Template-Based Named Entity Recognition Using BART

Leyang Cui[†◇♡], **Yu Wu**[‡], **Jian Liu**[◇♡], **Sen Yang**[◇♡], **Yue Zhang**[◇♡] *
[†]Zhejiang University
[‡]Microsoft Research Asia
[◇]School of Engineering, Westlake University
[♡]Institute of Advanced Technology, Westlake Institute of Advanced Study
{cuileyang,liujian,yangsen,zhangyue}@westlake.edu.cn Wu.Yu@microsoft.com

Abstract

There is a recent interest in investigating few-shot NER, where the low-resource target domain has different label sets compared with a resource-rich source domain. Existing methods use a similarity-based metric. However, they cannot make full use of knowledge transfer in NER model parameters. To address the issue, we propose a template-based method for NER, treating NER as a language model ranking problem in a sequence-to-sequence framework, where original sentences and statement templates filled by candidate named entity span are regarded as the source sequence and the target sequence, respectively. For inference, the model is required to classify each candidate span based on the corresponding template scores. Our experiments demonstrate that the proposed method achieves 92.55% F1 score on the CoNLL03 (rich-resource task), and significantly better than fine-tuning BERT 10.88%, 15.34%, and 11.73% F1 score on the MIT Movie, the MIT Restaurant, and the ATIS (low-resource task), respectively.

1 Introduction

Named entity recognition (NER) is a fundamental task in natural language processing, which identifies mention spans from text inputs according to pre-defined entity categories (Tjong Kim Sang and De Meulder, 2003), such as location, person, organization, etc. The current dominant methods use a sequential neural network such as BiLSTM (Hochreiter and Schmidhuber, 1997) and BERT (Devlin et al., 2019) is used to represent the input text, and softmax (Chiu and Nichols, 2016; Strubell et al., 2017; Cui and Zhang, 2019) or CRF (Lample et al., 2016; Ma and Hovy, 2016; Luo et al., 2020) output layers to assign named entity tags (e.g. organization, person and location) or non-entity tags on each input token. Such a system is illustrated in Figure 2(a).

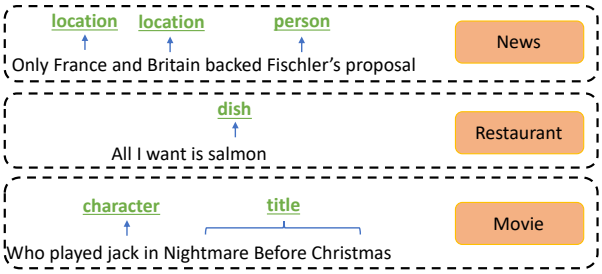


Figure 1: Example of NER on different domains.

Neural NER models require large labeled training data, which can be available for certain domains such as news, but scarce in most other domains. Ideally, it would be desirable to transfer knowledge from the resource-rich news domain so that a model can be used in target domains based on a few labeled instances. In practice, however, a challenge is that entity categories can be different across different domains. As shown in Figure 1, the system is required to identify *location* and *person* in the news domain, but *character* and *title* in the movie domain. Both a softmax layer and CRF layer require a consistent label set between training and testing. As a result, given a new target domain, the output layer needs adjustment and training must be conducted again using both source and target domain, which can be costly.

A recent line of work investigates the setting of few-shot NER by using distance metrics (Wiseman and Stratos, 2019; Yang and Katiyar, 2020; Ziyadi et al., 2020). The main idea is to train a similarity function based on instances in the source domain, and then make use of the similarity function in the target domain as a nearest neighbor criterion for few-shot NER.

Compared with traditional methods, distance-based methods largely reduce the domain adapta-

预训练和微调的名实体label要一样，否则就要调整输出层，用源，目的域训练

*Corresponding Author

tion cost, especially for scenarios where the number of target domains is large. Their performance under standard in-domain settings, however, is relatively weak. In addition, their domain adaptation power is also limited in two aspects. First, labeled instances in the target domain are used to find the best hyper-parameter settings for heuristic nearest neighbor search, but are not for updating the network parameters of the NER model. While being less costly, these methods cannot improve the neural representation for cross-domain instances. Second, these methods rely on similar textual patterns between the source domain and the target domain. This strong assumption may hinder the model performance when the target-domain writing style is different from the source domain.

To address these issues, we investigate a template-based method for exploiting the few-shot learning potential of generative pre-trained language models to sequence labeling. Specifically, as shown in Figure 2, BART (Lewis et al., 2020) is fine-tuned with pre-defined templates filled by corresponding labeled entities. For example, we can define templates such as "⟨`candidate_span`⟩ *is a* ⟨`entity_type`⟩ *entity*", where ⟨`entity_type`⟩ can be "*person*" and "*location*", etc. Given the sentence "*ACL will be held in Bangkok*", where "*Bangkok*" has a gold label "*location*", we can train BART using a filled template "*Bangkok is a location entity*" as the decoder output for the input sentence. In terms of non-entity spans, we use a template "⟨`candidate_span`⟩ *is not a named entity*", so that negative output sequences can also be sampled. During inference, we enumerate all possible text spans in the input sentence as named entity candidates, classifying them into entities or non-entities based on BART scores on templates.

