Simple Questions Generate Named Entity Recognition Datasets

Hyunjae Kim[1] **Jaehyo Yoo**[1] **Seunghyun Yoon**[2] **Jinhyuk Lee**[1*] **Jaewoo Kang**[1,3]
[1]Korea University [2]Adobe Research [3]AIGEN Sciences
`{hyunjae-kim,jaehyoyoo,jinhyuk_lee,kangj}@korea.ac.kr`
`syoon@adobe.com`

Abstract

Recent named entity recognition (NER) models often rely on human-annotated datasets, requiring the significant engagement of professional knowledge on the target domain and entities. This research introduces an *ask-to-generate* approach that automatically generates NER datasets by asking questions in simple natural language to an open-domain question answering system (e.g., "*Which disease?*"). Despite using fewer in-domain resources, our models, solely trained on the generated datasets, largely outperform strong low-resource models by an average F1 score of 19.4 for six popular NER benchmarks. Furthermore, our models provide competitive performance with rich-resource models that additionally leverage in-domain dictionaries provided by domain experts. In few-shot NER, we outperform the previous best model by an F1 score of 5.2 on three benchmarks and achieve new state-of-the-art performance. The code and datasets are available at https://github.com/dmis-lab/GeNER.

1 Introduction

Named entity recognition (NER) is the task of extracting named entities of specific types from text. An NER dataset essentially reflects the need to extract specific entity types. For instance, NCBI-disease (Doğan et al., 2014) was created for extracting *disease* entities from text. Recent NER models have provided robust performance when trained on carefully designed human-annotated datasets (Lample et al., 2016; Li et al., 2020; Lee et al., 2020). However, suppose that we want to build an NER model for extracting *bacteria* or other specific types for which human-annotated datasets are insufficient. Whenever we extract such entity types, should we rely on professional knowledge to create new datasets?

Previous weakly supervised NER models (Shang et al., 2018b; Liang et al., 2020) tackled this problem using rich in-domain dictionaries (e.g., The Comparative Toxicogenomics Database) and unlabeled training sentences (i.e., in-domain sentences), where entities in the dictionary are used to annotate the training sentences. However, these approaches easily fail in practice because in-domain dictionaries and sentences are often unavailable or expensive to construct for many entity types. It will be challenging to build NER models for *enzyme* or *astronomical object* entities without expert-level knowledge required for building dictionaries and searching for a large number of in-domain sentences for annotation.

In this study, we introduce GeNER, an automated dataset **Ge**neration framework for **NER**, which automatically constructs high-quality NER datasets. In particular, concrete needs for NER are described using simple natural language questions such as "*Which* `[TYPE]`*?*", where `[TYPE]` is substituted by the required entity type (e.g., "*Which disease?*"). Such questions do not require professional knowledge of the target domain and allow even non-experts to easily build domain-specific NER datasets. Using a phrase retrieval model designed for open-domain question answering (QA) (Lee et al., 2021a), GeNER first retrieves candidate entities (i.e., phrases) and evidence sentences from a large-scale open-domain corpus (e.g., Wikipedia). The retrieved entities form a pseudo-dictionary, which is used to annotate the evidence sentences to create the dataset. We then train standard NER models on our generated dataset using a recent self-training method (Liang et al., 2020). As shown in Table 1, this type of *ask-to-generate* approach significantly reduces dependency on the in-domain resources while outperforming the strong low-resource model, TALLOR (Li et al., 2021), and being comparable with the rich-resource model, BOND (Liang et al., 2020).

[*] JL currently works at Google Research. The collaboration started before he joined Google.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6220–6236
December 7-11, 2022 ©2022 Association for Computational Linguistics

Model	**Resource** $\mathbf{X}_{\text{train}}$ / $\mathbf{Y}_{\text{train}}$ / $\mathcal{V}$	**Wikigold**	**NCBI-disease**
Rich-resource models (w/ training label or in-domain dict.)			
Fully supervised	✓ / ✓ / ✗	86.8	88.6
BOND	✓ / ✗ / ✓	59.8	71.4
Low-resource models			
TALLOR	✓ / ✗ / △	30.3	44.3
GeNER (**ours**)	✗ / ✗ / ✗	72.5	67.9

Table 1: Comparison of existing approaches in NER. Each method is categorized based on how much it relies on in-domain resources during training. $\mathbf{X}_{\text{train}}$: (unlabeled) training sentences. $\mathbf{Y}_{\text{train}}$: human-annotated training labels. $\mathcal{V}$: in-domain dictionaries by domain experts. In-domain resources are either fully used (✓), partially used (△), or not used (✗). While using the fewest in-domain resources, GeNER shows strong performance on various domains and entity types.

We demonstrate the effectiveness of GeNER using six popular NER benchmarks across four domains: news (Tjong Kim Sang and De Meulder, 2003), Wikipedia (Balasuriya et al., 2009), Twitter (Strauss et al., 2016), and biomedicine (Doğan et al., 2014; Li et al., 2016; Krallinger et al., 2015). Models solely trained on our generated datasets from GeNER significantly outperformed TALLOR on all benchmarks by an average F1 score of 19.4. Although our models did not use rich in-domain dictionaries, they sometimes outperformed the previous best weakly supervised model, BOND (Liang et al., 2020), on two benchmarks. Moreover, GeNER achieved new state-of-the-art results on three few-shot NER benchmarks, outperforming the previous best model QUIP (Jia et al., 2022) by an F1 score of 5.2. Finally, we conducted extensive ablation studies and analyses to highlight important factors for low-resource NER.

Our contributions are summarized as follows:

- To the best of our knowledge, GeNER is the first attempt to automatically generate NER datasets for various low-resource domains using a general-purpose QA system.

- GeNER significantly reduces the dependency on in-domain training resources required by previous weakly supervised models, such as human-annotated labels, in-domain dictionaries, and in-domain sentences.

- GeNER outperformed the strong baseline model TALLOR by an F1 score of 19.4 on six benchmarks. In few-shot NER, GeNER outperformed the previous best model QUIP by an F1 score of 5.2 on three benchmarks, thereby achieving new state-of-the-art results.

2 Background

2.1 Named Entity Recognition

NER aims to identify named entities of pre-defined types in text. Let $\mathcal{D} = \{\mathbf{X}, \mathbf{Y}\}$ be a dataset, where $\mathbf{X} = \{x_n\}_{n=1}^{N}$ is the set of unlabeled sentences, $\mathbf{Y} = \{y_n\}_{n=1}^{N}$ is the set of corresponding token-level labels[1] for each sentence, and N is the size of the dataset. In supervised learning, $\mathcal{D}$ is split into $\mathcal{D}_{\text{train}} = \{\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}\}$, $\mathcal{D}_{\text{valid}} = \{\mathbf{X}_{\text{valid}}, \mathbf{Y}_{\text{valid}}\}$, and $\mathcal{D}_{\text{test}} = \{\mathbf{X}_{\text{test}}, \mathbf{Y}_{\text{test}}\}$, which are then used to train NER models, select hyperparameters, and evaluate the models, respectively.

Weakly supervised NER Instead of using human-annotated labels $\mathbf{Y}_{\text{train}}$, weakly supervised NER models rely on in-domain dictionaries $\mathcal{V}$ built by domain experts (Yang et al., 2018; Shang et al., 2018b; Cao et al., 2019; Yang and Katiyar, 2020; Liang et al., 2020). In-domain dictionaries are used to generate weak labels $\hat{\mathbf{Y}}_{\text{train}}$ for (unlabeled) training sentences $\mathbf{X}_{\text{train}}$ by annotating any occurrences of named entities from the dictionary. Models are then trained on $\hat{\mathcal{D}}_{\text{train}} = \{\mathbf{X}_{\text{train}}, \hat{\mathbf{Y}}_{\text{train}}\}$ and evaluated on $\mathcal{D}_{\text{test}}$. Instead of relying on training resources such as $\mathbf{X}_{\text{train}}, \mathbf{Y}_{\text{train}}$, and $\mathcal{V}$, we propose to automatically generate a new dataset $\tilde{\mathcal{D}}_{\text{train}} = \{\tilde{\mathbf{X}}_{\text{train}}, \tilde{\mathbf{Y}}_{\text{train}}\}$ with minimal human effort by asking simple questions to the QA model.

