# Named Entity Recognition with Context-Aware Dictionary Knowledge

**Chuhan Wu**[†], **Fangzhao Wu**[‡], **Tao Qi**[†], **Yongfeng Huang**[†]
[†]Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
[‡]Microsoft Research Asia, Beijing 100080, China
{wuchuhan15, wufangzhao, taoqi.qt}@gmail.com
yfhuang@tsinghua.edu.cn

## Abstract

Named entity recognition (NER) is an important task in the natural language processing field. Existing NER methods heavily rely on labeled data for model training, and their performance on rare entities is usually unsatisfactory. Entity dictionaries can cover many entities including both popular ones and rare ones, and are useful for NER. However, many entity names are context-dependent and it is not optimal to directly apply dictionaries without considering the context. In this paper, we propose a neural NER approach which can exploit dictionary knowledge with contextual information. We propose to learn context-aware dictionary knowledge by modeling the interactions between the entities in dictionaries and their contexts via context-dictionary attention. In addition, we propose an auxiliary term classification task to predict the types of the matched entity names, and jointly train it with the NER model to fuse both contexts and dictionary knowledge into NER. Extensive experiments on the CoNLL-2003 benchmark dataset validate the effectiveness of our approach in exploiting entity dictionaries to improve the performance of various NER models.

## 1 Introduction

Named entity recognition (NER) aims to extract entity names from texts and classify them into several pre-defined categories, such as person, location and organization (Levow, 2006). It is an important task in natural language processing, and a prerequisite for many downstream applications such as entity linking (Derczynski et al., 2015) and relation extraction (Lin et al., 2016; Luo et al., 2018; Zeng et al., 2018). Thus, NER is a hot research topic. In this paper, we focus on the English NER task.

Many methods have been proposed for English NER, and most of them model this task as a word-level sequence labeling problem (Chiu and Nichols, 2016). For example, Ma and Hovy (2016) proposed a CNN-LSTM-CRF model for English NER. They used CNN to learn word representations from characters, LSTM to model the contexts of words, and CRF to decode labels. These existing NER methods usually rely on massive labeled data for model training, which is costly and time-consuming to annotate. When training data is scarce, their performance usually significantly declines (Peng et al., 2019). In addition, their performance on recognizing entities that rarely or do not appear in training data is usually unsatisfactory (Wang et al., 2019).

Fortunately, many large-scale entity dictionaries such as Wikipedia (Higashinaka et al., 2012) and Geonames[1] are off-the-shelf, and they can be easily derived from knowledge bases and webpages (Neelakantan and Collins, 2014). These entity dictionaries contain both popular and rare entity names, and can provide important information for NER models to identify these entity names. There are a few researches on incorporating entity dictionary into NER (Liu et al., 2019; Magnolini et al., 2019) and most of them are based on dictionary matching features. For example, Wang et al. (2019) proposed to combine token matching features with token embeddings and LSTM outputs. However, in many cases entities are context-dependent. For instance, in Table 1, the word "Jordan" can be a person name or a location name in different contexts. Thus, it is not optimal to directly apply entity dictionaries to NER without considering the contexts.

[1]https://www.geonames.org

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

| | | |
|---|---|---|
| 1 | **Jordan** won against **Houston**.<br>He will give talks in **Jordan** and **Houston**. | Red: PER<br>Orange: ORG<br>Blue: LOC |
| 2 | **Brown** is the former prime minister.<br>**Brown** shoes are my favourate. | |

Table 1: Two examples of context-dependent entities.

In this paper, we propose a neural approach for named entity recognition with context-aware dictionary knowledge (CADK). We propose to exploit dictionary knowledge in a context-aware manner by modeling the relatedness between the entity names matched by entity dictionaries and their contexts. In addition, we propose an auxiliary term classification task to predict the types of the matched entity names in different contexts. Besides, we propose a unified framework to jointly train the NER model and the term classification model to incorporate entity dictionary knowledge and contextual information into the NER model. Extensive experiments show our approach can effectively exploit entity dictionaries to improve the performance of various NER models and reduce their dependence on labeled data.

## 2 Related Work

Named entity recognition is usually modeled as a sequence labeling problem (Wan et al., 2011). Many traditional NER methods are based on statistical sequence modeling methods, such as Hidden Markov Models (HMM) and Conditional Random Fields (CRF) (Cohen and Sarawagi, 2004; Ratinov and Roth, 2009; Passos et al., 2014; Arora et al., 2019). Usually, a core problem in these methods is how to build the feature vector for each word, and these features are traditionally constructed via manual feature engineering (Ratinov and Roth, 2009). For example, Ratinov and Roth (2009) used many features such as word n-grams, gazetteers and prediction histories as the word features. Passos et al. (2014) used features such as character n-grams, word types, capitalization pattern and lexicon matching features. They also incorporated lexicon embedding learned by skip-gram model to enhance the word representations. Designing these hand-crafted features usually needs a huge amount of domain knowledge. In addition, the feature vectors may be very sparse and their dimensions can be huge.

