GPT-NER: Named Entity Recognition via Large Language Models

Shuhe Wang[▲], **Xiaofei Sun**[♦], **Xiaoya Li**[♣], **Rongbin Ouyang**[♠]
Fei Wu[♦], **Tianwei Zhang**[♥], **Jiwei Li**[♦], **Guoyin Wang**[★]

Abstract

Despite the fact that large-scale Language Models (LLM) have achieved SOTA performances on a variety of NLP tasks, its performance on NER is still significantly below supervised baselines. This is due to the gap between the two tasks the NER and LLMs: the former is a sequence labeling task in nature while the latter is a text-generation model.

In this paper, we propose GPT-NER to resolve this issue. GPT-NER bridges the gap by transforming the sequence labeling task to a generation task that can be easily adapted by LLMs e.g., the task of finding location entities in the input text *Columbus is a city* is transformed to generate the text sequence @@*Columbus## is a city*, where special tokens @@## marks the entity to extract. To efficiently address the *hallucination* issue of LLMs, where LLMs have a strong inclination to over-confidently label NULL inputs as entities, we propose a self-verification strategy by prompting LLMs to ask itself whether the extracted entities belong to a labeled entity tag.

We conduct experiments on five widely adopted NER datasets, and GPT-NER achieves comparable performances to fully supervised baselines, which is the first time as far as we are concerned. More importantly, we find that GPT-NER exhibits a greater ability in the low-resource and few-shot setups, when the amount of training data is extremely scarce, GPT-NER performs significantly better than supervised models. This demonstrates the capabilities of GPT-NER in real-world NER applications where the number of labeled examples is limited.

[▲] The University of Melbourne, [♦] Zhejiang University, [♣] University of Washington, [♠] Peking University, [♥] Nanyang Technological University, [★] 01.AI
Email: shuhewang@student.unimelb.edu.au
Codes are available at https://github.com/ShuheWang1998/GPT-NER.

1 Introduction

Large-scale language models (LLMs) (Brown et al., 2020; Smith et al., 2022; Du et al., 2022; Rae et al., 2021; Thoppilan et al., 2022; Hoffmann et al., 2022; Chowdhery et al., 2022; Touvron et al., 2023) have shown an impressive ability for in-context learning: with only a few task-specific examples as demonstrations, LLMs are able to generate results for a new test input. Under the framework of in-context learning, LLMs have achieved promising results in a variety of NLP tasks, include machine translation (MT) (Vilar et al., 2022; Vidal et al., 2022; Moslem et al., 2023), question answering (QA) (Robinson et al., 2022; Li et al., 2022; Lazaridou et al., 2022) and named entity extraction (NEE) (Chowdhery et al., 2022; Brown et al., 2020).

Despite the progress, LLMs' performances on the task of NER are still well below supervised baselines. This is because of the intrinsic gap between the two tasks of NER and LLMs: NER is a sequence labeling task in nature, where the model needs to assign an entity-type label to each token within a sentence, while LLMs are formalized under a text generation task. The gap between the semantic labeling task and the text generation model leads to inferior performance when applying LLMs to resolve the NER task.

In this paper, we propose GPT-NER to resolve this issue. GPT-NER transforms the NER task to a text-generation task that can be easily adapted by LLMs. Specifically, the task of finding location entities in the input text *Columbus is a city* is transformed to generate the text sequence @@*Columbus## is a city*, where special tokens @@## marks the entity. We find that, compared with other formalizations, the proposed strategy, can significantly decrease the difficulty in generating text that fully encodes label information of the input sequence, as the model only needs to mark the position for entities and make copies for all the rest tokens. Experi-

Findings of the Association for Computational Linguistics: NAACL 2025, pages 4257–4275
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

ments show that the proposed strategy significantly improves the performance.

Another big problem with LLMs for NER is the *hallucination* issue, where LLMs have a strong inclination to over-confidently label NULL inputs as entities. To address this issue, we propose a self-verification strategy, which is placed right after the entity extraction stage, prompting LLMs to ask itself whether an extracted entity belongs to a labeled entity tag. The self-verification strategy acts as a regulating function to counteract the excessive confidence of LLMs, which we find effective in addressing the *hallucination* issue, leading to a significant performance boost.

We conduct experiments on five widely-adopted NER datasets, both flat NER and nested NER. GPT-NER achieves comparable performances to fully supervised baselines, which is the first time as far as we are concerned. Additionally, we find that the performance hasn't plateaued when we reach the GPT-3 token limit with respect to the number of demonstrations. This means that there is still room for improvement when the 4,096 token limits of GPT-3 are released, e.g., using GPT-4 whose token limit is more than 20K. What is particularly noteworthy is that GPT-NER exhibits impressive proficiency in low-resource and few-shot NER setups: when the amount of training data is extremely scarce, GPT-NER performs significantly better than supervised models. This illustrates the potential of GPT-NER to be employed in real-world NER applications even when the quantity of labeled samples is scant.

2 Related Work

Named Entity Recognition. Named Entity Recognition (NER) is a task to identify key information in the text and classify it into a set of predefined categories. A common approach to resolve NER is to formulate it as a sequence labeling task. Hammerton (2003) used unidirectional LSTMs to obtain token-level representations and feed them to the softmax classifier obtaining the results. Collobert et al. (2011) used CNN to embed each input word and leverage CRF to decode each embedding into a certain entity. Chiu and Nichols (2016) used a character CNN and Devlin et al. (2018) used BERT to obtain token-level representations for classifications. Lample et al. (2016) combined the bidirectional LSTMs with CRFs to augment the prediction. Sarzynska-Wawer et al. (2021) improved the quality of each word via a large-scale pre-training model. Li et al. (2019a,b) formulated the NER task as an MRC task and further leveraged dice loss to improve the performance of the MRC model, and Wang et al. (2022) proposed the GNN-SL model to allow a general NER model to refer to training examples at test time.

Large Language Models and In-context Learning. Large language models (LLMs) (Brown et al., 2020; Rae et al., 2021; Smith et al., 2022; Hoffmann et al., 2022; Chowdhery et al., 2022) have obtained significant performance boosts on a variety of natural language processing tasks (Hegselmann et al., 2022; Vilar et al., 2022; Perez et al., 2021; Pietrzak et al., 2021; Wei et al., 2021). Strategies to use LLMs for downstream tasks can be divided into two categories: fine-tuning and in-context learning. The fine-tuning strategy takes a pre-trained model as initialization and runs additional epochs on the downstream supervised data (Raffel et al., 2020; Gururangan et al., 2018; Roberts et al., 2020; Guu et al., 2020), while in-context learning (ICL) prompts LLMs to generate texts under few-shot demonstrations (Radford et al., 2019; Brown et al., 2020; Chowdhery et al., 2022; Perez et al., 2021; Lu et al., 2021).

3 GPT-NER

In this work, we propose GPT-NER, which uses large language models (LLMs) to resolve the NER task. GPT-NER follows the general paradigm of in-context learning and can be decomposed into three steps: (1) Prompt Construction: for a given input sentence X, we construct a prompt (denoted by Prompt(X)) for X; (2) feeding the constructed prompt to LLMs to obtain the generated text sequence $W = \{w_1, ..., w_n\}$; (3) transforming the text sequence W to a sequence of entity labels to obtain the final results.

3.1 Prompt Construction

Figure 1 is an example of the prompt used in GPT-NER, which consists of three parts:

3.1.1 Task Description

Task Description gives an overview of the task, which can be decomposed into three components:

(1) the first sentence of the task description,

"*I am an excellent linguist*"

is a constant telling LLMs to produce the output using linguistic knowledge;

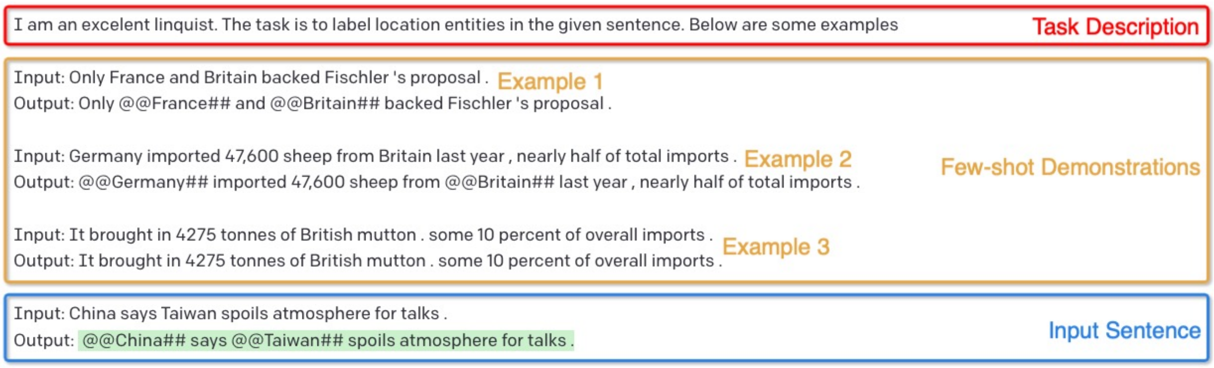

Figure 1: The example of the prompt of GPT-NER. Suppose that we need to recognize location entities for the given sentence: *China says Taiwan spoils atmosphere for talks*. The prompt consists of three parts: **(1) Task Description**: It's surrounded by a red rectangle, and instructs the GPT-3 model that the current task is to recognize **Location** entities using linguistic knowledge. **(2) Few-shot Demonstrations**: It's surrounded by a yellow rectangle giving the GPT-3 model few-shot examples for reference. **(3) Input Sentence**: It's surrounded by a blue rectangle indicating the input sentence, and the output of the GPT-3 model is colored green.