泛化能力强

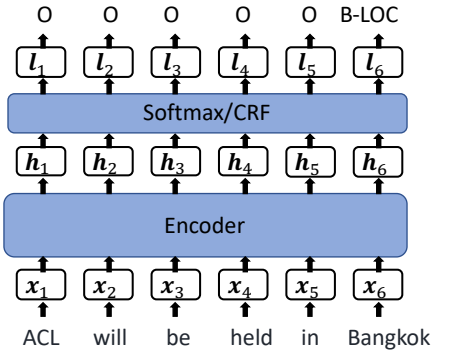
The proposed method has three advantages. First, due to the good generalization ability of pre-trained models (Brown et al., 2020; Gao et al., 2020), the network can effectively leverage labeled instances in the new domain for tine-tuning. Second, compared with distance-based methods, our method is more robust even if the target domain and source domain have a large gap in writing style. Third, compared with traditional methods (pre-trained model with a softmax/CRF), our method can be applied to arbitrary new categories of named entities without changing the output layer, and therefore allows continual learning (Lin et al., 2020).

We conduct experiments in both resource-rich and few-shot settings. Results show that our methods give competitive results with state-of-the-art label-dependent approaches on the news dataset CoNLL03 (Tjong Kim Sang and De Meulder, 2003), and significantly outperforms Wiseman and Stratos (2019), Ziyadi et al. (2020) and Huang et al. (2020) when it comes to few-shot settings. To the best of our knowledge, we are the first to employ a generative pre-trained language model to address a few-shot sequence labeling problem. We release our code at `https://github.com/Nealcly/templateNER`.

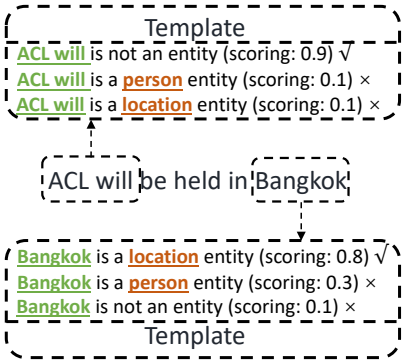
2 Related Work

Neural methods have given competitive performance in NER. Some methods (Chiu and Nichols, 2016; Strubell et al., 2017) treat NER as a local classification problem at each input token, while other methods use CRF (Ma and Hovy, 2016) or a sequence-to-sequence framework (Zhang et al., 2018; Liu et al., 2019). Cui and Zhang (2019) and Gui et al. (2020) use a label attention network and Bayesian neural networks, respectively. Yamada et al. (2020) use entity-aware pre-training and obtain state-of-the-art results on NER. These approaches are similar to ours in the sense that parameters can be tuned in supervised learning, but unlike our method, they are designed for prescribed named entity types, which makes their domain adaptation costly with new few-shot entity types.

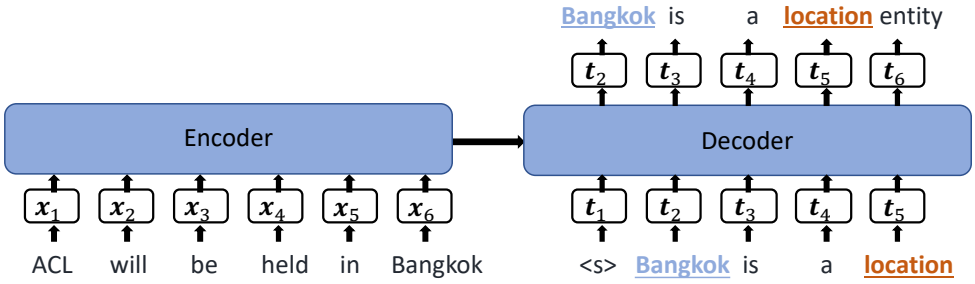
Our work is motivated by distance-based few-shot NER, which aims to minimize domain-adaptation cost. Wiseman and Stratos (2019) copy the token-level label from nearest neighbors by retrieving a list of labeled sentences. Yang and Katiyar (2020) improve Wiseman and Stratos (2019) by using a Viterbi decoder to capture label dependencies estimated from the source domain. Ziyadi et al. (2020) follow a two-step approach (Lin et al., 2019; Xu et al., 2017), which first detects spans boundary and then recognizes entity types by comparing the similarity with the labeled instance. While not updating the network parameters for NER, these methods rely on similar name entity patterns between the source domain and the target domain. One exception is Huang et al. (2020), who investigate noisy supervised pre-training and self-training method by using external noisy web NER data. Compared to their method, our method does not



(a) Traditional sequence labeling method.



(b) Inference of template-based method.



(c) Training of template-based method. The template we use here is "$\langle x_{i:j} \rangle$ is a $\langle y_k \rangle$ entity".