In recent years, many neural network based NER methods have been proposed (Collobert et al., 2011; Lample et al., 2016; Chiu and Nichols, 2016; Ma and Hovy, 2016; Peters et al., 2017; Li et al., 2017; Rei, 2017; Peters et al., 2018; Akbik et al., 2018; Lin and Lu, 2018; Clark et al., 2018; Chen et al., 2019; Zhu and Wang, 2019; Devlin et al., 2019). For example, Lample et al. (2016) proposed to use LSTM to learn the contextual representation of each token based on global context in sentences and use CRF for joint label decoding. Chiu and Nichols (2016) proposed to use CNN to learn word representations from original characters and then learn contextual word representation using Bi-LSTM. Ma and Hovy (2016) proposed to combine the CNN-LSTM framework with CRF for better performance. Peters et al. (2017) proposed a semi-supervised approach named TagLM for NER by pre-training a language model on a large corpus to provide contextualized word representations. Devlin et al. (2019) proposed a bidirectional pre-trained language model named BERT, which can empower downstream tasks like NER by using deep Transformers (Vaswani et al., 2017) to model contexts accurately. However, these neural network based methods heavily rely on labeled sentences to train NER models, which need heavy effort of manual annotation. In addition, their performance on recognizing entities which rarely or do not appear in labeled data is usually unsatisfactory (Wang et al., 2019).

There are several approaches on utilizing entity dictionaries for named entity recognition (Cohen and Sarawagi, 2004; Lin et al., 2007; Yu et al., 2008; Rocktäschel et al., 2013; Passos et al., 2014; Song et al., 2015; Wang et al., 2019; Liu et al., 2019). In traditional methods, dictionaries are often incorporated as additional features. For example, Cohen et al. (2004) proposed to extract dictionary features based on entity matching and similarities, and they incorporated these features into an HMM based model. There are also a few methods to incorporate dictionary knowledge into neural NER models (Chiu and Nichols, 2016; Wang et al., 2019; Liu et al., 2019). For example, Wang et al. (2019) proposed to incorporate dictionaries into neural NER model for detecting clinical entities. They manually designed several features

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

based on the matches with a clinical dictionary and then concatenated these features with the embedding vector as the input of the LSTM-CRF model. These methods rely on domain knowledge to design these dictionary based features, and these handcrafted features may not be optimal. Different from these methods, in our approach we introduce a term-level classification task to exploit the useful information in entity dictionary without manual feature engineering. We jointly train our model in both the NER and term classification tasks to enhance the performance of NER model in an end-to-end manner.

There are also a few methods that explore to incorporate dictionary knowledge into Chinese NER models in an end-to-end manner by using graph neural networks (Sui et al., 2019; Gui et al., 2019). For example, Sui et al. (2019) propose a character-based collaborative graph neural network to learn the representations of characters and words matched by dictionaries from three word-character graphs, i.e., a containing graph that describes the connection between characters and matched words, a transition graph that builds the connections between characters and the nearest contextual matched words, and a Lattice graph that connects each word with its boundary characters. However, these methods mainly model the interactions between matched entities and their local contexts, while ignore the relations between entities and their long-distance contexts. Different from these methods, our approach can model the interactions between the matched terms with the global contexts via entity-dictionary attention.

## 3 CADK Approach for NER

In this section, we introduce our NER approach with Context-Aware Dictionary Knowledge (CADK). The architecture of our approach is illustrated in Fig. 1. Our approach mainly contains five components, i.e., *text representation*, *term representation*, *context-dictionary attention*, *term classification* and *sequence tagging*. Next, we introduce the details of each module as follows.

### 3.1 Text Representation

The first module is a text representation model, which is used to learn the contextual representation of each word in an input text. It can be implemented by various neural text representation models, such as CNN (Zhu and Wang, 2019), LSTM (Huang et al., 2015) and GRU (Peters et al., 2017) or pre-trained language models like ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). We denote the word sequence of the input text as $[w_1, w_2, ...w_N]$, where $N$ is the number of words. The text representation model outputs a sequence that contains the contextual representation of each word, which is denoted as $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N]$.

### 3.2 Term Representation

The second module is *term representation*, which is used to obtain the representations of the terms matched by the entity dictionaries. Usually, entity dictionaries contain both popular (e.g., America) and rare entity names (e.g., Chatham), and can help NER models recognize these entity names correctly. Thus, entity dictionaries have the potential to improve the performance of NER and reduce the dependence on labeled data. To incorporate useful information in entity dictionaries, we use them to match the input text and obtain a candidate list with $M$ entity terms. We denote the word sequence of the $i_{th}$ term as $[w_{i1}, w_{i2}, ...w_{iP}]$, where $P$ represents the number of words in this term. In the *term representation* module, we first use a word embedding layer to convert the sequence of words in each term into a sequence of low-dimensional vectors. The word embedding parameters in this layer are shared with the *text representation* model. The word embedding sequence of the $i_{th}$ term is denoted as $[\mathbf{w}_{i1}, \mathbf{w}_{i2}, ...\mathbf{w}_{iP}]$. Then, we apply a word-level Bi-GRU network to the word embedding sequence of each term to learn a hidden term representation. The GRU layer scans the word embedding sequence of each term in two directions, and combines the last hidden states in both directions as the representation of this term. For the $i_{th}$ term, its representation is denoted as $\mathbf{t}_i$. We denote the sequence of the representations of the $M$ matched terms as $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_M]$.

