ConsistNER: Towards Instructive NER Demonstrations for LLMs with the Consistency of Ontology and Context

Chenxiao Wu[1*], **Wenjun Ke**[1, 3*†], **Peng Wang**[1, 2, 3†], **Zhizhao Luo**[4], **Guozheng Li**[1], **Wanyi Chen**[2]

[1]School of Computer Science and Engineering, [2]School of Cyber Science and Engineering, Southeast University
[3]Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China
[4]Beijing Institute of Computer Technology and Application
yanzu0311@gmail.com, {kewenjun, pwang, liguozheng, wanyichen}@seu.edu.cn, qibai-aluminum@outlook.com

Abstract

Named entity recognition (NER) aims to identify and classify specific entities mentioned in textual sentences. Most existing superior NER models employ the standard fully supervised paradigm, which requires a large amount of annotated data during training. In order to maintain performance with insufficient annotation resources ($i.e.$, low resources), in-context learning (ICL) has drawn a lot of attention, due to its plug-and-play nature compared to other methods ($e.g.$, meta-learning and prompt learning). In this manner, how to retrieve high-correlated demonstrations for target sentences serves as the key to emerging ICL ability. For the NER task, the correlation implies the consistency of both ontology ($i.e.$, generalized entity type) and context ($i.e.$, sentence semantic), which is ignored by previous NER demonstration retrieval techniques. To address this issue, we propose ConsistNER, a novel three-stage framework that incorporates ontological and contextual information for low-resource NER. Firstly, ConsistNER employs large language models (LLMs) to pre-recognize potential entities in a zero-shot manner. Secondly, ConsistNER retrieves the sentence-specific demonstrations for each target sentence based on the two following considerations: (1) Regarding ontological consistency, demonstrations are filtered into a candidate set based on ontology distribution. (2) Regarding contextual consistency, an entity-aware self-attention mechanism is introduced to focus more on the potential entities and semantic-correlated tokens. Finally, ConsistNER feeds the retrieved demonstrations for all target sentences into LLMs for prediction. We conduct experiments on four widely-adopted NER datasets, including both general and specific domains. Experimental results show that ConsistNER achieves a 6.01%-26.37% and 3.07%-21.18% improvement over the state-of-the-art baselines on Micro-F1 scores under 1- and 5-shot settings, respectively.

Introduction

NER is a fundamental natural language processing (NLP) task for various downstream tasks such as entity linking (Sevgili and Shelmanov 2020), event extraction (Xiang and Wang 2019), and Q&A (Kolomiyets and Moens 2011). Existing NER methods with pre-trained language models (PLMs) have achieved outstanding performance by employing the standard fully supervised paradigm (Peters et al. 2017; Souza, Nogueira, and Lotufo 2019). However, such a supervised paradigm heavily depends on large-scale annotated data. Hence, in real-world scenarios, existing methods tend to struggle when recognizing new entities with insufficient annotation resources ($i.e.$, low resources).

Several types of methods have been proposed to alleviate the challenge of low-resource NER, varying in meta-learning (Wu et al. 2020; de Lichy, Glaude, and Campbell 2021; Ma et al. 2022b), prompt learning (Ma et al. 2022a; Liu et al. 2022; Chen et al. 2022) and in-context learning (Brown et al. 2020; Smith et al. 2022; Du et al. 2022). Among these, in-context learning (ICL), which concatenates a query and few-shot demonstrations to prompt LLMs for prediction, is plug-and-play and does not require additional inductive bias learning or sophisticated template design. Generally, the primary research of ICL can be grouped into two directions: query forms (Wang et al. 2023b; Mishra et al. 2022; Wang et al. 2022; Wei et al. 2023; Wang et al. 2023a) and demonstration retrieval techniques (Ma et al. 2023; Jimenez Gutierrez et al. 2022; Lee et al. 2022; Wang et al. 2023a). The variances in query forms have the potential to cause significant differences in model predictions. Compared to directly applying the straightforward instructions as queries (Wang et al. 2023b; Mishra et al. 2022; Wang et al. 2022), multi-turn Q&A according to entity types could improve NER performance (Wei et al. 2023; Wang et al. 2023a). However, queries are often manually created based on human introspection (Petroni et al. 2019; Brown et al. 2020; Schick and Schütze 2021b,a,c), which results in query design becoming an engineering problem that requires extensive human experience and time (Shin et al. 2021).

On detailed analysis, the key to emerging ICL ability lies in how to retrieve high-correlated demonstrations. Current retrieval techniques have shown certain promising results in this regard (Ma et al. 2023; Jimenez Gutierrez et al. 2022; Wang et al. 2023a). However, these techniques fail to locate the delicate demonstration considering the consistency of ontology ($i.e.$, generalized entity type) and context ($i.e.$, sentence semantic), simultaneously. (1) Regarding the lack of ontological consistency, Ma et al. (2023) and Gutiérrez et al. (2022) use CLS embeddings of PLMs

*These authors contributed equally.
†Corresponding authors.

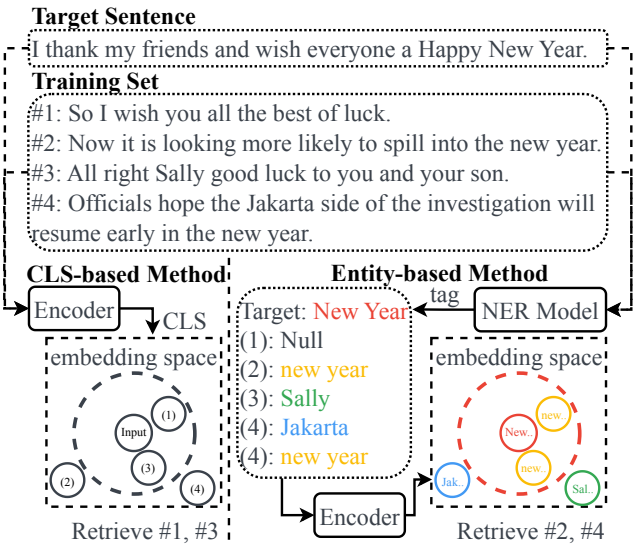


Figure 1: An illustration of different demonstration retrieval techniques, where different colors denote different generalized entity types. For the CLS-based Method, examples #1, #3 are retrieved based on the sentence embedding (CLS) similarity. For the Entity-based Method, after extracting entities with a NER tagging model, examples #2, #4 are retrieved based on the entity (*New Year*) embedding similarity.

to select semantically similar training examples as demonstrations. In this manner, the retrieved demonstrations might lack ontological consistency with target sentences, whose pre-given entity types still remain inconsistent after being generalized. As Figure 1 shown, considering the CLS-based method, the retrieved examples #1, #3, which only contain a *PER* entity *Sally*, could not provide sufficient assistance in recognizing the *EVENT* entity *New Year* in the target sentence. (2) Regarding the lack of contextual consistency, Wang et al. (2023a) suggest retrieving demonstrations based on the entity similarity, which could not guarantee the contextual consistency between the demonstrations and target sentences. Considering the entity-based method in Figure 1, the *DATE* entity *new year* in examples #2, #4 conveys different meanings compared to the *EVENT* entity *New Year* in the target sentence. This could mislead LLMs into recognizing the *New Year* in the target sentence as a *DATE* entity.