Figure 2: Overview of NER methods.

rely on self training on external data, yet yields better results.

There is a line of work using templates to solve natural language understanding tasks. The basic idea is to leverage information from pre-trained models, by defining specific sentence templates in a language modeling task. Brown et al. (2020) first use prompt for few-shot learning in text classification tasks. Schick and Schütze (2020) rephrase inputs as cloze questions for text classification. Schick et al. (2020) and Gao et al. (2020) extend Schick and Schütze (2020) by automatically generating label words and templates, respectively. Petroni et al. (2019) extract relation between entities from BERT by constructing cloze-style templates. Sun et al. (2019) use templates to construct auxiliary sentences, and transform aspect sentiment task as a sentence-pair classification task. Our work is in line with exploiting pre-trained language model for templates-based NLP. While previous work considers sentence-level task as masked language modeling or uses language models to score a whole sentence, our method uses a language model to assign a score for each span given an input sentence. To our knowledge, we are the first to apply template-based method to sequence labeling.

3 Background

We give the formal definition of few shot named entity recognition in Section 3.1 and traditional sequence labeling methods in Section 3.2.

3.1 Few shot Named Entity Recognition

Suppose that we have a rich-resource NER dataset $\mathbb{H} = \{(\mathbf{X}_1^H, \mathbf{L}_1^H), ..., (\mathbf{X}_I^H, \mathbf{L}_I^H)\}$, where $\mathbf{X}^H = \{x_1^H, \ldots, x_n^H\}$ is a sentence and $\mathbf{L}^H = \{l_1^H, \ldots, l_n^H\}$ is its corresponding label sequence. We use $\mathcal{V}^H$ to denote the label set of the rich-resource dataset ($\forall l_i^H, l_i^H \in \mathcal{V}^H$). In addition, we have a low-resource NER dataset, $\mathbb{L} = \{(\mathbf{X}_1^L, \mathbf{Y}_1^L), ..., (\mathbf{X}_J^L, \mathbf{Y}_J^L)\}$, and the number of its labelled sequence pairs is quite limited compared with the rich-resource NER dataset (i.e., $J \ll I$). Regarding the low-resource domain, the target label vocabulary $\mathcal{V}^L$ ($\forall l_i^L, l_i^L \in \mathcal{V}^L$) might be different from $\mathcal{V}^H$ (Figure 1). Our goal is to train an accurate and robust NER model with $\mathbb{L}$ and $\mathbb{H}$ for the low-resource domain.

3.2 Traditional Sequence Labeling Methods.

Traditional methods (Figure 2(a)) regard NER as a sequence labeling problem, where each output

label consists of a sequence segmentation component B (beginning of an entity), I (internal word in an entity), O (not an entity), and an entity type tag such as "*person*" and "*location*". For example, the tag "*B-person*" indicates the first word in a person type entity and the tag "*I-location*" indicates a token of a location entity not at the beginning. Formally, given $x_{1:n}$, the sequence labeling method calculates

$$\begin{aligned} \mathbf{h}_{1:n} &= \textsc{Encoder}(x_{1:n}) \\ p(\hat{l}_c) &= \textsc{softmax}(\mathbf{h}_c \mathbf{W}_{\mathcal{V}^R} + \mathbf{b}_{\mathcal{V}^R}) \ (c \in [1, ..., n]) \end{aligned} \tag{1}$$

where d_h is the hidden dimension of the encoder, $\mathbf{W}_{\mathcal{V}^R} \in \mathbb{R}^{d_h \times |\mathcal{V}^R|}$ and $\mathbf{b}_{\mathcal{V}^R} \in \mathbb{R}^{|\mathcal{V}^R|}$ are trainable parameters, and $\hat{l}_c$ is the label estimation for x_c. We use BERT (Devlin et al., 2019) and BART (Lewis et al., 2020) as our ENCODER to learn the sequence representation.

A standard method for NER domain adaptation is to train a model using source-domain data $\mathbb{R}$ first, before further tuning the model using target domain instances $\mathbb{P}$, if available. However, since the label sets can be different, and consequently the output layer parameters ($\mathbf{W}_{\mathcal{V}^R} \in \mathbb{R}^{d_h \times |\mathcal{V}^R|}$, $\mathbf{b}_{\mathcal{V}^R} \in \mathbb{R}^{|\mathcal{V}^R|}$ and $\mathbf{W}_{\mathcal{V}^P} \in \mathbb{R}^{d_h \times |\mathcal{V}^P|}$, $\mathbf{b}_{\mathcal{V}^P} \in \mathbb{R}^{|\mathcal{V}^P|}$) can be different across domains. We train $\mathbf{W}_{\mathcal{V}^P}$ and $\mathbf{b}_{\mathcal{V}^P}$ from scratch using $\mathbb{P}$. However, this method does not fully exploit label associations (e.g., the association between "*person*" and "*character*"), nor can it be directly used for zero-shot cases, where no labeled data in the target domain is available.