### 3.3 Context-Dictionary Attention

The third module is *context-dictionary attention*. Many entity names are context-dependent. For example, in the sentence "Jordan is a famous NBA player", the word "Jordan" is in a person name, while it is



Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

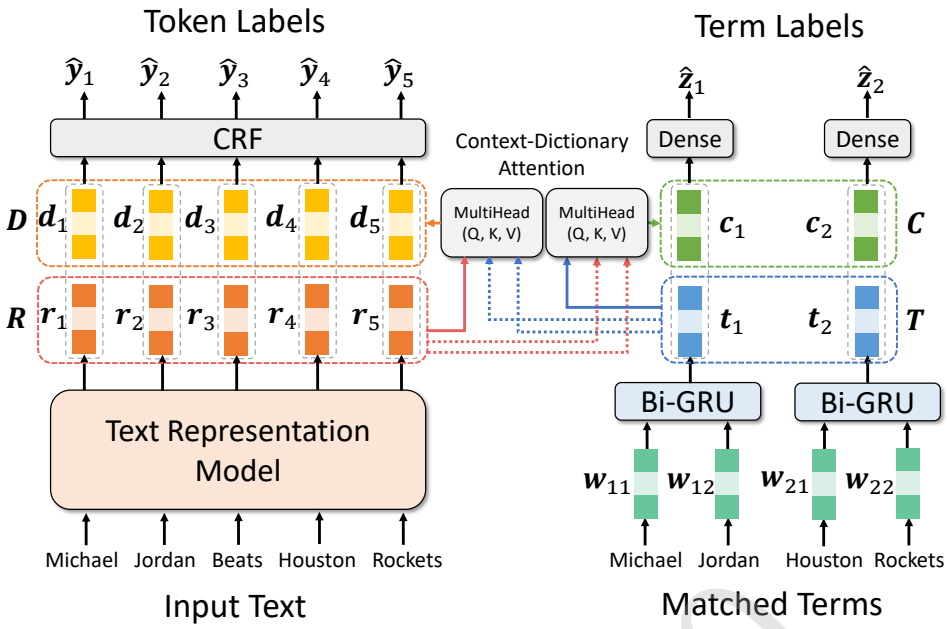



Figure 1: The architecture of our CADK approach.

also frequently used as a location name. Thus, we propose to incorporate dictionary knowledge in a context-aware manner by modeling the relationships between the matched entity terms and their contexts. It is used to model the interactions between terms matched by dictionaries with the contexts in sentences. Usually, entity names may interact with other words in the same text, and such interactions are important for recognizing these entities. For example, in the sentence "Jordan is a basketball player", the interaction between the entity "Jordan" and the word "player" is very informative for identifying the type of this entity is "person". In addition, an entity may interact with multiple words. For instance, in the sentence "He travels from Houston to Seattle", the interactions between the entity "Houston" and its contexts like "travels" and "Seattle" are useful clues for recognizing this entity. Motivated by these observations, we propose a context-dictionary attention module to model the interactions between the terms matched by dictionaries with all words in texts. The context-dictionary attention network takes both the sequences of word representations $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N]$ and term representations $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_M]$ ($N$ and $M$ are numbers of words and terms) as inputs, and outputs dictionary-aware representations of words in texts (denoted as $\mathbf{D}$) and context-aware representations of terms (denoted as $\mathbf{C}$). We use the multi-head productive attention mechanism (Vaswani et al., 2017) to model the interactions between terms and contexts. The dictionary-aware word representation sequence $\mathbf{D}$ is computed as follows:

$$\mathbf{D}^i = \text{Softmax}[\mathbf{W}_Q^i \mathbf{R}(\mathbf{W}_K^i \mathbf{T})^T](\mathbf{W}_V^i \mathbf{T}), \tag{1}$$

$$\mathbf{D} = \text{Concat}(\mathbf{D}^1, \mathbf{D}^2, ..., \mathbf{D}^h), \tag{2}$$

where $\mathbf{W}_Q^i$, $\mathbf{W}_K^i$, and $\mathbf{W}_V^i$ respectively stand for the parameters in the $i_{th}$ head for transforming the query, key and value, $h$ represents the number of parallel attention heads. The context-aware term representation sequence $\mathbf{C}$ is computed in a similar way as follows:

$$\mathbf{C}^i = \text{Softmax}[\mathbf{U}_Q^i \mathbf{T}(\mathbf{U}_K^i \mathbf{R})^T](\mathbf{U}_V^i \mathbf{R}), \tag{3}$$

$$\mathbf{C} = \text{Concat}(\mathbf{C}^1, \mathbf{C}^2, ..., \mathbf{C}^h), \tag{4}$$

where $\mathbf{U}_Q^i$, $\mathbf{U}_K^i$, and $\mathbf{U}_V^i$ are parameters. We concatenate $\mathbf{D}$ with the word representations $\mathbf{R}$, and $\mathbf{C}$ with the term representations $\mathbf{T}$, in a position-wise manner. In this way, entity dictionary with contextual information can be incorporated into a neural NER model.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