(2) The second sentence

"*The task is to label [Entity Type] entities in the given sentence*"

is a variable sentence indicating the category of entities to be extracted, *[Entity Type]* represents the type of entity to extract, e.g., **Location** in the example of Figure 1. It is worth noting that, in this way, for each input sentence, we need to iterate over all entity labels, which is equivalent to transforming an N-class classification task to N binary classification tasks. The reason behind this is as follows: for most current LLMs, e.g., GPT-3, there is a hard limit on the length of the prompt (e.g, 4096 tokens for GPT-3[1]) due to the hardware restrictions. Given this limited number of tokens, it is impossible to include descriptions and demonstrations for all entity types in a single prompt. Therefore, for each input sentence, we construct the prompt N times, each of which corresponds to each entity type;

(3) the third sentence

"*Below are some examples*"

marks the end of the description and points out the position of few-shot demonstrations.

3.1.2 Few-shot Demonstration

The few-shot demonstration is appended to the prompt. It serves as the following two purposes: (1) it regulates the format of the LLM outputs for each test input, as LLMs will (very likely) generate outputs that mimic the format of demonstrations. This is vital for the NER task as we need the output format to be consistent so that we can parse the output in the form of natural language to NER results; (2) it provides the LLM with direct evidence about the task and references to make predictions.

The Format of LLM Output. The format of each labeled sentence W, which is a text sequence, is of vital importance and should satisfy the following conditions: (1) it needs to contain the information for each word label, and can be easily transformed into the entity type sequence; (2) it needs to be smoothly and easily generated by LLMs to boost the models' final accuracy.

To resolve this issue, we propose LLMs output taking the format: if the input sequence does not contain any entity, W just copies the input X; for an entity/entities in the input sequence, we use special tokens "@@" and "##" to surround it/them. As the example 1 in Figure 1, the two words "*@@France##*" and "*@@Britain##*" are surrounded with special tokens, while others are just copied.

The proposed strategy significantly bridges the gap between the format of the sequence labeling task and the generation model: it significantly decreases the difficulty in the generated text that fully encodes label information, as the LLM only needs to mark the position for entities and make copies for all the rest. As we will show in the ablation study section 5.1, the proposed strategy yields significant performance boosts over other formats.

3.1.3 Input Sentence

This part feeds the current input sentence into the LLM and expects the LLM to generate the output sequence according to the defined format in Sec 3.1.2, where "*Output*" is blanked and denotes the

[1]Each English word corresponds to 1.3 tokens on average.

flag that the LLM begins to generate the labeled sequence. Here comes the end of the prompt construction, and next we will describe strategies to retrieve demonstration examples.

3.2 Few-shot Demonstrations Retrieval

3.2.1 Random Retrieval

The most straightforward strategy is randomly select k examples from the training set. The shortcoming is obvious: there is no guarantee that retrieved examples are semantically close to the input.

3.2.2 kNN-based Retrieval

To resolve the relatedness issue in Sec 3.2.1, we can retrieve k nearest neighbor (kNN) of the input sequence from the training set (Vilar et al., 2022; Liu et al., 2021): we first compute representations for all training examples, based on which we obtain the k nearest neighbors for an input test sequence.

kNN based on Sentence-level Representations. To find kNN examples in the training set, one straightforward method is to use text similarity models such as SimCSE (Gao et al., 2021): we first obtain sentence-level representations for training examples and the input sequence, and use cosine similarity to find kNN.

The shortcoming of kNN based on sentence-level representations is obvious: NER is a token-level task that focuses more on local evidence rather than a sentence-level task, which is concerned with sentence-level semantics. a retrieved sentence (e.g., *he is a soldier*) that is semantically similar to the input (e.g., *John is a soldier*) might shed no light on the NER the input contains: in the example above, the retrieved sentence contains no NER and thus provides no evidence for tagging the input.

Entity-level Embedding. To resolve the issue above, we need to retrieve kNN examples based on token-level representations rather than sentence-level representations. We first extract entity-level representations for all tokens of all training examples as the datastore using a fine-tuned NER tagging model. For a given input sequence with length N, we first iterate over all tokens within the sequence to find kNNs for each token, obtaining $K \times N$ retrieved tokens. Next, we select the top k tokens from the $K \times N$ retrieved tokens, and use their associated sentences as demonstrations. We select several examples to better illustrate demonstrations of three retrieval strategies in Appendix D.

3.3 Self-verification

LLMs significantly suffer from the *hallucination* or overprediction issue (Braverman et al., 2020; Jiang et al., 2021; Zhao et al., 2021). Specifically for NER, LLMs have a strong inclination to over-confidently label NULL inputs as entities, even with demonstrations. To address this issue, we propose the self-verification strategy. Given an extracted entity by LLMs, we ask the LLM to further verify whether the extracted entity is correct, answered by *yes* or *no*.

Take the extraction of location entities as an example. The prompt starts with the task description:

"*The task is to verify whether the word is a location entity extracted from the given sentence*".

Again, we need few-shot demonstrations to boost the accuracy of the self-verifier. Shown in the yellow rectangle in Figure 2, each demonstration consists of three lines:

(1) "*The input sentence: Only France and Britain backed Fischler's proposal*",
(2) "*Is the word "France" in the input sentence a location entity? Please answer with yes or no*".
(3) *Yes*.

We pack multiple demonstrations in the prompt in the few-shot setup. Demonstrations are followed by the test example, and fed to the LLM to obtain the output.

Demonstration Selection. Since the center of the self-verification task is asking about whether an extracted entity is a specific entity type, we need to select training examples that are semantic to the extracted entity rather than overall sentence-level semantics.

Therefore, we use the entity-level embedding described in Sec 3.2.2 for kNN demonstration search rather than sentence-level representations: (1) firstly, we construct the datastore by extracting entity-level representations for all training examples using a fine-tuned NER model; (2) then, we use the same fine-tuned NER model to extract representation for the queried word; (3) finally, we use the representation of the queried word to select k examples from the datastore as few-shot demonstrations, whose answer is "*Yes*" if the retrieved entity belongs to the queried entity type, and "*no*" otherwise.

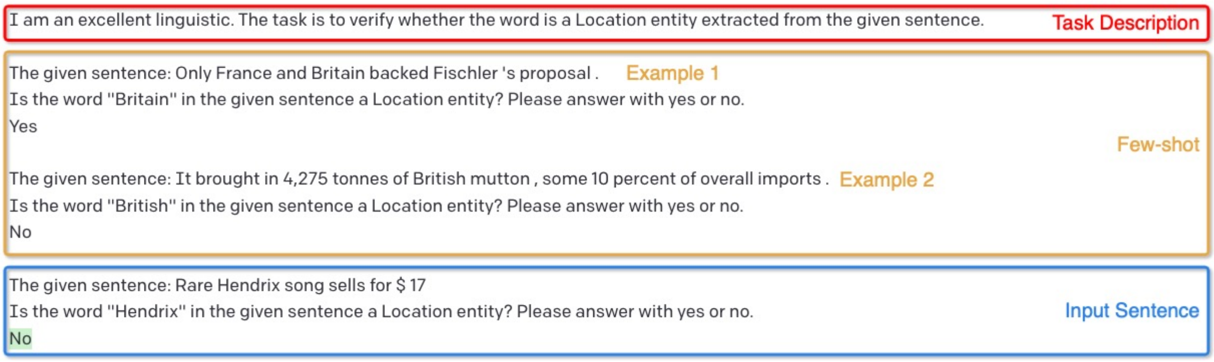

Figure 2: The example of the prompt of verification using the GPT-3. Supposed that we need to verify whether the word "*Hendrix*" in the given sentence "*Rare Hendrix song sells for $ 17*" is a **Location** entity. The prompt consists of three parts: **(1) Task Description** (Red Rectangle): It gives the definition of the current task: to discriminate whether the specified word in the given sentence belongs to **Location** entity. **(2) Few-shot** (Yellow Rectangle): It provides several examples for the GPT-3 to reference.**(3) Input Sentence** (Blue Rectangle): It indicates the current word that needs to be verified and the sentence it belongs to, and the output of the GPT-3 is colored green.

4 Experiments

We use GPT-3 (Brown et al., 2020) (davinci-003) as the LLM backbone for all experiments. For davinci-003 parameters, we set the maximum output length to 512 tokens. Temperature is set to 0, top_p to 1, frequency_penalty to 0, presence_penalty to 0, and best_of to 1.

4.1 Results on the Full Training Set

4.1.1 Results on Flat NER

For flat NER, entities can't overlap with each other. We conduct experiments on the two widely-used flat-NER datasets, English CoNLL2003 and OntoNotes5.0, which can be found in Appendix, using span-level precision, recall, and F1 score as evaluation metrics. We put more details about our used datasets and baselines in Appendix A.1.

Due to the fact that accessing davinci-003 can be expensive, in addition to the full test set, we randomly selected 100 test instances to make it easier for the community to replicate our results. We report performances on both the full and the partial test sets.