4 Template-Based Method

We consider NER as a language model ranking problem under a seq2seq framework. The source sequence of the model is an input text $\mathbf{X} = \{x_1, \ldots, x_n\}$ and the target sequence $\mathbf{T}_{y_k, x_{i:j}} = \{t_1, \ldots, t_m\}$ is a template filled by candidate text span $x_{i:j}$ and the entity type y_k. We first introduce how to create templates in Section 4.1, and then show the inference and training details in Section 4.2 and Section 4.3, respectively.

4.1 Template Creation

We manually create the template, which has one slot for `candidate_span` and another slot for the `entity_type` label. We set a one to one mapping function to transfer the label set $\mathbf{L} = \{l_1, \ldots, l_{|L|}\}$ (e.g., l_k="LOC") to a natural word set $\mathbf{Y} = \{y_1, \ldots, y_{|L|}\}$ (e.g. y_k="*location*"), and use words to define templates $\mathbf{T}^+_{y_k}$ (e.g. ⟨`candidate_span`⟩ is a location entity.). In addition, we create a non-entity template $\mathbf{T}^-$ for none of the named entity (e.g., ⟨`candidate_span`⟩ is not a named entity.). This way, we can obtain a list of templates $\mathbf{T} = [\mathbf{T}^+_{y_1}, \ldots, \mathbf{T}^+_{y_{|L|}}, \mathbf{T}^-]$. In Figure 2(c), the template $\mathbf{T}_{y_k, x_{i:j}}$ is "$\langle x_{i:j} \rangle$ is a $\langle y_k \rangle$" and $\mathbf{T}^-_{x_{i:j}}$ is "$\langle x_{i:j} \rangle$ is not a named entity", where $x_{i:j}$ is a candidate text span.

一个span最多有八个模板

4.2 Inference

We first enumerate all possible spans in the sentence $\{x_1, \ldots, x_n\}$ and fill them in the prepared templates. For efficiency, we restrict the number of n-grams for a span from one to eight, so $8n$ templates are created for each sentence. Then, we use the fine-tuned pre-trained generative language model to assign a score for each template $\mathbf{T}_{y_k, x_{i:j}} = \{t_1, \ldots, t_m\}$, formulated as

$$f(\mathbf{T}_{y_k, x_{i:j}}) = \sum_{c=1}^{m} \log p(t_c | t_{1:c-1}, \mathbf{X}) \tag{2}$$

We calculate a score $f(\mathbf{T}^+_{y_k, x_{i:j}})$ for each entity type and $f(\mathbf{T}^-_{x_{i:j}})$ for the none entity type by employing any pre-trained generative language model to score ~~templates. Then we assign $x_{i:j}$ the entity~~ type with the largest score to the text span. In this paper, we take BART as the pre-trained generative language models.

Our datasets do not contain nested entities. If two spans have text overlap and are assigned different labels in the inference, we choose the span with higher score as the final decision to avoid possible prediction contradictions. For instance, given the sentence "*ACL will be held in Bangkok*", the $n-$gram "*in Bangkok*" and "*Bangkok*" can be labeled "ORG" and "LOC", respectively, by using local scoring function $f(\cdot)$. In this case, we compare $f(\mathbf{T}^+_{\text{ORG,"in Bangkok"}})$ and $f(\mathbf{T}^+_{\text{LOC,"Bangkok"}})$, and choose the label which has a larger score to make the global decision.

4.3 Training

Gold entities are used to create template during training. Suppose that the entity type of $x_{i:j}$ is y_k. We fill the text span $x_{i:j}$ and the entity type y_k into $\mathbf{T}^+$ to create a target sentence $\mathbf{T}^+_{y_k, x_{i:j}}$. Similarly, if the entity type of $x_{i:j}$ is a none entity text span, the target sentence $\mathbf{T}^-_{x_{i:j}}$ is obtained by filling $x_{i:j}$ into $\mathbf{T}^-$. We use all gold entities in the training set to construct $(\mathbf{X}, \mathbf{T}^+)$ pairs, and additionally create negative samples $(\mathbf{X}, \mathbf{T}^-)$ by randomly sampling

Entity Template $\mathbf{T}^+$	None-Entity Template $\mathbf{T}^-$	Dev F1
⟨candidate_span⟩ is a ⟨entity_type⟩ entity	⟨candidate_span⟩ is not a named entity	95.27
The entity type of ⟨candidate_span⟩ is ⟨entity_type⟩	The entity type of ⟨candidate_span⟩ is none entity	95.15
⟨candidate_span⟩ belongs to ⟨entity_type⟩ category	⟨candidate_span⟩ belongs to none category	88.42
⟨candidate_span⟩ should be tagged as ⟨entity_type⟩	⟨candidate_span⟩ should tagged as none entity	76.80

Table 1: Resulting using different templates.

non-entity text spans. The number of negative pairs is 1.5 times that of positive pairs.