### 3.4 Term Classification

The fourth module is *term classification*, which is used to classify the types of the terms matched by dictionaries based on the representations of terms and their interactions with the contexts. To fully exploit the useful information in the entity dictionary, we propose an auxiliary term classification task which predicts the type of the entity names matched by the entity dictionary. For example, in the sentence "Michael Jordan Beats Houston Rockets", if the terms "Michael Jordan" and "Houston Rockets" are matched by the dictionary, our model is required to classify the types of these terms in the context of this sentence. We use a dense layer with the softmax activation function to classify the type of each term as follows:

$$\hat{z}_i = \text{softmax}(\mathbf{U}[\mathbf{c}_i; \mathbf{t}_i] + \mathbf{v}), \tag{5}$$

where $\mathbf{U}$ and $\mathbf{v}$ are parameters, $\mathbf{c}_i$ is the context-aware representation of the $i_{th}$ term, and $\hat{z}_i$ is the predicted type label of this term. The gold type label of the matched term can be derived from the token labels of the input sentence. For example, if the label sequence of a sentence is "O-BLOC-ELOC-O", we can know that the gold type of the entity in this sentence is "location". The loss function of the term classification task is the cross-entropy of the gold and the predicted labels of all terms, which is evaluated as follows:

$$\mathcal{L}_{Term} = -\sum_{i=1}^{S}\sum_{j=1}^{M}\sum_{k=1}^{K} \hat{z}_{ijk} \log(z_{ijk}), \tag{6}$$

where $S$ is the number of sentences for model training, $K$ is the number of entity categories, $z_{ijk}$ and $\hat{z}_{ijk}$ are the gold and predicted type labels of the $j_{th}$ term from the $i_{th}$ sentence in the $k_{th}$ category.

### 3.5 Sequence Tagging

The last module is *sequence tagging*. Usually the label at each position may have relatedness with the previous ones. For example, in the *BIOES* tagging scheme, the label "I-LOC" can only appear after "B-LOC" and "I-LOC". Thus, a CRF layer is usually employed to jointly decode the label sequence. Given a tag sequence $\mathbf{y} = [y_1, y_2, ..., y_N]$, the score of the tag sequence $\mathbf{y}$ in sentence $\mathbf{x}$ is defined as:

$$\mathbf{s}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} U_{i,y_i} + \sum_{i=1}^{N-1} A_{y_i,y_{i+1}}, \tag{7}$$

where $U_{i,y_i}$ is the unary score of assigning the tag $y_i$ to the $i_{th}$ token, and $A_{y_i,y_{i+1}}$ represents the score of jumping from tag $y_i$ to $y_{i+1}$. The unary score $\mathbf{U}_i$ is calculated as:

$$\mathbf{U}_i = \mathbf{W}_u \mathbf{h}_i + \mathbf{b}_u, \tag{8}$$

where $\mathbf{W}_u$ and $\mathbf{b}_u$ are trainable parameters. In CRF, the likelihood probability of the tag sequence $\mathbf{y}$ is formulated as:

$$p(\mathbf{y}|\mathbf{x}) = \frac{e^{\mathbf{s}(\mathbf{x},\mathbf{y})}}{\sum_{\mathbf{y}' \in \mathcal{Y}_\mathbf{x}} e^{\mathbf{s}(\mathbf{x},\mathbf{y}')}}, \tag{9}$$

where $\mathcal{Y}_\mathbf{x}$ represents the set of all possible tag sequences. Then the loss function of the NER task is evaluated as:

$$\mathcal{L}_{NER} = -\sum_{y_i \in \mathcal{S}} \log(p(\mathbf{y}_i|\mathbf{x}_i)), \tag{10}$$

where $\mathcal{S}$ denotes the training dataset, and $\mathbf{y}_i$ is the ground-truth tag sequence of sentence $\mathbf{x}_i$.

To incorporate the useful information in entity dictionary into NER models more effectively, we propose a unified framework based on multi-task learning to jointly train our model in both NER and term classification tasks. The final loss function is the weighted summation of the NER and term classification loss, which is formulated as follows:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{NER} + \lambda \mathcal{L}_{Term}, \tag{11}$$

where $\mathcal{L}_{NER}$ is the loss of CRF model, $\lambda \in [0, 1]$ is a coefficient to control the relative importance of the term classification task.



Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - Novermber 1, 2020.

| Model | 10% | | | 25% | | | 100% | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F | P | R | F | P | R | F |
| LSTM-CRF | 84.23 | 88.22 | 86.18 | 87.75 | 87.86 | 87.81 | 90.75 | 90.14 | 90.36 |
| LSTM-CRF+Feature | 84.90 | 89.02 | 86.91 | 88.33 | 88.40 | 88.37 | 91.14 | 90.18 | 90.66 |
| LSTM-CRF+GNN | 85.54 | 88.74 | 87.11 | 88.53 | 88.56 | 88.54 | 90.99 | 90.51 | 90.75 |
| LSTM-CRF+CADK | 85.94 | 89.27 | 87.58 | 89.34 | 88.72 | 89.03 | 91.58 | 90.81 | 91.19 |
| TagLM | 85.63 | 88.70 | 87.14 | 88.64 | 89.05 | 88.85 | 92.01 | 91.40 | 91.71 |
| TagLM+Feature | 85.77 | 90.14 | 87.90 | 89.44 | 89.25 | 89.35 | 92.41 | 91.64 | 92.02 |
| TagLM+GNN | 86.27 | 90.02 | 88.10 | 89.79 | 89.34 | 89.56 | 92.62 | 91.91 | 92.26 |
| TagLM+CADK | 86.56 | 90.68 | 88.57 | 89.98 | 90.14 | 90.06 | 93.03 | 92.33 | 92.68 |
| ELMo | 85.34 | 89.24 | 87.25 | 88.76 | 89.13 | 88.95 | 92.42 | 92.23 | 92.30 |
| ELMo+Feature | 86.01 | 89.96 | 87.94 | 89.51 | 89.39 | 89.45 | 92.73 | 92.19 | 92.46 |
| ELMo+GNN | 86.71 | 89.97 | 88.31 | 89.70 | 89.65 | 89.68 | 92.92 | 92.28 | 92.60 |
| ELMo+CADK | 87.09 | 90.36 | 88.70 | 90.40 | 89.82 | 90.11 | 93.49 | 92.57 | 93.03 |
| BERT | 84.76 | 87.87 | 86.29 | 87.91 | 88.11 | 88.01 | 91.89 | 91.23 | 91.49 |
| BERT+Feature | 85.48 | 88.86 | 87.14 | 88.60 | 88.43 | 88.51 | 91.99 | 91.41 | 91.70 |
| BERT+GNN | 85.73 | 88.72 | 87.20 | 88.65 | 88.90 | 88.77 | 92.12 | 91.64 | 91.88 |
| BERT+CADK | 86.20 | 89.30 | 87.72 | 89.19 | 89.32 | 89.26 | 92.40 | 92.00 | 92.20 |