Main Results. Table 1 shows results on the full test set for flat NER, and due to the limitation of pages, we put results on the partial test set on the Appendix B. Observations are as follows:

(1) kNN retrieval is of vital importance for the NER task. For the random retrieval strategy where demonstrations are randomly selected rather than through kNN search, performances are only 72.62 and 61.58 on the full CoNLL2003 and OntoNotes5.0 sets. Results skyrocket to 84.36 and 75.32 on the full CoNLL2003 and OntoNotes5.0 when sentence-level embeddings are used for the

English CoNLL2003 (FULL)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
BERT-Tagger (Devlin et al., 2018)	-	-	92.8
BERT-MRC (Li et al., 2019a)	92.33	94.61	93.04
GNN-SL (Wang et al., 2022)	93.02	93.40	93.2
ACE+document-context (Wang et al., 2020)	-	-	**94.6 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	77.04	68.69	72.62
GPT-3 + *sentence-level embedding*	81.04	88.00	84.36
GPT-3 + *entity-level embedding*	88.54	91.4	89.97
Self-verification (zero-shot)			
+ GPT-3 + *random retrieval*	77.13	69.23	73.18
+ GPT-3 + *sentence-level embedding*	83.31	88.11	85.71
+ GPT-3 + *entity-level embedding*	89.47	91.77	90.62
Self-verification (few-shot)			
+ GPT-3 + *random retrieval*	77.50	69.38	73.44
+ GPT-3 + *sentence-level embedding*	83.73	88.07	85.9
+ GPT-3 + *entity-level embedding*	89.76	92.06	90.91
English OntoNotes5.0 (FULL)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
BERT-Tagger (Devlin et al., 2018)	90.01	88.35	89.16
BERT-MRC (Li et al., 2019a)	92.98	89.95	91.11
GNN-SL (Wang et al., 2022)	91.48	91.29	91.39
BERT-MRC+DSC (Li et al., 2019b)	91.59	92.56	**92.07 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	58.8	64.36	61.58
GPT-3 + *sentence-level embedding*	71.87	78.77	75.32
GPT-3 + *entity-level embedding*	79.17	84.29	81.73
Self-verification (zero-shot)			
+ GPT-3 + *random retrieval*	59.14	64.44	61.79
+ GPT-3 + *sentence-level embedding*	72.29	78.81	75.55
+ GPT-3 + *entity-level embedding*	79.64	84.52	82.08
Self-verification (few-shot)			
+ GPT-3 + *random retrieval*	59.23	64.65	61.94
+ GPT-3 + *sentence-level embedding*	72.35	78.79	75.57
+ GPT-3 + *entity-level embedding*	79.89	84.51	82.20

Table 1: Results of full data for two **Flat** NER datasets: CoNLL2003 and OntoNotes5.0.

kNN demonstration retrieval.

(2) We observe a significant improvement by changing the sentence-level embedding to token-level embedding for the kNN demonstration search: 84.36 v.s. 89.97 on CoNLL2003 dataset and 75.32 v.s. 81.73 on OntoNotes5.0. This phenomenon is because NER is a token-level task that focuses more on local evidence rather than a sentence-level task: the two sentences "he is a soldier" and "John is a soldier" are semantically similar but don't share any identical entities. Using token-level representation for the kNN search help retrieve more similar demonstrations with respect to the specific entity type, leading to better performances.

(3) We observe further improvements by adding self-verification: on the full CoNLL2003 dataset with entity-level embedding, 89.97 v.s. 90.62 respectively for without and with self-verification for zero-shot learning and 84.97 v.s. 85.91 for few-shot learning. The results prove the effectiveness of self-verification in alleviating overprediction of the GPT-3.

(4) LLM-based systems obtain comparable results to supervised baselines using BERT, i.e., 90.91 v.s. 92.8 on the full CoNLL2003 dataset and 82.20 v.s. 89.16 on the full OntoNotes5.0 dataset. We observe that there still remains a gap between the supervised SOTA model: 94.6 v.s. 90.91 on the full CoNLL2003 dataset and 92.07 v.s. 82.20 on the full OntoNotes5.0 dataset. As will be shown in the ablation study section, we find that the performance hasn't plateaued when we reach the GPT-3 token limit with respect to the number of KNN demonstrations. This means that the token limit is released, e.g., using GPT-4 whose token limit is more than 20K tokens, there is still room for improvement. We will update performances when GPT-4 API is accessible.

4.1.2 Results on Nested NER

For nested NER, entities in each sentence may overlap with each other, like: two geographical entities "*Chinese*" and "*France*" overlap with the facility entity "*The Chinese embassy in France*".

We conduct experiments on the three widely-used nested NER datasets: ACE2004, ACE2005 and GENIA, and use span-level precision, recall, and F1 score for evaluation. We put more details about our used datasets and baselines in Appendix A.2.

ACE2004 (FULL)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
BERT-MRC (Li et al., 2019a)	85.05	86.32	85.98
Triaffine+BERT (Yuan et al., 2021)	87.13	87.68	87.40
Triaffine+ALBERT (Yuan et al., 2021)	88.88	88.24	88.56
BINDER (Zhang et al., 2022)	88.3	89.1	**88.7 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	55.04	41.76	48.4
GPT-3 + *sentence-level embedding*	65.31	53.67	60.68
GPT-3 + *entity-level embedding*	72.23	75.01	73.62
Self-verification (zero-shot)			
GPT-3 + *random retrieval*	55.44	42.22	48.83
GPT-3 + *sentence-level embedding*	69.64	54.98	62.31
GPT-3 + *entity-level embedding*	73.58	74.74	74.16
Self-verification (few-shot)			
GPT-3 + *random retrieval*	55.63	42.49	49.06
GPT-3 + *sentence-level embedding*	70.17	54.87	62.52
GPT-3 + *entity-level embedding*	73.29	75.11	74.2
ACE2005 (FULL)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
Triaffine+BERT (Yuan et al., 2021)	86.70	86.94	86.82
BERT-MRC (Li et al., 2019a)	87.16	86.59	86.88
Triaffine+ALBERT (Yuan et al., 2021)	87.39	90.31	88.83
BINDER (Zhang et al., 2022)	89.1	89.8	**89.5 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	45.5	46.24	45.37
GPT-3 + *sentence-level embedding*	58.04	58.97	58.50
GPT-3 + *entity-level embedding*	71.72	74.2	72.96
Self-verification (zero-shot)			
GPT-3 + *random retrieval*	45.06	46.62	45.84
GPT-3 + *sentence-level embedding*	59.49	60.17	59.83
GPT-3 + *entity-level embedding*	72.63	75.39	73.46
Self-verification (few-shot)			
GPT-3 + *random retrieval*	45.49	46.73	46.11
GPT-3 + *sentence-level embedding*	59.69	60.35	60.02
GPT-3 + *entity-level embedding*	72.77	75.51	73.59
GENIA (FULL)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
Triaffine+BERT (Yuan et al., 2021)	80.42	82.06	81.23
BERT-MRC (Li et al., 2019a)	85.18	81.12	**83.75 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	44.1	38.64	41.37
GPT-3 + *sentence-level embedding*	63.43	44.17	51.68
GPT-3 + *entity-level embedding*	61.38	66.74	64.06
Self-verification (zero-shot)			
GPT-3 + *random retrieval*	44.31	38.79	41.55
GPT-3 + *sentence-level embedding*	59.54	44.26	51.9
GPT-3 + *entity-level embedding*	61.77	66.81	64.29
Self-verification (few-shot)			
GPT-3 + *random retrieval*	44.68	38.98	41.83
GPT-3 + *sentence-level embedding*	59.87	44.39	52.13
GPT-3 + *entity-level embedding*	61.89	66.95	64.42

Table 2: Results of full data for three **Nested** NER datasets: ACE2004, ACE2005 and GENIA.

Main Results. Results are shown in Table 2, and phenomenon is similar to flat NER is observed:

(1) Again, kNN retrieval is of vital importance: on the full ACE2004 dataset, 48.4 for random retrieval v.s. 73.62 for entity-level embedding re-

trieval using KNN search.

(2) For the kNN demonstration search, a significant improvement is observed by changing the sentence-level embedding to entity-level embedding: 60.68 v.s. 73.62 on ACE2004 and 56.68 v.s. 69.06 on GENIA.

(3) Further performance boost is obtained by adding self-verification, i.e., on the full ACE2004 dataset with sentence-level embedding, 60.68 v.s. 62.31 for zero-shot learning and 60.68 v.s. 62.52 for few-shot learning.

We also observe that the gap between GPT-NER and SOTA models is greater than flat NER. This is because:

(1) Nested NER datasets contain more similar entity types, e.g., the location entities (LOC) and the geographical entities (GPE). Since only a limited number of demonstrations is allowed, it is harder for GPT-3 to distinguish between them,

(2) The annotation guidelines for the three nested NER datasets are more complex and less straightforward. For example, the substring of "*The bodies of six people*" within the sentence "*The bodies of six people were found in the region*" is annotated as a person entity. It is easier for a supervised model fine-tuned on the full training set to learn these complex rules, while much harder for an LLM model with a limited number of demonstrations.

