

# Few-shot Named Entity Recognition with Self-describing Networks

Jiawei Chen<sup>1,3,\*</sup>, Qing Liu<sup>1,3,\*</sup>, Hongyu Lin<sup>1,†</sup>, Xianpei Han<sup>1,2,4,†</sup>, Le Sun<sup>1,2</sup>

<sup>1</sup>Chinese Information Processing Laboratory <sup>2</sup>State Key Laboratory of Computer Science  
Institute of Software, Chinese Academy of Sciences, Beijing, China

<sup>3</sup>University of Chinese Academy of Sciences, Beijing, China

<sup>4</sup>Beijing Academy of Artificial Intelligence, Beijing, China

{jiawei2020, liuqing2020, hongyu, xianpei, sunle}@iscas.ac.cn

## Abstract

Few-shot NER needs to effectively capture information from limited instances and transfer useful knowledge from external resources. In this paper, we propose a self-describing mechanism for few-shot NER, which can effectively leverage illustrative instances and precisely transfer knowledge from external resources by describing both entity types and mentions using a universal concept set. Specifically, we design *Self-describing Networks* (SDNet), a Seq2Seq generation model which can universally describe mentions using concepts, automatically map novel entity types to concepts, and adaptively recognize entities on-demand. We pre-train SDNet with large-scale corpus, and conduct experiments on 8 benchmarks from different domains. Experiments show that SDNet achieves competitive performances on all benchmarks and achieves the new state-of-the-art on 6 benchmarks, which demonstrates its effectiveness and robustness.

## 1 Introduction

Few-shot named entity recognition (FS-NER) aims to identify entity mentions corresponding to new entity types (i.e., novel types) with only a few illustrative examples. FS-NER is a promising technique for open-domain NER which contains various unforeseen types and very limited examples and therefore has attracted great attention in recent years (Huang et al., 2020; Wang et al., 2021).

The main challenge of FS-NER is how to accurately model the semantics of unforeseen entity types using only a few illustrative examples. To achieve this, FS-NER needs to effectively capture information in few-shot examples, meanwhile exploiting and transferring useful knowledge from external resources. Unfortunately, information entailed in illustrative examples is very limited, i.e., the **limited information challenge**. And external

\*Equally Contribution.

†Corresponding authors.

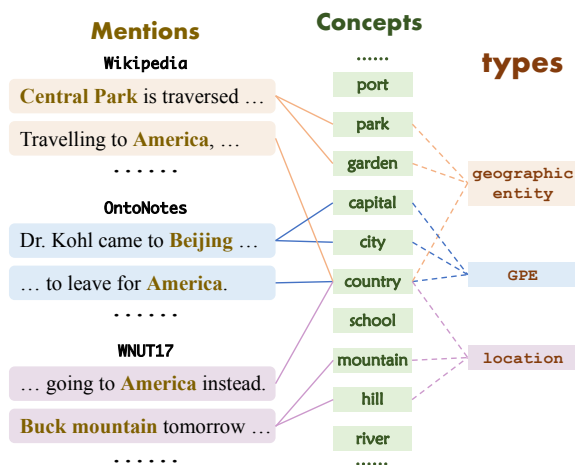


Figure 1: Examples of concept description. Wikipedia, OntoNotes, WNUT17 can transfer knowledge to each other by describing mentions and types using a universal concept set.

knowledge usually doesn't directly match with the new task because it may contain irrelevant, heterogeneous, or even conflicting knowledge (Beryozkin et al., 2019; Yu and Yang, 2020) – which we refer as **knowledge mismatch challenge**. For example, the schemas in Wikipedia, OntoNotes (Ralph et al., 2013) and WNUT17 (Derczynski et al., 2017) are conflicting, where “America” is geographic entity in Wikipedia, GPE in OntoNotes, and location in WNUT17. Such a knowledge mismatch problem makes it unsuitable to directly transfer external knowledge to downstream tasks. Consequently, how to sufficiently leverage limited few-shot examples and precisely transfer external knowledge are the critical challenges for FS-NER.

To this end, this paper proposes a self-describing mechanism for FS-NER. The main idea behind self-describing mechanism is that all entity types can be described using the same set of concepts, and the mapping between types and concepts can be universally modeled and learned. In this way, the knowledge mismatch challenge can be resolved by uniformly describing different entity types using

the same concept set. For example, in Figure 1 the types in different schemas are mapped to the same concept set  $\{park, garden, country, \dots\}$ , therefore the knowledge in different sources can be universally described and transferred. Furthermore, because the concept mapping is universal, the few examples are only used to construct the mapping between novel types and concepts, the limited information problem can be effectively addressed.

Based on the above idea, we propose Self-describing Networks – SDNet, a Seq2Seq generation network which can universally describe mentions using concepts, automatically map novel entity types to concepts, and adaptively recognize entities on-demand. Specifically, to capture the semantics of a mention, SDNet generates a set of universal concepts as its description. For example, generate  $\{capital, city\}$  for “Dr. Kohl came to [Beijing], ...”. To map entity types to concepts, SDNet generates and fuses the concept description of the mentions with the same entity type. For example, map GPE to  $\{country, capital, city\}$  using its mentions “Beijing” and “America”. To recognize entity, SDNet directly generates all entities in a sentence via a concept-enriched prefix prompt, which contains the target entity types and their concept descriptions. For example, recognizing entity in “France is beautiful.” by generating “France is GPE” using prefix prompt “[EG] GPE:  $\{country, capital, city\}$ ”. Because the concept set is universal, we pre-train SDNet on large-scale, easily accessible web resources. Concretely, we collect a pre-training dataset which contains 56M sentences with more than 31K concepts by leveraging the links from Wikipedia anchor words to the Wikidata items.

By projecting both mentions and entity types to a universal concept space, SDNet can effectively enrich entity types to resolve the limited information problem, universally represent different schemas to resolve the knowledge mismatch problem, and can be effectively pre-trained in a unified way. Moreover, all the above tasks are modeled in a single generation model by using prefix prompt mechanism to distinguish different tasks, which makes the model controllable, universal and can be continuously trained.