Table 2: Performance of different NER methods under different ratios of training data. P, R, F respectively stand for precision, recall and Fscore.

## 4 Experiments

### 4.1 Dataset and Experimental Settings

Our experiments were conducted on the CoNLL-2003 dataset (Tjong Kim Sang and De Meulder, 2003), which is a widely used benchmark dataset for NER. This dataset contains four different types of named entities, i.e., locations, persons, organizations, and miscellaneous entities that do not belong in the three previous categories. Following previous works (Ratinov and Roth, 2009), we used the BIOES labeling scheme. In our experiments, we used an entity dictionary provided by (Higashinaka et al., 2012), which is derived from the WikiPedia database. This dictionary contains 297,073,139 entity names. The coefficient $\lambda$ in Eq. (11) was 0.4. We used Adam (Kingma and Ba, 2014) with gradient norms clipped at 5.0 as the optimizer for model training, and the learning rate was 0.001. The batch size was 64. These hyperparameters were tuned on the validation set. Each experiment was repeated 5 times independently, and the average performance in terms of precision, recall and Fscore were reported.

### 4.2 Comparison with Baseline Methods

To verify the effectiveness of the proposed *CADK* method, we compare several popular models and their variants using different methods for incorporating entity dictionaries. The methods to be compared including: (1) LSTM-CRF (Huang et al., 2015), a neural NER method that uses LSTM to learn word representations and CRF to decode labels; (2) TagLM (Peters et al., 2017), a neural NER model which uses GRU networks and a pre-trained language model to learn word representations, and uses CRF to decode labels; (3) ELMo (Devlin et al., 2019), a pre-trained language model with bidirectional deep LSTM network. We apply an LSTM-CRF network based on the contextualized word embeddings generated by the ELMo model; (4) BERT (Devlin et al., 2019), a pre-trained language model with bidirectional transformers. We fine-tune the BERT-base version in the NER task; The methods for incorporating entity dictionaries including: (a) Feature (Wang et al., 2019), incorporating entity dictionaries using feature engineering. We combines the dictionary matching features with the hidden representations learned by the aforementioned methods; (b) GNN (Sui et al., 2019), using graph neural networks to incorporate entity dictionary knowledge; (c) CADK, our proposed method with context-aware dictionary knowledge.

We randomly sampled different ratios (i.e., 10%, 25% and 100%) of samples from the data for model

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - Novermber 1, 2020.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

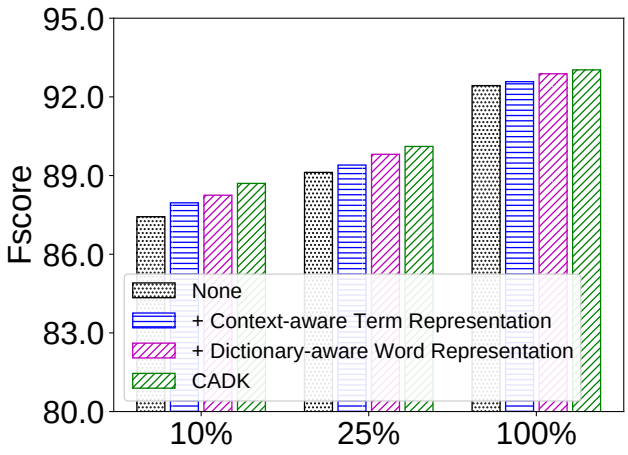



Figure 2: Effectiveness of the context-dictionary attention module.

training to evaluate these methods under different amounts of labeled data. The results are summarized in Table 2.[2] From Table 2, we find that when the training data is scarce, the performance of the methods without dictionary knowledge declines significantly. This is probably because these neural network based methods are data-intensive and require a large amount of labeled data for model training. When training data is scarce, many entities in the test set are unseen in the training data, making it difficult for existing NER methods to recognize them. Compared with methods without dictionaries, the methods that consider dictionary knowledge achieve better performance, and their advantage is larger when training data is more scarce. This is probably because incorporating dictionary knowledge can help recognize unseen or rare entities more effectively, which can reduce the dependency on labeled data. In addition, compared with the methods using dictionary matching features, the methods that can model the contexts of matched entities (*GNN* and *CADK*) perform better. This is probably because manually crafted features may be not optimal to utilize entity dictionaries, and the contexts of the matched entity names in different texts are not considered. Besides, our *CADK* method is better than the *GNN* method in exploiting dictionary knowledge for NER. Different from the *GNN* method that can only model the local contexts of matched entity names, in our approach we use the context-dictionary attention model to capture the global contexts of the matched terms, and we jointly train our model in both NER and term classification tasks to incorporate dictionary knowledge in a unified framework. Thus, our method can exploit dictionary information more accurately to improve neural NER model.