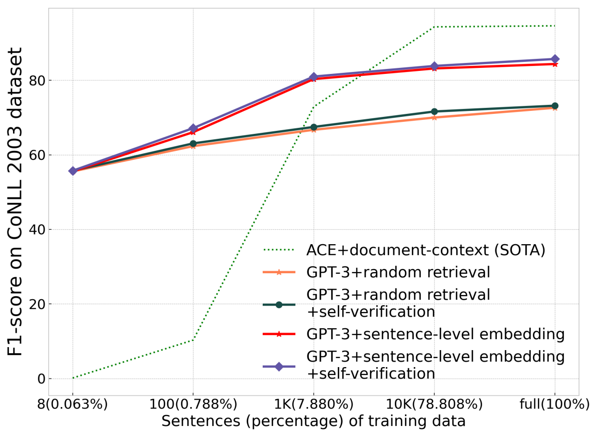

Figure 3: Low-resource comparisons on CoNLL2003 dataset.

4.2 Results on Low-resource Scenario

We conduct experiments to estimate the performance of GPT-NER in low resource setups on the English CoNLL2003 dataset. In order to mimic the low-resource scenario, we randomly select a subset of the full training data as the training set: (a) 8 training sentences (0.063%); (b) 100 training sentences (0.788%); (c) 1K sentences (7.880%); and (d) 10K sentences (78.808%). For the setup with 8 training sentences, the dataset is constructed to ensure that each entity type contains one positive and one negative example. Evaluations are performed on the full test set.

Setups. We use the same GPT parameters as in Sec 4. For baselines, we train the ACE model (Wang et al., 2020) (which is the current SOTA model) on different training subsets. For GPT-NER, we use random demonstration retrieval and sentence-level embedding-based demonstration retrieval for demonstration selection in the few-shot learning stage. For the self-verification stage, we only use zero-shot learning where no demonstration is needed.

Main Results. Results are shown in Figure 3. Observations are as follows:

(1) When the size of the training set is extremely small (i.e., 8 or 100 sentences), and the performance of the supervised model is far below GPT-3. Specifically, with only 8 training examples, the F1 score of GPT-NER is already about 60 while the performance of supervised models is around 0. This demonstrates the significantly better generalization ability of GPT-NER over supervised baselines in the low-resource setup.

(2) With the increase of the training data, the performance of KNN search grows faster than random retrieval, which is in accord with our expectations: for random retrieval, where all demonstrations are randomly selected, the impact of increasing the size of training data is minimal: the outcomes of selecting K demonstration from 100 and 1000 sets are similar since they are all randomly selected. But for kNN demonstration search, increasing the size of training data means selected demonstrations are more likely to be related to the input, leading to better performances.

(3) When the amount of data reaches 10%, as the size of training data increases, the performance of the supervised model will significantly improve, while the result of GPT-3 will increase marginally. This phenomenon indicates that for in-context learning, instead of focusing on increasing the amount of training data, it is more effective to focus on improving the quality of retrieved demonstrations (e.g., random retrieval to kNN based retrieval) and prompt structure (e.g., adding self-verication).

5 Ablation Study

5.1 Varying the Format of LLM Output

In Sec 3.1.2, we propose to use special tokens "@@" and "##" to regulate the format of the GPT-3 output. We compare the proposed output format with the following two formats:

BMES directly outputs the beginning, middle, end, and singleton indicator for input each token:

Input:White House is in Washington

Output:B-ORG E-ORG O O O

Entity+Position asks LLMs to output the entity within the sentence along with its position:

Input:White House is in Washington

Output:White House (0)

where "White House (0)" means that "White House" is an entity and its starting position is 0 at the input sentence.

To enable apple-to-apple comparisons, we use the same setup for the three formats and conduct experiments on the 100-sample CoNLL2003 with 32 few-shots. The F1-score for the proposed ##@@ strategy, *BMES* and *Entity+Position* are respective 92.68, 29.75 and 38.73, where *BMES* and *Entity+Position* significantly underperform the proposed ##@@ strategy. This is because:

For the **BMES** strategy, the LLM needs to learn the alignment between each input word and each **BMES** label: *White* to *B-ORG*, *House* to *E-ORG*, *is* to *O*, *in* to *O*, *Washington* to *O*. By analyzing the error samples, we find that it is usually even hard for the LLM to output a BMES string with the correct length, especially when the input sentence is long, leading to poor final evaluation performances.

For the **Entity+Position** strategy, we find that the LLM usually confuses the meaning of the position index (e.g., whether it is character index or word index), leading to incorrect entity position. This problem can be partially alleviated by demonstrations but still exists considering the 4096 token limit for GPT-3. Incorrect position indexes make it hard to map the output to the sequence labeling format, leading to poor final evaluation performances.

5.2 The Number of Few-shot Demonstrations

We conduct experiments to estimate the effect of the number of demonstrations. Experiments, shown in Figure 4, are conducted on the 100-sample CoNLL 2003 dataset. We can observe as k increases, all three LLM-based results keep rising. As we approach the 4096 token limit, the result still hasn't plateaued, which means performance will still rise if more demonstrations are allowed.

An interesting phenomenon is observed that when the number of demonstrations is small, i.e., $k = 2, 4$, kNN-based strategies underperform the random retrieval strategy. The explanation is as follows: kNN-based retrieval tends to select demonstrations that are very similar to the input sentence. Therefore, if the input sentence doesn't contain any entity, the retrieved demonstrations are most to contain no entity either. In this case, demonstrations don't contain the output format information we wished, leading LLMs to output arbitrary format.

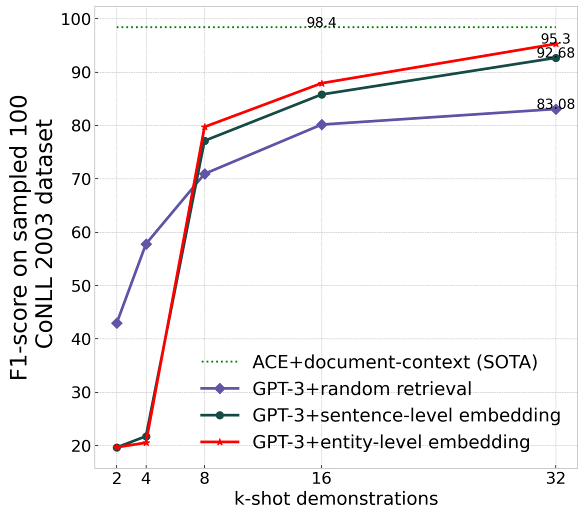

Figure 4: Comparisons by varying k-shot demonstrations.

6 Conclusion

In this paper, we propose GPT-NER to adapt LLMs to the NER task. To bridge the gap between the sequence labeling task and the text generation task, we instruct the LLM to generate a labeled sequence by surrounding entities with special tokens. Additionally, we propose a self-verification strategy to alleviate the hallucination issue of the LLM model. We conduct experiments on both flat and nested NER datasets, and achieve comparable performances to fully supervised baselines. Besides that, we find that GPT-NER has a remarkable ability in the low-resource scenario, that when the amount of training data is extremely scarce, the results of GPT-NER are significantly better than that of the supervised model.

7 Limitation

In this work, we have demonstrated that for in-context learning, instead of focusing on increasing the amount of training data, it is more effective to focus on improving the quality of retrieved demonstrations and prompt structure. Therefore, our next step will attempt more representations extracted with different strategies.

Additionally, we find that the performance of GPT-NER hasn't plateaued when we reach the GPT-3 token limit with respect to the number of KNN demonstrations. This means that the token limit is released, e.g., using GPT-4 whose token limit is more than 20K tokens, there is still room for improvements. We will update performances when GPT-4 API is unlimited accessible.

References

Mark Braverman, Xinyi Chen, Sham Kakade, Karthik Narasimhan, Cyril Zhang, and Yi Zhang. 2020. Calibration, entropy rates, and memory in language models. pages 1089–1099.

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.

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

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Walker Christopher, Strassel Stephanie, Medero Julie, and Maeda Kazuaki. 2006. Ace 2005 multilingual training corpus.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(ARTICLE):2493–2537.

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*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. 2(1).

Nan Du, Yanping Huang, Andrew M Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, et al. 2022. Glam: Efficient scaling of language models with mixture-of-experts. pages 5547–5569.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. *arXiv preprint arXiv:2104.08821*.

Suchin Gururangan, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R Bowman, and Noah A Smith. 2018. Annotation artifacts in natural language inference data. *arXiv preprint arXiv:1803.02324*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. pages 3929–3938.

James Hammerton. 2003. Named entity recognition with long short-term memory. pages 172–175.

Stefan Hegselmann, Alejandro Buendia, Hunter Lang, Monica Agrawal, Xiaoyi Jiang, and David Sontag. 2022. Tabllm: Few-shot classification of tabular data with large language models. *arXiv preprint arXiv:2210.10723*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. 9:962–977.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Angeliki Lazaridou, Elena Gribovskaya, Wojciech Stokowiec, and Nikolai Grigorev. 2022. Internet-augmented language models through few-shot prompting for open-domain question answering. *arXiv preprint arXiv:2203.05115*.

Junlong Li, Zhuosheng Zhang, and Hai Zhao. 2022. Self-prompting large language models for open-domain qa. *arXiv preprint arXiv:2212.08635*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019a. A unified mrc framework for named entity recognition. *arXiv preprint arXiv:1910.11476*.

Xiaoya Li, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2019b. Dice loss for data-imbalanced nlp tasks. *arXiv preprint arXiv:1911.02855*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Yasmin Moslem, Rejwanul Haque, and Andy Way. 2023. Adaptive machine translation with large language models. *arXiv preprint arXiv:2301.13294*.