Given a sequence pair $(\mathbf{X}, \mathbf{T})$, we feed the input $\mathbf{X}$ to the encoder of the BART, and then we obtain hidden representations of the sentence

$$\mathbf{h}^{enc} = \textsc{Encoder}(x_{1:n}) \tag{3}$$

At the c th step of the decoder, $\mathbf{h}^{enc}$ and previous output tokens $t_{1:c-1}$ are then as inputs, yielding a representation using attention (Vaswani et al., 2017)

$$\mathbf{h}_c^{dec} = \textsc{Decoder}(\mathbf{h}^{enc}, t_{1:c-1}) \tag{4}$$

The conditional probability of the word t_c is defined as:

$$p(t_c|t_{1:c-1}, \mathbf{X}) = \textsc{softmax}(\mathbf{h}_c^{dec}\mathbf{W}_{lm} + \mathbf{b}_{lm}) \tag{5}$$

where $\mathbf{W}_{lm} \in \mathbb{R}^{d_h \times |\mathcal{V}|}$ and $\mathbf{b}_{lm} \in \mathbb{R}^{|\mathcal{V}|}$. $|\mathcal{V}|$ represents the vocab size of pre-trained BART. The cross-entropy between the decoder's output and the original template is used as the loss function.

$$\mathcal{L} = -\sum_{c=1}^{m} \log p(t_c|t_{1,c-1}, \mathbf{X}) \tag{6}$$

4.4 Transfer Learning

Given a new domain $\mathbb{P}$ with few-shot instances, the label set $\mathbf{L}^P$ (Section 4.1) can be different from what has been used for training the NER model. We thus fill the templates with the new domain label set for both training and testing, with the rest of the model and algorithms unchanged. In particular, given a small amount of $(\mathbf{X}^P, \mathbf{T}^P)$, we create sequence pairs with the method described above for the low-resource domain, and fine-tuning the NER model trained on the rich-source domain. This process has low cost, yet can effectively transfer label knowledge. Because the output of our method is a natural sentence instead of specific labels, both resource-rich and low-resource label vocabulary are subset of the pre-trained language model vocabulary ($\mathcal{V}^R, \mathcal{V}^P \subsetneqq \mathcal{V}$). This allows our method to make use of label correlations such as "*person*" and "*character*", and "*location*" and "*city*", for enhancing the effect of transfer learning across domains.

5 Experiments

We compare template-based BART with several baselines on both resource-rich settings and few-shot settings. We use the CoNLL2003 (Tjong Kim Sang and De Meulder, 2003) as the resource-rich dataset. Following Ziyadi et al. (2020) and Huang et al. (2020), we use MIT Movie Review (Liu et al., 2013), MIT Restaurant Review (Liu et al., 2013) and ATIS (Hakkani-Tur et al., 2016) as the cross-domain few-shot dataset. Regarding the cross-domain transfer, there are unseen entity types in the three target few-shot datasets. Details of our training details and dataset statistics are shown in Appendix.

5.1 Template Influence

There can be different templates for expressing the same meaning. For instance "⟨candidate_span⟩ *is a person*" can also be expressed by "⟨candidate_span⟩ *belongs to the person category*". We investigate the impact of manual templates using the CoNLL03 development set. Table 1 shows the performance impact of different choice of templates. For instance, "⟨candidate_span⟩ *should be tagged as* ⟨entity_type⟩" and "⟨candidate_span⟩ *is a* ⟨entity_type⟩ *entity*" give 76.80% and 95.27% F1 score, respectively, indicating the template is a key factor that influences the final performance. Based on the development results, we use the top performing template "⟨candidate_span⟩ is a ⟨entity_type⟩ entity" in our experiments.

5.2 CoNLL03 Results

Standard NER setting. We first evaluate the performance under the standard NER setting on CoNLL03. The results are shown in Table 2, where state-of-the-art methods are also compared. In particular, the sequence labeling BERT gives a strong baseline, F1 score at 91.73%. We can see that even though the template-based BART is designed for few-shot named entity recognition, it performs competitively in resource-rich setting as well. For instance, our method outperforms sequence label-

ing BERT by 1.80% on recall, which shows that our method is more effective in identifying the named entity, but also selecting irrelevant span. Noted that though both sequence labeling BART and template-based BART make use of BART decoder representations, their performances have a large gap, where the latter outperforms the former by absolutely 1.30% on F1 score, demonstrating the effectiveness of the template-based method. The observation is consistent with that of Lewis et al. (2020), which shows that BART is not the most competitive for sequence classification. This may result from the nature of its seq2seq-based denoising autoencoder training, which is different from masked language modeling for BERT.

bart在sequence label中不是最好的

To explore if templates are complementary for each other, we train three models using the first three templates reported in Table 1, and adopt an entity-level voting method to ensemble these three models. There is a 1.21% precision increase using ensemble, which shows that different templates capture different type of knowledge. Finally, method achieves a 92.55 % F1 score by leveraging three templates, which is highly competitive with the best reported score. For computational efficiency, we use a single model for the subsequent few-shot experiments.