We conduct experiments on 8 few-shot NER benchmarks with different domains. Experiments show that SDNet leads to very competitive performance and achieves the new state-of-the-art on 6

of these benchmarks.<sup>1</sup>

Generally speaking, the contributions of this paper are:

- We propose a self-describing mechanism for FS-NER, which can effectively resolve the limited information challenge and the knowledge mismatch challenge by describing both entity types and mentions using a universal concept set.
- We propose Self-describing Networks – SDNet, a Seq2Seq generation network which can universally describe mentions using concepts, automatically map novel entity types to concepts, and adaptively recognize entities on-demand.
- We pre-train SDNet on the large-scale open dataset, which provides a universal knowledge for few-shot NER and can benefit many future NER studies.

## 2 Related Work

To deal with the limited information challenge, current FS-NER studies mostly focus on leveraging external knowledge, many knowledge resources are used: 1) PLMs. Early FS-NER studies (Tong et al., 2021; Wang et al., 2021) mainly use PLMs for better encoding. And prompt-based NER formulation is proposed to exploit the PLMs’ knowledge more effectively (Xin et al., 2018; Obeidat et al., 2019; Dai et al., 2021; Ding et al., 2021; Yan et al., 2021a; Liu et al., 2021; Cui et al., 2021; Ma et al., 2021; Lee et al., 2021). 2) Existing annotation datasets. These studies (Fritzler et al., 2019; Hou et al., 2020; Yang and Katiyar, 2020; Li et al., 2020a,b; Tong et al., 2021; Das et al., 2021) focus on reusing annotations in existing datasets, and the annotations can be used to pre-train NER models. 3) Distantly annotated datasets. Some works (Mengge et al., 2020; Huang et al., 2020; Jiang et al., 2021) try to automatically construct NER datasets via distant supervision, but which often suffer from the partially-labeled (Yang et al., 2018; Nooralahzadeh et al., 2019; Peng et al., 2019) and noise label (Shang et al., 2018; Peng et al., 2019; Zhang et al., 2021b,a) problem.

To deal with the knowledge mismatch problem, Kim et al. (2015); Reed et al. (2016); Qiao et al. (2016); Xian et al. (2019); Hou et al. (2020) employ label project methods which project labels

<sup>1</sup>Our source codes are openly available at <https://github.com/chen700564/sdnet>

in different schemas. Rei and Søgaard (2018); Li et al. (2020c); Wang et al. (2021); Aly et al. (2021) enrich the semantics of labels using manually label descriptions. Beryozkin et al. (2019); Yu and Yang (2020) merge the labels in different schemas into the same taxonomy for knowledge sharing. And Jiang et al. (2021) relabels the external noisy datasets using current labels. Compared with these methods, we resolve the knowledge mismatch problem by mapping all entity types to a universal concept set, and the concept mapping and target entities are automatically generated using a self-describing networks.

### 3 Self-describing Networks for FS-NER

In this section, we describe how to build few-shot entity recognizers and recognize entities using Self-describing networks. Figure 3 (b) shows the entire procedure. Specifically, SDNet is a Seq2Seq network which performs two generation tasks successively 1) *Mention describing*, which generates the concept descriptions of mentions; 2) *Entity generation*, which adaptively generates entity mentions corresponding to desirable novel type one by one. Using SDNet, NER can be directly performed through the entity generation process by putting type descriptions into its prompt. Given a novel type, its type description is built through mention describing upon its illustrative instances. In the following, we will first introduce SDNet, then describe how to construct type descriptions and build few-shot entity recognizers.

#### 3.1 Self-describing Networks

SDNet is a Seq2Seq network that can perform two generation tasks: mention describing and entity generation. Mention describing is to generate the concept descriptions of mentions and entity generation is to adaptively generate entity mentions. To guide the above two processes, SDNet uses different task prompts  $\mathbf{P}$  and generates different outputs  $\mathcal{Y}$ . Figure 2 shows their examples. For mention describing, the prompt contains a task descriptor [MD], and the target entity mentions. For entity recognition, the prompt contains a task descriptor [EG], and a list of target novel types and their corresponding descriptions. Taking prompt  $\mathbf{P}$  and sentence  $X$  as input, SDNet will generate a sequence  $\mathcal{Y}$  which contains the mention describing or entity generation results. The above two processes can be viewed as **symmetrical processes**:

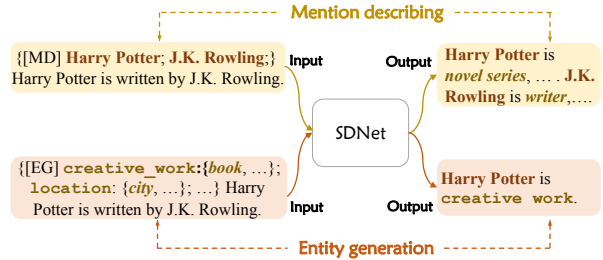


Figure 2: Examples of input and output of mention describing and entity generation.

one is to capture concept semantics of given entities, the other is to identify entities containing the specific concepts.

Specifically, SDNet first concatenates prompt  $\mathbf{P}$  and sentence  $\mathbf{X}$  into a sequence  $\mathcal{I} = \mathbf{P} \oplus \mathbf{X}$  and then fed  $\mathcal{I}$  into an encoder to obtain the hidden state representation  $\mathcal{H}$ :

$$\mathcal{H} = \text{Encoder}(\mathcal{I}).$$

Then  $\mathcal{H}$  will be fed into a decoder, and the decoder will sequentially generate a sequence  $\mathcal{Y}$ . At time step  $t$ , the probability  $\mathbf{p}_t$  of generating tokens in vocabulary is calculated by:

$$\mathbf{p}_t = \text{Decoder}(\mathcal{H}, \mathcal{Y}_{<t}).$$

We use the greedy decoding here and therefore the word in the target vocabulary with maximum value in  $\mathbf{p}_t$  is generated until [EOS] is generated.

By modeling different tasks in a single model, the generation is controllable, learning is uniform, and the model can be continuously trained.