2.2 Open-domain Question Answering

Open-domain QA finds answers from a large-scale corpus that is not limited to specific domains (Chen et al., 2017). Among several other open-domain QA approaches, Seo et al. (2019) proposed *phrase retrieval*, which formulates question answering as retrieving pre-indexed phrases, significantly beneficial to scalability because the similarity search can be efficiently implemented. Recent advancements in phrase retrieval (Lee et al., 2021a) have designed the retrieval purely with dense vectors as follows:

$$\begin{aligned} &\mathbf{s} = E_s(s, \mathcal{W}), \quad \mathbf{q} = E_q(q), \\ &s^* = \operatorname*{arg\,max}_{s \in \mathcal{S}(\mathcal{W})} (\mathbf{s}^\top \mathbf{q}), \end{aligned} \tag{1}$$

where $\mathcal{W}$ is a large-scale open-domain corpus; $\mathcal{S}(\mathcal{W})$ is the set of all phrases in $\mathcal{W}$; q is the input

[1] We follow the BIO tagging (Ramshaw and Marcus, 1995).

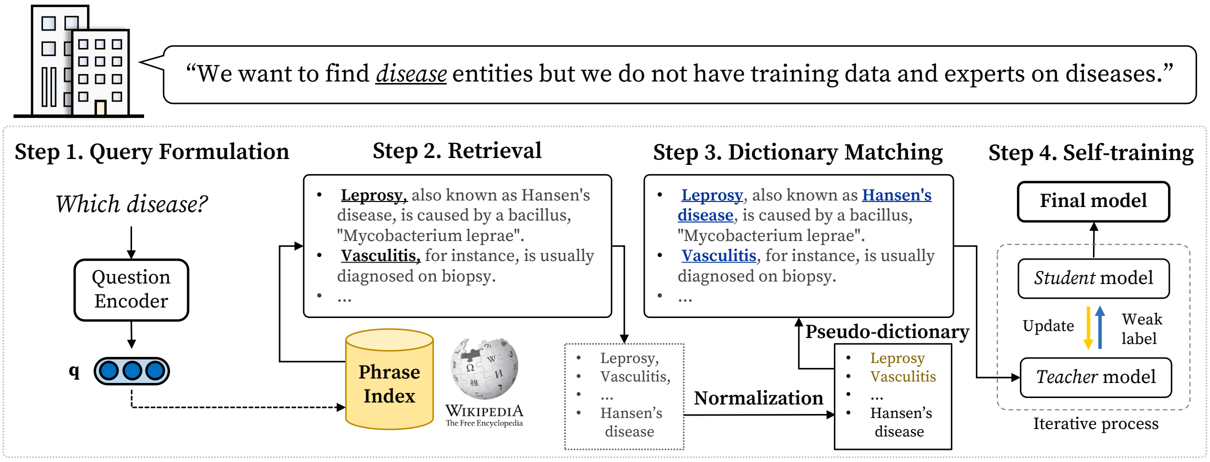

"We want to find disease entities but we do not have training data and experts on diseases."
Step 1. Query Formulation
Which disease?
Question Encoder
q
Step 2. Retrieval
Leprosy, also known as Hansen's disease, is caused by a bacillus, "Mycobacterium leprae".
Vasculitis, for instance, is usually diagnosed on biopsy.
...
Phrase Index
Wikipedia
The Free Encyclopedia
Leprosy,
Vasculitis,
...
Hansen's disease
Normalization
Step 3. Dictionary Matching
Leprosy, also known as Hansen's disease, is caused by a bacillus, "Mycobacterium leprae".
Vasculitis, for instance, is usually diagnosed on biopsy.
...
Pseudo-dictionary
Leprosy
Vasculitis
...
Hansen's disease
Step 4. Self-training
Final model
Student model
Update
Weak label
Teacher model
Iterative process

Figure 1: Overview of GeNER. Given the needs of extracting disease entities from text, GeNER automatically generates an NER dataset for the disease entities without resorting to human-annotated training data or domain experts. (1) Query formulation: the needs for disease NER are first formulated as simple natural language questions. (2) Retrieval: we use an open-domain QA model to retrieve relevant phrases (i.e., entities) as well as sentences to annotate. (3) Dictionary matching: retrieved sentences are annotated by the normalized phrases. (4) Self-training: we train NER models purely on our generated dataset using self-training. See Section 3 for more details.

question; and E_q and E_s are question and phrase encoders, respectively, that produce dense vectors $\mathbf{q}$ and $\mathbf{s}$, respectively. The candidate answer s^* can be retrieved along with an evidence document or sentence because of its extractive nature (Lee et al., 2021b). In this study, we used the phrase retrieval model DensePhrases (Lee et al., 2021a) as our open-domain QA model because of its strong open-domain QA accuracy and run-time efficiency while retrieving thousands of phrases and sentences to construct NER datasets. We use evidence sentences that contain each s^* as our training sentences $\tilde{\mathbf{X}}_{\text{train}}$ and leverage the retrieved phrases s^* to generate the weak labels $\tilde{\mathbf{Y}}_{\text{train}}$.

3 Method

This section describes GeNER, which automatically generates NER datasets from scratch and provides NER models trained on these datasets. Figure 1 provides an overview of the GeNER.

3.1 Query Formulation

In GeNER, we first formulate the need to recognize specific types of named entities as natural language questions. For instance, we ask, "*Which disease?*" to extract disease entities. Based on the ablation study in Section 5.1, we provide a template "*Which* [TYPE]*?*," where [TYPE] is substituted by an entity type. An input question q is then encoded as question vector $\mathbf{q}$ as shown in Equation (1). In real-world scenarios, practitioners have a small-sized entity-type ontology or list that defines entity types of interest; therefore, they can easily determine query terms (i.e., [TYPE]) with minimal effort.

It should be noted that clear needs should be reflected in our questions. For instance, rather than a single, broad, unspecified question, such as "*Which organization?*", multiple distinct sub-questions such as "*Which sports team?*" or "*Which company?*", that represent the target entities specifically would be a better option (see Section 5.1.).

3.2 Retrieval

Based on the question vector for each sub-question, we mine the unlabeled training sentences, $\tilde{\mathbf{X}}_{\text{train}}$, and generate the pseudo-dictionary, $\tilde{\mathcal{V}}$. Given L sub-questions for all entity types, we use DensePhrases to retrieve relevant phrases and evidence sentences. While other domain-specific corpora, such as PubMed, can be used for domain-specific NER, we use Wikipedia[2] as our corpus for retrieval because it covers many different domains, making GeNER generally applicable.

Normalization We retrieve the top phrases, s^*, for each sub-question and normalize them to refine their spans. The set of normalized phrases comprises the pseudo-dictionary, $\tilde{\mathcal{V}}$. Normalization rules are required because of different annotation guidelines of the datasets or the inherent characteristics of different entity types. For instance, phrases containing the conjunction "and" should be split into two different phrases for CoNLL-2003,

[2] We use the 2018-12-20 Wikipedia snapshot.

Domain	Dataset (# Types)	Training # Sents	 # Labels	Validation # Sents	 # Labels	Test # Sents	 # Labels
News	CoNLL-2003 (3)	14,987	20,061	3,469	5,022	3,685	4,947
Wikipedia	Wikigold (3)	1,142	1,842	280	523	274	484
Twitter	WNUT-16 (9)	2,394	1,271	1,000	529	3,850	2,889
Biomedicine	NCBI-disease (1)	5,432	5,134	923	787	942	960
	BC5CDR (2)	4,582	9,387	4,602	9,596	4,812	9,809
	CHEMDNER (1)	30,884	29,530	30,841	29,543	26,561	25,388

Table 2: Statistics of NER datasets. # Types: number of entity types. # Sents: number of sentences. # Labels: number of entity-level annotations. Note that we do not use any training sentences and labels from benchmarks but only the generated dataset from GeNER in the low-resource NER experiment (Section 4.1).

whereas this rule should not be applied to biomedical NER datasets because such phrases are considered a single *composite mention*. Furthermore, some entity types (e.g., song) can begin with the article "the," whereas other entity types usually have no article in their names. Therefore, we provide ten simple rules (Appendix A) generally applicable to most entity types. Practically, these rules can be determined by practitioners in a top-down manner or treated as hyperparameters.