### 4.3 Effectiveness of Context-Dictionary Attention

In this section, we conduct several ablation studies to validate the effectiveness of the context-dictionary attention module in our *CADK* method. Since it mainly aims to generate the dictionary-aware word representation and the context-aware term representation, we compare the performance of *ELMo-CADK* under different ratios of training data by removing one or both of them. The results are shown in Fig. 3. According to the results, we find that the dictionary-aware word representation can effectively improve the performance of our approach. This is because the dictionary-aware word representation encodes the information of the entities matched by dictionaries, which is helpful for recognizing them more accurately. In addition, incorporating the context-aware term representation can also improve the model performance. This is because many entities are context-dependent, and modeling their relations with the contexts is beneficial for NER. These results show the effectiveness of context-dictionary attention in injecting context-aware dictionary knowledge into neural NER models.

[2]The performance of BERT is surprisingly unsatisfactory though we used the officially released model and carefully tuned hyperparameters.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

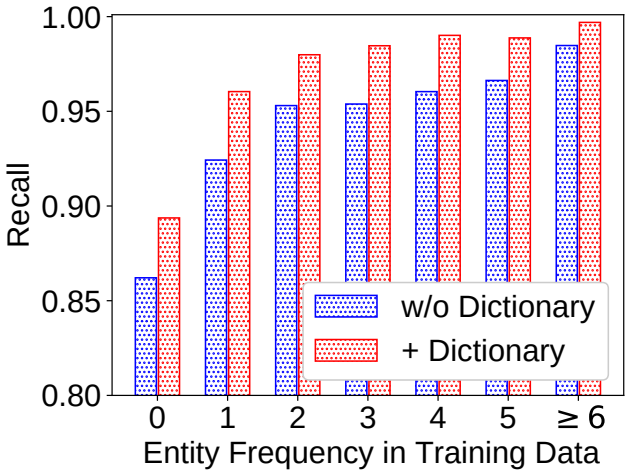



Figure 3: Recall of the entities in the test set with different frequencies in the training data.

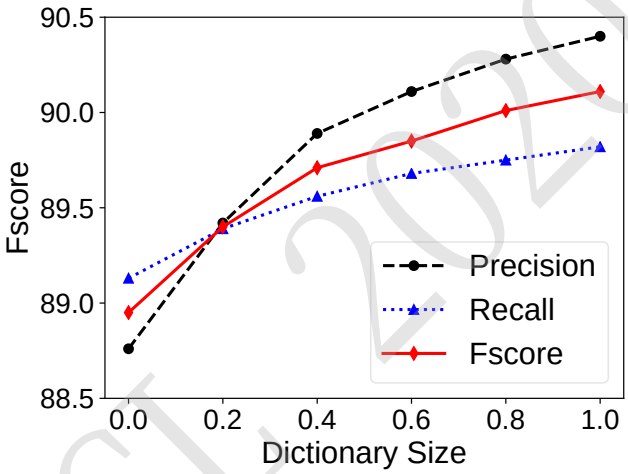



Figure 4: Model performance under different dictionary size.

### 4.4 Performance on Rare Entities

In this section, we explore the influence of incorporating dictionary knowledge on recognizing the entities rarely appearing in the training data. We evaluate the recall of the entities in the test set with different appearance times in the training data. We conduct experiments under 25% of training data and the results of the *ELMo+CADK* model are shown in Fig. 3, which reveals two findings. First, the performance on entities that do not or rarely appear in the training data is much lower than recognizing common entities. This result shows that rare entities are more difficult to recognize. Second, our approach can effectively improve the performance on entities that rarely appear in the training data. This is because our approach can utilize dictionary knowledge to help neural NER model recognize these rare entities more accurately.

### 4.5 Influence on Dictionary Size

In this section, we study the influence of the size of entity dictionaries. We randomly sampled different ratios of entities from the dictionary for entity matching and compare the performance of the *ELMo-CADK* model under 25% of training data. The results are shown in Fig. 4. We find that the model performance consistently improves when the dictionary size grows. This is because a larger dictionary usually has better entity coverage, and our approach can exploit richer information from the entity dictionary to help recognize entities more accurately.