Ethan Perez, Douwe Kiela, and Kyunghyun Cho. 2021. True few-shot learning with language models. *Advances in neural information processing systems*, 34:11054–11070.

Ben Pietrzak, Ben Swanson, Kory Mathewson, Monica Dinculescu, and Sherol Chen. 2021. Story centaur: Large language model few shot learning as a creative writing tool.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

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

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Joshua Robinson, Christopher Michael Rytting, and David Wingate. 2022. Leveraging large language models for multiple choice question answering. *arXiv preprint arXiv:2210.12353*.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.

Justyna Sarzynska-Wawer, Aleksander Wawer, Aleksandra Pawlak, Julia Szymanowska, Izabela Stefaniak, Michal Jarkiewicz, and Lukasz Okruszek. 2021. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Research*, 304:114135.

Shaden Smith, Mostofa Patwary, Brandon Norick, Patrick LeGresley, Samyam Rajbhandari, Jared Casper, Zhun Liu, Shrimai Prabhumoye, George Zerveas, Vijay Korthikanti, et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. 2022. Lamda: Language models for dialog applications. *arXiv preprint arXiv:2201.08239*.

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

Blanca Vidal, Albert Llorens, and Juan Alonso. 2022. Automatic post-editing of mt output using large language models. pages 84–106.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2022. Prompting palm for translation: Assessing strategies and performance. *arXiv preprint arXiv:2211.09102*.

Shuhe Wang, Yuxian Meng, Rongbin Ouyang, Jiwei Li, Tianwei Zhang, Lingjuan Lyu, and Guoyin Wang. 2022. Gnn-sl: Sequence labeling based on nearest examples via gnn. *arXiv preprint arXiv:2212.02017*.

Xinyu Wang, Yong Jiang, Nguyen Bach, Tao Wang, Zhongqiang Huang, Fei Huang, and Kewei Tu. 2020. Automated concatenation of embeddings for structured prediction. *arXiv preprint arXiv:2010.05006*.

Jason Wei, Maarten Bosma, Vincent Y 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. *arXiv preprint arXiv:2109.01652*.

Zheng Yuan, Chuanqi Tan, Songfang Huang, and Fei Huang. 2021. Fusing heterogeneous factors with triaffine mechanism for nested named entity recognition. *arXiv preprint arXiv:2110.07480*.

Sheng Zhang, Hao Cheng, Jianfeng Gao, and Hoifung Poon. 2022. Optimizing bi-encoder for named entity recognition via contrastive learning. *arXiv preprint arXiv:2208.14565*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. pages 12697–12706.

A Datasets

A.1 Flat NER

CoNLL2003. CoNLL2003 (Sang and De Meulder, 2003) contains four types of named entities: Location, Organization, Person and Miscellaneous. Table 3 shows annotations for each entity type and Table 5 shows the number of sentences, tokens and entities in CoNLL2003.

OntoNotes5.0. OntoNotes (Pradhan et al., 2013) contains 18 types of named entities, and Table 4 lists each entity and its annotation. The number of sentences, tokens and entities of OntoNotes5.0 is shown in Table 5.

Baselines. We adopt currently widely-used NER systems as baselines including:

- **BERT-Tagger** (Devlin et al., 2018) fine-tunes BERT on the full training dataset.
- **MRC-NER** (Li et al., 2019a) formulates the NER task as a machine reading comprehension (MRC) task and trains the MRC-NER model on the full training dataset.
- **MRC-NER+DSC** (Li et al., 2019b) is the current SOTA model on the OntoNotes5.0 dataset, leveraging dice loss in replacement of the standard cross-entropy loss during training.
- **GNN-SL** (Wang et al., 2022) fine-tunes RoBERTa (Liu et al., 2019) on the full training dataset and using GNN to refer to the whole training examples at test time.
- **ACE+document-context** (Wang et al., 2020) is the current SOTA model on the CoNLL2003 dataset, optimizing the controller to find better concatenations of embeddings on the full training dataset.

A.2 Nested NER

ACE2004 and ACE2005. ACE2004 (Doddington et al., 2004) and ACE2005 (Christopher et al., 2006) are two English nested NER datasets containing seven entity types: geographical political entities (GPE), organization entities (ORG), person entities (PER), facility entities (FAC), vehicle entities (VEH), location entities (LOC) and weapon entities (WEA). Annotations for each entity type are shown in Table 7, and the number of sentences, entities, nested entities and nested percentages are shown in Table 6.

Entities Annotations of English CoNLL2003	
Entity Type	**Annotation**
ORG	organization entities are limited to named corporate, governmental, or other organizational entities
PER	person entities are named persons or family
LOC	location entities are the name of politically or geographically defined locations such as cities, provinces, countries, international regions, bodies of water, mountains, etc
MISC	miscellaneous entities include events, nationalities, products and works of art

Table 3: Entity annotations of the flat NER dataset CoNLL2003.

Entities Annotations of English OntoNotes5.0	
Entity Type	**Annotation**
PERSON	People, including fictional
NORP	Nationalities or religious or political groups
FAC	Buildings, airports, highways, bridges, etc
ORG	Companies, agencies, institutions, etc
GPE	Countries, cities, states
LOC	Non-GPE locations, mountain ranges, bodies of water
PRODUCT	Vehicles, weapons, foods, etc
EVENT	Named hurricanes, battles, wars, sports events, etc
WORK_OF_ART	Titles of books, songs, etc
LAW	Named documents made into laws
LANGUAGE	Any named language
DATE	Absolute or relative dates or periods
TIME	Times smaller than a day
PERCENT	Percentage (including "%")
MONEY	Monetary values, including unit
QUANTITY	Measurements, as of weight or distance
ORDINAL	"first", "second", etc
CARDINAL	Numerals that do not fall under another type

Table 4: Entity annotations of the flat NER dataset OntoNotes5.0.

Statistics on English CoNLL2003			
Dataset	**Sentences**	**Tokens**	**Entities**
Training set	14,987	203,621	23,499
Development set	3,466	51,362	5,942
Test set	3,684	46,435	5,648
Statistics on OntoNotes5.0			
Dataset	**Sentences**	**Tokens**	**Entities**
Training set	59,924	1,088,503	81,828
Development set	8,528	147,724	11,066
Test set	8,262	152,728	11,257

Table 5: Number of sentences, tokens and entities of the flat NER dataset English CoNLL2003 and OntoNotes5.0.

Statistics on ACE2004				
Dataset	**Sentences**	**Entities**	**Nested Entities**	**Nested Percentage**
Training set	6,200	22,204	10,149	45.71%
Development set	745	2,514	1,092	46.69%
Test set	812	3,035	1,417	45.61%
Statistics on ACE2005				
Dataset	**Sentences**	**Entities**	**Nested Entities**	**Nested Percentage**
Training set	7,194	24,441	9,389	38.41%
Development set	969	3,200	1,112	34.75%
Test set	1,047	2,993	1,118	37.35%
Statistics on GENIA				
Dataset	**Sentences**	**Entities**	**Nested Entities**	**Nested Percentage**
Training set	16,692	50,509	9,064	17.95%
Development set	-	-	-	-
Test set	1,854	5,506	1,199	21.78%

Table 6: Number of sentences, entities, nested entities, and nested percentage of the nested NER dataset ACE2004, ACE2005 and GENIA.

GENIA. GENIA is an English nested NER dataset within the molecular biology domain and contains five entity types: cell line, cell type, DNA, RNA and protein. Literally, each entity is named according to the biological meaning, and the number of sentences, entities, nested entities and nested percentages are shown in Table 6.

Baselines. For baselines, four widely used supervised models are included:

Entities Annotations of English ACE2004 and ACE2005	
Entity Type	**Annotation**
GPE	geographical political entities are geographical regions defined by political and or social groups such as countries, nations, regions, cities, states, government and its people
ORG	organization entities are limited to companies, corporations, agencies, institutions and other groups of people
PER	a person entity is limited to human including a single individual or a group
FAC	facility entities are limited to buildings and other permanent man-made structures such as buildings, airports, highways, bridges
VEH	vehicle entities are physical devices primarily designed to move, carry, pull or push the transported object such as helicopters, trains, ship and motorcycles
LOC	location entities are limited to geographical entities such as geographical areas and landmasses, mountains, bodies of water, and geological formations
WEA	weapon entities are limited to physical devices such as instruments for physically harming such as guns, arms and gunpowder

Table 7: Entity annotations of the dataset ACE2004 and ACE2005.

- **BERT-MRC** (Li et al., 2019a): the current SOTA model on the GENIA dataset, formulating the NER task as a machine reading comprehension (MRC) task and training the MRC-NER model on the full training dataset.

- **Triaffine+BERT** (Yuan et al., 2021): fine-tuning BERT (Devlin et al., 2018) on the full training set and fusing heterogeneous factors for span representations and classification.

- **Triaffine+ALBERT** (Yuan et al., 2021): fine-tuning ALBERT (Lan et al., 2019) on the full training set and fusing heterogeneous factors for span representations and classification.

- **BINDER** (Zhang et al., 2022): the current SOTA model on the ACE2004 dataset and ACE2005 dataset, leveraging a bi-encoder framework to apply contrastive learning to map candidate text spans and entity types into the same vector representation space for representation and classification.

B Results on the Partial Test Set

Table 13 shows results on the partial. Observations are as follows:

(1) kNN retrieval is of vital importance for the NER task.