Training sentences We obtain the top k_l unique sentences for each sub-question that contains each phrase s^*. We gather $k_1 + \cdots + k_L$ sentences in total, which we consider as our unlabeled training sentences $\tilde{\mathbf{X}}_{\text{train}}$. We do not have a hyperparameter for the size of the pseudo-dictionary, $\tilde{\mathcal{V}}$, and the size changes based on the number of unique sentences to be retrieved.[3]

3.3 Dictionary Matching

This stage annotates the unlabeled sentences, $\tilde{\mathbf{X}}_{\text{train}}$, and generates $\tilde{\mathbf{Y}}_{\text{train}}$ using pseudo-dictionary $\tilde{\mathcal{V}}$ to prevent possible false negatives that might occur while annotating only initially retrieved phrases as entities. For every phrase (or entity) in $\tilde{\mathcal{V}}$, every occurrence of the phrase in $\tilde{\mathbf{X}}_{\text{train}}$ is annotated to generate $\tilde{\mathbf{Y}}_{\text{train}}$. If a phrase has more than two entity types, the resulting label ambiguity is dealt with using the probability of the phrase appearing with a particular entity type. For instance, if "Washington" is retrieved three times for the location type and seven times for the person type, we annotate 30% of all occurrences with location and the remaining 70% with person. Finally, we obtain the training data, $\tilde{\mathcal{D}}_{\text{train}} = \{\tilde{\mathbf{X}}_{\text{train}}, \tilde{\mathbf{Y}}_{\text{train}}\}$, which reflects our specific needs.

3.4 Self-training

As the weak labels generated by dictionary matching are often noisy and incomplete, directly training NER models using $\tilde{\mathcal{D}}_{\text{train}}$ is not optimal. Therefore, we train our models using a current self-training method (Liang et al., 2020) to mitigate noise. First, we initialize *teacher* and *student* models using the same pre-trained weights of a language model. We train the teacher model with the generated $\tilde{\mathcal{D}}_{\text{train}}$ for T_{begin} steps. The teacher model then re-annotates $\tilde{\mathbf{X}}_{\text{train}}$ and the student model is trained on the re-annotated corpus. The teacher model is replaced with the student model for every T_{update} step, where T_{update} denotes the update period. This process is iterated until the maximum epoch is reached. We use the student model with the best validation F1 score during the process as the final NER model. It should be noted that GeNER is a model-agnostic framework; therefore, other recent techniques (Liu et al., 2021a; Meng et al., 2021) can be adopted to correct mislabeling.

4 Experiments

We evaluated GeNER in two scenarios wherein training resources are scarce: low-resource NER (Section 4.1) and few-shot NER (Section 4.2). We used entity-level precision (P), recall (R), and F1 score (F1) as the evaluation metrics.

Query term selection Evaluating our models requires investigating the needs inherent in the benchmark datasets. In most NER benchmarks, entity types are coarsely defined and can be classified into many subtypes. These subtypes are often defined differently based on the datasets. For instance, the organization type of CoNLL-2003 mostly includes sports teams and companies, but that of Wikigold additionally covers bands. Therefore, to understand the needs of each NER dataset, we sampled 100 ex-

[3] If we have more than two phrases contained in the same sentence, we retain all of them in $\tilde{\mathcal{V}}$.

Model	Resource	CoNLL-2003			Wikigold			WNUT-16		
	$\mathbf{X}_{\text{train}}$ / $\mathbf{Y}_{\text{train}}$ / $\mathcal{V}$	P	R	F1	P	R	F1	P	R	F1
Rich-resource models (w/ training label or in-domain dict.)										
RoBERTa (Liu et al., 2019)	✓ / ✓ / ✗	93.1	93.9	93.5	86.5	87.2	86.8	57.3	60.8	59.0
BOND (Liang et al., 2020)	✓ / ✗ / ✓	80.6	82.4	81.5	58.2	61.5	59.8	47.0	48.4	47.7
Low-resource models										
Seed Entities	✗ / ✗ / △	**95.1***	2.8	5.3	**90.5***	3.9	7.5	**67.7***	1.5	2.9
Neural Tagger	✓ / ✗ / △	71.8	13.6	22.9	58.8*	4.1	7.7	0.5	7.4	1.0
Self-training	✓ / ✗ / △	43.0	31.6	36.4	32.8	17.4	22.7	25.0	19.6	22.3
TALLOR† (Li et al., 2021)	✓ / ✗ / △	64.3	64.1	64.2	-	-	-	-	-	-
TALLOR (Li et al., 2021)	✓ / ✗ / △	59.3	58.4	60.2	35.0	26.8	30.3	32.0	23.7	27.2
GeNER (**ours**)	✗ / ✗ / ✗	73.1	**69.0**	**71.0**	65.8*	**79.9***	**72.5***	44.8	**54.0***	**48.5***

Model	Resource	NCBI-disease			BC5CDR			CHEMDNER		
	$\mathbf{X}_{\text{train}}$ / $\mathbf{Y}_{\text{train}}$ / $\mathcal{V}$	P	R	F1	P	R	F1	P	R	F1
Rich-resource models (w/ training label or in-domain dict.)										
BioBERT (Lee et al., 2020)	✓ / ✓ / ✗	86.6	90.5	88.5	86.7	90.5	88.6	91.4	91.1	91.2
BOND (Liang et al., 2020)	✓ / ✗ / ✓	87.5	60.3	71.4	81.0	80.3	80.6	-	-	-
Low-resource models										
Seed Entities	✗ / ✗ / △	**88.8***	10.7	19.1	**95.7***	3.6	6.9	**93.7**	12.2	21.5
Neural Tagger	✓ / ✗ / △	75.2	24.9	37.4	93.1*	9.7	17.6	74.8	21.6	33.5
Self-training	✓ / ✗ / △	67.5	35.1	46.2	73.3	12.7	21.6	41.2	44.7	42.9
TALLOR† (Li et al., 2021)	✓ / ✗ / △	-	-	-	66.5	66.9	66.7	63.0	60.2	61.6
TALLOR (Li et al., 2021)	✓ / ✗ / △	61.5	34.7	44.3	65.6	56.8	61.9	61.6	51.5	56.1
GeNER (**ours**)	✗ / ✗ / ✗	75.0	**62.1***	**67.9**	71.9	**76.8**	**74.3**	60.3	**64.4**	**62.3**

Table 3: Performance of NER models on six datasets. $\mathbf{X}_{\text{train}}$: (unlabeled) training sentences. $\mathbf{Y}_{\text{train}}$: human-annotated training labels. $\mathcal{V}$: in-domain dictionaries by domain experts. In-domain resources are either fully used (✓), partially used (△), or not used (✗). †: utilizes n-gram statistics from the test set ($\mathbf{Y}_{\text{test}}$). Among low-resource models, best scores are marked in boldface and scores higher than that of BOND are denoted as *.

amples from each validation set and analyzed them to formulate adequate sub-questions. For instance, we used nine sub-questions for CoNLL-2003. All sub-questions for each dataset are presented in Table B.2 (Appendix) owing to space limitations.

NER models For the teacher and student models, we used RoBERTa (Liu et al., 2019) with a simple linear classifier for the token-level prediction in most experiments. For biomedical-domain datasets, BioBERT (Lee et al., 2020) was used as the backbone language model.

4.1 Low-resource NER

This experiment assumed that human-annotated training labels and dictionaries are not available. Following Li et al. (2021), we used the validation sets to search for the best hyperparameters and model checkpoints.