In domain few-shot NER setting. We construct a few-shot learning scenario on the CoNLL03, where the number of training instances for some specific categories is quite limited by down-sampling. In particular, we set "MISC" and "ORG" as the resource-rich entities, and "LOC" and "PER" as the low-resource entities. We down-sample the CoNLL03 training set, yielding 3,806 training instances, which includes 3,925 "ORG", 1,423 "MISRC", 50 "LOC" and 50 "PER". Since the text style is consistent in rich-resource and low-resource entity categories, we call the scenario in domain few-shot NER.

As shown in Table 3, sequence labeling BERT and template-based BART show similar performance in resource-rich entity types, while our method significantly outperforms BERT by 11.26 and 12.98 F1 score in "LOC" and "MISC", respectively. It demonstrates that our method has a stronger modeling capability for in-domain few-shot NER, and indicates that the proposed method can better transfer the knowledge between different entity categories.

Traditional Models	P	R	F
Yang et al. (2018)	-	-	90.77
Ma and Hovy (2016)	-	-	91.21
Gui et al. (2020)	-	-	92.02
Yamada et al. (2020)*	-	-	94.30
Sequence Labeling BERT	91.93	91.54	91.73
Sequence Labeling BART	89.60	91.63	90.60
Few-shot Friendly Models	**P**	**R**	**F**
Wiseman and Stratos (2019)	-	-	89.94
Template BART	90.51	93.34	91.90
multi-template BART	91.72	93.40	92.55

Table 2: Model performance on the CoNLL03 .The original result of BERT (Devlin et al., 2019) was not achieved with the current version of the library as discussed and reported by Stanislawek et al. (2019), Akbik et al. (2019) and Gui et al. (2020). * indicates training on external data.

Models	PER	ORG	LOC*	MISC*	Overall
BERT	75.71	77.59	60.72	60.39	69.62
Ours	84.49	72.61	71.98	73.37	75.59

Table 3: In-domain Few-shot performance on the CoNLL03. * indicates it is a few-shot entity type.

5.3 Cross-domain Few-Shot NER Result

We evaluate the model performance when the target entity types are different from the source-domain, and only a small amount of labeled data is available for training. We simulate the cross-domain low-resource data scenarios by random sampling training instances from a large training set as the training data in the target domain. We use different numbers of instances for training, randomly sampling a fixed number of instances per entity type (10, 20, 50, 100, 200, 500 instances per entity type for MIT Movie and MIT restaurant, and 10, 20, 50 instances per entity type for ATIS). If an entity has a smaller number of instances than the fixed number to sample, we use all of them for training. The results on few-shot experiments using MIT Movie, MIT Restaurant and ATIS are shown in Table 4, where the methods of Wiseman and Stratos (2019), Ziyadi et al. (2020) and Huang et al. (2020) are also compared.

We first consider a training-from-scratch setting, where no source-domain data is used. Distance-based methods cannot suit this setting. Compared with the traditional sequence labeling BERT method, our method can make better use of few-shot data. In particular, with as few as 20 instances per entity type, our method gives a F1 score of 57.1%, higher than BERT using 100 instances per entity type on MIT Restaurant.

We further investigate how much knowledge can

Source	Methods	10	20	50	100	200	500
MIT Movie							
None	Sequence Labeling BERT	25.2	42.2	49.64	50.7	59.3	74.4
	Template-based BART	37.3	48.5	52.2	56.3	62.0	74.9
CoNLL03	Wiseman and Stratos (2019)	3.1	4.5	4.1	5.3	5.4	8.6
	Ziyadi et al. (2020)	40.1	39.5	40.2	40.0	40.0	39.5
	Huang et al. (2020)*	36.4	36.8	38.0	38.2	35.4	38.3
	Sequence Labeling BERT	28.3	45.2	50.0	52.4	60.7	76.8
	Sequence Labeling BART	13.6	30.4	47,8	49.1	55.8	66.9
	Template-based BART	42.4	54.2	59.6	65.3	69.6	80.3
MIT Restaurant							
None	Sequence Labeling BERT	21.8	39.4	52.7	53.5	57.4	61.3
	Template-based BART	46.0	57.1	58.7	60.1	62.8	65.0
CoNLL03	Wiseman and Stratos (2019)	4.1	3.6	4.0	4.6	5.5	8.1
	Ziyadi et al. (2020)	27.6	29.5	31.2	33.7	34 .5	34.6
	Huang et al. (2020)	46.1	48.2	49.6	50.0	50.1	
	Sequence Labeling BERT	27.2	40.9	56.3	57.4	58.6	75.3
	Sequence Labeling BART	8.8	11.1	42.7	45.3	47.8	58.2
	Template-based BART	53.1	60.3	64.1	67.3	72.2	75.7
ATIS							
None	Sequence Labeling BERT	44.1	76.7	90.7	-	-	-
	Template-based BART	71.7	79.4	92.6	-	-	-
CoNLL03	Wiseman and Stratos (2019)	6.7	8.8	11.1	-	-	-
	Ziyadi et al. (2020)	17.4	19.8	22.2	-	-	-
	Huang et al. (2020)	71.2	74.8	76.0	-	-	-
	Sequence Labeling BERT	53.9	78.5	92.2	-	-	-
	Sequence Labeling BART	51.3	74.4	89.9	-	-	-
	Template-based BART	77.3	88.9	93.5	-	-	-