We can see that, few-shot entity recognition can be effectively performed using the above two generation processes. For entity recognition, we can put the descriptions of target entity types into the prompt, then entities will be adaptively generated through the entity generation process. To construct the entity recognizer of a novel type, we only need its type description, which can be effectively built by summarizing the concept descriptions of their illustrative instances.

#### 3.2 Entity Recognition via Entity Generation

In SDNet, entity recognition is performed by the entity generation with the given entity generation prompt  $\mathbf{P}_{\text{EG}}$  and sentence  $X$ . Specifically,  $\mathbf{P}_{\text{EG}}$  starts with a task descriptor [EG], and the descriptor is followed by a list of target types and their corresponding descriptions, i.e.,  $\mathbf{P}_{\text{EG}} = \{[\text{EG}] t_1 : \{l_1^1, \dots, l_1^{m_1}\}; t_2 : \{l_2^1, \dots, l_2^{m_2}\}; \dots\}$ ,

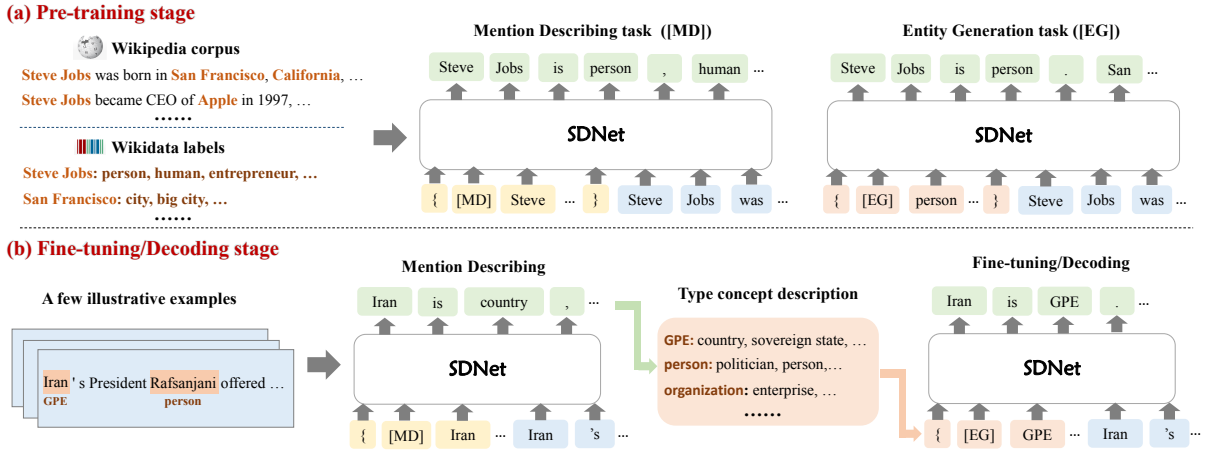


Figure 3: Overview of the process of SDNet. The upper part is the pre-training stage, and the lower part is the fine-tuning/decoding stage. In pre-training stage, the external data is used to jointly train mention describing and entity generation tasks. In fine-tuning/decoding stage, SDNet first conducts mention describing to summarize type concept descriptions, and then conducts entity generation based on the generated descriptions.

where  $l_i^j$  is the  $j$ -th concept of  $i$ -th type  $t_i$ . Prompt  $\mathbf{P}_{EG}$  and sentence  $X$  will be fed to SDNet as Section 3.1 described. Then, SDNet will generate text  $\mathcal{Y}$  in the format as “ $e_1$  is  $t_{y_1}$ ; ...;  $e_n$  is  $t_{y_n}$ .”, where  $t_{y_i}$  is the type of  $i$ -th entity  $e_i$ . Based on the generated text  $\mathcal{Y}$ , the recognized entities are obtained, i.e.,  $\{ \langle e_1, t_{y_1} \rangle \dots \langle e_n, t_{y_n} \rangle \}$ .

We can see that, in SDNet, the entity generation process can be controlled on-the-fly using different prompts. For example, given a sentence “Harry Potter is written by J.K. Rowling.”, if we want to identify entity of person type, put  $\{ [EG] \text{ person: } \{actor, writer\} \}$  to  $\mathbf{P}_{EG}$ , SDNet will generate “J.K. Rowling is person”, while if we want to identify entity of creative\_work type, put  $\{ [EG] \text{ creative\_work: } \{book, music\} \}$  to  $\mathbf{P}_{EG}$ , SDNet will generate “Harry Potter is creative\_work”.

### 3.3 Type Description Construction via Mention Describing

SDNet is controlled on-the-fly to generate different types of entities by introducing different corresponding type descriptions to  $\mathbf{P}_{EG}$ . For example, the description  $\{actor, doctor, \dots\}$  for and the description is  $\{city, state, \dots\}$  for location.

To build the type description for novel types with several illustrative examples, SDNet first obtains the concept description of each mention in illustrative examples via mention describing. Then the type description of each type is constructed by summarizing all the concept descriptions of its illustrative examples. In the following, we describe

them in detail.

**Mention Describing.** In SDNet, mention describing is a generation process, whose input is mention describing prompt  $\mathbf{P}_{MD}$  and an illustrative instance  $X$ . Specifically, given an illustrative example  $X$  which contains entity mentions  $\{e_1, e_2, \dots\}$  of novel types,  $\mathbf{P}_{MD}$  starts with a task descriptor  $[MD]$ , and the descriptor is followed by target entity mentions. i.e.,  $\mathbf{P}_{MD} = \{ [MD] e_1; e_2; \dots \}$ . Prompt  $\mathbf{P}_{MD}$  and sentence  $X$  will be fed to SDNet as Section 3.1 described. And then SDNet will generate the text  $\mathcal{Y}$  in the format as “ $e_1$  is  $l_1^1, \dots, l_1^{m_1}$ ;  $e_2$  is  $l_2^1, \dots, l_2^{m_2}$ ; ...”, where  $l_i^j$  is the  $j$ -th concept for the  $i$ -th entity mention. The concept set  $\{l_i^1, l_i^2, \dots, l_i^{m_i}\}$  will be considered as the semantic concepts reflected by entity mention  $e_i$ .