(2) We observe a significant improvement by changing the sentence-level embedding to token-level embedding for the kNN demonstration search.

(3) We observe further improvements by adding self-verification: on the full CoNLL2003 dataset with entity-level embedding.

(4) LLM-based systems obtain comparable results to supervised baselines using BERT.

C Error Cases of Format BMES and Entity-position

We select several examples on sample-100 CoNLL2003 dataset to better illustrate the ineffectiveness of these two formats BMES and entity-position. For BMES format is shown in Table 8, and for entity-position format is shown in 9.

From these examples, we can obviously observe that (1) for BMES format, it is difficult for GPT-3 to generate the output with the same length as the input sentence, especially when the input sentence is long; (2) for entity-position format, it is confused for GPT-3 to generate the correct position information.

D Examples

To better illustrate demonstrations of our GPT-NER, we select several examples for random retrieval in Table 10, for sentence-level embedding 11 and for entity-level embedding 12. From these results, we can observe that:

(1) For random retrieval in Table 10, we can observe that all sentences have the same opportunity to appear as an example in the few-shot demonstration, and the input sentence and each retrieved example usually do not contain similar examples.

(2) For sentence-level embedding in Table 11, we can observe that the retrieved examples are semantically similar to the input sentence, but may not focus on the same local entities as the input sentence.

(3) For entity-level embedding in Table 12, we can observe that the retrieved examples do focus on the same local entities as the input sentence to lead the prediction progress of GPT-3 more easily. This phenomenon emphasizes the effectiveness of the quality of demonstrations in-context learning.

Example 1 **Length Error**

Task Description
I am an excellent linguist. The task is to label organization entities. Below are some examples.

Sentence-level Demonstrations
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O O O O O O
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O O O O O O
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O O O O O O
Input: Soccer - Italian cup second round results .
Output: O O O O O O O O

Input Sentence and GPT-3 Output
Input: Soccer - Japan get lucky win , China in surprise defeat .
Output: O O O O O O O O O O O
Expected Output: O O O O O O O O O O O O

Example 2 **Length Error and Entity Error**

Task Description
I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.

Sentence-level Demonstrations
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O B-MISC E-MISC O O O
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O B-MISC E-MISC O O O
Input: Soccer - results of South Korean PRO-SOCCER games .
Output: O O O O B-MISC E-MISC O O O
Input: Soccer - Italian cup second round results .
Output: O O B-MISC E-MISC O O O O

Input Sentence and GPT-3 Output
Input: Soccer - Japan get lucky win , China in surprise defeat .
Output: O O B-MISC E-MISC O O O O B-MISC E-MISC O O
Expected Output: O O O O O O O O O O O O

Example 3 **Length Error**

Task Description
I am an excellent linguist. The task is to label person entities. Below are some examples.

Sentence-level Demonstrations
Input: Dubai 1996-08-26
Output: O O
Input: Dubai 1996-08-29
Output: O O
Input: Dubai 1996-08-29
Output: O O
Input: Dubai 1996-08-22
Output: O O

Input Sentence and GPT-3 Output
Input: AL-AIN , United Arab Emirates 1996-12-06
Output: O O
Expected Output: O O O O O O

Example 4 **Length Error and Entity Error**

Task Description
I am an excellent linguist. The task is to label location entities. Below are some examples.

Sentence-level Demonstrations
Input: Azerbaijan beat Switzerland 1-0 (halftime 1-0) in their World Cup soccer European group three qualifying match on Saturday .
Output: S-LOC O S-LOC O O O O O O O O O O O O O O O O O O
Input: Nijmeh of Lebanon beat Nasr of Saudi Arabia 1-0 (halftime 1-0) in their Asian club championship second round first leg tie on Saturday .
Output: O O S-LOC O O O B-LOC E-LOC O O O O O O O O O O O O O O O O O O
Input: Slovakia beat the Faroe Islands 2-1 (halftime 1-0) in their World Cup soccer European group six qualifying match on Saturday .
Output: S-LOC O O B-LOC E-LOC O O O O O O O O O O O O O O O O O O
Input: Canada beat Panama 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 World Cup on Friday .
Output: S-LOC O S-LOC O O O O O O O O O O O O O O O O O O O O

Input Sentence and GPT-3 Output
Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday .
Output: S-LOC O O O O O O O O O O O O B-LOC E-LOC O O O O O O O O O O O O O O
Expected Output: S-LOC O O O O O O O O O O O O O O S-LOC O O O O O O O O O

Table 8: Examples for the BMES output format on sample-100 CoNLL2003 dataset, where the error information is colored red and the expected correct output is colored blue.

Example 1 **Position Error and Entity Error**
Task Description I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.
Sentence-level Demonstrations Input: Soccer - results of South Korean PRO-SOCCER games . Output: South Korean(4) Input: Soccer - results of South Korean PRO-SOCCER games . Output: South Korean(4) Input: Soccer - results of South Korean PRO-SOCCER games . Output: South Korean(4) Input: Soccer - Italian cup second round results . Output: Italian cup(2)
Input Sentence and GPT-3 Output Input: Soccer - Japan get lucky win , China in surprise defeat . Output: Japan(4), China(4) *Expected Output*: None
Example 2 **Position Error and Entity Error**
Task Description I am an excellent linguist. The task is to label organization entities. Below are some examples.
Sentence-level Demonstrations Input: Dubai 1996-08-26 Output: None Input: Dubai 1996-08-29 Output: None Input: Dubai 1996-08-29 Output: None Input: Dubai 1996-08-22 Output: None
Input Sentence and GPT-3 Output Input: AL-AIN , United Arab Emirates 1996-12-06 Output: AL-AIN, United Arab Emirates *Expected Output*: None
Example 3 **Position Error**
Task Description I am an excellent linguist. The task is to label location entities. Below are some examples.
Sentence-level Demonstrations Input: Third seed Arantxa Sanchez Vicario , the 1994 champion , and eighth-seeded Olympic gold medalist Lindsay Davenport dropped three game each en route to the second round . Output: None Input: Dutch champions Ajax Amsterdam faltered in their second league match of the season on Saturday losing 2-0 away at Heerenveen . Output: None Input: Soccer - disappointing Ajax slump 2-0 at Heerenveen . Output: Heerenveen(7) Input: Australian Open runner-up Anke Huber of Germany , the sixth seed , was undone by an unlucky draw that put her against 17th ranked South African Amanda Coetzer in her opening match . Output: Germany(6)
Input Sentence and GPT-3 Output Input: But China saw their luck desert them in the second match of the group , crashing to a surprise 2-0 defeat to newcomers Uzbekistan . Output: China(2), Uzbekistan(9) *Expected Output*: China(1), Uzbekistan(23)
Example 4 **Position Error and Entity Error**
Task Description I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples. *Sentence-level Demonstrations* Input: Azerbaijan beat Switzerland 1-0 (halftime 1-0) in their World Cup soccer European group three qualifying match on Saturday . Output: World Cup(10), European(13) Input: Nijmeh of Lebanon beat Nasr of Saudi Arabia 1-0 (halftime 1-0) in their Asian club championship second round first leg tie on Saturday . Output: Asian(15) Input: Slovakia beat the Faroe Islands 2-1 (halftime 1-0) in their World Cup soccer European group six qualifying match on Saturday . Output: World Cup(12), European(15) Input: Canada beat Panama 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 World Cup on Friday . Output: World Cup(18)
Input Sentence and GPT-3 Output Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday . Output: Asian(14) *Expected Output*: Asian Cup(6)

Table 9: Examples for the entity-position output format on sample-100 CoNLL2003 dataset, where the error information is colored red and the expected correct output is colored blue.

Example 1

Task Description
I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.

Sentence-level Demonstrations
Input: Seattle at Boston
Output: Seattle at Boston
Input: 3. Carla Sacramento (Portugal) 4:08.96
Output: 3. Carla Sacramento (Portugal) 4:08.96
Input: Director Budge Weidman , who has shepherded the project from the beginning , predicts it will take up to a decade to complete .
Output: Director Budge Weidman , who has shepherded the project from the beginning , predicts it will take up to a decade to complete .
Input: Hull 0 Barnet 0
Output: Hull 0 Barnet 0
Input: Scott Draper (Australia) vs. Galo Blanco (Spain)
Output: Scott Draper (Australia) vs. Galo Blanco (Spain)
Input: Standings in the French first
Output: Standings in the @@French## first
Input: He said only the removal of the government and an early election could save Pakistan from disaster . "
Output: He said only the removal of the government and an early election could save Pakistan from disaster . "
Input: Stock markets
Output: Stock markets

Input Sentence and GPT-3 Output
Input: Soccer - Japan get lucky win , China in surprise defeat .
Output: Soccer - Japan get lucky win , China in surprise defeat .
Expected Output: Soccer - Japan get lucky win , China in surprise defeat .

Example 2

Task Description
I am an excellent linguist. The task is to label organization entities. Below are some examples.