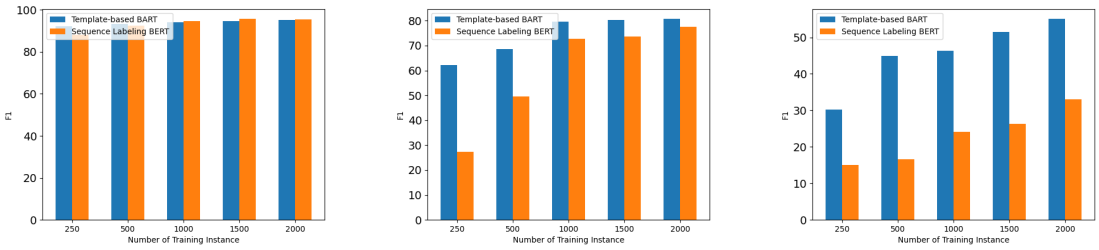
Table 4: Cross-domain few-shot NER performance on different test sets. * indicates training on external data. 10 indicates 10 instances for each entity types.

be transferred from the news domain (CoNLL03). In this setting, we further train the model which is trained on the news domain. It can be seen from the Table 4 that on all the three datasets, the few-short learning methods outperform sequence label- RT and BART methods when the number ing instances is small. For example, when re only 10 training instances, the method di et al. (2020) gives a F1 score of 40.1% on MIT Movie, as compared to 28.3% by BERT, despite that BERT requires re-training with a different output layer on both CoNLL03 and MIT Movie. However, as the number of training instances increase, the advantage of baseline few-shot methods decreases. When the number of instances grows as large as 500, BERT outperforms all existing methods. Our method is effective in both 10 instances and 500 instances, outperforming both BERT and baseline few-shot methods.

Compared with the distance-based method (Wiseman and Stratos, 2019; Ziyadi et al., 2020; Huang et al., 2020), our method shows more improvement when the number of target-domain labeled data increases, because the distance-based just optimizes its searching threshold rather dating its neural network parameters. We that the performance of distance-based methods remains the same as the labeled data increasing. For example, the performance of Huang et al. (2020) increases only 1.9% F1 score when the number of instances per entity type increase from 10 to 500. Both BERT and our method perform better than training from scra odel average increases 6.6, 6.9 and 5.4 MIT restaurant, MIT movie and ATI ely, which is significantly higher than 3.1, 1.9 and 4.3 F1 score in BERT. This shows that our model is more successful in transferring knowledge learned from the source domain. One possible explanation is that our model makes more use of the correlations between different entity type labels in the vocabulary as mentioned earlier, which BERT cannot achieve due to treating the output as discrete class labels.

5.4 Discussion

Impact of entity frequencies in training data. To explore the relation between recognition accuracy and the frequency of an entity type in training, we split ATIS test set into three subset based on the entity frequency in training. The most 33% frequency entities are put into *high frequency* subset, the last 33% frequency entities are put into *low frequency* subset, and the remaining are put into *mid frequency* subset. Figure 3 shows the F1 score of



(a) High Frequency. (b) Mid Frequency. (c) Low Frequency.

Figure 3: Comparison of F1 with different frequency entity types on ATIS.

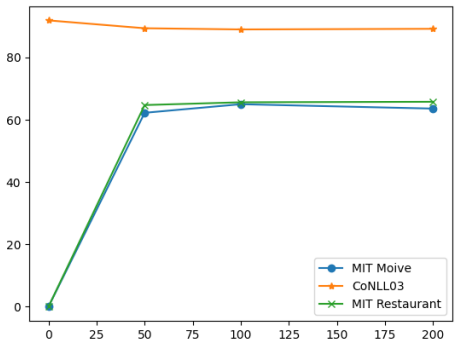
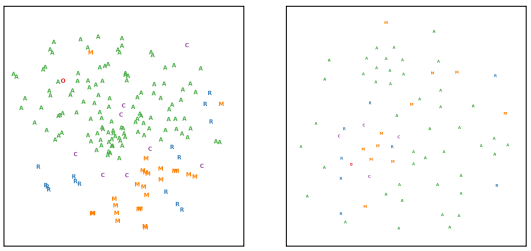


Figure 4: Continual learning experiments.



(a) Sequence labeling BERT. (b) Template-based BART.

Figure 5: Visualization of the output embedding. "C"–CoNLL03, "A"–ATIS, "M"–MIT Movie, "R"–MIT Restaurant.