Datasets We used six popular NER benchmarks across four domains: [4] (1) **CoNLL-2003** (Tjong Kim Sang and De Meulder, 2003) comprises Reuters news articles with three entity types: person, location, and organization. (2) **Wikigold** (Balasuriya et al., 2009) has the same entity types as CoNLL-2003, but their subcategories are drastically different because of domain differences. In addition, its size is relatively small compared to that of others. (3) **WNUT-16** (Strauss et al., 2016) comprises nine entity types annotated on tweets, such as TV show, movie, and musician. (4) **NCBI-disease** (Doğan et al., 2014) is a corpus of 793 PubMed abstracts with manually annotated disease entities. (5) **BC5CDR** (Li et al., 2016) comprises 1,500 manually annotated PubMed abstracts with disease and chemical entities. (6) **CHEMDNER** (Krallinger et al., 2015) is a corpus of 10,000 PubMed abstracts with manually annotated chemical entities; it is the largest corpus in our experiments. Table 2 lists the benchmark statistics.

[4] Following Li et al. (2021), we exclude the *miscellaneous* and *others* types because the needs for entities are not clarified.

Baselines We compared GeNER with other *low-resource* models that do not use a full-size in-domain dictionary. Among the previous low-resource models, TALLOR (Li et al., 2021) uses the least amount of in-domain resources: unlabeled training sentences $\mathbf{X}_{\text{train}}$ and the set of seed entities, which is a small dictionary $\mathcal{V}$ that contains 20-60 manually selected (i.e., $\mathcal{V} = \triangle$). In addition to TALLOR, we provide baselines that use similar in-domain resources as TALLOR, i.e., Seed Entities, Neural Tagger, and Self-training. More details of each baseline are presented in Appendix C.1. Additionally, we also report the performance of *rich-resource* models that have access to either human-annotated training labels $\mathbf{Y}_{\text{train}}$ or rich in-domain dictionaries $\mathcal{V}$ constructed by domain experts. This type of model includes fully supervised RoBERTa (Liu et al., 2019), BioBERT (Lee et al., 2020), and the previous best weakly supervised model BOND (Liang et al., 2020).

Results As shown in Table 3, despite using fewer in-domain resources, GeNER outperformed all low-resource models in terms of F1 score. GeNER significantly outperformed the strongest low-resource model, TALLOR, by an average F1 score of 19.4 (macro averaged over six datasets). Although GeNER used unlabeled sentences retrieved from Wikipedia, it delivered excellent performance on noisy user-generated text (i.e., WNUT-16) and scientific literature (i.e., biomedical-domain datasets), thereby indicating that our model is applicable to various text genres and domains.

Interestingly, GeNER even outperformed BOND on Wikigold and WNUT-16 by F1 scores of 12.3 and 0.8, respectively.[5] These results indicate that our approach, which automatically generates a pseudo-dictionary, is promising and can be comparable to methods that use an expert-provided dictionary.

4.2 Few-shot NER

Few-shot NER is another approach for addressing low-resource problems. Unlike the use of the entire training dataset, few-shot NER models use a smaller number of training sentences and their labels (that is, $\mathbf{X}_{\text{train}} = \triangle$ and $\mathbf{Y}_{\text{train}} = \triangle$). We evaluated (1) **GeNER**, which does not use even a small number of human-annotated examples, and (2) **GeNER + Fine-tuning**, which is initialized by the best checkpoint of GeNER and then fine-tuned with the token-level prediction objective using the few-shot training examples until it converges.

Model	CoNLL -2003	Wikigold	BC5CDR
Supervised	53.5‡	47.0‡	55.0
+ NSP	61.4‡	64.0‡	-
+ Self-training	65.4‡	68.4‡	-
QUIP (Standard)	70.0‡	67.6	61.8
QUIP (Q-prompt)	74.0‡	70.6	65.7
GeNER	71.0	72.5	74.3
+ Fine-tuning	**75.0**	**73.3**	**77.7**

Table 4: Performance of few-shot NER models on three NER datasets. F1 score is reported. ‡ indicates that scores are from Huang et al. (2021) and Jia et al. (2022).

Settings We compared GeNER with the methods of Huang et al. (2021) and Jia et al. (2022) using three NER datasets used in the prior works: CoNLL-2003, Wikigold, and BC5CDR. Details of the baseline models are presented in Appendix C.2. In total, 20 training examples were provided for CoNLL-2003 and Wikigold, whereas only 10 were provided for BC5CDR. All the results were averaged over five different sampled datasets with the same number of examples. Unlike the low-resource NER experiment, the *miscellaneous* type was included in the experiment for a fair comparison with the baselines.

Results Table 4 shows the performance of the few-shot NER models and GeNER. GeNER outperformed the previous best model, QUIP (Jia et al., 2022), on two datasets, even before its fine-tuning. When fine-tuned on the same set of few-shot examples, GeNER achieved a new state-of-the-art performance on all datasets.

5 Analysis

5.1 Ablation Study

Question templates We tested five different question templates in GeNER and compared them in terms of their phrase retrieval quality and final NER performance. To measure the retrieval quality, we manually checked how many of the top 100 phrases for each sub-question were entities of correct types and computed the precision (P@100). Furthermore, we measured the number of unique phrases in the top-100 retrievals (i.e., Diversity). Table 5 shows that "*Which* [TYPE]*?*" has the highest P@100 and

[5] It is difficult to train BOND on CHEMDNER owing to the lack of rich dictionaries, which indicates the limitation of previous weakly supervised methods.

Template	P@100	Diversity	F1 Score	
			$k_l = 100$	$5,000$
Which [TYPE]?	**97.4**	44.6	52.3	**72.7**
list of [TYPE]	79.4	56.3	53.6	72.1
example of [TYPE]	66.4	50.9	49.7	57.9
What [TYPE]?	90.7	48.7	53.3	61.0
[TYPE]	69.6	**58.9**	**56.1**	67.4

Table 5: Retrieval quality and NER performance of different question templates. P@100 and diversity are macro-averaged over different types, and F1 score on the CoNLL-2003 validation set is reported.

[TYPE]	CoNLL-2003	Wikigold
organization	27.3	35.8
sports team	49.9	46.8
+ company	53.3	57.2
+ band	**55.3**	**60.7**

Table 6: Performance of GeNER with different sets of sub-questions for the organization type on the CoNLL-2003 and Wikigold validation sets. F1 score on the organization type is reported. Each [TYPE] is used with the question template "*Which* [TYPE]*?*".

"[TYPE]" has the best diversity. Although the diversity measure correlates well with the performance while retrieving $k_l = 100$ sentences for each sub-question, retrieving a larger number of sentences ($k_l = 5,000$) mitigates the low-diversity problem and provides the best overall performance.

Effect of sub-questions GeNER uses sub-questions to better reflect the needs inherent in each NER dataset. In Table 6, we report the performance of GeNER on the CoNLL-2003 and Wikigold validation sets with different sets of sub-questions. Using multiple sub-questions provides better performance while being more explicit about the needs than the performance while using only the "*Which organization?*" question. Interestingly, although CoNLL-2003 does not contain many band names unlike Wikigold, both datasets benefit from using "band" as an additional sub-question, implying that their context may help generalize to other organizational entities.

Effect of self-training Figure 2 illustrates the effect of self-training with three different dictionaries: seed entities from TALLOR, an in-domain dictionary from BOND, and a pseudo-dictionary from GeNER.[6] The performance at step=0 represents the performance immediately after the teacher model is first initialized. Although our pseudo-dictionary is initially incomplete compared with the in-domain dictionary, self-training largely closes the gap, which does not occur for the seed entities.