Sentence-level Demonstrations
Input: Jakob Hlasek (Switzerland) beat Alberto Berasategui (Spain) 7-6 (7-5) 7-6 (9-7) 6-0
Output: Jakob Hlasek (Switzerland) beat Alberto Berasategui (Spain) 7-6 (7-5) 7-6 (9-7) 6-0
Input: After bogeying the 10th hole to move to four-over for the round , he rallied for birdies on 15 and 18 .
Output: After bogeying the 10th hole to move to four-over for the round , he rallied for birdies on 15 and 18 .
Input: Abidjan 1996-08-29
Output: Abidjan 1996-08-29
Input: Falkirk 1 Partick 0
Output: @@Falkirk## 1 @@Partick## 0
Input: Williams seized two wickets in two deliveries and left-armer Ilott also captured two as Gloucestershire , 252 behind on first innings , slumped to 27 for four at the close on the third day of the four-day game at Colchester .
Output: Williams seized two wickets in two deliveries and left-armer Ilott also captured two as @@Gloucestershire## , 252 behind on first innings , slumped to 27 for four at the close on the third day of the four-day game at Colchester .
Input: South Queensland 21 4 0 17 210 460 8
Output: @@South Queensland## 21 4 0 17 210 460 8
Input: In Skopje : Sloga Jugomagnat (Macedonia) 0 Kispest Honved
Output: In Skopje : @@Sloga Jugomagnat## (Macedonia) 0 Kispest Honved
Input: Call C 98.00 pct 0.47 Dem 3.30 pct 202.90 X
Output: Call C 98.00 pct 0.47 Dem 3.30 pct 202.90 X

Input Sentence and GPT-3 Output
Input: AL-AIN , United Arab Emirates 1996-12-06
Output: @@AL-AIN## , United Arab Emirates 1996-12-06
Expected Output: AL-AIN , United Arab Emirates 1996-12-06

Example 3

Task Description
I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.

Sentence-level Demonstrations
Input: Serbian policeman shot dead in Kosovo province .
Output: @@Serbian## policeman shot dead in Kosovo province .
Input: British Labour Party leader Tony Blair won a narrow victory on Saturday when the party 's Scottish executive voted 21-18 in favour of his plans for a referendum on a separate parliament for Scotland .
Output: British Labour Party leader Tony Blair won a narrow victory on Saturday when the party 's @@Scottish## executive voted 21-18 in favour of his plans for a referendum on a separate parliament for Scotland .
Input: Newcastle 24 Western Reds 20
Output: Newcastle 24 Western Reds 20
Input: WSRL is part of the Welspun group which has a presence in the cotton yarn , terry towels and polyester yarn industry , the statement said .
Output: WSRL is part of the Welspun group which has a presence in the cotton yarn , terry towels and polyester yarn industry , the statement said .
Input: In Chicago , Erik Hanson outdueled Alex Fernandez , and Jacob Brumfield drove in Otis Nixon with the game 's only run in the sixth inning as the Toronto Blue Jays blanked the White Sox 1-0 in a game shortened to six innings due to rain .
Output: In Chicago , Erik Hanson outdueled Alex Fernandez , and Jacob Brumfield drove in Otis Nixon with the game 's only run in the sixth inning as the Toronto Blue Jays blanked the White Sox 1-0 in a game shortened to six innings due to rain .
Input: (Corrects that Habsudova is sixth seed) .
Output: (Corrects that Habsudova is sixth seed) .
Input: San Francisco at New York
Output: San Francisco at New York
Input: – The short-term price objective is $ 5 a share and the long-term objective is $ 9 .
Output: – The short-term price objective is $ 5 a share and the long-term objective is $ 9 .

Input Sentence and GPT-3 Output
Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday .
Output: Japan began the defence of their @@Asian## Cup title with a lucky 2-1 win against @@Syria## in a Group C championship match on Friday .
Expected Output: Japan began the defence of their @@Asian Cup## title with a lucky 2-1 win against Syria in a Group C championship match on Friday .

Table 10: Examples on the CoNLL2003 datasets with the random retrieval.

Example 1
Task Description
I am an excellent linguist. The task is to label location entities. Below are some examples.
Sentence-level Demonstrations Input: Dubai 1996-08-26 Output: @@Dubai## 1996-08-26 Input: Dubai 1996-08-29 Output: @@Dubai## 1996-08-29 Input: Dubai 1996-08-29 Output: @@Dubai## 1996-08-29 Input: Dubai 1996-08-22 Output: @@Dubai## 1996-08-22 Input: Dubai 1996-08-25 Output: @@Dubai## 1996-08-25 Input: Baghdad 1996-08-24 Output: @@Baghdad## 1996-08-24 Input: Baghdad 1996-08-27 Output: @@Baghdad## 1996-08-27 Input: Baghdad 1996-08-28 Output: @@Baghdad## 1996-08-28
Input Sentence and GPT-3 Output Input: AL-AIN , United Arab Emirates 1996-12-06 Output: @@AL-AIN## , @@United Arab Emirates## 1996-12-06 *Expected Output*: @@AL-AIN## , @@United Arab Emirates## 1996-12-06
Example 2
Task Description I am an excellent linguist. The task is to label location entities. Below are some examples.
Sentence-level Demonstrations Input: Azerbaijan beat Switzerland 1-0 (halftime 1-0) in their World Cup soccer European group three qualifying match on Saturday . Output: @@Azerbaijan## beat @@Switzerland## 1-0 (halftime 1-0) in their World Cup soccer European group three qualifying match on Saturday . Input: Nijmeh of Lebanon beat Nasr of Saudi Arabia 1-0 (halftime 1-0) in their Asian club championship second round first leg tie on Saturday . Output: Nijmeh of @@Lebanon## beat Nasr of @@Saudi Arabia## 1-0 (halftime 1-0) in their Asian club championship second round first leg tie on Saturday . Input: Slovakia beat the Faroe Islands 2-1 (halftime 1-0) in their World Cup soccer European group six qualifying match on Saturday . Output: @@Slovakia## beat the @@Faroe Islands## 2-1 (halftime 1-0) in their World Cup soccer European group six qualifying match on Saturday . Input: Canada beat Panama 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 World Cup on Friday . Output: @@Canada## beat @@Panama## 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 World Cup on Friday . Input: Soccer - Azerbaijan beat Switzerland in world cup Qualifier . Output: Soccer - @@Azerbaijan## beat @@Switzerland## in world cup Qualifier . Input: Soccer - Wales beat San Marino in world cup Qualifier . Output: Soccer - @@Wales## beat @@San Marino## in world cup Qualifier . Input: The United States edged Austria in Salzburg 3-2 in the opening round in April , and then blanked Japan 5-0 in Nagoya last month in the semifinals . Output: The @@United States## edged @@Austria## in @@Salzburg## 3-2 in the opening round in April , and then blanked @@Japan## 5-0 in @@Nagoya## last month in the semifinals . Input: Soccer - Slovakia beat Faroes in world cup Qualifier . Output: Soccer - @@Slovakia## beat @@Faroes## in world cup Qualifier .
Input Sentence and GPT-3 Output Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday . Output: @@Japan## began the defence of their Asian Cup title with a lucky 2-1 win against @@Syria## in a Group C championship match on Friday . *Expected Output*: @@Japan## began the defence of their Asian Cup title with a lucky 2-1 win against @@Syria## in a Group C championship match on Friday .
Example 3
Task Description I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.
Sentence-level Demonstrations Input: Azerbaijan beat Switzerland 1-0 (halftime 1-0) in their World Cup soccer European group three qualifying match on Saturday . Output: Azerbaijan beat Switzerland 1-0 (halftime 1-0) in their @@World Cup## soccer @@European## group three qualifying match on Saturday . Input: Nijmeh of Lebanon beat Nasr of Saudi Arabia 1-0 (halftime 1-0) in their Asian club championship second round first leg tie on Saturday . Output: Nijmeh of Lebanon beat Nasr of Saudi Arabia 1-0 (halftime 1-0) in their @@Asian## club championship second round first leg tie on Saturday . Input: Slovakia beat the Faroe Islands 2-1 (halftime 1-0) in their World Cup soccer European group six qualifying match on Saturday . Output: Slovakia beat the Faroe Islands 2-1 (halftime 1-0) in their @@World Cup## soccer @@European## group six qualifying match on Saturday . Input: Canada beat Panama 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 World Cup on Friday . Output: Canada beat Panama 3-1 (halftime 2-0) in their CONCACAF semifinal phase qualifying match for the 1998 @@World Cup## on Friday . Input: Soccer - Azerbaijan beat Switzerland in world cup Qualifier . Output: Soccer - Azerbaijan beat Switzerland in @@world cup## Qualifier . Input: Soccer - Wales beat San Marino in world cup Qualifier . Output: Soccer - Wales beat San Marino in @@world cup## Qualifier . Input: The United States edged Austria in Salzburg 3-2 in the opening round in April , and then blanked Japan 5-0 in Nagoya last month in the semifinals . Output: The United States edged Austria in Salzburg 3-2 in the opening round in April , and then blanked Japan 5-0 in Nagoya last month in the semifinals . Input: Soccer - Slovakia beat Faroes in world cup Qualifier . Output: Soccer - Slovakia beat Faroes in @@world cup## Qualifier .
Input Sentence and GPT-3 Output Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday . Output: Japan began the defence of their @@Asian Cup## title with a lucky 2-1 win against Syria in a Group C championship match on Friday . *Expected Output*: Japan began the defence of their @@Asian Cup## title with a lucky 2-1 win against Syria in a Group C championship match on Friday .

Table 11: Examples on the CoNLL2003 datasets with the sentence-level embedding.

Example 1

Task Description
I am an excellent linguist. The task is to label location entities. Below are some examples.