To address the above two issues, $i.e.$ the inconsistency of ontology and context, we propose ConsistNER, a novel three-stage framework that incorporates ontological and contextual information for low-resource NER. Firstly, LLMs are employed in a zero-shot manner to pre-recognize potential entities, enabling the utilization of local information ($i.e.$, entities and entity types) for later. Secondly, to retrieve the sentence-specific demonstrations for each target sentence, we consider both ontological and contextual consistency. Motivated by the bag of words (BoW) (Zhang, Jin, and Zhou 2010), we derive the bag of ontology (BoO) and the ontology distribution (OD) representation to filter demonstrations and construct the candidate set to maintain ontological consistency. Inspired by prototypical network (Snell, Swersky, and Zemel 2017), we design a dual self-attention to derive the entity-aware contextual (EAC) representation, which pays more attention to the potential entities and semantic-correlated tokens, to maintain contextual consistency. Finally, after all target sentences have retrieved their own sentence-specific demonstrations, ConsistNER feeds dataset-specific demonstrations, which are selected based on the occurrences, into LLMs to solve NER.

Experimental results on four benchmark datasets from CoNLL2003, OntoNotes5.0, NCBI and BC5CDR demonstrate the superior performance of ConsistNER over state-of-the-art low-resource NER models, with improvements on F1 scores of 6.01%, 26.37%, 19.56% and 21.44% on the four datasets, respectively. The effectiveness of maintaining ontological and contextual consistency between demonstrations and target sentences is also verified. Additionally, we analyze the theoretical boundary of ConsistNER, which reveals the importance of pre-recognition quality. To sum up, our major contributions are three-fold:

- We propose ConsistNER, a framework devoted to retrieving high-correlated demonstrations, in order to tackle the low-resource NER task with LLMs.
- We devise a novel sample correlation measure mechanism that considers both entity type ontology and contextual semantic, specifically designed for the NER task.
- Extensive experiments demonstrate the superiority of ConsistNER over state-of-the-art baselines. Further analysis verify effectiveness of ConsistNER in maintaining ontological and contextual consistency.

Method

We formulate the low-resource NER in the typical N-way-K-shot form. Given the training set $\mathcal{D}$ and the target sentence set $\mathcal{T}$ (both unlabeled), for N entity types, ConsistNER first retrieves and annotates K examples for each entity type from $\mathcal{D}$ to form the support set $\mathcal{S} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^{N\times K}$, where $\boldsymbol{x}^i = \{x_1^i, x_2^i, ..., x_m^i\}$ is an m-token text and $\boldsymbol{y}^i = \{(e_j^i, y_j^i)\}_{j=1}^{l}, y \in \mathcal{Y}$ is a list of l tuples, where e and y denote contained entity and its entity type, and $\mathcal{Y}$ is a pre-defined entity type set. ConsistNER then feeds $\mathcal{S}$ into LLMs for prediction and outputs a list of l' recognized tuples $\{(e_j, y_j)\}_{j=1}^{l'}$ for each target sentence in $\mathcal{T}$.

The overall architecture of ConsistNER is shown in Figure 2. In the first stage, ConsistNER pre-recognizes all texts from $\mathcal{D}$ and $\mathcal{T}$ in a zero-shot manner to extract local information ($i.e.$, entities and entity types), obtaining the pre-recognized training set $\hat{\mathcal{D}}$ and target sentence set $\hat{\mathcal{T}}$. In the second stage, for the target #i from $\hat{\mathcal{T}}$, ConsistNER first filters demonstrations from $\hat{\mathcal{D}}$ and constructs the candidate set $\mathcal{C}_i$ for it based on the OD representation. Subsequently, we build a dual (target- and train-level) attention mechanism to derive the EAC representation and retrieve target #i-specific demonstrations $\hat{\mathcal{C}}_i$ from $\mathcal{C}_i$. In the third stage, we select the dataset-specific demonstrations $\hat{\mathcal{S}}$ from $\{\hat{\mathcal{C}}_i\}_{i=1}^{|\hat{\mathcal{T}}|}$ based on the occurrences, which are then annotated as $\mathcal{S}$ to prompt LLMs to recognize entities in $\mathcal{T}$.