**Type Description Construction.** SDNet then summarizes the generated concepts to describe the precise semantics of specific novel types. Specifically, all concept descriptions of mentions with the same type  $t$  will be fused to  $C$  and regarded as the description of type  $t$ . And the type descriptions  $\mathcal{M} = \{(t, C)\}$  are constructed. Then the constructed type descriptions are incorporated to  $\mathbf{P}_{EG}$  to guide entity generation.

**Filtering Strategy.** Because of the diversified downstream novel types, SDNet may not have sufficient knowledge for describing some of these types, and therefore forcing SDNet to describe them can result in the inaccurate descriptions. To resolve this problem, we introduce a filtering strategy to



make SDNet able to reject generating unreliable descriptions. Specifically, SDNet is trained to generate `other` as the concept description for those uncertain instances. Given a novel type and a few illustrative instances, we will count the frequency of `other` in the concept descriptions from these instances. If the frequency of generating `other` on illustrative instances is greater than 0.5, we will remove the type description, and directly use the type name as `PEG`. We will describe how SDNet learns the filtering strategy in Section 4.1.

## 4 Learning

In this section, we first describe how to pre-train SDNet using large-scale external data, so that the common NER ability can be captured through the mention describing ability and the entity generation ability. Then we describe how to quickly adapt and transfer NER knowledge via fine-tuning. Figure 3 shows the two processes and we describe them as follows.

### 4.1 SDNet Pre-training

In SDNet, the NER ability consists of mention describing ability and entity generation ability, which can be effectively pre-trained by constructing corresponding datasets. This paper constructs datasets and pre-trains SDNet using the easily available and large-scale Wikipedia and Wikidata data.

**Entity Mention Collection.** For SDNet pre-training, we need to collect  $\langle e, T, X \rangle$  triples, where  $e$  is entity mention,  $T$  is entity types and  $X$  is sentence, such as  $\langle \text{J.K. Rowling}; \text{person, writer, ...}; \text{J.K. Rowling writes ...} \rangle$ . To this end, we use the 20210401 version of Wikipedia and Wikidata dump and collect triples by aligning facts in Wikidata and documents in Wikipedia and process as follows. 1) Firstly, we construct an entity type dictionary from Wikidata. We regard each item in Wikidata as an entity and use the “instance of”, “subclass of” and “occupation” property values as its corresponding entity types. To learn general NER knowledge, we use all entity types except whose instances are  $< 5$ . For the types whose names are longer than 3 tokens, we use their head words as the final type for simplicity, e.g., “state award of the Republic of Moldova” is converted to “state award”. In this way, we obtain a collection  $\mathbb{T}$  of 31K types which can serve as a solid foundation for universal NER. 2) Secondly, we collect the mentions of each entity

using its anchor texts in Wikipedia and the top 3 frequent noun phrase occurrences of its entry page (Li et al., 2010). Then for each mention, we identify its entity types by linking it to its Wikidata item’s types. If its Wikidata item doesn’t have a type, we assign its type as `other`. For each Wikipedia page, we split the text to sentences<sup>2</sup> and filter out sentences that have no entities. Finally, we construct a training dataset containing 56M instances.

**Type Description Building.** To pre-train SDNet, we need the concept descriptions  $\mathcal{M}^P = \{(t_i, C_i)\}$ , where  $t_i \in \mathbb{T}$ ,  $C_i$  is the related concepts of type  $t_i$ . This paper uses the collected entity types above as concepts, and builds the type description as follows. Given an entity type, we collect all its co-occurring entity types as its describing concepts. For example, `Person` can be described as  $\{\text{businessman, CEO, musician, musician...}\}$  by collecting the types of “Steve Jobs”:  $\{\text{person, businessman, CEO}\}$  and “Beethoven”:  $\{\text{person, musician, pianist}\}$ . In this way, for each entity type we have a describing concept set. Because some entity types have a very large describing concept set, we randomly sample no more than  $N$  (10 in this paper) concepts during pre-training for efficiency.

**Pre-training via Mention Describing and Entity Generation.** Given a sentence  $X$  with its mention-type tuples  $\{(e_i, T_i) | e_i \in E, T_i \subset \mathbb{T}\}$ , where  $T_i = \{t_i^1, \dots, t_i^{n_i}\}$  is the set of types of  $i$ -th entity mention  $e_i$ ,  $t_i^j$  is the  $j$ -th type of the  $e_i$ ,  $E = \{e_1, e_2, \dots\}$  is the set of entity mentions contained in  $X$ . Then we construct type descriptions, and transform these triples to pre-training instances. Specifically, for mention describing, some target entity mentions  $E'$  are sampled from  $E$  to put into prompt `PMD`. Then SDNet will take `PMD` and  $X$  to generate the corresponding types of sampled mentions  $E'$  as described in Section 3.3. For entity generation, positive type  $T_p$  and negative type  $T_n$  are sampled to construct the target-sampled type set  $T' = T_p \cup T_n$ , where  $T_p \subset T_1 \cup T_i \dots \cup T_k$ ,  $T_n \subset \mathbb{T} \setminus \{T_1 \cup T_i \dots \cup T_k\}$ . Next, the type set  $T'$  and their sampled concept description will be put into prompt `PEG`. Then SDNet will take prompt `PEG` and sentence  $X$  to generate the sequence as described in Section 3.2.