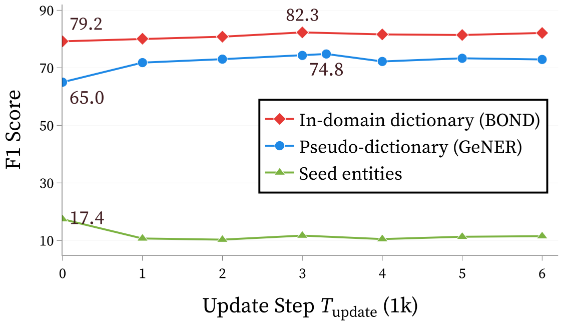


Figure 2: Effect of self-training with three different dictionaries from TALLOR, BOND, and GeNER. F1 score on the BC5CDR validation set is reported. Best scores for each dictionary are labeled.

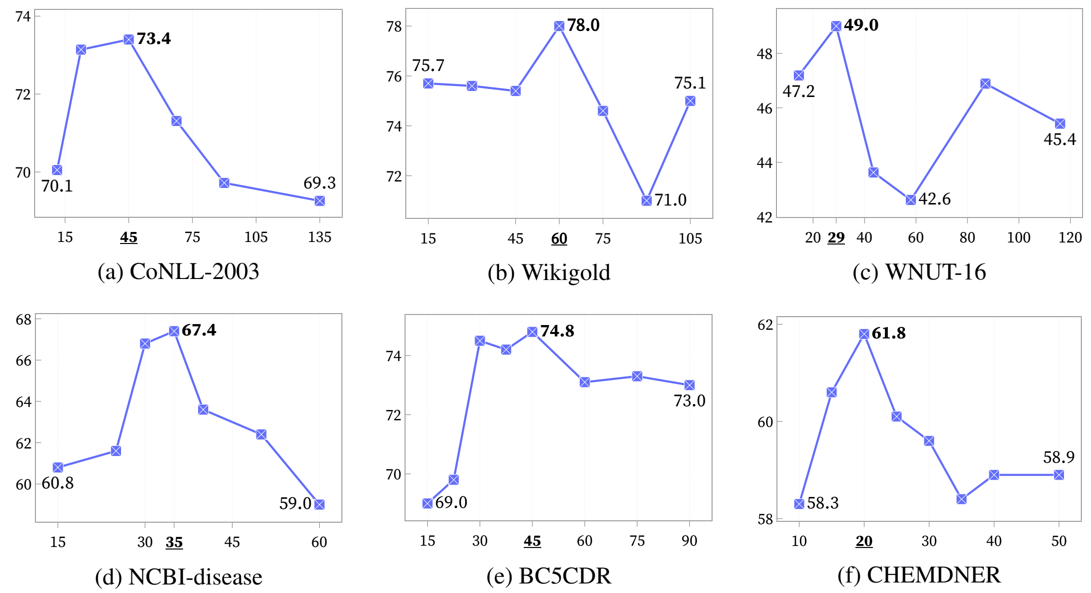
Effect of number of sentences Figure 3 shows how the performance changes when the total number of retrieved sentences for each dataset is increased. The performance tends to improve at first but degrades after the highest is reached, indicating that simply increasing the number of entities does not help. We suspect that the number of phrases with incorrect types may be increasing, causing performance degradation.

5.2 Qualitative Analysis

Visualization The top phrases of GeNER are retrieved based on the similarity score between the phrase (i.e., entity) vectors and our question vector. To understand how GeNER works, we visualized question vectors $\mathbf{q}$ used by GeNER and the entity vectors computed from the CoNLL-2003 validation set. The question vectors are encoded from the question encoder of DensePhrases, whereas its phrase encoder is used to compute the entity vectors of the annotated entities in CoNLL-2003. From Figure 4, it can be observed that the entities in the validation set are well separated based on their entity types, indicating that the phrase encoder of DensePhrases provides high-quality entity representations for NER. In addition, we observe that our question vectors cover different groups of entity vectors, which eventually retrieve entities of correct types.

Context diversity Dense representations of text can capture subtle semantic relationships between the context and question (Karpukhin et al., 2020; Lee et al., 2021a); therefore, our simple questions

[6] The size of the dictionary from BOND is more than 300k, and ours is 15k. The set of seed entities comprises 20 entities.



(a) CoNLL-2003

(b) Wikigold

(c) WNUT-16

(d) NCBI-disease

(e) BC5CDR

(f) CHEMDNER

Figure 3: Performance of GeNER with different number of retrieved sentences on six datasets. The X and Y axes of the graphs indicate the total number of sentences (1k) and the F1 scores on the validation sets, respectively.

TaLLoR (**entities** and their context for each rule)
Rule: `POStag`="`NOUN`" ∧ `PostNgram`="attack" [1] Acute **hepatitis** attack after exposure to telithromycin. [2] This is not consistent with a CNS origin of **migraine** attack.
Rule: `PreNgram`="in patients with" ∧ `PostNgram`="'s disease" [1] . . . an increased mortality in patients with **Parkinson's disease** (PD) . . . [2] . . . in the treatment of psychosis and disruptive behaviors in patients with **Alzheimer's disease**.

GeNER (**entities** and their context for "*Which disease?*")
[1] **Leprosy** has affected humanity for thousands of years. [2] **Heart disease** is one of the leading causes of death in the world. [3] During this war an outbreak of **syphilis** occurred among the French troops. [4] . . . , **typhus** being at once the most contagious and the most preventable of diseases, . . . [5] When **syphilis** was first definitely recorded in Europe in 1495, its pustules often . . .

Table 7: Comparison of extracted entities and their context from TaLLoR (Li et al., 2021) and GeNER (ours) for disease NER. While TaLLoR relies on explicit rules based on POS tags or n-grams, GeNER discovers named entities more implicitly, which appear in more diverse context. Note that the context of TaLLoR is from the BC5CDR training set (sentences from PubMed) while that of GeNER is mined from Wikipedia.

often retrieve sentences with diverse contexts. We found that almost half of the retrieved sentences for "*Which disease?*", do not contain "disease" in their context. As shown in Table 7, our retrieved sentences have a considerably diverse context than sentences from the rule-based model (Li et al., 2021).

5.3 Complexity Analysis

From automatic dataset generation to model training, GeNER is highly efficient. The dataset generation steps (Steps 1, 2, and 3) mostly required approximately 10 min in total, while the self-training step (Step 4) required approximately 30 min.

6 Discussion and Conclusion

GeNER is the first attempt to automatically generate NER datasets using a general-purpose QA system. GeNER, while using fewer in-domain resources, largely outperformed existing low-resource models on all six benchmarks, sometimes even outperforming the rich-resource model BOND. GeNER achieved a new state-of-the-art performance on three benchmarks upon evaluating the few-shot setting. Our code and datasets have been made publicly available to facilitate further research. We discuss some of the important aspects of GeNER that have not been explored in depth and provide possible future directions.

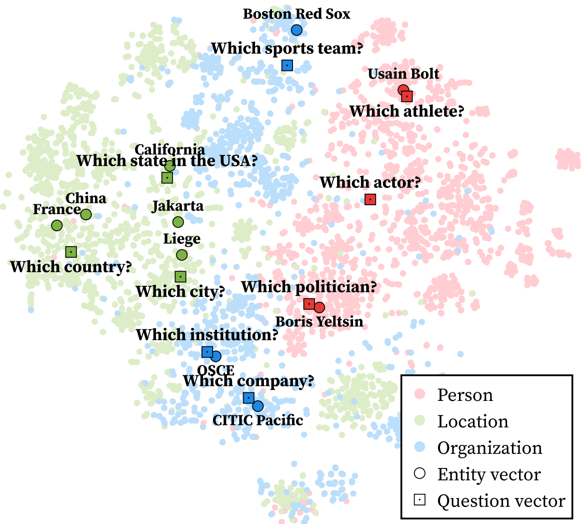


Figure 4: Visualization of GeNER question vectors and entities in the CoNLL-2003 validation set. Vectors are visualized with t-SNE.