Sentence-level Demonstrations
Input: AL-RAM , West Bank 1996-08-30
Output: @@AL-RAM## , @@West Bank## 1996-08-30
Input: AL-MUNTAR , West Bank 1996-08-26
Output: @@AL-MUNTAR## , @@West Bank## 1996-08-26
Input: Teravainen (U.S.) , Jean Van de Velde (France) , Oyvind Rojahn
Output: Teravainen (@@U.S.##) , Jean Van de Velde (@@France##) , Oyvind Rojahn
Input: The greatest declines in the volume of help-wanted advertising were in the New England , Mountain and West South Central regions .
Output: The greatest declines in the volume of help-wanted advertising were in the @@New England## , @@Mountain## and @@West South Central## regions .
Input: Doug Flach (U.S.) beat Gianluca Pozzi (Italy) 7-5 7-6 (7-5) 2-6 7-6 (8-6)
Output: Doug Flach (@@U.S.##) beat Gianluca Pozzi (@@Italy##) 7-5 7-6 (7-5) 2-6 7-6 (8-6)
Input: Jeff Tarango (U.S.) beat Alex Radulescu (Romania) 6-7 (5-7) 6-4 6-1 retired , heat exhaustion
Output: Jeff Tarango (@@U.S.##) beat Alex Radulescu (@@Romania##) 6-7 (5-7) 6-4 6-1 retired , heat exhaustion
Input: Chelsea , 16 , was at President Bill Clinton 's side as he rode the rails through parts of West Virginia , Kentucky and Ohio , and was introduced at every stop .
Output: Chelsea , 16 , was at President Bill Clinton 's side as he rode the rails through parts of @@West Virginia## , @@Kentucky## and @@Ohio## , and was introduced at every stop .
Input: Clinton said on Saturday he had ordered U.S. forces in the Gulf to go on high alert and was reinforcing them in response to Iraqi attacks on Kurdish dissidents in northern Iraq .
Output: Clinton said on Saturday he had ordered @@U.S.## forces in the @@Gulf## to go on high alert and was reinforcing them in response to Iraqi attacks on Kurdish dissidents in northern @@Iraq## .

Input Sentence and GPT-3 Output
Input: AL-AIN , United Arab Emirates 1996-12-06
Output: @@AL-AIN## , @@United Arab Emirates## 1996-12-06
Expected Output: @@AL-AIN## , @@United Arab Emirates## 1996-12-06

Example 2

Task Description
I am an excellent linguist. The task is to label miscellaneous entities. Below are some examples.

Sentence-level Demonstrations
Input: The armed hijackers of the Airbus 310 Flight 150 , which is expected to arrive about 4 a.m. (0300 GMT) , have said they intend to surrender and seek political asylum in Britain .
Output: The armed hijackers of the @@Airbus 310## @@Flight 150## , which is expected to arrive about 4 a.m. (0300 @@GMT##) , have said they intend to surrender and seek political asylum in Britain .
Input: Toronto-based Barrick , the world 's third largest gold producer , sweetened its July 11 bid to C$ 30 a share from C$ 27 on August 16 after a fresh batch of drill results from the Pierina deposit .
Output: @@Toronto-based## Barrick , the world 's third largest gold producer , sweetened its July 11 bid to @@C$## 30 a share from @@C$## 27 on August 16 after a fresh batch of drill results from the Pierina deposit .
Input: The club , who put Manchester United out of last year 's UEFA Cup , were fined $ 1,000 .
Output: The club , who put Manchester United out of last year 's @@UEFA Cup## , were fined $ 1,000 .
Input: Shr C$ 0.12 C$ 0.15
Output: Shr @@C$## 0.12 @@C$## 0.15
Input: Shr C$ 0.04 C$ 0.08
Output: Shr @@C$## 0.04 @@C$## 0.08
Input: An Iraqi Kurdish group on Wednesday said it had agreed a new U.S.-brokered ceasefire with a rival faction after a previous accord was shattered by sporadic fighting between the groups in recent days .
Output: An @@Iraqi Kurdish## group on Wednesday said it had agreed a new @@U.S.-brokered## ceasefire with a rival faction after a previous accord was shattered by sporadic fighting between the groups in recent days .
Input: On Friday , Metro Holdings topped gainers , soaring by S$ 1.55 to close at S$ 6.05 on market rumours of a takeover bid by First Capital Corp .
Output: On Friday , Metro Holdings topped gainers , soaring by @@S$## 1.55 to close at @@S$## 6.05 on market rumours of a takeover bid by First Capital Corp .
Input: We 're looking for it to stabilise now , " said one Euromark options trader at a U.S. bank .
Output: We 're looking for it to stabilise now , " said one @@Euromark## options trader at a U.S. bank .

Input Sentence and GPT-3 Output
Input: Japan began the defence of their Asian Cup title with a lucky 2-1 win against Syria in a Group C championship match on Friday .
Output: Japan began the defence of their @@Asian Cup## title with a lucky 2-1 win against Syria in a Group C championship match on Friday .
Expected Output: Japan began the defence of their @@Asian Cup## title with a lucky 2-1 win against Syria in a Group C championship match on Friday .

Example 3

Task Description
I am an excellent linguist. The task is to label location entities. Below are some examples.

Sentence-level Demonstrations
Input: In April , China quashed a draft resolution by the U.N. Human Rights Commission expressing concern over continuing reports of Beijing 's violations of fundamental freedoms .
Output: In April , @@China## quashed a draft resolution by the U.N. Human Rights Commission expressing concern over continuing reports of @@Beijing## 's violations of fundamental freedoms .
Input: China thanks Gabon for support on human rights .
Output: @@China## thanks @@Gabon## for support on human rights .
Input: China says Taiwan spoils atmosphere for talks .
Output: @@China## says @@Taiwan## spoils atmosphere for talks .
Input: Asked what India would do if the pact were forwarded to the United Nations General Assembly , Gujral said : " That bridge I will cross when I come to it . "
Output: Asked what @@India## would do if the pact were forwarded to the United Nations General Assembly , Gujral said : " That bridge I will cross when I come to it . "
Input: China says militant Japan must face war past .
Output: @@China## says militant @@Japan## must face war past .
Input: The victory against Japan marked the Fed Cup debut of Monica Seles , who became a naturalised U.S. citizen in 1994 .
Output: The victory against @@Japan## marked the Fed Cup debut of Monica Seles , who became a naturalised @@U.S.## citizen in 1994 .
Input: The constitutional monarch , who last visited China in 1993 , was scheduled to meet Chinese President Jiang Zemin and Premier Li Peng during his visit , they said .
Output: The constitutional monarch , who last visited @@China## in 1993 , was scheduled to meet Chinese President Jiang Zemin and Premier Li Peng during his visit , they said .
Input: Atheist China officially bans missionary activities but often turns a blind eye to religious activities of people nominally employed as foreign language teachers , particularly in remote areas that are unable to attract other candidates .
Output: Atheist @@China## officially bans missionary activities but often turns a blind eye to religious activities of people nominally employed as foreign language teachers , particularly in remote areas that are unable to attract other candidates .

Input Sentence and GPT-3 Output
Input: But China saw their luck desert them in the second match of the group , crashing to a surprise 2-0 defeat to newcomers Uzbekistan .
Output: But @@China## saw their luck desert them in the second match of the group , crashing to a surprise 2-0 defeat to newcomers @@Uzbekistan##
Expected Output: But @@China## saw their luck desert them in the second match of the group , crashing to a surprise 2-0 defeat to newcomers @@Uzbekistan##

Table 12: Examples on the CoNLL2003 datasets with the entity-level embedding.

English CoNLL2003 (Sampled 100)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
ACE+document-context (Wang et al., 2020)	97.8	98.28	**98.04 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	88.18	78.54	83.08
GPT-3 + *sentence-level embedding*	90.47	95	92.68
GPT-3 + *entity-level embedding*	94.06	96.54	95.3
Self-verification (zero-shot)			
+ GPT-3 + *random retrieval*	88.95	79.73	84.34
+ GPT-3 + *sentence-level embedding*	91.77	96.36	94.01
+ GPT-3 + *entity-level embedding*	94.15	96.77	95.46
Self-verification (few-shot)			
+ GPT-3 + *random retrieval*	90.04	80.14	85.09
+ GPT-3 + *sentence-level embedding*	92.92	95.45	94.17
+ GPT-3 + *entity-level embedding*	94.73	96.97	95.85
English OntoNotes5.0 (Sampled 100)			
Model	**Precision**	**Recall**	**F1**
Baselines (Supervised Model)			
BERT-MRC+DSC (Li et al., 2019b)	93.81	93.95	**93.88 (SOTA)**
GPT-NER			
GPT-3 + *random retrieval*	64.21	65.51	64.86
GPT-3 + *sentence-level embedding*	76.08	83.06	79.57
GPT-3 + *entity-level embedding*	78.38	83.9	81.14
Self-verification (zero-shot)			
+ GPT-3 + *random retrieval*	64.94	65.90	65.42
+ GPT-3 + *sentence-level embedding*	77.33	83.29	80.31
+ GPT-3 + *entity-level embedding*	79.05	83.71	81.38
Self-verification (few-shot)			
+ GPT-3 + *random retrieval*	65.21	66.25	65.73
+ GPT-3 + *sentence-level embedding*	77.64	83.22	80.43
+ GPT-3 + *entity-level embedding*	79.25	83.73	81.49

Table 13: Results of sampled 100 pieces of data for two **Flat** NER datasets: CoNLL2003 and OntoNotes5.0.