For each instance, SDNet generates two kinds of sequences:  $\tilde{\mathcal{Y}}_m^p$  for mention describing, and  $\tilde{\mathcal{Y}}_e^p$  for

<sup>2</sup>nltk.tokenize.punkt

		CoNLL	WNUT	Res	Movie1	Movie2	Re3d	I2B2	Onto	AVE
<b>Baselines</b>	RoBERTa (Huang et al., 2020)	53.5	25.7	48.7	51.3	/	/	36.0	57.7	/
	RoBERTa-DS (Huang et al., 2020)*	61.4	34.2	49.1	53.1	/	/	38.5	68.8	/
	Proto (Huang et al., 2020)	58.4	29.5	44.1	38.0	/	/	32.0	53.3	/
	Proto-DS (Huang et al., 2020)*	60.9	35.9	48.4	43.8	/	/	36.6	57.0	/
	spanNER (Wang et al., 2021)	71.1	25.8	49.1	/	65.4	/	/	67.3	/
	spanNER-DS (Wang et al., 2021)*	<b>75.6</b>	38.5	51.2	/	67.8	/	/	<b>71.6</b>	/
<b>Baselines [in-house]</b>	Bert-base	58.6	23.2	47.6	52.4	66.3	57.0	47.6	61.1	51.7
	T5-base	60.0	36.6	59.4	57.9	69.9	57.1	39.9	62.0	55.3
	T5-base-prompt	55.4	34.2	58.4	58.7	67.1	60.7	61.8	59.8	57.0
	T5-base-DS	68.2	34.9	59.7	58.4	70.8	56.0	34.1	58.8	55.1
<b>Ours</b>	SDNet	71.4	<b>44.1</b>	<b>60.7</b>	<b>61.3</b>	<b>72.6</b>	<b>65.4</b>	<b>64.3</b>	71.0	<b>63.8</b>

Table 1: Micro-F1 scores on 8 datasets in 5-shot setting. \* means these approaches use external distant supervision datasets to pre-train model different from SDNet. AVE are the average scores of these datasets.

entity generation. We use cross-entropy (CE) loss to train SDNet:

$$\mathcal{L}_p = \text{CE}(\widetilde{\mathcal{Y}}_m^p, \mathcal{Y}_m^p) + \text{CE}(\widetilde{\mathcal{Y}}_e^p, \mathcal{Y}_e^p) \quad (1)$$

Note that when constructing the target generation sequence  $\mathcal{Y}_e^p$ , the order of mentions depends on the order they appear in the original text.

## 4.2 Entity Recognition Fine-tuning

As described above, SDNet can directly recognize entities using manually designed type descriptions. But SDNet can also automatically build type descriptions using illustrative instances and be further improved by fine-tuning. Specifically, given annotated  $\langle e, T, X \rangle$  instances, we first construct the descriptions of different types, next build an entity generation prompt  $\mathbf{P}_{\text{EG}}$ , then generate sequence  $\mathcal{Y}_n^f$ . We fine-tune SDNet by optimizing:

$$\mathcal{L}_f = \text{CE}(\widetilde{\mathcal{Y}}_n^f, \mathcal{Y}_n^f) \quad (2)$$

We can see that, by fine-tuning SDNet, the entity generation process can better capture the associations between mentions and entity types.

# 5 Experiments

## 5.1 Settings

**Datasets.** Following previous studies, we use 8 benchmarks from different domains: 1) CoNLL2003 (Sang and Meulder, 2003); 2) WNUT17 (Derczynski et al., 2017); 3) Re3d (Science and Laborator, 2017); 4) MIT corpus (Liu et al., 2013a,b) includes three datasets: Res, Movie1(trivial10k13 version) and Movie2; 5) I2B2 (Stubbs and Uzuner, 2015); 6) OntoNotes5 (Ralph et al., 2013). Appendix shows detailed statistics of these datasets.

**Evaluation.** We conduct main experiments on 5-shot setting as previous work (Huang et al., 2020; Wang et al., 2021), and also ranging the shot size from 5 to 100, as well as full shot for further analysis. For k-shot setting, we sample k instances for each entity type from training set as support set to fine-tune models. Specifically, all pre-trained models are trained 300k steps, all datasets are fine-trained 50 epochs and more hyperparameters are shown in Appendix. The performance is evaluated by micro-F1 on test set, and a predicted entity is correct if its entity type and offsets both match the golden entity. To obtain the offset of each mention, we extract entity mentions and their types from the generated sentence, and locate them in the original sentence. And if they are repeated, we match them in order, that is, the i-th same mention in the generated sentence will be matched to the i-th same utterances in the original sentence. We run 10 times for each dataset and report the average F1 score as Huang et al. (2020) and Wang et al. (2021) did.

**Baselines.** We compare with following baselines: To evaluate the effect of pre-training for few-shot NER, we compare with baselines without NER-specific pre-training: 1) **BERT-base**, a traditional sequential BIO-based NER tagger (Wang et al., 2021) using pre-trained bert-base-uncased model. 2) **T5-base**, a generation-based NER baseline which uses the same generation format as SDNet but only using original t5-base model for generation. 3) **T5-base-prompt**, the prompt-extended version of T5-base which use entity types as prompt. To compare the effect of different knowledge transfer ways, we construct a distant supervision based baseline: 4) **T5-base-DS**, we further pre-train T5-base using the dataset collected in Section 4.1 as distantly supervised dataset. We

	WNUT			Re3d			Res			Movie1		
	P	R	F	P	R	F	P	R	F	P	R	F
<b>SDNet</b>	<b>54.78</b>	37.08	<b>44.06</b>	<b>63.67</b>	<b>67.22</b>	<b>65.39</b>	<b>63.99</b>	<b>57.88</b>	<b>60.74</b>	<b>63.54</b>	59.30	<b>61.33</b>
w/o desp	48.78	<b>39.51</b>	43.54	62.15	65.87	63.95	62.60	57.44	59.88	62.93	<b>59.61</b>	61.22
w/o joint	50.68	37.46	42.96	62.99	65.01	63.97	63.15	57.23	60.01	62.71	58.64	60.60
w/o filter	53.57	35.01	42.23	63.49	66.63	65.00	63.31	57.40	60.17	62.99	59.07	60.96

Table 2: Ablation experiments. SDNet is the full model, SDNet w/o filter is the same model fine-tuned without filtering strategy, SDNet w/o desp is the same model pre-trained and fine-tuned without description, and SDNet w/o joint is two models trained to perform mention description and entity generation separately.

also compare with several recent few-shot NER methods: 5) RoBERTa-based few-shot classifier **RoBERTa** and its distantly-supervised pre-trained version **RoBERTa-DS** (Huang et al., 2020). 6) Prototypical network based RoBERTa model **Proto** and its distantly supervised pre-training version **Proto-DS** (Huang et al., 2020). 7) MRC model **SpanNER** which needs to design the description for each label and its distantly supervised pre-training version **SpanNER-DS** (Wang et al., 2021). Notice that these methods mostly only focus on task-specific entity types, by contrast, this paper focuses on building a general few-shot NER model which can recognize entities universally.