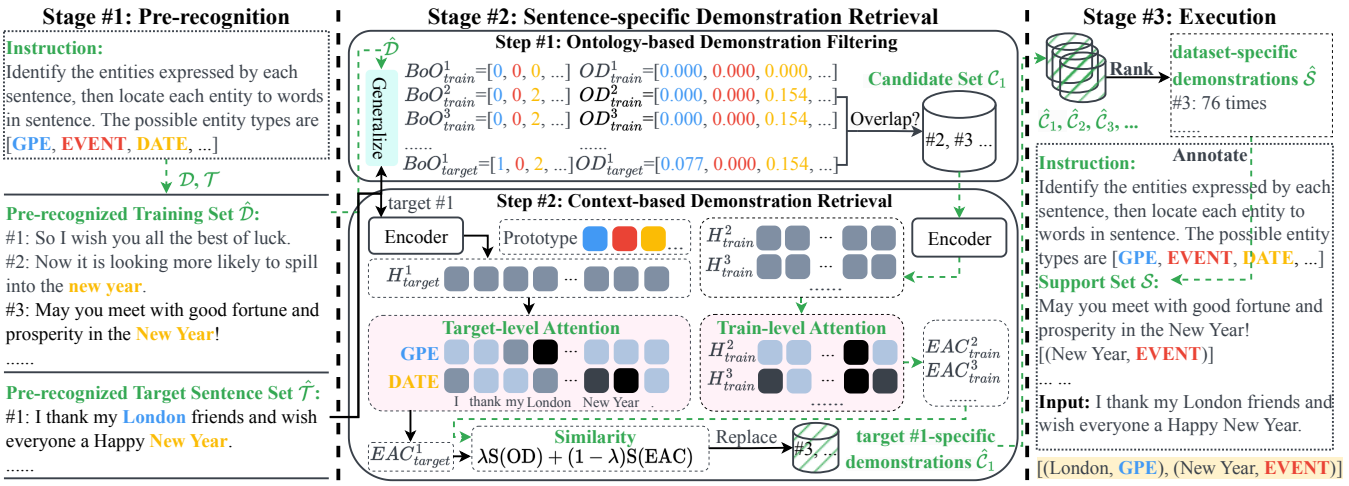


Figure 2: Overview of ConsistNER *w.r.t.* the target sentence *I thank my London friends and wish everyone a Happy New Year*.

Pre-recognition

Previous work viewed NER as a token-level classification task, which emphasized the significance of local information (Yan et al. 2021). Therefore, a high-correlated NER demonstration should be consistent with the target sentence at the token level. Specifically, for NER, the most crucial local information is entities and entity types. To enable the utilization of such information in subsequent stages, we adopt LLMs in a zero-shot manner (*i.e.*, without demonstrations) (Kojima et al. 2022) to identify potential entities in $\mathcal{D}$ and $\mathcal{T}$ to obtain the pre-recognized $\hat{\mathcal{D}}$ and $\hat{\mathcal{T}}$. For example, as shown in Stage #1 in Figure 2, the pre-recognition results of target #1 in $\hat{\mathcal{T}}$ are *London*(*GPE*) and *New Year*(*DATE*). But in fact, *New Year* is an *EVENT* entity representing a traditional festival rather than a *DATE* entity.

Sentence-specific Demonstration Retrieval

With $\hat{\mathcal{D}}$ and $\hat{\mathcal{T}}$ from Stage #1, we can retrieve sentence-specific demonstrations $\hat{\mathcal{C}}_i$ from $\hat{\mathcal{D}}$ for target #i in $\hat{\mathcal{T}}$ considering both ontological and contextual consistency.

Ontology-based Demonstration Filtering Since entity types grouped under the same ontology are often correlated (Roche 2003), we assume that not only demonstrations with the same entity types could assist in recognition, but also demonstrations sharing the same ontology could achieve this. Based on this consideration, we first leverage a pre-defined mapping schema to generalize pre-given entity types into the ontology level. As shown in Step #1 of Stage #2 in Figure 2, generalized pre-given entity types *GPE*, *EVENT* and *DATE* belong to different ontologies represented by different colors.

Subsequently, motivated by the bag of words (BoW) where a text is represented as the bag of its words (Zhang, Jin, and Zhou 2010), we propose the ontology-level BoW, the bag of ontology (BoO), to describe the occurrences of each ontology in a text. As shown in Step #1, the BoO for target #1 is $[1, 0, 2, ...]$. Additionally, with the text length, the ontology distribution (OD) representation φ could be derived by computing the BoO values per unit text length, which denotes the density of each ontology in a text. The OD representation for target #1 is $[0.077, 0.000, 0.154, ...]$.

With the OD representation, for the target #i in $\hat{\mathcal{T}}$, we include training examples from $\hat{\mathcal{D}}$ that have overlapping ontology in the demonstration candidate set $\mathcal{C}_i$. As shown in Step #1, examples #2 and #3 are included in the target #1 candidate set $\mathcal{C}_1$. In this way, each included demonstration could provide assistance in predicting the entities belonging to overlapped ontology. Notably, when the target sentence does not contain any ontology, we also include training examples without ontology in its demonstration candidate set.

Context-based Demonstration Retrieval The vanilla sentence representation (*e.g.*, the CLS embeddings or the mean of all token embeddings (Huang et al. 2021)) treats each token equally, which is not appropriate for NER that pays more attention to the potential entities and semantic-correlated tokens. Hence, we build a dual (train-level and target-level) self-attention mechanism to derive the entity-aware contextual (EAC) representation for training examples and target sentences, respectively.

For train-level attention, we first employ BERT (Devlin et al. 2019) to encode training examples. Given the p^{th} m-token training example $\boldsymbol{x}^p = \{x_1^p, x_2^p, ..., x_m^p\} \in \hat{\mathcal{D}}$, BERT will map all tokens into the hidden embedding representation $\boldsymbol{H}^p = \{\boldsymbol{h}_1^p, \boldsymbol{h}_2^p, ..., \boldsymbol{h}_m^p\}$, where $\boldsymbol{h} \in \mathbb{R}^{d_h}$ is the representation of x and d_h is its dimension.

Based on the pre-recognition, we could derive the raw entity semantic representation by averaging the embeddings of the potential entity span. Given the p^{th} training sample $\boldsymbol{x}^p$ with l potential entities $\{(e_1^p, y_1^p), ..., (e_l^p, y_l^p)\}$, where e_i^p and y_i^p denote the i^{th} entity and its entity type, the raw entity semantic representation $\boldsymbol{h}_{e_i}^p \in \mathbb{R}^{d_h}$ of e_i^p is as follows:

$$\boldsymbol{h}_{e_i}^p = \frac{1}{|\mathcal{M}_i^p|} \sum_{j \in \mathcal{M}_i^p} \boldsymbol{h}_j^p \tag{1}$$

where $\mathcal{M}_i^p$ denotes the index set of the i^{th} entity span.

Since each token in the context contributes differently when recognizing entities, we assume that the tokens semantically correlated to the entity could provide more assistance. Therefore, when computing the sentence representation, we not only pay more attention to the potential entities but also to semantic-correlated tokens. To achieve this, raw entity semantic representation is utilized to sift out the entity-specific information through the sentence attentively (Cong et al. 2022). The entity-specific information $\hat{\boldsymbol{h}_i^p} \in \mathbb{R}^{d_h}$ of e_i^p is as follows:

$$\hat{\boldsymbol{h}_i^p} = \text{softmax}(\frac{\boldsymbol{h}_{e_i}^p \boldsymbol{H}^{pT}}{\sqrt{d_h}})\boldsymbol{H}^p \tag{2}$$

Since a sentence may contain multiple potential entities, we consider the average of the entity-specific information of all potential entities as the EAC representation. The EAC representation $\boldsymbol{\varepsilon}^p \in \mathbb{R}^{d_h}$ of $\boldsymbol{x}^p$ is as follows:

$$\boldsymbol{\varepsilon}^p = \frac{1}{l}\sum_{i=1}^{l} \hat{\boldsymbol{h}_i^p} \tag{3}$$

where l denotes the number of potential entities in $\boldsymbol{x}^p$.