Step	Time
1. Query Formulation	0.7s
2. Retrieval	1m 22.5s
3. Dictionary Matching	9m 16.3s
4. Self-training	29m 4.2s
Total Time	39m 43.7s

Table 8: Complexity analysis of GeNER on CoNLL-2003. For the self-training step, time taken for finding the best model on each validation set is reported. Time per each step is measured on a server with Intel Xeon(R) Silver 4210 CPU @ 2.20GHz and a single 24GB GPU (RTX 3090).

Better QA models GeNER is a model-agnostic framework; therefore, we can employ stronger open-domain QA models that often rely on the retriever-reader approach (Fajcik et al., 2021). However, because of the large number of phrases that are required to be retrieved (e.g., 5,000), it is highly convenient to use phrase retrieval models with better run-time efficiency and strong accuracy.[7] Whether the advancement of open-domain QA models can translate to the improvement of GeNER is an interesting research direction.

Other applications Other applications of GeNER can include relation extraction. For instance, if we want to train a relation extraction model for the drug-disease relationship, we can simply ask "*Which drug is effective for disease?*" and use the retrieved sentences as the positive training instances. We can use retrieved phrases as objects (drugs) and leverage NER models to identify subject entities (diseases) in the evidence sentence. It will be interesting to compare this approach with distantly supervised approaches (Mintz et al., 2009) in future research. In addition, we discuss the potential of GeNER for fine-grained and zero-shot NER tasks in Appendices D and E.

Limitations

Until a superior QA model improves GeNER in the future, it will continue to inherit the limitations of the current QA model. For instance, it is difficult to adapt our framework to languages with limited resources other than English because DensePhrases does not support other languages; moreover, other QA models are being primarily developed for English. In this regard, we believe that future research on generating NER datasets for low-resource languages would be valuable and interesting.

Although our template "*Which* `[TYPE]`*?*" can be generally applied to various *named* entities, it requires modification for some entity types. For instance, for extracting *numerical* entity types such as money, date, time, duration, and quantity, questions beginning with "when," "what time," and "how much/many," which are tailored to the specific types, would be required. Our study focuses on named entities, and identifying these specialized entity types will be a part of of future research.

Acknowledgements

We thank Jungsoo Park, Gyuwan Kim, Mujeen Sung, Sungdong Kim, Yonghwa Choi, Wonjin Yoon, and Gangwoo Kim for the helpful feedback. This research was supported by (1) National Research Foundation of Korea (NRF-2020R1A2C3010638), (2) the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2021-2020-0-01819) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation), and (3) a grant of the Korea Health Technology R&D Project through the Korea Health Industry Development Institute (KHIDI), funded by the Ministry of Health & Welfare, Republic of Korea (grant number: HR20C0021).

[7] Our preliminary experiments of using DPR (Karpukhin et al., 2020) showed much lower retrieval performance (P@100) and slower inference speed. It is also difficult to scale DPR to extract more than a hundred phrases.

References

Rami Aly, Andreas Vlachos, and Ryan McDonald. 2021. Leveraging type descriptions for zero-shot named entity recognition and classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1516–1528, Online. Association for Computational Linguistics.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 261–270, Hong Kong, China. Association for Computational Linguistics.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada. Association for Computational Linguistics.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

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

Martin Fajcik, Martin Docekal, Karel Ondrej, and Pavel Smrz. 2021. R2-D2: A modular baseline for open-domain question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 854–870, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Abbas Ghaddar and Phillippe Langlais. 2017. WiNER: A Wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2021. Few-shot named entity recognition: An empirical baseline study. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10408–10423, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Robin Jia, Mike Lewis, and Luke Zettlemoyer. 2022. Question answering infused pre-training of general-purpose contextualized representations. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 711–728, Dublin, Ireland. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

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

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. 2021a. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6634–6647, Online. Association for Computational Linguistics.

Jinhyuk Lee, Alexander Wettig, and Danqi Chen. 2021b. Phrase retrieval learns passage retrieval, too. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiacheng Li, Haibo Ding, Jingbo Shang, Julian McAuley, and Zhe Feng. 2021. Weakly supervised named entity tagging with learnable logical rules. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4568–4581, Online. Association for Computational Linguistics.

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

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

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1054–1064.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021a. Noisy-labeled NER with confidence estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3437–3445, Online. Association for Computational Linguistics.

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*.

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

Yu Meng, Yunyi Zhang, Jiaxin Huang, Xuan Wang, Yu Zhang, Heng Ji, and Jiawei Han. 2021. Distantly-supervised named entity recognition with noise-robust learning and language model augmented self-training. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10367–10378, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Lance Ramshaw and Mitch Marcus. 1995. Text chunking using transformation-based learning. In *Third Workshop on Very Large Corpora*.

Minjoon Seo, Jinhyuk Lee, Tom Kwiatkowski, Ankur Parikh, Ali Farhadi, and Hannaneh Hajishirzi. 2019. Real-time open-domain question answering with dense-sparse phrase index. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4430–4441, Florence, Italy. Association for Computational Linguistics.

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018a. Automated phrase mining from massive text corpora. *IEEE Transactions on Knowledge and Data Engineering*, 30(10):1825–1837.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018b. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2054–2064, Brussels, Belgium. Association for Computational Linguistics.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

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

Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. Learning from language description: Low-shot named entity recognition via decomposed framework. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1618–1630, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised NER with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

A Normalization Rules

This section details the normalization rules defined by us. They are generally applicable to several domains without considerable modification. They can be treated as hyperparameters (i.e., whether a rule is applied or not) and tuned based on the model performance on the validation sets. They can also be determined without validation sets, based on the common knowledge of practitioners regarding target entities. They must be applied in order for reproducing the same results.

- **Rule 1** Some retrieved phrases contain multiple entities linked by the conjunction "and." We simply split such composite mentions based on "and." This rule should be applied based on the annotation guidelines of the datasets. For instance, biomedical NER datasets such as NCBI-disease consider composite mentions (e.g., "colorectal, endometrial, and ovarian cancers") as one entity;[8] therefore, we do not use this rule for these datasets. Moreover, it should not be applied to movie entities (e.g., "Harry Potter and the Sorcerer's Stone").

- **Rule 2** DensePhrases frequently returns phrases with punctuation at the start or end of the string, such as commas or quotation marks. We remove these noises using simple post-processing (e.g., "Leprosy**,**" → "Leprosy"). However, for some entities such as songs, punctuation may not be noise but can be a part of the name.

- **Rule 3** We exclude phrases that are entirely in lowercase. Many named entities in the real world contain one or more uppercase letters (e.g., the first letter of a person's name is capitalized.). Thus, retrieved phrases, entirely in lowercase, are lowercase are more likely to be noisy results. However, because lowercase entities are common in biomedical datasets, this rule should be carefully applied depending on the datasets and entity types.

- **Rule 4** We remove definite article "the" from the string (e.g., "the Boston Red Sox" → "Boston Red Sox"). This rule should be applied depending on the annotation guidelines of the datasets or the superficial characteristics of the entities. For instance, because band entities sometimes include "the" in their name, this rule should not be applied to such entities.

- **Rule 5** We exclude phrases with a length of less than three in our dictionary because short strings can cause significant noise.

- **Rule 6** We exclude phrases whose lowercase strings are in the stopword list such as "WAS" (Wiskott-Aldrich Syndrome) and "US" (United States) because they can cause considerable amount of false-positive noise in the dictionary matching process. Simultaneously, this rule can produce false-negative noise in the generated dataset; however, self-training mitigates this noise.

- **Rule 7** We exclude phrases that are the same as [TYPE] in the sub-question. For example, if we ask a question "*Which disease?*," and the resulting phrase is "disease," we do not use it.

- **Rule 8** Because named entities are often abbreviated, it is important to annotate abbreviations so as to avoid false-negative noise from them. We detected the abbreviations of retrieved phrases using the ScispaCy abbreviation tool.[9] For instance, when the phrase "Crohn's disease" is retrieved with the evidence sentence "*Crohn's disease (CD) is one of the two main forms of inflammatory bowel disease.*," its abbreviation "CD" is detected. It should be noted that abbreviations are not added to the dictionary because they usually have short forms, which can lead to considerable noise in dictionary matching.