## 5.2 Main Results

Table 1 shows the performances of SDNet and all baselines on 8 datasets. We can see that:

1) **By universally modeling and pre-training NER knowledge in a generation architecture, the self-describing network can effectively handle few-shot NER.** Compared with previous baselines, SDNet achieves competitive performance on all 8 datasets (new state-of-the-art on 6 datasets), and its performance is robust on different datasets.

2) **Due to the limited information problem, transferring external knowledge to few-shot NER models are critical.** Compared with BERT-base, T5-base, and T5-base-prompt, SDNet achieves 24%/16%/11% F1 improvements, which verified that SDNet provides a universal knowledge-enhanced foundation for NER and can adaptively transfer universal knowledge to enhance novel type recognition.

3) **Due to the knowledge mismatch, it is challenging to transfer external knowledge effectively to novel downstream types.** Using the same external knowledge sources, SDNet can achieve a 16% F1 improvement than T5-base-DS. We believe it is due to the noise, partially-labeled and heterogeneous problems in the external knowl-

edge sources, and SDNet can effectively address these issues.

## 5.3 Effects of Shot Size

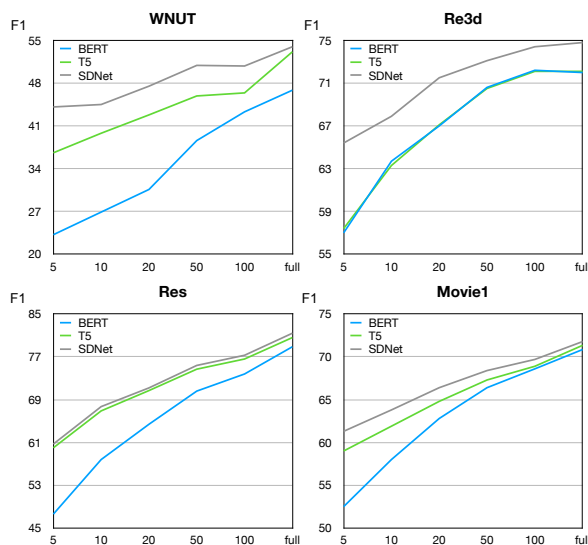


Figure 4: Performances of BERT, T5, SDNet with k-shot samples on WNUT, Re3d, Res, Movie1 dataset.

To verify the performance of SDNet under different shot settings, we compare the performance of BERT, T5, and SDNet with k-shot samples where k ranges from 5 to 100. From Figure 4 we can see that 1) SDNet can achieve better performance under all different shot settings. Furthermore, the improvements are more significant on low shot settings, which verified the intuitions behind SDNet; 2) Generation-based models usually achieve better performance than classifier-based BERT model. We believe this is because generation-based model can more efficiently capture the semantics of types by leveraging the label utterances, and therefore can achieve much better performance, especially in low-shot settings. 3) SDNet significantly outperforms T5 on almost all datasets except *Res*. This shows the effectiveness of the proposed self-describing mechanism. For *Res*, we find that the main reason why T5 can achieve close performance

to SDNet is the huge domain shifting between *Res* and Wikipedia. Such domain shifting makes SDNet frequently generate *other* for type descriptions, and therefore SDNet degrades to T5 in many cases. However, SDNet can still perform better than T5 on *Res*, which verifies the robustness of the proposed type description and the filtering strategy.

#### 5.4 Ablation Study

To analyze the effectiveness of type description, multi-task modeling and type description filtering, we conduct following ablation experiments: 1) SDNet w/o desp: we directly use entity type as prompt, without the universal concept description, e.g. { [EG] person; location; ... }; 2) SDNet w/o joint: we split SDNet into two individual generation network, one for mention description, the other for entity generation, and trained them using same resources as SDNet; 3) SDNet w/o filter: we use all the generated concept descriptions with no filtering strategy. From Table 2 we can see that:

**1) Type description is critical for SDNet to transfer knowledge and capture type semantics.**

By removing type description, the F1 of all datasets will decrease. We believe this is because 1) type description provides a common base for knowledge transferring, where all entity types are described using the same set of concepts; 2) the concept descriptions capture the semantics of entity types more accurately and precisely, which can better guide the NER process.

**2) Joint learning mention describing and entity generation processes in a unified generation network is effective to capture type semantics.** Compared with modeling two tasks separately, SDNet can achieve better performance. We believe this is because the two processes are symmetrical, and they can complement and promote each other.

**3) Filtering strategy can effectively alleviate the transferring of mismatched knowledge.** Removing the filtering strategy will undermine the performance on all 4 datasets. We believe this is because there exist some instances that can not be described based on the pre-trained SDNet knowledge. As a result, introducing filtering strategy can effectively prevent the mistaken knowledge transferring to these instances.

#### 5.5 Zero-shot NER with Manual Description

In this section, we adapt SDNet to zero-shot setting, to investigate whether SDNet can achieve

Types	w/o desp	Artifact	Few-shot
person	51.29	46.27	59.80
corporation	31.30	34.43	33.21
location	46.15	50.36	52.64
creative-work	12.29	14.38	27.69
group	19.93	24.45	24.10
product	19.10	23.08	23.63

Table 3: F1 score of each label in WNUT under zero-shot setting. **w/o desp** is the model pre-trained without description. **Artifact** is SDNet with manually designed concept description based on guideline.

<b>Text</b>	[Chris Hill] <sub>person</sub> was in [China] <sub>GPE</sub> [a few days ago] <sub>date</sub> .
<b>Input1</b>	{ [EG] GPE: {state, country, city, democracy, republic, community}; date: {}; } Chris Hill was in China ...
<b>Output1</b>	China is GPE. a few days ago is date.
<b>Input2</b>	{ [EG] person: {politician, actor, lawyer}; organization: {business, company}; } Chris Hill was in China ...
<b>Output2</b>	Chris Hill is person.

