate words in $\tilde{X}$. Our experimental results in Table 6 demonstrate that the optimization can significantly enhance efficiency, achieving up

Dataset	Domain	Types	#Train	#Valid	#Test	Download Link
Pile-NER (Zhou et al., 2023)		13,020	45,889	0	0	link
CoNLL2003 (Sang and De Meulder, 2003)		4	14,041	3,250	3,453	link
conllpp (Wang et al., 2019)		4	14,041	3,250	3,453	link
CrossNER AI (Liu et al., 2021)		13	100	350	431	link
CrossNER literature (Liu et al., 2021)		11	100	400	416	link
CrossNER music (Liu et al., 2021)		12	100	380	465	link
CrossNER politics (Liu et al., 2021)	General	8	200	541	651	link
CrossNER science (Liu et al., 2021)		16	200	450	543	link
MultiNERD (Tedeschi and Navigli, 2022)		16	144,144	10,000	10,000	link
Ontonotes (Weischedel et al., 2013)		18	59,924	8,528	8,262	link
PolyglotNER (Al-Rfou et al., 2015)		3	403,982	10,000	10,000	link
WikiANN en (Pan et al., 2017)		3	20,000	10,000	10,000	link
WikiNeural (Tedeschi et al., 2021)		3	92,720	11,590	11,597	link
AnatEM (Pyysalo and Ananiadou, 2014)		1	5,861	2,118	3,830	link
bc2gm (Smith et al., 2008)		1	12,500	2,500	5,000	link
bc4chemd (Krallinger et al., 2015)	Biomed	1	30,682	30,639	26,364	link
bc5cdr (Li et al., 2016)		2	4,560	4,581	4,797	link
ncbi (Doğan et al., 2014)		1	5,432	923	940	link
HarveyNER (Chen et al., 2022)		4	3,967	1,301	1,303	link
Broad Tweet Corpus (Derczynski et al., 2016)		3	6,338	1,001	2,001	link
TweetNER7 (Ushio et al., 2022)	Social media	7	7,111	886	576	link
mit-movie (Liu et al., 2013)		12	9,775	2,443	2,443	link
mit-restaurant (Liu et al., 2013)		8	7,660	1,521	1,521	link
FabNER (Kumar and Starly, 2022)	STEM	12	9,435	2,183	2,064	link
FindVehicle (Guan et al., 2023)	Transportation	21	21,565	20,777	20,777	link

Table 8: Statistics of datasets in our collected datasets.

to a 17.3 times speedup for long sequences compared to the naive $O(N^2)$ implementation.

Back Tokenization One notable problem in the matching process is the missing words in the vocabulary, as detailed in our case study in Appendix C. For example, "antropología" in the original text becomes 'antropologa' in the model's predictions, resulting in an inaccurate match in the matching process. To address this, we employ back tokenization, which involves tokenizing each word in the original text and then detokenizing it to match a word in the model's vocabulary, thereby creating a more resilient matching condition.

E Self-correction Mechanism

In this section, we conduct a case study to explore (1) the reasons behind the reduced effectiveness of entity-centric methods like UniNER (Zhou et al., 2023) when beam search is applied, and (2) the specific enhancements of the self-correction mechanism in our task schema. Upon comparing UniNER's performance with and without beam search, we observe that beam search leads to the model responding with the same answers across a variety of entity-type queries. For our models, we provide representative examples in Table 10 to

illustrate the self-correction mechanism's impact, showcasing (1) enhanced precision in determining entity boundaries (cases 1 and 2), (2) the use of contextual clues to recognize inherent entities (cases 3, 6 and 7), and correct mistakes (cases 4 and 5).

F Detailed Evaluation Results

We detail the performance of our models across all datasets in Table 11, including error bars for zero-shot performance derived from the variance of five separate runs. For the supervised settings, we do not conduct multiple runs due to the extensive size of the datasets, where the training and inference process can be very time-consuming. Our trials with smaller models indicate that the variability, or error bars, for models in the supervised settings is minimal, approximately around 0.15.

G Environmental Impact

Training huge models can have a negative impact on the environment. All our models are trained on the hardware of a single A100 node (8× Nvidia-A100-80G-SXM4) with approximately 800 GPU hours in total. The carbon footprint estimation is 135.3 kg CO_2eq according to Wu et al. (2022).

Model	Issue	Case	Type	Prediction
GNER LLaMA	Omission (39%)	1	raw	who directed **the film** the lorax
			pred.	who directed the lorax
		2	raw	any reasonably priced indian restaurants in **the** theater district
			pred.	any reasonably priced indian restaurants in theater district
	Addition (3%)	3	raw	the conservative regionalist navarra suma finished first and …
			pred.	the conservative regionalist **regionalist** navarra suma finished first and …
	Substitution (58%)	4	raw	which five star italian restaurants in manattan have the best reviews
			pred.	which five star italian restaurants in **manhattan** have the best reviews
		5	raw	polyethylene terephthalate (pet) bottles are made from ethylene and p-xylene .
			pred.	polyethylene terephthalate **(p e t)** bottles are made from ethylene and p-xylene .
GNER T5	Omission (23%)	6	raw	… whose debut album tol cormpt norz norz **norz** rock hard journalist wolf-rüdiger mühlmann considers a part of war metal 's roots .
			pred.	… whose debut album tol cormpt norz norz rock hard journalist wolf-rüdiger mühlmann considers a part of war metal 's roots .
		7	raw	jennifer lien starred in this action film of the **the** last six years that received a really good rating
			pred.	jennifer lien starred in this action film of the last six years that received a really good rating
	Addition (2%)	8	raw	… performed by wet wet wet that remained at number 1 …
			pred.	… performed by wet wet wet **wet** that remained at number 1 …
		9	raw	… liked by many people that starred william forsythe
			pred.	… liked by many people that starred william forsythe **the**
	Substitution (75%)	10	raw	four more children followed : charlotte brontë , (1816-1855) , branwell brontë (1817-1848) , emily brontë , (1818-1848) and anne (1820-1849) .
			pred.	four more children followed : charlotte **bront** , (1816-1855) , branwell **bront** (1817-1848) , emily **bront** , (1818-1848) and anne (1820-1849) .