- **Rule 9** During dictionary matching, phrases in the dictionary and sentences are converted to lowercase by default. We prevent lowercase single tokens in the sentence from being matched with phrases in the dictionary because single lowercase tokens tend to be noisy (compared with multi-tokens).

- **Rule 10** We use the phrase mining tool, AutoPhrase (Shang et al., 2018a), to refine entity boundaries in the dictionary matching stage. Specifically, if the span of a retrieved phrase is

[8] https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/Guidelines.html

[9] https://github.com/allenai/scispacy

included in that of the phrase detected by AutoPhrase, we expand the span of the retrieved phrase to that of the detected phrase.

B Implementation Details

Seed entities For CoNLL-2003, BC5CDR, and CHEMDNER, we used the same seed entities as Li et al. (2021). For the other benchmarks, because there are no pre-defined seed entities, we manually selected frequent and high-precision entities in the training sets, following Li et al. (2021). Note that this selection process relies on in-domain resources, such as training sentences or professional knowledge of experts. The seed entities are listed in Table B.1.

In-domain dictionary BOND requires rich in-domain dictionaries to achieve a high performance. For CoNLL-2003, Wikigold, and WNUT-16, the model used Wikidata that comprises more than 96 million entities (as of October 21, 2021) and multiple gazetteers from different websites for each dataset. For instance, 10 websites were used for CoNLL-2003, including Random Name and Intergovernmental Organization.[10] For NCBI-disease and BC5CDR, we used dictionaries provided by Shang et al. (2018b), which are derived from the MeSH database and the Comparative Toxicogenomics Database, which comprises more than 300k disease and chemical entities. The dictionaries were additionally tailored to the target corpora using techniques such as corpus-aware dictionary tailoring (Shang et al., 2018b).

TALLOR Although we used the official code base,[11] the model we implemented was lower than the reported performance (scores with † in Table 3). This is because, in the original implementation, n-gram statistics of the test set was used in the entity candidate generation process of TALLOR. On the other hand, we implemented the model using only the training corpus for fair comparison.

Other details The entire subquestions we selected are listed in Table B.2. We used public PyTorch implementation provided by Liu et al. (2019) and Lee et al. (2020)[12] for implementing the Neural Tagger baselines and our fine-tuning models. We used the pre-trained weights of the `densephrases-multi-query-multi` model for the question and phrase encoders of DensePhrases.[13] For BOND, we used the official code base provided by the authors.[14] Also, we used the same code base for the self-training step in GeNER. The best hyperparameters in self-training of our models are detailed in Table B.3.

C Baseline Models

C.1 Low-resource NER

Seed Entities This model directly matches seed entities with the test corpus. Seed entities are a small number of entities, which are manually selected by experts. For instance, OSCE, NATO, Honda, Interfax, and Marseille are seed entities for the organization type of CoNLL-2003.

Neural Tagger We first annotate in-domain sentences by dictionary matching using the seed entity set as the dictionary, and then train RoBERTa or BioBERT on the generated corpus.

Self-training Similar to our model, this model is trained through self-training on weak labels, but the labels are generated by dictionary matching between in-domain sentences and seed entities. This can be viewed as a weak version of BOND (Liang et al., 2020) that uses a small dictionary (i.e., the seed entity set).

TALLOR (Li et al., 2021) This model is a strong baseline model that starts with 20-60 initial labeling rules (called seed rules) and automatically expands its labeling rule set. Seed rules are defined as string matching between entity candidates (i.e., spans of text) in in-domain sentences and pre-defined seed entities. A neural model that is initially trained on the sentences annotated by seed rules generates weak labels. Rule candidates are selected based on the in-domain sentences and weak labels, and several top-ranked rules are added to the rule set. This process is performed iteratively.

BOND (Liang et al., 2020) This framework first generates weak labels by dictionary matching between the in-domain sentences and in-domain dictionary, and then trains NER models based on self-training, which is the previous best weakly supervised method. Dictionaries are created using rich external resources such as Wikidata and online websites. Appendix B provides more information on the in-domain dictionaries. In the first iteration,

[10] See Liang et al. (2020) for the entire list.

[11] https://github.com/JiachengLi1995/TALLOR

[12] https://github.com/dmis-lab/biobert-pytorch

[13] https://github.com/princeton-nlp/DensePhrases

[14] https://github.com/cliang1453/BOND

Dataset	Entity Type	# Seeds	Seed Entities
CoNLL-2003	*person*	7	wasim akram, waqar younis, mushraq ahmed, aamir sohail, saeed anwar, bill clinton, mother teresa
	location	8	britain, italy, russia, sweden, belgium, iraq, south africa, united states
	organization	5	osce, nato, honda, interfax, marseille
Wikigold	*person*	5	cabral, bobick, belgrano, behe, moses mendelssohn
	location	10	england, indonesia, old goa, chicago, ontario, aabenraa county, illinois, hay street, b & sr, cal anderson park
	organization	10	maaa, ncaa, 139th, major league baseball, cbs cable, bcit, montreal hockey club, 882 6pr, konami, 30 seconds to mars
WNUT-16	*person*	2	lindsay lohan, scooter braun
	location	3	belgium, toronto, arizona
	product	8	ipad, htc desire z, iphone, pumpkin moonshine, coke, flip minohd, club penguin, xbox 360
	facility	5	visions lounge, frat house hattiesburg, empire state building, disney world, club blu
	company	3	twitter, youtube, facebook
	sports team	1	jv soccer
	TV show	1	friday night lights
	movie	1	iron man 2
	music artist	1	kings of leon
NCBI-disease	*disease*	20	dmd, pws, myotonic dystrophy, g6pd deficiency, hd, pku, aniridia, duchenne muscular dystrophy, fap, a - t, tay - sachs disease, tsd, fmf, prader - willi syndrome, amn, wiskott - aldrich syndrome, huntington disease, pelizaeus - merzbacher disease, bmd
BC5CDR	*disease*	10	proteinuria, esrd, thrombosis, tremor, hepatotoxicity, hypertensive, thrombotic microangiopathy, thrombocytopenia, akathisia, confusion
	chemical	10	nicotine, morphine, haloperidol, warfarin, clonidine, creatinine, isoproterenol, cyclophosphamide, sirolimus, tacrolimus
CHEMDNER	*chemical*	60	glucose, cholesterol, glutathione, ethanol, androgen, graphene, glutamate, dopamine, cocaine, serotonin, estrogen, nicotine, tyrosine, resveratrol, nitric oxide, cisplatin, alcohol, superoxide, curcumin, metformin, amino acid, testosterone, flavonoids, camp, methanol, amino acids, fatty acids, polyphenols, nmda, silica, 5-ht, oxygen, calcium, copper, cadmium, arsenic, zinc, mercury, (1) h, ca (2+)

Table B.1: Seed entities used in our experiments. All seed entities are in lowercase. We use the seed entities provided by Li et al. (2021) for CoNLL-2003, BC5CDR, and CHEMDNER. For the remaining datasets where seed entities are not provided, we manually select frequent and high-precision entities, following Li et al. (2021).

the *teacher* and *student* models are initialized to standard language models (e.g. RoBERTa), and the teacher model is fine-tuned on the weak labels. The teacher model then re-annotates the in-domain sentences, and the student model is trained on the newly generated labels by the teacher model. For every T_{update} iterations (i.e., the period of the update), the teacher model is updated as a (trained) student model.

C.2 Few-shot NER

We used the models from the recent two studies (Huang et al., 2021; Jia et al., 2022) as baselines. We excluded some few-shot NER models because they use a sufficient amount of source data (Yang and Katiyar, 2020; Cui et al., 2021), which differs from our setting.

Supervised This model is trained directly on few-shot examples. RoBERTa (for CoNLL-2003 and Wikigold) and BioBERT (for BC5CDR) were used.