Table 4: Examples of the outputs of entity generation process. SDNet is fine-tuned on OntoNotes and the type description is automatically generated by SDNet via mention describing.

promising zero-shot performance without any illustrative instances. To this end, we conduct an experiment on WNUT by introducing manually created concepts as type descriptions based on annotation guideline, and the designed descriptions are shown in Appendix. Then we compare with the baseline without using type description, to see the effectiveness of the descriptions and whether SDNet can well-adapted to manually created descriptions.

From Table 3, we can see that SDNet can benefit from manual description significantly. Compared with SDNet without description, incorporating manual description can improve zero-shot performance on the majority of types. Furthermore, SDNet with manual description on zero-shot setting can achieve comparable performance with few-shot settings in many entity types. This demonstrates that type description is an effective way for model to capture the semantic of novel types, which verifies the intuition of SDNet.

#### 5.6 Effect of Entity Generation Prompt

Table 4 shows that by putting different types and its corresponding type descriptions to prompt, SDNet can generate different outputs according to the prompt. This verifies that SDNet can be controlled on-the-fly to generate different types of entities.



## 6 Conclusions

In this paper, we propose Self-describing Networks, a Seq2Seq generation model which can universally describe mentions using concepts, automatically map novel entity types to concepts, and adaptively recognize entities on-demand. A large-scale SDNet model is pre-trained to provide universal knowledge for downstream NER tasks. Experiments on 8 datasets show that SDNet is effective and robust. For future work, we will extend self-describing mechanism to other NLP tasks like event extraction (Paolini et al., 2021; Lu et al., 2021) and complex NER tasks like nested (Lin et al., 2019) or discontinuous NER (Yan et al., 2021b).

## 7 Acknowledgments

We thank all reviewers for their valuable comments. This work is supported by the National Key Research and Development Program of China (No. 2020AAA0106400), the National Natural Science Foundation of China under Grants no. U1936207, 62122077 and 62106251, and the Project of the Chinese Language Committee under Grant no. YB2003C002.

## 8 Ethics Consideration

This paper has no particular ethic consideration.

## 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1516–1528. Association for Computational Linguistics.
- Genady Beryozkin, Yoel Drori, Oren Gilon, Tzvika Hartman, and Idan Szpektor. 2019. [A joint named-entity recognizer for heterogeneous tag-sets using a tag hierarchy](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 140–150. 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, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 1835–1845. Association for Computational Linguistics.
- Hongliang Dai, Yangqiu Song, and Haixun Wang. 2021. [Ultra-fine entity typing with weak supervision from a masked language model](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 1790–1799. Association for Computational Linguistics.
- Sarkar Snigdha Sarathi Das, Arzoo Katiyar, Rebecca J Passonneau, and Rui Zhang. 2021. [Container: Few-shot named entity recognition via contrastive learning](#). *arXiv preprint arXiv:2109.07589*.
- Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. [Results of the WNUT2017 shared task on novel and emerging entity recognition](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text, NUT@EMNLP 2017, Copenhagen, Denmark, September 7, 2017*, pages 140–147. Association for Computational Linguistics.
- Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. [Prompt-learning for fine-grained entity typing](#). *CoRR*, abs/2108.10604.
- Alexander Fritzier, Varvara Logacheva, and Maksim Kretov. 2019. [Few-shot classification in named entity recognition task](#). In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, pages 993–1000.
- Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. [Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1381–1393. Association for Computational Linguistics.
- Jiaxin Huang, Chunyuan Li, Krishan Subudhi, Damien Jose, Shobana Balakrishnan, Weizhu Chen, Baolin Peng, Jianfeng Gao, and Jiawei Han. 2020. [Few-shot named entity recognition: A comprehensive study](#). *CoRR*, abs/2012.14978.
- Haoming Jiang, Danqing Zhang, Tianyu Cao, Bing Yin, and Tuo Zhao. 2021. [Named entity recognition with small strongly labeled and large weakly labeled data](#). 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 1775–1789, Online. Association for Computational Linguistics.
- Young-Bum Kim, Karl Stratos, Ruhi Sarikaya, and Minwoo Jeong. 2015. [New transfer learning techniques for disparate label sets](#). In *Proceedings*

- of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 473–482, Beijing, China. Association for Computational Linguistics.
- Dong-Ho Lee, Mahak Agarwal, Akshen Kadakia, Jay Pujara, and Xiang Ren. 2021. [Good examples make A faster learner: Simple demonstration-based learning for low-resource NER](#). *CoRR*, abs/2110.08454.
- Jing Li, Billy Chiu, Shanshan Feng, and Hao Wang. 2020a. Few-shot named entity recognition via meta-learning. *IEEE Transactions on Knowledge and Data Engineering*.
- Jing Li, Shuo Shang, and Ling Shao. 2020b. Metaner: Named entity recognition with meta-learning. In *Proceedings of The Web Conference 2020*, pages 429–440.
- Peng Li, Jing Jiang, and Yinglin Wang. 2010. [Generating templates of entity summaries with an entity-aspect model and pattern mining](#). In *ACL 2010, Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, July 11-16, 2010, Uppsala, Sweden*, pages 640–649. The Association for Computer Linguistics.
- Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020c. [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.
- Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019. [Sequence-to-nuggets: Nested entity mention detection via anchor-region networks](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5182–5192. Association for Computational Linguistics.
- Jingjing Liu, Panupong Pasupat, Scott Cyphers, and James R. Glass. 2013a. [Asgard: A portable architecture for multilingual dialogue systems](#). In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013, Vancouver, BC, Canada, May 26-31, 2013*, pages 8386–8390. IEEE.
- Jingjing Liu, Panupong Pasupat, Yining Wang, Scott Cyphers, and James R. Glass. 2013b. [Query understanding enhanced by hierarchical parsing structures](#). In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding, Olomouc, Czech Republic, December 8-12, 2013*, pages 72–77. IEEE.
- Qing Liu, Hongyu Lin, Xinyan Xiao, Xianpei Han, Le Sun, and Hua Wu. 2021. [Fine-grained entity typing via label reasoning](#). In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 4611–4622. Association for Computational Linguistics.
- Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. [Text2event: Controllable sequence-to-structure generation for end-to-end event extraction](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2795–2806. Association for Computational Linguistics.
- Ruotian Ma, Xin Zhou, Tao Gui, Yiding Tan, Qi Zhang, and Xuanjing Huang. 2021. [Template-free prompt tuning for few-shot NER](#). *CoRR*, abs/2109.13532.
- Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. [Coarse-to-Fine Pre-training for Named Entity Recognition](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.
- Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. 2019. [Reinforcement-based denoising of distantly supervised NER with partial annotation](#). In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP, DeepLo@EMNLP-IJCNLP 2019, Hong Kong, China, November 3, 2019*, pages 225–233. Association for Computational Linguistics.
- Rasha Obeidat, Xiaoli Z. Fern, Hamed Shahbazi, and Prasad Tadepalli. 2019. [Description-based zero-shot fine-grained entity typing](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 807–814. Association for Computational Linguistics.
- Giovanni Paolini, Ben Athiwaratkun, Jason Krone, Jie Ma, Alessandro Achille, Rishita Anubhai, Cícero Nogueira dos Santos, Bing Xiang, and Stefano Soatto. 2021. [Structured prediction as translation between augmented natural languages](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. [Distantly supervised named entity recognition using positive-unlabeled learning](#). In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 2409–2419. Association for Computational Linguistics.