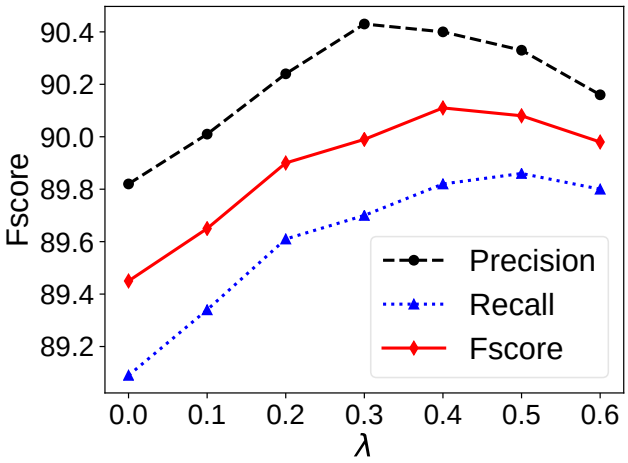



Figure 5: The performance of our approach with different $\lambda$ values under different ratios of training data.

### 4.6 Influence of Hyper-parameters

In this section we explore the influence of an important hyper-parameter on our approach, i.e., $\lambda$ in Eq. (11), which is used to control the relative importance of the term classification loss. The experimental results on $\lambda$ using the *ELMo-CADK* model with 25% of training data are shown in Fig. 5. According to Fig. 5, the performance of our approach improves when $\lambda$ increases. However, when $\lambda$ becomes too large the performance declines. This is because when $\lambda$ is too small, the useful information in the term classification task is not fully exploited. Thus, the performance is sub-optimal. When $\lambda$ goes too large, the auxiliary task is dominant and the NER task is not fully respected. Thus, the performance of our approach is also sub-optimal. These results lead to a moderate selection of $\lambda$ (e.g., 0.4).

### 4.7 Case Study

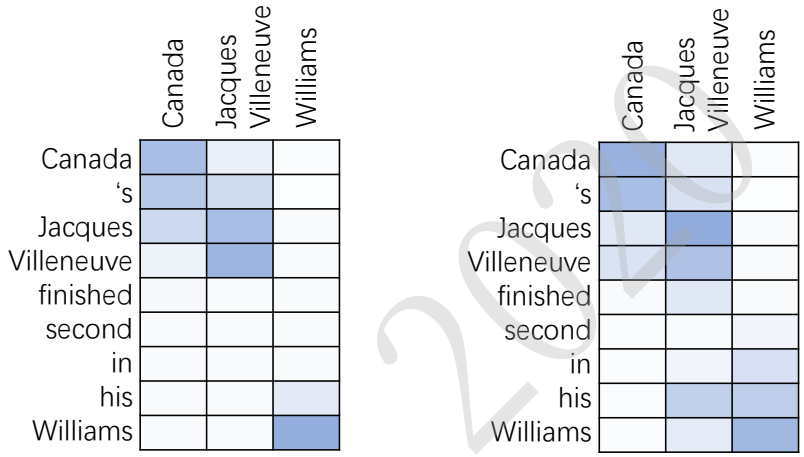
In this section, we conducted several case studies to better understand our approach in incorporating dictionary knowledge in a context-aware manner. Several representative samples are shown in Table 3. This experiment is conducted using 10% of training data. According to Table 3, incorporating entity dictionaries can help a NER model better recognize rare entities. For example, "Partizan" is a name of a football team, which only appears once in the training set. The basic NER model recognized it as a person name, while the approaches using dictionaries can make correct predictions. Our approach can also correctly recognize the context-dependent entities which the basic model and the model based on dictionary features fail to recognize. For example, the entity "Florida" is recognized as a location by *ELMo* and *ELMo+Feature*, since it is usually used as a location name. Our approach can recognize this entity correctly based on its contexts. These results show that our approach can effectively exploit the useful information in entity dictionaries with contextual information.

Next, we visualize the attention weights in the context-dictionary attention to better understand the interactions between contexts and matched terms. The visualization results are shown in Fig. 6. According tothe results, we can see that our approach can effectively model the interactions between entity terms and contexts. For example, in Fig. 6(a), the interaction between the word "Jacques" and the term "Jacques Villeneuve" is highlighted, which is important for identifying the word "Jacques" belongs to an entity name. In addition, in Fig. 6(b), the interaction between the term "Jacques Villeneuve" and the word "his" is also highlighted, which is an important clue for inferring the type of this entity is "person". These results indicate that our approach can effectively capture the relationships between the entity names matched by dictionaries and their contexts to learn context-aware dictionary knowledge.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - November 1, 2020.

| Example | Method | NER result |
|---|---|---|
| 1 | ELMo | Third one-day match : December 8, in Karachi. |
| | ELMo+Feature | Third one-day match : December 8, in Karachi. |
| | ELMo+CADK | Third one-day match : December 8, in Karachi. |
| 2 | ELMo | Partizan - Dejan Koturovic 21 |
| | ELMo+Feature | Partizan - Dejan Koturovic 21 |
| | ELMo+CADK | Partizan - Dejan Koturovic 21 |
| 3 | ELMo | Bolesy (Florida manager John Boles) told me ... |
| | ELMo+Feature | Bolesy (Florida manager John Boles) told me ... |
| | ELMo+CADK | Bolesy (Florida manager John Boles) told me ... |

Table 3: Several named entity recognition examples. Red, orange, and blue words represent the predicted person, location and organization entities respectively.




(a) Using word representations as attention query and term representations as key.

(b) Using term representations as attention query and word representations as key.

Figure 6: Visualization of the attention weights in the context-dictionary attention network.