Inspired by prototypical network (Snell, Swersky, and Zemel 2017), we compute the prototype for each entity type by averaging the raw entity semantic representation of that type in the training set $\hat{\mathcal{D}}$. The prototype representation $\boldsymbol{\sigma}^k \in \mathbb{R}^{d_h}$ of the k^{th} entity type is as follows:

$$\boldsymbol{\sigma}^k = \frac{1}{|\mathcal{S}_k|}\sum_{e_i^p \in \mathcal{S}_k} \boldsymbol{h}_{e_i}^p \tag{4}$$

where $\mathcal{S}_k$ is the entity set of the k^{th} entity type in $\hat{\mathcal{D}}$.

For target-level attention, we also employ BERT to encode target sentences. Given the q^{th} m-token target sentence $\boldsymbol{x}^q = \{x_1^q, x_2^q, ..., x_m^q\} \in \hat{\mathcal{T}}$, BERT maps all tokens into the hidden embedding representation $\boldsymbol{H}^q = \{\boldsymbol{h}_1^q, \boldsymbol{h}_2^q, ..., \boldsymbol{h}_m^q\}$.

Given that recognizing entity types is simpler than recognizing entities, we tend to rely more on the entity types pre-recognized by LLMs rather than the entities. Hence, when computing the EAC representation of target sentences, we only utilize the pre-recognized entity types.

Given the q^{th} target sentence $\boldsymbol{x}^q$ with l potential entities $\{(e_1^q, y_1^q), ..., (e_l^q, y_l^q)\}$, if it contains k^{th} entity type y_k, we use the corresponding prototype representation $\boldsymbol{\sigma}^k$ to gather prototype-specific semantics through the sentence attentively. The prototype-specific information $\hat{\boldsymbol{h}_k^q} \in \mathbb{R}^{d_h}$ of $\boldsymbol{\sigma}^k$ is as follows:

$$\hat{\boldsymbol{h}_k^q} = \text{softmax}(\frac{\boldsymbol{\sigma}^k \boldsymbol{H}^{qT}}{\sqrt{d_h}})\boldsymbol{H}^q \tag{5}$$

Similarly, we consider the average of the prototype-specific information of the contained entity types as the EAC representation. The EAC representation $\boldsymbol{\varepsilon}^q \in \mathbb{R}^{d_h}$ of $\boldsymbol{x}^q$ is as follows:

$$\boldsymbol{\varepsilon}^q = \frac{1}{|\mathcal{Y}^q|}\sum_{y_k \in \mathcal{Y}^q} \hat{\boldsymbol{h}_k^q} \tag{6}$$

where $\mathcal{Y}^q$ is the entity type set of $\boldsymbol{x}^q$.

As shown in Step #2 of Stage #2 in Figure 2, when computing the EAC representation of target #1, the pre-recognized entity types *GPE* and *DATE* will pay more attention (deeper color) to *London* and *New Year*, respectively. Eventually, with the OD ($\boldsymbol{\varphi}$) obtained in Step #1 and EAC ($\boldsymbol{\varepsilon}$) representation for each training example and target sentence, we consider the weighted sum of their respective similarities as the similarity between the p^{th} training example and the q^{th} target sentence:

$$\text{sim}^{p,q} = \lambda \text{S}(\boldsymbol{\varphi}^p, \boldsymbol{\varphi}^q) + (1-\lambda)\text{S}(\boldsymbol{\varepsilon}^p, \boldsymbol{\varepsilon}^q) \tag{7}$$

where S denote pre-defined similarity measures ($e.g.$, cosine similarity), and λ is the hyperparameter weight. Accordingly, we can retrieve demonstrations from the candidate set $\mathcal{C}_i$ for the target #i,. As shown in Step #2, example #3 has been retrieved for target #1.

Since all above computations depend on pre-recognition, errors originating from Stage #1 would propagate throughout the entire pipeline. As shown in Stage #1, after LLMs incorrectly pre-recognize *New Year* as a *DATE* entity rather than an *EVENT* entity in target #1, most retrieved demonstrations will contain *DATE* entities. Inevitably, these demonstrations would lead LLMs to predict *DATE*, resulting in inaccurate predictions. Therefore, similar to (Ma et al. 2023; Jimenez Gutierrez et al. 2022), we select top-k semantically similar training examples to replace part of the retrieved demonstrations, serving as the final sentence-specific demonstrations $\hat{\mathcal{C}}_i$.

Execution

After all target sentences have retrieved their own sentence-specific demonstrations $\{\hat{\mathcal{C}}_i\}_{i=1}^{|\hat{\mathcal{T}}|}$, we can not directly annotate them to form the support set $\mathcal{S}$ due to the low-resource settings. Therefore, we select the ones that occur the most frequently per entity type as the dataset-specific demonstrations $\hat{\mathcal{S}}$. Subsequently, we manually annotate these demonstrations to form $\mathcal{S}$, which is then fed into LLMs for prediction. For each target sentence in $\mathcal{T}$, ConsistNER outputs a list of l' recognized tuples $\{(e_j, y_j)\}_{j=1}^{l'}$. As shown in Stage #3 in Figure 2, ConsistNER outputs [(*London*, *GPE*), (*New Year*, *EVENT*)], where the recognition of *New Year* in target #1 has been corrected to *EVENT*.

Experiments

Datasets and Experimental Settings

For the general domain, our experiments consider two datasets. (1) CoNLL2003 (Sang and Meulder 2003) consists of data taken from Reuters news stories. (2) OntoNotes5.0[1] is a large corpus comprising various genres of text, such as news, weblogs, etc. For the specific domain, our experiments consider two datasets. (1) NCBI (Doğan, Leaman, and Lu 2014) consists of 793 PubMed abstracts annotated with disease-related mentions. (2) BC5CDR (Li et al. 2016) consists of 1,500 PubMed articles annotated with chemical

[1]https://catalog.ldc.upenn.edu/LDC2013T19

Model	CoNLL2003		OntoNotes5.0		NCBI		BC5CDR	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
ProtoBERT	49.90±8.6†	61.30±9.1†	20.30±6.9†	36.70±4.1†	17.24‡	34.18‡	23.61‡	40.58‡
NNShot	61.20±10.4†	74.10±2.3†	27.80±9.4†	50.50±4.1†	11.82‡	16.22‡	32.96‡	39.30‡
StructShot	62.40±10.5†	74.80±2.4†	27.90±9.7†	52.90±4.8†	4.63‡	13.89‡	16.09‡	30.97‡
CONTaiNER	61.20±10.7†	75.80±2.7†	32.10±9.8†	56.20±5.0†	16.51‡	26.83‡	37.25‡	41.21‡
COPNER	67.00±3.8†	74.90±2.9†	-	-	15.54‡	24.23‡	36.30‡	42.78‡
VQ + Random	68.57±2.7	73.51±2.3	55.29±1.5	-	25.54±2.3	34.34±4.0	51.59±3.4	59.50±0.9
VQ + CLS	70.66±0.2	75.49±0.1	56.87±0.3	-	27.23±0.1	41.92±0.2	54.10±0.2	60.19±0.1
VQ + ConsistNER	**73.01±0.1**	**78.87±0.1**	**58.47±0.2**	-	34.85±0.2	42.73±0.2	57.98±0.2	61.20±0.1
G-N* + Random	57.26±2.2	63.87±2.0	49.52±1.3	-	28.65±3.9	37.95±3.6	55.88±2.1	60.20±2.0
G-N* + CLS	60.94±0.2	67.94±0.2	50.83±0.5	-	35.15±0.3	41.44±0.4	56.73±0.3	62.65±0.2
G-N* + ConsistNER	61.10±0.3	71.83±0.2	50.95±0.7	-	**36.80±0.2**	**43.87±0.5**	**58.69±0.2**	**63.96±0.3**

Table 1: Performance (%) of different Models on the CoNLL2003, OntoNotes5.0, NCBI and BC5CDR datasets under the 1- and 5-shot settings, where VQ and G-N* denote the vanilla query and GPT-NER*, respectively. The best results are in bold. Due to the 4096 token limit, the 5-shot setting on OntoNotes5.0 is not feasible. Results with † and ‡ are retrieved from original papers and (Li and Zhang 2023), respectively.

and disease mentions. Since the entity types in CoNLL2003 and BC5CDR are sufficiently abstract, we only formulate mapping schemas for OntoNotes5.0 and NCBI, which generalize entity types from the concrete to the abstract. For example, we generalize OntoNotes5.0 entity types *FAC*, *GPE* and *LOC* into the *LOC* ontology.

We employ ChatGPT[2] (gpt-3.5-turbo) as the LLM backbone in our experiments. We adopt two query forms. (1) `Vanilla query` (`VQ`) (Wang et al. 2023b; Mishra et al. 2022; Wang et al. 2022) uses straightforward instructions to recognize entities in a sentence. (2) GPT-NER (Wang et al. 2023a) transforms the sequence labeling task into a generation task, which uses special symbols ($i.e.$, @@ and ##) to mark entities in a sentence. However, for each input, GPT-NER needs to enquire n times, where n denotes the number of entity types, resulting in significant overhead. Hence, similar to (Li and Zhang 2023), we use </type> and <type> to mark entities and only need to enquire 1 time for each input, referred to as `GPT-NER*` (`G-N*`). We adopt two demonstration retrieval techniques to compare with `ConsistNER`. (1) `Random` retrieval strategy indicates randomly selecting k examples from the training set as demonstrations. (2) `CLS` retrieval strategy indicates selecting k nearest neighbor (kNN) (Ma et al. 2023; Jimenez Gutierrez et al. 2022) of the input from the training set as demonstrations, where the distance is measured by the cosine similarity between the CLS embeddings. We use the span-level Micro-F1 score for evaluation and report the mean and associated standard deviation over 5 runs on the full testing set.

Baselines

We choose five representative low-resource NER models as baselines for comparison. `ProtoBERT` (Snell, Swersky, and Zemel 2017) combines prototype-based learning with BERT (Devlin et al. 2019). `NNShot` (Wiseman and Stratos 2019) is a simple method based on token-level nearest neighbor classification. `StructShot` (Yang and Katiyar 2020) adopts an additional Viterbi decoder (Forney 1973) based on NNShot. `CONTaiNER` (Das et al. 2022) leverages contrastive learning to infer the distributional distance of Gaussian embeddings of entities. `COPNER` (Huang et al. 2022) proposes to leverage class-specific words from natural language to serve as the agents of corresponding entity types.

Main Results

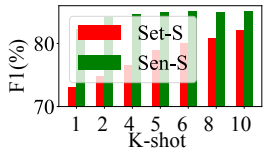
The main results are shown in Table 1. Firstly, regardless of different query forms or demonstration retrieval techniques, LLMs consistently outperform other fine-tuning baselines, which indicates the effectiveness of in-context learning in low-resource scenarios. Specifically, under the 1-shot setting, `ConsistNER` achieves performance gains up to 6.01%, 26.37%, 19.56% and 21.44% over the strongest baselines on the four datasets, respectively. Under the 5-shot setting, `ConsistNER` achieves performance gains up to 3.07%, 9.69% and 21.18% over the strongest baselines on CoNLL2003, NCBI and BC5CDR, respectively. This demonstrates the growing LLMs superiority as annotation resources decrease. Moreover, LLM methods show lower standard deviations than fine-tuning baselines, indicating their stability with limited annotations.

Secondly, for both `VQ` and `G-N*`, `CLS` consistently outperforms `Random` by up to 4.10%, 1.58%, 7.58% and 2.51% on the four datasets, which shows the importance of contextual consistency. And greater improvements can be observed in specific domains, demonstrating that LLMs lack the domain-specific knowledge ($e.g.$, disease and chemical) which can be provided by appropriate demonstrations.

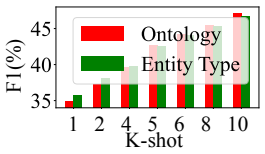
Thirdly, for both `VQ` and `G-N*`, `ConsistNER` consistently outperforms `CLS` by up to 3.89%, 1.60%, 7.62% and 3.88% on the four datasets, which demonstrates the significance of maintaining ontological consistency and focusing more on potential entities and semantic-correlated tokens.

The `VQ` outperforms `G-N*` on CoNLL2003 and OntoNotes5.0, but performs worse on NCBI and BC5CDR. Despite expectations, `G-N*` should consistently excel due to better query form design. However, `VQ` tends to overconfidently label Null tokens as entities. This results in the `VQ` over-recognizing entities and performing poorly on the

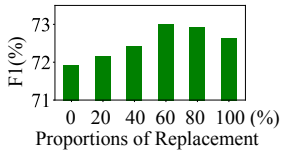
[2]https://openai.com/blog/chatgpt



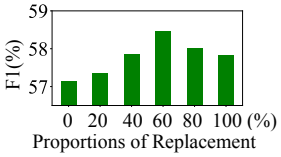
(a) Performance gains with more demonstrations for Set-S and Sen-S on CoNLL2003.



(b) Performance comparison (ontology-based v.s. entity type-based) on NCBI.



(c) The impact of different proportions of replacement on CoNLL2003.



(d) The impact of different proportions of replacement on OntoNotes5.0.

Figure 3: Results of ablation experiments.

Model	CoNLL	OntoNotes	NCBI	BC5CDR
ConsistNER	73.01	58.47	34.85	57.98
w/o OC	71.12	57.75	32.97	57.27
w/o OC & Attn.	70.66	56.87	27.23	54.10
w/o OC & CC	68.57	55.29	25.54	51.59
w/o Repl.	71.93	57.13	33.06	55.15

Table 2: Results (%) of ablation experiments under the 1-shot setting, where OC, Attn., CC and Repl. denote ontological consistency, attention mechanism, contextual consistency and demonstration replacement.

NCBI and BC5CDR which contain fewer entities in a sentence. Unlike `G-N*`, which omits numerous entities, it performs poorly on CoNLL2003 and OntoNotes5.0 due to their high entity density.

Ablation Experiments

Effect of Different Demonstration Numbers We conduct experiments on CoNLL2003 to explore the effect of different demonstration numbers. As shown in Figure 3a, `Set-S` and `Sen-S` represent dataset-specific and sentence-specific demonstrations, respectively. We observe that as the number of demonstrations increases, the performance of both `Set-S` and `Sen-S` is on an upward trend, which is due to the additional information provided by more demonstrations. Notably, after the 5-shot, the `Sen-S` performance is converged, while the `Set-S` performance continues to rise. This phenomenon owes to the fact that the current constraint on `Set-S` performance is no longer insufficient demonstrations, but rather errors within the pre-recognition. These errors result in continually selecting inconsistent demonstrations, thereby preventing further performance improvement.

About Ontological Consistency To verify the effectiveness of maintaining ontological consistency, we remove Step #1 of Stage #2 and set λ in Equation 7 to 0. As shown in Table 2, comparing `ConsistNER` and `w/o OC`, we observe the performance degradation by up to 1.89% on CoNLL2003, which shows the significance of maintaining the consistency of ontology. As mentioned before, we select candidate demonstrations based on ontology overlap. Intuitively, using entity type overlap as the criterion seems more reasonable. As shown in Figure 3b, although using entity type-based criterion has advantages when demonstrations are limited, as the number of demonstrations increases, the ontology-based criterion gradually surpasses it. This is because the former criterion overly emphasizes low-level entity type consistency, thus losing the diversity of retrieved demonstrations or even omitting maintaining contextual consistency. Besides, LLMs often struggle to distinguish entity types belonging to the same ontology. Therefore, demonstrations with overlapping ontology can enhance LLMs' understanding of each entity type under the same ontology and thus reducing incorrect recognition.

About Contextual Consistency To verify the effectiveness of maintaining contextual consistency, we remove the entire Stage #2. As shown in Table 2, comparing `w/o OC` and `w/o CC & OC`, we observe the performance degradation by up to 7.43% on NCBI, which shows the significance of maintaining the consistency of context. Meanwhile, to verify the effectiveness of the self-attention mechanism, we remove Step #1 of Stage #2 and the self-attention mechanism. As shown in Table 2, comparing `w/o OC` and `w/o OC & Attn.`, we observe that even without maintaining ontological consistency, the entity-aware self-attention still improves performance by up to 5.74% on NCBI, which shows importance of paying more attention to the potential entities and semantic-correlated tokens for NER.

About Demonstration Replacement To verify the effectiveness of demonstration replacement, we remove the replacement operation in Stage #2. As shown in Table 2, comparing `ConsistNER` and `w/o Repl.`, we observe the performance degradation by up to 2.83% on BC5CDR, which shows the significance of the replacement. We also study the impact of different proportions of replacement. As shown in Figure 3c and 3d, we observe that for CoNLL2003 and OntoNotes5.0, the suitable replacement proportion is around 60%. A proportion that is too low would hinder the mitigation of error propagation, while too high would limit the effectiveness of the self-attention mechanism.

Analysis Experiments

Set-S and Sen-S To investigate the performance influenced by `Set-S` and `Sen-S`, we conduct experiments to compare their performance. As shown in Table 3, `Set-S` consistently underperforms `Sen-S`, with performance degradation as high as 14.54%, 10.67%, 12.26%, and 9.22% on four datasets, respectively. This shows that each target sentence has its own high-correlated demonstrations that maintain ontological and contextual consistency, while `Set-S` needs to balance all target sentences, unavoidably leading to a decrease in performance.

Theoretical Boundary As mentioned earlier, the pre-recognition quality could affect demonstration retrieval. In

Model	CoNLL2003		OntoNotes5.0		NCBI		BC5CDR	
	Set-S	Sen-S	Set-S	Sen-S	Set-S	Sen-S	Set-S	Sen-S
VQ + CLS	70.66±0.2	80.14±0.1	56.87±0.3	59.97±0.3	27.23±0.1	34.07±0.2	54.10±0.2	59.12±0.2
VQ + ConsistNER	73.01±0.1	82.22±0.1	58.47±0.2	64.21±0.4	34.85±0.2	38.21±0.1	57.98±0.2	62.48±0.1
G-N* + CLS	60.94±0.2	70.80±0.0	50.83±0.5	53.40±0.3	35.15±0.3	40.77±0.1	56.73±0.3	59.72±0.1
G-N* + ConsistNER	61.10±0.3	75.64±0.1	50.95±0.7	61.62±0.6	36.80±0.2	49.06±0.1	58.69±0.2	67.91±0.2

Table 3: Performance (%) comparison between Set-S and Sen-S under the 1-shot setting, where Set-S and Sen-S denote set-specific and sentence-specific demonstrations, respectively. The rest abbreviations herein are the same as in Table 1.