- Ruizhi Qiao, Lingqiao Liu, Chunhua Shen, and Anton van den Hengel. 2016. [Less is more: Zero-shot learning from online textual documents with noise suppression](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2249–2257. IEEE Computer Society.
- Weischedel Ralph, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue et al. 2013. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium*.
- Scott E. Reed, Zeynep Akata, Honglak Lee, and Bernt Schiele. 2016. [Learning deep representations of fine-grained visual descriptions](#). In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 49–58. IEEE Computer Society.
- Marek Rei and Anders Søgaard. 2018. [Zero-shot sequence labeling: Transferring knowledge from sentences to tokens](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 293–302, New Orleans, Louisiana. Association for Computational Linguistics.
- 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, CoNLL 2003, Held in cooperation with HLT-NAACL 2003, Edmonton, Canada, May 31 - June 1, 2003*, pages 142–147. ACL.
- Defence Science and Technology Laborator. 2017. Relationship and entity extraction evaluation dataset.
- Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. [Learning named entity tagger using domain-specific dictionary](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2054–2064. Association for Computational Linguistics.
- Amber Stubbs and Özlem Uzuner. 2015. [Annotating longitudinal clinical narratives for de-identification: The 2014 i2b2/uthealth corpus](#). *J. Biomed. Informatics*, 58:S20–S29.
- Meihan Tong, Shuai Wang, Bin Xu, Yixin Cao, Minghui Liu, Lei Hou, and Juanzi Li. 2021. [Learning from miscellaneous other-class words for few-shot named entity recognition](#). 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 6236–6247, Online. Association for Computational Linguistics.
- Yaqing Wang, Haoda Chu, Chao Zhang, and Jing Gao. 2021. [Learning from language description: Low-shot named entity recognition via decomposed framework](#). *CoRR*, abs/2109.05357.
- Yongqin Xian, Subhabrata Choudhury, Yang He, Bernt Schiele, and Zeynep Akata. 2019. [Semantic projection network for zero- and few-label semantic segmentation](#). In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 8256–8265. Computer Vision Foundation / IEEE.
- Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. 2018. [Put it back: Entity typing with language model enhancement](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 993–998. Association for Computational Linguistics.
- Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021a. [A unified generative framework for various NER subtasks](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.
- Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021b. [A unified generative framework for various NER subtasks](#). In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 5808–5822. Association for Computational Linguistics.
- 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, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2159–2169. 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 2020, Online, November 16-20, 2020*, pages 6365–6375. Association for Computational Linguistics.
- Keunwoo Peter Yu and Yi Yang. 2020. [One model to recognize them all: Marginal distillation from NER models with different tag sets](#). *CoRR*, abs/2004.05140.

Wenkai Zhang, Hongyu Lin, Xianpei Han, and Le Sun. 2021a. *De-biasing distantly supervised named entity recognition via causal intervention*. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 4803–4813. Association for Computational Linguistics.

Wenkai Zhang, Hongyu Lin, Xianpei Han, Le Sun, Huidan Liu, Zhicheng Wei, and Nicholas Jing Yuan. 2021b. *Denoising distantly supervised named entity recognition via a hypergeometric probabilistic model*. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14481–14488. AAAI Press.

## A Experimental Details

	Pre-train	Fine-tune
batch size	16	4
Learning rate	5e-5	1e-4
Optimizer	AdamW	AdamW
schedule	-	linear schedule
warmup rate	-	6%

Table 5: Hyperparameter settings.

**Detailed Hyperparameters** SDNet is initialized with T5-base. Table 5 shows the hyperparameters of SDNet. When fine-tuning, we set the same hyperparameters for T5-base, T5-base-prompt and T5-base-DS as for SDNet. For Bert-base, the learning rate is 2e-5 and batch size is 8.

**Dataset Analysis** Table 6 shows the statistics of the dataset we use.

**Manually Designed Type Descriptions** Table 7 shows the manually designed type descriptions for WNUT.

Dataset	Domain	#Types	#Test
WNUT	Social Media	6	1287
CoNLL	News	4	3453
re3d	Defense	10	200
Res	Review	8	1521
Moive1	Review	12	1953
Movie2	Review	12	2443
I2B2	Medical	23	43697
Onto	General	18	8262

Table 6: Statistics of 8 public datasets.

<b>person</b>	writer, entrepreneur, association football player, actor, businessperson, baseball player, politician
<b>corporation</b>	digital media, website, organization, trademark, entrepreneur, airline, social media
<b>location</b>	port, park, city, country, road, province, state, mountain
<b>creative-work</b>	television program, audiovisual work, album, release, film
<b>group</b>	group, band, basketball team, football club, sports team
<b>product</b>	medication, chemical compound, electronic game, video game, smartphone model, chemical substance

Table 7: Manually designed concept descriptions in WNUT.