Noisy supervised pre-training (NSP) (Huang et al., 2021) NSP pre-trains models on the large-scale corpus WiNER (Ghaddar and Langlais, 2017), which comprises 2013 Wikipedia documents and weak labels for 113 fine-grained entity types. The labels are generated based on the anchor links and coreference resolution. The models pretrained by NSP were then fine-tuned using few-shot examples.

Self-training This model (Huang et al., 2021) follows the current semi-supervised learning method (Xie et al., 2020), where the model is initialized with few-shot examples and further (self-)trained using unlabeled training sentences.

QuIP (Standard) (Jia et al., 2022) QuIP is a contextualized representation model pre-trained with approximately 80 million question-answer pairs, which are automatically generated by the BART-large model (Lewis et al., 2020). QuIP (Standard) comprises a QuIP encoder with a randomly initialized linear output layer. The initialized

Dataset	Entity Type	[TYPE]	k_l	Rule
CoNLL-2003	*person*	athlete, politician, actor	5,000	1,3,4
	location	country, city, state in the USA	5,000	1,3,4
	organization	sports team, company, institution	5,000	1,3,4
Wikigold	*person*	athlete, politician, actor, director, musician	4,000	1,3,4
	location	country, city, state in the USA, road, island	4,000	1,3,4
	organization	sports team, company, institution, association, band	4,000	1,3,4
WNUT-16	*person*	athlete, politician, actor, author	1,000	1,3,4
	location	country, city, state in the USA	1,000	1,3,4
	product	mobile app software, operating system, car, smart phone	1,000 1,000	3 1,3,4
	facility	facility, cafe, restaurant, college, music venue sports facility	1,000 1,000	3 1,3,4
	company	company, technology company news agency, magazine	1,000 1,000	1,3,4 1,3
	sports team	sports team	1,000	1,3,4
	TV show	TV show	1,000	3
	movie	movie	1,000	3
	music artist	band, rapper, musician, singer	1,000	3
NCBI-disease	*disease*	disease	35,000	4,9
BC5CDR	*disease*	disease	15,000	4,9
	chemical	chemical compound, drug	15,000	4,9
CHEMDNER	*chemical*	chemical compound, drug	10,000	4,9
CrossNER	*enzyme*	enzyme	5,000	1,4,9
	astronomical object	astronomical object	5,000	1,3,4
	award	award	10,000	1,3,4
	conference	conference on artificial intelligence	5,000	3

Table B.2: Subquestions and hyperparameters used for NER benchmarks. Each sub-question is formulated as "*Which* [TYPE]*?*" and used for the retrieval. k_l: number of unique sentences retrieved for each sub-question. The total number of sentences for a dataset is calculated as the sum of the number of sentences for each sub-question. Normalization rules are detailed in Appendix A. Note that we omit Rules 2, 5, 6, 7, 8, and 10, because they are commonly applied.

QuIP model is fine-tuned on few-shot examples.

QuIP (Q-prompt) (Jia et al., 2022) Unlike QuIP (Standard), the output layer of QuIP (Q-prompt) is initialized as the embeddings for question prompts. For instance, for the organization type, "*What is an organization?*" is used as the question prompt, and the output layer is then initialized as QuIP's representation of the prompt. Jia et al. (2022) showed that this initialization strategy is effective for few-shot NER. As suggested, we used the same question prompts as Jia et al. (2022) for CoNLL-2003 and Wikigold. For BC5CDR, we used "*What is a disease?*" for the disease type and "*What is a chemical compound?* for the chemical type because "*What is a drug?*" is less effective.

D Fine-grained NER

The human annotations for fine-grained entity types are sparser than those for coarse-grained types. To determine whether the data sparsity problem is addressable by GeNER, we created four fine-grained datasets derived from CrossNER (Liu et al., 2021b): enzyme and astronomical object (natural science domain), award (literature domain), and conference (artificial intelligence domain), by removing labels for the other entity types from the sentences in CrossNER. We selected these four types because they were not coarse-grained, thus meeting the purpose of this experiment. Table D.1 presents the statistics of the datasets. We used a single sub-question for each dataset (See Table B.2.)

Model	Dataset	T_{begin}	T_{update}
GeNER	CoNLL-2003	900	300
	Wikigold	500	300
	WNUT-16	900	450
	NCBI-disease	900	300
	BC5CDR	500	200
	CHEMDNER	900	300
	Enzyme	350	700
	Astr.	500	300
	Award	350	400
	Conf.	200	100
BOND	CoNLL-2003	900	450
	Wikigold	900	300
	WNUT-16	900	300
	NCBI-disease	900	450
	BC5CDR	500	300
Self-training	CoNLL-2003	400	100
	Wikigold	350	200
	WNUT-16	500	100
	NCBI-disease	200	100
	BC5CDR	500	100
	CHEMDNER	900	450

Table B.3: Hyperparameter configuration in self-training of GeNER and baselines. T_{begin} is the early stopping step before updating the model, and T_{update} is the period of the update. For more detailed descriptions of T_{begin} and T_{update}, refer to Liang et al. (2020).

Dataset	# Sents (train / valid / test)	# Labels (train / valid / test)
Enzyme	200 / 450 / 543	22 / 48 / 80
Astro.	200 / 450 / 543	121 / 373 / 337
Award	100 / 400 / 416	34 / 124 / 141
Conference	100 / 350 / 431	24 / 89 / 93

Table D.1: Statistics of fine-grained NER datasets derived from the CrossNER dataset (Liu et al., 2021b). Astr.: astronomical object. # Types: number of entity types. # Sents: number of sentences. # Labels: number of entity-level annotations.

Model	Enzyme	Astr.	Award	Conf.
Fully supervised	56.4	78.0	75.4	49.4
GeNER	49.5	71.9	80.9	41.1
+ Fine-tuning	**63.1**	**86.8**	**81.6**	**64.0**

Table D.2: Performance of the fully supervised RoBERTa model and GeNER on fine-grained entity types. F1 score is reported. Astr. and Conf. indicate astronomical object and conference, respectively.

Results We compared GeNER with the fully supervised RoBERTa, as shown in Table D.2. We found that GeNER is highly comparable with the fully supervised model and sometimes even outperforms the fully supervised model (Award). Its performance can be further improved by fine-tuning on each small training set (+ Fine-tuning).

Retrieved Entities from GeNER	P@50	Div.
[TYPE]: *song nominated for the Grammy Awards*		
Hotline Bling, Love Me like You Do, Mystery of Love, Can't Stop the Feeling!, The Price is Wrong, . . .	0.96	0.90
[TYPE]: *dish made with eggs*		
Eggs Benedict, Pancakes, Shakshouka, Omelettes, Huevos rancheros, Chilaquiles, Menemen, . . .	0.80	0.78
[TYPE]: *satellite made by an American company*		
GE-2, AMC-2, Ariel 1, Syncom 3, CHIPSat, Telstar, Explorer 1, Westar 1, SkyTerra-1, . . .	0.88	0.82

Table D.3: Retrieval entities from GeNER for fine-grained entity types. "*Which* [TYPE]*?*" is used as a question. Div. indicates the diversity measure defined in Section 5.1.

Retrieved entities We further show the potential of GeNER on three entities that are extremely fine-grained such as "*satellite made by an American company*." As there are no human annotations for these entities, we manually measured retrieval performance using precision at 50 (P@50) and diversity, similar to Section 5.1. Table D.3 shows that accurate and diverse entities were retrieved for each question. Because of the flexibility of the natural language questions, GeNER can easily provide NER models for specialized entity types.

E Connection to Zero-shot NER

Zero-shot NER aims to build models that generalize to *unseen* entity types without corresponding labels. It has hard constraints that the entity types in $\mathbf{Y}_{\text{test}}$ are not observed during training over $\mathcal{D}_{\text{train}}$. To tackle this task, researchers have proposed to utilize external descriptions of the target entities (Aly et al., 2021; Wang et al., 2021). We expect that GeNER can support zero-shot NER by generating weak-labeled sentences for the target entity types, where the sentences can be used as semantic information. Leveraging GeNER in zero-shot NER would be interesting since it can remove the strong assumption that type descriptions are available.