Model	CoNLL2003		OntoNotes5.0		NCBI		BC5CDR	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
VQ + ER	73.01	78.87	58.47	-	34.85	42.73	57.98	61.20
VQ + TB	74.80(+1.79)	81.00(+2.13)	64.11(+5.64)	-	38.80(+3.95)	46.09(+3.36)	61.41(+3.43)	64.70(+3.50)
G-N* + ER	61.10	71.83	50.95	-	36.80	43.87	58.69	63.96
G-N* + TB	63.11(+2.01)	73.86(+2.03)	55.81(+4.86)	-	42.29(+5.49)	49.67(+5.80)	62.44(+3.75)	67.06(+3.10)

Table 4: Performance (%) comparison between ER and TB under the 1- and 5-shot settings, where ER and TB denote empirical results and theoretical boundaries of ConsistNER, respectively. The rest abbreviations herein are the same as in Table 1.

light of this, we replace the pre-recognition results with the ground truth, enabling the retrieved demonstrations to maximally emerge the ICL capabilities of LLMs, leading to the achievement of the theoretical performance boundaries. As shown in Table 4, we can observe that, regardless of datasets, number of demonstrations or query forms, the theoretical boundaries (TB) of ConsistNER are consistently 1.79%-5.64% above the empirical results (ER) of ConsistNER. Specifically, we notice that the performance improvement is more significant on OntoNotes5.0 compared to other datasets. This is because OntoNotes5.0 contains more entity types, thereby posing more challenges for LLMs to recognize entities. Accordingly, the pre-recognition on OntoNotes5.0 will include more errors, leading to more inappropriate demonstrations being retrieved to misguide LLMs. Thus, with the improved pre-recognition quality, there is a substantial performance boost on OntoNotes5.0. Given the TB, we could approximate it by improving pre-recognition quality. For example, we could use pre-built lexicons (Chiu and Nichols 2016; Collobert et al. 2011) or gazetteers (Liu, Yao, and Lin 2019) in pre-recognition to retrieve more instructive demonstrations.

Advantages of Ontology The major innovative advantages of ontology are in two aspects: generalization in concept hierarchy and generalization in diverse tasks. (1) *Generalization in concept hierarchy.* During pre-training, LLMs primarily encounter general knowledge while lacking domain-specific knowledge, which leads to their inability to differentiate excessively fine-grained entity types. In this case, pre-recognition which outputs entity types has a low accuracy. Inspired by the concept hierarchy of ontology in knowledge graphs, we generalize entity types upwards to ontology (e.g., Biden: president→official→person) and require pre-recognition only to output ontology. Thus pre-recognition is aligned with the pre-training tasks of LLMs and provides more solid prior knowledge for later stages. (2) *Generalization in diverse tasks.* When dealing with other information extraction tasks like relation extraction, properties and property hierarchy of ontology (e.g., relations and generalized relations) can also be useful. This open-ended question remains to be further explored in future research.

A Chaos Phenomenon Furthermore, we have discovered a chaos phenomenon, where erroneously pre-recognized training examples could still assist LLMs. For example, after both *Orlando* from the target sentence *LA LAKERS 92 Orlando 81* and *MINNESOTA* from the training example *Texas 13 MINNESOTA 2* have been erroneously pre-recognized as *LOC* entities, this training example will be retrieved as a demonstration for its both ontological and contextual consistency with the target sentence. However, the labels in demonstrations are correct, $i.e.$, *MINNESOTA* as an *ORG*, which could correct the misrecognition of *Orlando*. The superficial reason behind this phenomenon is that LLMs can pre-recognize *Orlando* and *MINNESOTA* as the same type of entity, but they cannot distinguish whether they are *LOC* or *ORG* entities. Essentially, for pre-recognition, ConsistNER aims to identify the fine-grained text information used to retrieve high-correlated demonstrations, rather than simply type entities. Therefore, the accuracy of pre-recognition is not crucial, as long as it labels entities of the same type with the same label. Thus this chaos phenomenon reveals the robustness of ConsistNER.

Conclusion

In this paper, we argue that the key to emerging ICL ability lies in how to retrieve high-correlated demonstrations, where correlation implies the consistency of both ontology and context for NER. Based on this motivation, we propose a three-stage framework, ConsistNER, to incorporate ontological and contextual information for low-resource NER. In this way, the retrieved demonstrations could maintain both ontological and contextual consistency. We conduct experiments on both general and specific domains, which demonstrates that ConsistNER not only effectively instructs LLMs, but also boosts NER with state-of-the-art performance.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. This work was supported by National Science Foundation of China (Grant Nos.62376057) and the Start-up Research Fund of Southeast University (RF1028623234). All opinions are of the authors and do not reflect the view of sponsors.

References

Brown, T.; Mann, B.; Ryder, N.; et al. 2020. Language Models Are Few-shot Learners. In *NeurIPS*.

Chen, X.; Li, L.; Deng, S.; et al. 2022. LightNER: A Lightweight Tuning Paradigm for Low-resource NER via Pluggable Prompting. In *COLING*.

Chiu, J. P.; and Nichols, E. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. In *TACL*.

Collobert, R.; Weston, J.; Bottou, L.; et al. 2011. Natural Language Processing (almost) from Scratch. *JMLR*, 12(2011): 2493–2537.

Cong, X.; Sheng, J.; Cui, S.; et al. 2022. Relation-guided Few-shot Relational Triple Extraction. In *SIGIR*.

Das, S. S. S.; Katiyar, A.; Passonneau, R. J.; and Zhang, R. 2022. CONTaiNER: Few-shot Named Entity Recognition via Contrastive Learning. In *ACL*.

de Lichy, C.; Glaude, H.; and Campbell, W. 2021. Meta-learning for Few-shot Named Entity Recognition. In *MetaNLP*.

Devlin, J.; Chang, M.-W.; Lee, K.; et al. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*.

Doğan, R. I.; Leaman, R.; and Lu, Z. 2014. NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47: 1–10.

Du, N.; Huang, Y.; Dai, A. M.; et al. 2022. Glam: Efficient Scaling of Language Models with Mixture-of-experts. In *ICML*.

Forney, G. D. 1973. The Viterbi Algorithm. *Proceedings of IEEE*, 61(3): 268–278.

Huang, J.; Tang, D.; Zhong, W.; et al. 2021. WhiteningBERT: An Easy Unsupervised Sentence Embedding Approach. In *Findings of EMNLP*.

Huang, Y.; He, K.; Wang, Y.; et al. 2022. Copner: Contrastive Learning with Prompt Guiding for Few-shot Named Entity Recognition. In *COLING*.

Jimenez Gutierrez, B.; McNeal, N.; Washington, C.; Chen, Y.; Li, L.; Sun, H.; and Su, Y. 2022. Thinking about GPT-3 In-Context Learning for Biomedical IE? Think Again. In *Findings of EMNLP*.

Kojima, T.; Gu, S. S.; Reid, M.; et al. 2022. Large Language Models Are Zero-shot Reasoners. In *NeurIPS*.

Kolomiyets, O.; and Moens, M.-F. 2011. A Survey on Question Answering Technology from an Information Retrieval Perspective. *Information Sciences*, 181(24): 5412–5434.

Lee, D.-H.; Kadakia, A.; Tan, K.; et al. 2022. Good Examples Make A Faster Learner: Simple Demonstration-based Learning for Low-resource NER. In *ACL*.

Li, J.; Sun, Y.; Johnson, R. J.; Sciaky, D.; Wei, C.-H.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Wiegers, T. C.; and Lu, Z. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Li, M.; and Zhang, R. 2023. How far is Language Model from 100% Few-shot Named Entity Recognition in Medical Domain. *arXiv preprint arXiv:2307.00186*.

Liu, A. T.; Xiao, W.; Zhu, H.; Zhang, D.; Li, S.-W.; and Arnold, A. 2022. QaNER: Prompting question answering models for few-shot named entity recognition. *arXiv preprint arXiv:2203.01543*.

Liu, T.; Yao, J.-G.; and Lin, C.-Y. 2019. Towards Improving Neural Named Entity Recognition with Gazetteers. In *ACL*.

Ma, R.; Zhou, X.; Gui, T.; et al. 2022a. Template-free Prompt Tuning for Few-shot NER. In *NAACL*.

Ma, T.; Jiang, H.; Wu, Q.; et al. 2022b. Decomposed Meta-learning for Few-shot Named Entity Recognition. In *Findings of ACL*.

Ma, Y.; Cao, Y.; Hong, Y.; and Sun, A. 2023. Large language model is not a good few-shot information extractor, but a good reranker for hard samples! *arXiv preprint arXiv:2303.08559*.

Mishra, S.; Khashabi, D.; Baral, C.; et al. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *ACL*.

Peters, M. E.; Ammar, W.; Bhagavatula, C.; and other. 2017. Semi-supervised Sequence Tagging with Bidirectional Language Models. In *ACL*.

Petroni, F.; Rocktäschel, T.; Lewis, P.; et al. 2019. Language Models as Knowledge Bases? In *EMNLP*.

Roche, C. 2003. Ontology: A Survey. *IFAC Proceedings Volumes*, 36(22): 187–192.

Sang, E. F. T. K.; and Meulder, F. D. 2003. Introduction to the CoNLL-2003 Shared Task: Language-independent Named Entity Recognition. In *NAACL*.

Schick, T.; and Schütze, H. 2021a. Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference. In *EACL*.

Schick, T.; and Schütze, H. 2021b. Few-shot Text Generation with Pattern-exploiting Training. In *EMNLP*.

Schick, T.; and Schütze, H. 2021c. It's Not Just Size That Matters: Small Language Models Are Also Few-Shot Learners. In *NAACL*.

Sevgili, O.; and Shelmanov, A. 2020. Neural Entity Linking: A Survey of Models Based on Deep Learning. *Semantic Web*, 13(3).

Shin, R.; Lin, C. H.; Thomson, S.; et al. 2021. Constrained Language Models Yield Few-Shot Semantic Parsers. In *EMNLP*.

Smith, S.; Patwary, M.; Norick, B.; LeGresley, P.; Rajbhandari, S.; Casper, J.; Liu, Z.; Prabhumoye, S.; Zerveas, G.; Korthikanti, V.; et al. 2022. Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. *arXiv preprint arXiv:2201.11990*.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical Networks for Few-shot Learning. In *NeurIPS*.

Souza, F.; Nogueira, R.; and Lotufo, R. 2019. Portuguese named entity recognition using BERT-CRF. *arXiv preprint arXiv:1909.10649*.

Wang, S.; Sun, X.; Li, X.; Ouyang, R.; Wu, F.; Zhang, T.; Li, J.; and Wang, G. 2023a. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*.

Wang, X.; Zhou, W.; Zu, C.; Xia, H.; Chen, T.; Zhang, Y.; Zheng, R.; Ye, J.; Zhang, Q.; Gui, T.; et al. 2023b. InstructUIE: Multi-task Instruction Tuning for Unified Information Extraction. *arXiv preprint arXiv:2304.08085*.

Wang, Y.; Mishra, S.; Alipoormolabashi, P.; et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *EMNLP*.

Wei, X.; Cui, X.; Cheng, N.; Wang, X.; Zhang, X.; Huang, S.; Xie, P.; Xu, J.; Chen, Y.; Zhang, M.; et al. 2023. Zero-shot information extraction via chatting with chatgpt. *arXiv preprint arXiv:2302.10205*.

Wiseman, S.; and Stratos, K. 2019. Label-agnostic Sequence Labeling by Copying Nearest Neighbors. In *ACL*.

Wu, Q.; Lin, Z.; Wang, G.; et al. 2020. Enhanced Meta-learning for Cross-lingual Named Entity Recognition with Minimal Resources. In *AAAI*.

Xiang, W.; and Wang, B. 2019. A Survey of Event Extraction from Text. *IEEE Access*, 7: 173111–173137.

Yan, H.; Gui, T.; Dai, J.; et al. 2021. A Unified Generative Framework for Various NER Subtasks. In *ACL*.

Yang, Y.; and Katiyar, A. 2020. Simple and Effective Few-shot Named Entity Recognition with Structured Nearest Neighbor Learning. In *EMNLP*.

Zhang, Y.; Jin, R.; and Zhou, Z.-H. 2010. Understanding Bag-of-words Model: A Statistical Framework. *JMLC*, 1: 43–52.