## 5 Conclusion

In this paper we propose a neural NER approach which can incorporate entity dictionaries with contextual information. In our approach, we propose a context-dictionary attention network to model the interactions between entity names matched by dictionaries and their contexts in texts. In addition, we propose an auxiliary term classification task to classify the types of the terms matched by dictionaries based on contexts, and we jointly train our model in both NER and term classification tasks to incorporate the information of entity dictionaries and contexts into NER. Extensive experiments on the CoNLL-2003 benchmark dataset show that our approach can effectively improve the performance of NER especially when training data is insufficient.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant number 2018YFC1604002, the National Natural Science Foundation of China under Grant numbers U1936208, U1936216, U1836204, and U1705261.

## References


Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING*, pages 1638–1649.

Ravneet Arora, Chen-Tse Tsai, Ketevan Tsereteli, Prabhanjan Kambadur, and Yi Yang. 2019. A semi-Markov structured support vector machine model for high-precision named entity recognition. In *ACL*.

Hui Chen, Zijia Lin, Guiguang Ding, Jianguang Lou, Yusen Zhang, and Borje Karlsson. 2019. Grn: Gated relation network to enhance convolutional neural network for named entity recognition. In *AAAI*.

Jason Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *TACL*, 4(1):357–370.

Kevin Clark, Minh-Thang Luong, Christopher D Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *EMNLP*, pages 1914–1925.

William W Cohen and Sunita Sarawagi. 2004. Exploiting dictionaries in named entity extraction: combining semi-markov extraction processes and data integration methods. In *KDD*, pages 89–98. ACM.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537.

Leon Derczynski, Diana Maynard, Giuseppe Rizzo, Marieke van Erp, Genevieve Gorrell, Raphaël Troncy, Johann Petrak, and Kalina Bontcheva. 2015. Analysis of named entity recognition and linking for tweets. *Information Processing & Management*, 51(2):32–49.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuan-Jing Huang. 2019. A lexicon-based graph neural network for chinese ner. In *EMNLP-IJCNLP*, pages 1039–1049.

Ryuichiro Higashinaka, Kugatsu Sadamitsu, Kuniko Saito, Toshiro Makino, and Yoshihiro Matsuo. 2012. Creating an extended named entity dictionary from wikipedia. In *COLING*, pages 1163–1178.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL-HLT*, pages 260–270.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117.

Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. 2017. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *EMNLP*, pages 2664–2669.

Bill Yuchen Lin and Wei Lu. 2018. Neural adaptation layers for cross-domain named entity recognition. In *EMNLP*, pages 2012–2022.

Hongfei Lin, Yanpeng Li, and Zhihao Yang. 2007. Incorporating dictionary features into conditional random fields for gene/protein named entity recognition. In *PAKDD*, pages 162–173. Springer.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*, pages 2124–2133.

Tianyu Liu, Jin-ge Yao, and Chin-Yew Lin. 2019. Towards improving neural named entity recognition with gazetteers. In *ACL*, pages 5301–5307.

Xiong Luo, Wenwen Zhou, Weiping Wang, Yueqin Zhu, and Jing Deng. 2018. Attention-based relation extraction with bidirectional gated recurrent unit and highway network in the analysis of geological data. *IEEE Access*, 6:5705–5715.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, volume 1, pages 1064–1074.

Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. 2019. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning*, pages 40–49.

Arvind Neelakantan and Michael Collins. 2014. Learning dictionaries for named entity recognition using minimal supervision. In *EACL*, pages 452–461.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. *CoNLL-2014*, page 78.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *ACL*, pages 2409–2419.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *ACL*, volume 1, pages 1756–1765.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *CoNLL*, pages 147–155.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *ACL*, pages 2121–2130.

Tim Rocktäschel, Torsten Huber, Michael Weidlich, and Ulf Leser. 2013. Wbi-ner: The impact of domain-specific features on the performance of identifying and classifying mentions of drugs. In *SemEval 2013*, volume 2, pages 356–363.

Min Song, Hwanjo Yu, and Wook-Shin Han. 2015. Developing a hybrid dictionary-based bio-entity recognition technique. *BMC medical informatics and decision making*, 15(1):S9.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *EMNLP-IJCNLP*, pages 3821–3831.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *NAACL-HLT*, pages 142–147.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

Xiaojun Wan, Liang Zong, Xiaojiang Huang, Tengfei Ma, Houping Jia, Yuqian Wu, and Jianguo Xiao. 2011. Named entity recognition in chinese news comments on the web. In *IJCNLP*, pages 856–864.

Qi Wang, Yangming Zhou, Tong Ruan, Daqi Gao, Yuhang Xia, and Ping He. 2019. Incorporating dictionaries into deep neural networks for the chinese clinical named entity recognition. *Journal of biomedical informatics*, 92:103133.

Xiaofeng Yu, Wai Lam, Shing-Kit Chan, Yiu Kei Wu, and Bo Chen. 2008. Chinese ner using crfs and logic for the fourth sighan bakeoff. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Weixin Zeng, Jiuyang Tang, and Xiang Zhao. 2018. Entity linking on chinese microblogs via deep neural network. *IEEE Access*, 6:25908–25920.

Yuying Zhu and Guoxin Wang. 2019. Can-ner: Convolutional attention network for chinese named entity recognition. In *NAACL-HLT*, pages 3384–3393.



Proceedings of the 19th China National Conference on Computational Linguistics, pages 916-927, Hainan, China, October 31 - Novermber 1, 2020.