Table 9: Representative examples concerning the word addition, omission, and substitution problems in the zero-shot evaluation. We remove the label information in the predictions for a clear comparison with the raw texts.

Model	Case	Type	Text Generations
GNER LLaMA	1	w/o beam search	who(O) is(O) directing(O) **the(O)** hobbit(B-title)
		w/ beam search	who(O) is(O) directing(O) **the(B-title) hobbit(I-title)**
	2	w/o beam search	what(O) is(O) the(O) plot(O) of(O) **the(O)** wild(B-title) bunch(I-title)
		w/ beam search	what(O) is(O) the(O) plot(O) of(O) **the(B-title) wild(I-title) bunch(I-title)**
	3	w/o beam search	was(O) there(O) a(O) **romantic(O) film(O) noir(O)**
		w/ beam search	was(O) there(O) a(O) **romantic(B-genre) film(I-genre) noir(I-genre)**
	4	w/o beam search	does(O) paymon(B-Restaurant Name) serves(O) **white(B-Cuisine) wine(I-Cuisine)**
		w/ beam search	does(O) paymon(B-Restaurant Name) serves(O) **white(B-Dish) wine(I-Dish)**
GNER T5	5	w/o beam search	… some(O) **batman(B-character)** movies(O) from(O) the(O) 1990s(B-year)
		w/ beam search	… some(O) **batman(B-title) movies(I-title)** from(O) the(O) 1990s(B-year)
	6	w/o beam search	where(O) was(O) **the(O)** presidio(B-title) filmed(O)
		w/ beam search	where(O) was(O) **the(B-title) presidio(I-title)** filmed(O)
	7	w/o beam search	… the(O) **third(O) harry(O) potter(O) movie(O)** called(O)
		w/ beam search	… the(O) **third(B-title) harry(I-title) potter(I-title) movie(I-title)** called(O)

Table 10: Representative examples in the self-correction mechanism via beam search.

Method Backbone # Params.	GNER-T5 Flan-T5-small 77M	GNER-T5 Flan-T5-base 248M	GNER-T5 Flan-T5-large 783M	GNER-T5 Flan-T5-xl 3B	GNER-T5 Flan-T5-xxl 11B	GNER-LLaMA LLaMA-7B 7B
Zero-shot Performance						
AI	50.18±0.9	56.83±0.4	62.56±0.2	62.09±0.3	68.19±0.3	63.11±0.2
Literature	49.78±1.5	58.68±0.8	58.20±0.4	64.94±1.1	68.66±0.2	68.20±0.3
Music	65.83±1.3	72.29±0.3	76.73±0.7	80.59±0.6	81.24±0.4	75.72±0.8
Politics	57.28±1.1	64.50±1.1	66.99±0.8	73.73±0.6	75.11±0.9	69.38±1.2
Science	62.68±1.9	68.00±1.2	72.60±0.2	68.74±1.2	76.70±1.0	69.93±0.4
Movie	37.38±1.8	54.52±0.2	58.59±0.1	62.96±0.4	62.52±0.5	68.63±0.5
Restaurant	14.30±1.4	41.41±1.2	48.61±0.5	49.82±0.2	51.04±0.4	47.49±1.1
Avg.	48.20±1.1	59.46±0.8	63.47±0.2	66.12±0.2	69.06±0.3	66.07±0.3
Supervised Performance						
AnatEM	81.02	86.99	90.22	90.29	90.30	90.24
bc2gm	69.02	79.11	83.10	84.25	84.29	83.18
bc4chemd	76.33	85.19	88.51	90.22	90.04	89.40
bc5cdr	82.02	87.16	88.81	89.83	89.95	90.27
Broad Twitter	80.09	81.59	82.61	84.34	84.56	83.74
CoNLL2003	89.12	91.82	93.14	93.14	93.28	93.60
FabNER	68.20	77.34	81.89	81.54	83.20	85.39
FindVehicle	90.64	93.61	95.71	95.97	97.37	98.62
HarveyNER	60.27	70.77	75.24	74.00	76.33	74.73
Movie	85.03	88.08	89.39	89.31	89.28	90.23
Restaurant	78.98	82.21	83.72	83.06	83.84	81.73
MultiNERD	90.94	93.17	94.24	94.51	94.35	94.30
ncbi	82.06	87.14	88.46	89.58	88.27	88.55
Ontonotes	86.36	89.33	90.54	91.63	91.83	90.69
PolyglotNER	45.27	62.13	66.16	67.15	66.90	67.52
TweetNER7	62.92	67.36	67.50	68.07	67.97	66.87
WikiANN	76.58	82.56	85.32	86.09	85.19	86.87
wikiNeural	88.97	92.24	93.56	93.85	93.71	93.71
Avg.	77.43	83.21	85.45	85.94	86.15	86.09

Table 11: Zero-shot and supervised evaluation results.