BERT and our method against the number of training instance in the three subsets. As the number of training instances increases, the performance of all models increases. Our method outperforms sequence labeling BERT by a large margin, especially on the *mid frequency* and *low frequency* subsets, which demonstrates that our method is more robust in few-shot settings.

ual Learning NER In continual learning (Lin et al., 2020), all the baselines that we Table 4 face limitations. The sequence labeling BERT method needs re-training using all training data each time a new entity type is encountered, which is highly expensive. The distance based methods cannot make use of all available data for improving representation learning. Figure 4 shows performance of our method on MIT movie, MIT restaurant and CoNLL03, when we continue to train our CoNLL03 model on both MIT movie and MIT restaurant. The performance on the CoNLL03 only slightly decrease when we continue training the model on the MIT Movie and MIT Restaurant dataset, demonstrating the robustness of our method in the continual learning setting.

Visualization. We explore why our model works well in the low-resource domain by visualizing the output layer. We train BERT and our method on all four datasets, and use t-SNE (van der Maaten and Hinton, 2008) to project the output layer into 2-dimensions, where the output layer for sequence labeling BERT and template-based BART are $\mathbf{W}_{\mathcal{V}^R}$ in Eq 1 and $\mathbf{W}_{lm}$ in Eq 5, respectively. In Figure 5, each dot represents a row in the output matrix (corresponding to a label embedding). We can see that output layer embeddings of BERT are clustered based on dataset while the vectors of template-based BART are sparsely distributed in the space. It indicates that our output matrix is more domain independent, and our method enjoys better generalization ability across different domains.

Error Types We find that most mistakes are caused by the domain distance between high-resource data and low-source NER data. As shown in Fig 5, Template-based methods rely on label semantics. If the embedding of the word with a few-shot labels is far from that with in-domain labels, the model shows lower performance on that label type. Taking 50 examples per entity type on MIT movie as an example, "ACTOR" is similar to

"PERSON" in CoNLL03, and achieves 84.81 F1. The embedding of "SONG" is far from the existing labels in CoNLL03, and only achieves 34.97 F1. In contrast, sequence labeling BERT does not suffer from this distance, because BERT cannot draw label correlation between two domains, it achieves 53.98 and 40.13 on "ACTOR" and "SONG", respectively.

6 Conclusion

We investigated template-based few-shot NER using BART as the backbone model. In contrast to the traditional sequence labeling methods, our method is more powerful on few-shot NER, since it can be fine-tuned for the target domain directly when new entity categories exist. Experiment results show that our model achieves competitive results on a rich-resource NER benchmark, and outperforms traditional sequence labeling methods and distance-based methods significantly on the cross-domain and few-shot NER benchmarks.

Acknowledgments

We thank all anonymous reviewers for their constructive comments. This work is supported by National Science Foundation of China (Grant No. 61976180).

References

Alan Akbik, Tanja Bergmann, and Roland Vollgraf. 2019. Pooled contextualized embeddings for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 724–728, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. 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 4115–4128, Hong Kong, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners.

Tao Gui, Jiacheng Ye, Qi Zhang, Zhengyan Li, Zichu Fei, Yeyun Gong, and Xuanjing Huang. 2020. Uncertainty-aware label refinement for sequence labeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2316–2326, Online. Association for Computational Linguistics.

Dilek Hakkani-Tur, Gokhan Tur, Asli Celikyilmaz, Yun-Nung Chen, Jianfeng Gao, Li Deng, and Ye-Yi Wang. 2016. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm. In *Proceedings of Interspeech*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

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.

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.

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.

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 Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192, Florence, Italy. Association for Computational Linguistics.

Shaofu Lin, Jiangfan Gao, Shun Zhang, Xiaobo He, Ying Sheng, and Jianhui Chen. 2020. A continuous learning method for recognizing named entities by integrating domain contextual relevance measurement and web farming mode of web intelligence. In *World Wide Web*.

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

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI-2020)*.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey E. Hinton. 2008. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? 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 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference.

Tomasz Stanislawek, Anna Wróblewska, Alicja Wójcicka, Daniel Ziembicki, and Przemyslaw Biecek. 2019. Named entity recognition - is there a glass ceiling? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 624–633, Hong Kong, China. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2670–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota. 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 at HLT-NAACL 2003*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008.

Sam Wiseman and Karl Stratos. 2019. Label-agnostic sequence labeling by copying nearest neighbors. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5363–5369, Florence, Italy. Association for Computational Linguistics.

Mingbin Xu, Hui Jiang, and Sedtawut Watcharawittayakul. 2017. A local detection approach for named entity recognition and mention detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1237–1247, Vancouver, Canada. Association for Computational Linguistics.

Ikuya Yamada, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yuji Matsumoto. 2020. LUKE: Deep contextualized entity representations with entity-aware self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*,

pages 3879–3889, 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.

Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. 2018. Learning tag dependencies for sequence tagging. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4581–4587. International Joint Conferences on Artificial Intelligence Organization.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition.