Coarse-to-Fine Pre-training for Named Entity Recognition

Mengge Xue[1,2] **Bowen Yu**[1,2] **Zhenyu Zhang**[1,2] **Tingwen Liu**[1,2*] **Yue Zhang**[3] and **Bin Wang**[4]
[1]Institute of Information Engineering, Chinese Academy of Sciences. Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences. Beijing, China
[3]School of Engineering, Westlake University, Hangzhou, China
[4]Xiaomi AI Lab, Xiaomi Inc., Beijing, China
{xuemengge, yubowen, zhangzhenyu1996, liutingwen}@iie.ac.cn
yue.zhang@wias.org.cn wangbin11@xiaomi.cn

Abstract

More recently, Named Entity Recognition has achieved great advances aided by pre-training approaches such as BERT. However, current pre-training techniques focus on building language modeling objectives to learn a general representation, ignoring the named entity-related knowledge. To this end, we propose a NER-specific pre-training framework to inject coarse-to-fine automatically mined entity knowledge into pre-trained models. Specifically, we first warm-up the model via an entity span identification task by training it with Wikipedia anchors, which can be deemed as general-typed entities. Then we leverage the gazetteer-based distant supervision strategy to train the model extract coarse-grained typed entities. Finally, we devise a self-supervised auxiliary task to mine the fine-grained named entity knowledge via clustering. Empirical studies on three public NER datasets demonstrate that our framework achieves significant improvements against several pre-trained baselines, establishing the new state-of-the-art performance on three benchmarks. Besides, we show that our framework gains promising results without using human-labeled training data, demonstrating its effectiveness in label-few and low-resource scenarios.[1]

1 Introduction

Named Entity Recognition (NER) is the task of discovering information entities and identifying their corresponding categories, such as mentions of people, organizations, locations, temporal and numeric expressions (Freitag, 2004). It is an essential component in many applications including machine translation (Babych and Hartley, 2003), relation extraction (Yu et al., 2019), entity linking (Xue et al., 2019a), and so on.

Recently, NER has seen remarkable advances with the help of pre-trained representation models, such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019). Providing contextual representation, these pre-trained models could be easily applied to NER applications as an encoder by just fine-tuning it. Despite refreshing the state-of-the-art performance of NER, the current pre-training techniques are not directly optimized for NER. Typically, these models build unsupervised training objectives to capture dependency between words and learn a general language representation (Tian et al., 2020), while rarely considering incorporating named entity information which can provide rich knowledge for NER. Due to little knowledge connection between NER and general language modeling, how to adapt public pre-trained models to be NER-specific remains an open problem.

To this end, injecting named entity knowledge during pre-training is a possible solution. However, this process of knowledge acquisition may be inefficient and expensive. In fact, there are extensive weakly labeled annotations that naturally exist on the web yet to be explored for NER model pre-training, which are relatively easier to obtain compared with labeled data (Cao et al., 2019). One can collect them from online resources, such as the Wikipedia anchors and gazetteers (named entity dictionaries). Although automatically derived corpora usually contain massive noisy data, it still contains some extend the valuable semantic information required for NER (Peng et al., 2019).

In this paper, we propose a *Coarse-to-Fine Entity knowledge Enhanced* (CoFEE) pre-training framework for NER task, aiming to gather and utilize knowledge related to named entities. In particular, we first extract anchors from Wikipedia and use them as training corpora for entity span identification. While anchors have no entity type information, the model could get general-typed entity

[*] Corresponding Author

[1] The source code can be obtained from https://github.com/strawberryx/CoFEE

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6345–6354,
November 16–20, 2020. ©2020 Association for Computational Linguistics

knowledge from them and learn to distinguish entity words and non-entity words. In the second phase, we use gazetteers and anchors to generate weakly labeled data for specific entity types and use it to train the model for extracting entities with coarse-grained type. Furthermore, another observation is that entities with the same coarse-grained type may belong to different fine-grained types. According to the cluster hypothesis (Chapelle et al., 2009), the features of entities with the same latent fine-grained label will cluster together in the semantic space. Intuitively, mining these latent cluster structures provides auxiliary information about the coarse-grained entity type, which could be beneficial to improve the NER performance. Based on such motivation, we finally devise a self-supervised method to exploit fine-grained type knowledge and tap the potential of weakly labeled data, which effectively train the NER model with clustering-generated pseudo labels.

We conduct experiments on three realistic NER benchmarks in this paper. Experimental results show that the proposed CoFEE pre-training framework significantly outperforms other competitive baselines, often by large margins. We also demonstrate that CoFEE pre-training can work well in more challenging, label-free and low-resource scenarios. Further ablation studies show the impact of each pre-training task in achieving these strong performance. To the best of our knowledge, this is the first work that has tackled NER-specific representation during pre-training.

2 Related Work

Entity Knowledge for NER. Recently, neural networks have been used for NER and achieved great success (Collobert et al., 2011; dos Santos and Guimarães, 2015; Huang et al., 2015; Ma and Hovy, 2016). Specifically, various types of entity knowledge, including lexical words, gazetteers and anchors in Wikipedia have been proved to be useful for a wide range of sentiment analysis tasks.

For supervised NER task, some researchers utilize lattice structure to incorporate the lexical information into character-based NER and avoid the segmentation error propagation of word (Zhang and Yang, 2018; Gui et al., 2019a; Xue et al., 2019b; Gui et al., 2019b; Sui et al., 2019). Additionally, gazetteers have long been regarded as a piece useful knowledge for NER, previous methods commonly incorporated gazetteers by either using them as handcraft features (Alan et al., 2011; Dominic et al., 2018) or auxiliary structural information (Ding et al., 2019; Liu et al., 2019).

For weakly supervised NER, a typical line of methods centres around transfer learning to extract source knowledge for target, such as cross-domain (Yang et al., 2017; Lin and Lu, 2018; Jia et al., 2019) or cross-lingual (Ni et al., 2017; Xie et al., 2018; Zhou et al., 2019). There are also a lot of weak labels lying on the web or gazetteers, which have not been explored. Consequently, a number of works focus on distantly supervised methods, using anchors or gazetteers to generate data by distant supervision (Liu et al., 2015; Yang et al., 2018; Cao et al., 2019; Peng et al., 2019).

Task Specific Pre-training. Unsupervised language model pre-training and task-specific fine-tuning achieve SOTA results on many NLP tasks, including NER (Peters et al., 2018; Devlin et al., 2019; Li et al., 2020). Recently, with the help of automatically minded knowledge lying in the web, researchers devoted them to the pre-training models for specific tasks, including word sense disambiguation (Huang et al., 2019), word-in-context tasks (Levine et al., 2020), entity-linking and relation classification (Zhang et al., 2019), sentiment classification (Tian et al., 2020).

3 Background

In this section, we give a brief introduction to MRC-NER (Li et al., 2020), which achieves satisfying performance in NER and thus is chosen as the foundation of our work. Given an input paragraph $X = \{x_1, x_2, \cdots, x_n\}$ where x_i denotes the i-th character, NER aims at discovering each entity $x_{start,end}$ in X and identify its corresponding type $y \in Y$, where Y is the set of predefined tags(e.g., PER, LOC). $x_{start,end} = \{x_{start}, x_{start+1}, \cdots, x_{end-1}, x_{end}\}$ is a substring of X satisfying *start* $\leq$ *end*. Specifically, MRC-NER formulates NER as a machine reading comprehension (MRC) problem. Each entity type y is characterized by a natural language query $Q_y = \{q_1^y, q_2^y, ..., q_m^y\}$, and entities are extracted by answering these queries given the contexts. For example, the task of assigning the PER label to "[Washington] was born into slavery on the farm" is formalized as answering the question "Find person including fictional". This strategy naturally introduces the natural language query which encodes significant prior knowledge about the entity

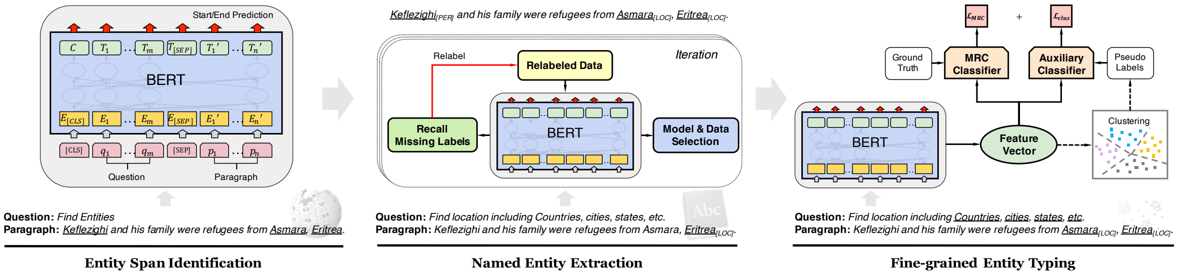

Start/End Prediction
C
T_1
T_m
T_[SEP]
T_1'
T_n'
BERT
E_[CLS]
E_1
E_m
E_[SEP]
E_1'
E_n'
[CLS]
q_1
q_m
[SEP]
p_1
p_n
Question
Paragraph
Question: Find Entities
Paragraph: Keflezighi and his family were refugees from Asmara, Eritrea.
Entity Span Identification
Keflezighi[PER] and his family were refugees from Asmara[LOC], Eritrea[LOC].
Relabel
Relabeled Data
Iteration
Recall Missing Labels
BERT
Model & Data Selection
Abc
Question: Find location including Countries, cities, states, etc.
Paragraph: Keflezighi and his family were refugees from Asmara, Eritrea[LOC].
Named Entity Extraction
L_MRC
+
L_clus
Ground Truth
MRC Classifier
Auxiliary Classifier
Pseudo Labels
Clustering
BERT
Feature Vector
Question: Find location including Countries, cities, states, etc.
Paragraph: Keflezighi and his family were refugees from Asmara[LOC], Eritrea[LOC].
Fine-grained Entity Typing

Figure 1: The overall architecture of CoFEE. In Fine-grained Entity Typing, the *solid line* represents the training phase and the *dotted line* represents the clustering phase. These two stages are iteratively done until the network converges.

category to extract.

Formally, MRC-NER model concatenates the query Q and paragraph X, forming string $\{[\text{CLS}], Q, [\text{SEP}], X\}$, where [CLS] and [SEP] are special tokens. Then BERT (Devlin et al., 2019) captures the contextual information for each token in the string via self-attention and produces the representation matrix $\mathbf{H} \in \mathbb{R}^{n \times d}$ of X, where d is the dimension of the last layer of BERT. To extract entity spans, the representation of each word is fed to two softmax layers to predict the probability of each token being a start or end index as follows:

$$P_{\text{start}}(\mathbf{y}_i^{\text{s}}|x_i) = softmax(\mathbf{W}_{\text{s}}\mathbf{h}_i + \mathbf{b}_{\text{s}}), \quad (1)$$

$$P_{\text{end}}(\mathbf{y}_i^{\text{e}}|\mathbf{x}_i) = softmax(\mathbf{W}_{\text{e}}\mathbf{h}_i + \mathbf{b}_{\text{e}}), \quad (2)$$

where $\mathbf{W}_{\text{s}}, \mathbf{W}_{\text{e}} \in \mathbb{R}^{d \times 2}$ and $\mathbf{b}_{\text{s}}, \mathbf{b}_{\text{e}} \in \mathbb{R}^2$ are trainable parameters. At training time, S associated with each question Q_y is paired with two label sequences $\mathbf{Y}_{\text{start}} = \{y_1^{\text{s}}, y_2^{\text{s}}, ..., y_n^{\text{s}}\}$ and $\mathbf{Y}_{\text{end}} = \{y_1^{\text{e}}, y_2^{\text{e}}, ..., y_n^{\text{e}}\}$, where y_i^{s} (y_i^{e}) is the ground-truth label of x_i being the start (end) index of a y-typed entity or not. The cross-entropy loss of start and end index predictions are therefore denoted as:

$$\mathcal{L}_{\text{start}}^{\mathcal{D}} = -\frac{1}{n}\sum_{i=1}^{n} y_i^{\text{s}} log(P_{\text{start}}(y_i^{\text{s}}|x_i)), \quad (3)$$

$$\mathcal{L}_{\text{end}}^{\mathcal{D}} = -\frac{1}{n}\sum_{i=1}^{n} y_i^{\text{e}} log(P_{\text{end}}(y_i^{\text{e}}|x_i)), \quad (4)$$

where $\mathcal{D}$ denotes the training dataset. Finally, the overall training objective to be minimized can be formulated as follows:

$$\mathcal{L}_{\text{MRC}}^{\mathcal{D}} = \mathcal{L}_{\text{start}}^{\mathcal{D}} + \mathcal{L}_{\text{end}}^{\mathcal{D}}. \quad (5)$$

4 Methodology

In this section, we introduce the overall framework of our coarse-to-fine pre-training. Figure 1 gives a brief illustration, which operates in three stages as follows: (1) Stage 1: identity entity span based on Wikipedia anchors; (2) Stage 2: extract coarse-grained entities based on gazetteers; (3) Stage 3: predict fine-grained entity types with a clustering-oriented self-supervised method.

4.1 Entity Span Identification

Pre-trained language Models such as BERT (Devlin et al., 2019) and XLNet (Yang et al., 2019) have been proven to capture rich language information from text. However, as the entity information of a text is seldom explicitly studied, it is hard to expect such pre-trained general representations to capture entity-centric knowledge. In order to better capture entity information and learn NER-specific representation, we propose the first pre-training task named Entity Span Identification (ESI). The entity-centric knowledge is automatically mined from the large scale Wikipedia corpus. In Wikipedia, an anchor $\langle m, e\rangle$ links a mention m to an entity e. Therefore, we assign an "Entity" tag to each anchor in the sentence and construct a General-typed weakly labeled NER dataset $\mathcal{D}_{\text{g}}$ without considering the entity type. To align with MRC-NER, the question of the generated dataset is set as "*Find Entities*". With the general labeled data, the MRC-NER model can be warmed-up with loss $\mathcal{L}_{\text{MRC}}^{\mathcal{D}_{\text{g}}}$. By integrating the general-typed named entity knowledge into the pre-training process, the learned representation would be incorporated with the structural information of crucial importance for NER.

4.2 Named Entity Extraction

After the ESI pre-training, the model has learned to distinguish entity words and non-entity words. Then we step into the second phase (i.e., NEE) in which the model is trained to extract typed entities with gazetteer-labeled data. To alleviate human

Type	Question
PER	人名和虚构的人物形象
ORG	组织包括公司，政府党派，学校，政府，新闻机构
LOC	山脉，河流自然景观的地点
GPE	按照国家，城市，州县划分的地理区域
HP	找出文中的商标，包括公司，品牌
HC	找出文中的产品，包括商品，作品，食品，用品，设施，副产品，农产品，制成品，软件产品，硬件产品，资讯产品，通讯产品，通信产品，电信产品，电脑产品，手机产品，电子产品，科技产品，其他产品
ORG	organization entities are limited to named corporate, governmental, or other organizational entities.
PRR	person entities are named persons or family.
LOC	location entities are the name of politically or geographically defined locations such as cities, provinces, countries, international regions, bodies of water, mountains, etc.

Table 1: Neural language questions for each entity type used in our model.

effort, gazetteer-based distant supervision has been applied to automatically generate labeled data and has gained successes in NER (Yang et al., 2018; Peng et al., 2019). A standard strategy is to scan through the anchor text in $\mathcal{D}_\mathrm{g}$ using the gazetteer of a given entity type y and treat anchors matched with entries of the given gazetteer as the entities with type y. In this way, we can obtain a specific-typed NER dataset $\mathcal{D}_\mathrm{s}$, which is then exploited to train the MRC-NER model by optimizing $\mathcal{L}^{\mathcal{D}_\mathrm{s}}_{\mathrm{MRC}}$. Besides, in order to meet the paradigm of MRC-NER, we also generate a natural language query for each entity type. This procedure is critical since queries encode prior knowledge about labels. Inspired by (Li et al., 2020), we take annotation guideline notes as references to construct queries and illustrate all of the queries used in our model in Table 1. They are theoretical description of the tag categories, thus having the ability to make the model incorporate the information within the label categories unambiguously and completely.

However, as most existing gazetteers only cover part of entities, the automatically derived dataset usually contains massive noisy data including missing labels, incorrect boundaries and types. To address this issue, we propose an iterative self-picking strategy. At the beginning (iteration 0), the model starts with training from the original noisy label set. At the end of each iteration, the model determines the next label set by making predictions on $\mathcal{D}_\mathrm{s}$. Concretely, a new entity will be extracted with type y if the probabilities of its start and end indices being predicted as y are both greater than a picking threshold δ. In the next iteration, we use the new derived dataset as input for the model training. Considering that we aim to recall the missing labels, we set $\delta < 0.5$. The model is trained until we find the best model w.r.t. the performance on the validation set. And the final derived dataset is denoted as $\mathcal{D}_\mathrm{s}^{\mathrm{best}}$.

4.3 Fine-grained Entity Typing

NEE pre-training focuses on teaching the model named entity knowledge about coarse-grained entity types. However, one coarse-grained entity type may be composed of a set of fine-grained entity types. For example, the coarse-grained type *Location* includes City, Country, Bodies of water, etc. These fine-grained types can provide auxiliary information to help us understand the meaning of *Location*. With this in mind, it is intuitive to group the extracted entities with a cluster miner, and use the subsequent cluster assignments as pseudo labels to mine the fine-grained NER knowledge. One of the most well-studied clustering algorithms is k-Means, and the simplicity and efficiency have established it as a popular means for performing clustering across different disciplines.

Formally, in order to partition the entity set $E = \{e_1, e_2, \cdots, e_M\}$ in $\mathcal{D}_\mathrm{s}^{\mathrm{best}}$ into pre-defined K distinct clusters $\{C_k\}_{k=1}^{K}$, k-Means minimizes the sum of the intra-cluster variances $\sum_{k=1}^{K} \mathcal{V}_k$, where $\mathcal{V}_k = \sum_{i=1}^{M} \delta_{ik} ||\mathbf{e}_i - \mathbf{m}_k||^2$ and $\mathbf{m}_k = \sum_{i=1}^{M} \delta_{ik}\mathbf{e}_i / \sum_{i=1}^{M} \delta_{ik}$ are the variance and the center of the k-th cluster, respectively, $\mathbf{e}_i = \mathit{sumpool}([\mathbf{h}_{start_i}, \mathbf{h}_{start_i+1}, ..., \mathbf{h}_{end_i}])$ denotes the representation of the i-th entity, and δ_{ik} is a cluster indicator variable with $\delta_{ik} = 1$ if $e_i \in C_k$

and 0 otherwise. Clustering proceeds by alternating between assigning instances to their closest center and recomputing the centers, until a local minimum is reached. The cluster assignments are used as pseudo labels to guide the transformation of $\mathcal{D}_{\text{s}}^{\text{best}}$ to a pseudo-labeled fine-grained dataset $\mathcal{D}_{\text{c}} = \{(e_i, y_i^{\text{c}})\}$, where y_i^{c} is the pseudo label of e_i Then we can take the negative log-likelihood of the pseudo-labeled tags as the training objective:

$$P_c(\mathbf{y}_i^{\text{c}}|e_i) = softmax(\mathbf{W}_{\text{c}}\mathbf{e}_i + \mathbf{b}_{\text{c}}) \tag{6}$$

$$\mathcal{L}_{\text{clus}} = -\frac{1}{M}\sum_{i=1}^{M} y_i^{\text{c}} log(P_{\text{c}}(y_i^{\text{c}}|e_i)) \tag{7}$$

where $\mathbf{W}_{\text{c}} \in \mathbb{R}^{K \times d}$ and $\mathbf{b}_{\text{c}} \in \mathbb{R}^{K}$ are trainable parameters, $P_{\text{c}}(y_i^{\text{c}}|e_i)$ denotes the probability of entity e_i being predicted to the y_i^{c}-th cluster. Recall that our purpose is to pre-train the NER model to discover typed entities belonging to Y rather than fine-grained entities, so $\mathcal{L}_{\text{clus}}$ can be deemed as an auxiliary task to assist the model to mine the fine-grained NER knowledge and regularize the optimization of $\mathcal{L}_{\text{MRC}}^{\mathcal{D}_{\text{s}}}$. So the training objective in this stage is defined as:

$$\mathcal{L}_{\text{FET}} = \mathcal{L}_{\text{MRC}}^{\mathcal{D}_{\text{s}}^{\text{best}}} + \gamma\mathcal{L}_{\text{clus}}, \tag{8}$$

where γ is the trade-off parameter.

While optimizing with pseudo labels created by the cluster miner seems reasonable, the inevitable label noise caused by the clustering procedure is ignored. To this end, we propose a variance-weighted cross-entropy loss to alleviate the influence of noisy pseudo labels. Obviously, the inverse of $\mathcal{V}_k$ ($\mathcal{V}_k^{-1}$) represents the intra-cluster compactness of the k-th cluster. If the features of instances in the k-th cluster are close together, $\mathcal{V}_k^{-1}$ will be large, meantime the confidence of assigning pseudo label y_i to these instances should also be high and vice versa. Thus we re-formulate Equation 7 as:

$$\mathcal{L}_{\text{clus}} = -\frac{1}{M}\sum_{i=1}^{M} \alpha_{y_i^{\text{c}}} y_i^{\text{c}} log(P_{\text{c}}(y_i^{\text{c}}|e_i)), \tag{9}$$

$$\alpha_{y_i^{\text{c}}} = \frac{exp(\mathcal{V}_{y_i^{\text{c}}}^{-1})}{\sum_{k=1}^{K} exp(\mathcal{V}_n^{-1})}. \tag{10}$$

Finally, we iterate the above clustering-optimizing process by putting back the model to output new representations, generate new pseudo labels $\mathcal{D}_{\text{c}}$ and start the next iteration.

Algorithm 1 Coarse-to-fine Pre-training

```
Require: Wikipedia corpus;
Require: Specific typed gazetteers;
Require: Specific typed validation data $\mathcal{D}_s^{val}$;
Require: Initialize Model Parameters $\theta$ with BERT.
Construct $\mathcal{D}_g$ based on Wikipedia anchors
for $epoch \leftarrow 1$ to $e_1$ do                         ▷ Stage 1.
    Update $\theta$ w.r.t. $\mathcal{L}_{MRC}^{\mathcal{D}_g}$
end for
Construct $\mathcal{D}_s$ by matching $\mathcal{D}_g$ to gazetteer.
for $epoch \leftarrow 1$ to $e_2$ do                         ▷ Stage 2.
    Update $\theta$ w.r.t. $\mathcal{L}_{MRC}^{\mathcal{D}_s}$.
    if $score(\theta, \mathcal{D}_s^{val}) > best\_score$ then
        $\theta_{best} \leftarrow \theta$; $\mathcal{D}_s^{best} \leftarrow \mathcal{D}_s$;
        $best\_score = score(\theta, \mathcal{D}_s^{val})$
    end if
    Re-label $\mathcal{D}_s$ with $\theta$.
end for
$\theta \leftarrow \theta_{best}$, $best\_score \leftarrow 0$
Construct $\mathcal{D}_c$ by clustering entities in $\mathcal{D}_s^{best}$.
for $epoch \leftarrow 1$ to $e_3$ do                         ▷ Stage 3.
    Update $\theta$ w.r.t. $\mathcal{L}_{FET}$
    if $score(\theta, \mathcal{D}_s^{val}) > best\_score$ then
        $\theta_{best} \leftarrow \theta$;
        $best\_score = score(\theta, \mathcal{D}_s^{val})$
    end if
    Re-cluster entities in $\mathcal{D}_s^{best}$ and construct new $\mathcal{D}_c$
end for
return $\theta_{best}$
```

4.4 Algorithm Workflow

In this subsection, we introduce the overall procedure of our framework. Algorithm 1 gives the scratch. First, we construct general-typed NER data $\mathcal{D}_{\text{g}}$ based on Wikipedia anchors, and pre-train the model to extract general typed entities with loss $\mathcal{L}_{\text{MRC}}^{\mathcal{D}_{\text{g}}}$. Then we leverage the gazetteer-based distant supervision strategy to construct a specific-typed NER dataset $\mathcal{D}_{\text{s}}$, and propose an iterative self-picking method to alleviate the data missing problem. In each iteration, the model is optimized to fit the data labeled by the previous iteration. When the performance on the validation set starts to decline, the iteration is ended and the best-performed model is passed to the third stage, where a cluster miner is deployed to group the entities extracted from the second stage into fine-grained types, and the model is trained to simultaneously distinguish fine-grained entities and extract specific-typed entities. Also, we iteratively cluster the features from the last iteration to gradually refine the fine-grained pseudo labels for current.

5 Experiments

We evaluate the CoFEE framework under two settings: (i) supervised setting (ii) weakly supervised setting. In the supervised setting, the pre-trained model is fine-tuned on human-labeled datasets while in the weakly-supervised setting, the model pre-trained with CoFEE is directly applied to perform NER without fine-tuning. Next, we describe these experiments in detail.

5.1 Datasets

Our experiments are conducted on three benchmarks. (1) **Chinese Ontonotes 4.0** consists of newswire text and published by Ralph et al. (2011). It is annotated by four types: PER (Person), ORG (Organization), GPE (Geo-Political Entity) and LOC (Location) for Chinese named entity. It contains 15.7k sentences for training and 4.3k for testing. (2) **E-commerce** is a Chinese NER dataset collected from the e-commerce domain and released by Ding et al. (2019). It is annotated by PROD (product) and BRAN (brand) types. The training and test datasets contain 273k and 53k lines, respectively. (3) **Twitter** is an English NER dataset (Qi et al., 2018), following (Peng et al., 2019), we only use textual information to perform NER and make entity detection on PER, LOC and ORG. It contains 4,000 tweets for training and 3,257 tweets for testing.

5.2 Pre-training Corpora

Wikipedia. We use 20200401 Chinese and English Wikipedia dumps[2][3] for data construction, where we set the max sentence length as 250 and remove the sentences which contain three or fewer anchors. The resulting Chinese corpora contains 1,116,514 sentences and 6,383,142 anchors (entity mentions), and the English corpora contains 3,911,059 sentences and 37,755,176 anchors.

Gazetteer. For Chinese PER, ORG, GPE, and LOC, we collect the gazetteers from the crowd-source dictionaries used by Chinese Input Method ”Sougou”[4], which contain 2,314 person names, 2,649 organization names, 895 geopolitical entities, and 628 location names. For Chinese PROD and BRAN, we use the gazetteers provided by Ding et al.(2019), which contain 628 brand names and 2,974 product names. For English PER, ORG and LOC, we collect the gazetteers using the method released by Peng et al.(2019), which contain 2,795 person names, 1,825 organization names and 1,408 location names.

5.3 Baselines

We chose two types of baselines: supervised methods and the weakly supervised methods. We call our proposed CoFEE pre-training framework with MRC-NER backbone as CoFEE-MRC. In addition, to demonstrate the model-agnostic and generic property of CoFEE, we also implemented another competitive baseline by replacing the MRC-NER backbone with a widely used BERT model (Devlin et al., 2019) without any change in the training procedure, denoted as CoFEE-BERT. We used open-source release of https://github.com/huggingface/transformers.

Supervised Setting. We fine-tune CoFEE-MRC and CoFEE-BERT on supervised NER data and compare with the following baselines to learn how improvement can be achieved for supervised models. **BiLSTM-CRF** (Huang et al., 2015) is a classical neural-network-based baseline for NER, which usually achieves competitive performance in supervised NER. **BERT-Tagger** (Devlin et al., 2019) uses the outputs from the last layer of model BERT$_{base}$ as the character-level enriched contextual representations to make sequence labeling. **MRC-NER** (Li et al., 2020) formulates NER as a machine reading comprehension task and uses BERT as the basic encoder.

Weakly Supervised Setting. We investigate the effect of CoFEE-MRC for solving the NER task without any human annotations, and compare the model to some weakly supervised NER models. For fair comparison, we implemented baselines with the same gazetteers constructed in Section 5.2. **Gazetteer Matching** applies the constructed gazetteers to the test set directly to obtain entity mentions with exactly the same surface name. By comparing with it, we can check the improvements of neural models over the distant supervision itself. **MRC-NER** uses the MRC-NER backbone to perform weakly supervised NER task with gazetteer labeled training data. Furthermore, we explore the influence of our proposed pre-training tasks by removing entity span identification pre-training (**-ESI**) and fine-grained entity typing pre-training (**-FTP**) from CoFEE-MRC.

[2] https://dumps.wikimedia.org/zhwiki/20200401/zhwiki-20200401-pages-articles.xml.bz2

[3] https://dumps.wikimedia.org/enwiki/20200401/enwiki-20200401-pages-articles.xml.bz2

[4] https://pinyin.sogou.com/dict/

Chinese OntoNotes 4.0			
Model	P	R	F1
BiLSTM-CRF (Huang et al., 2015)	72.0	75.1	73.5
BERT (Devlin et al., 2019)	78.01	80.35	79.16
MRC-NER (Li et al., 2020)	**82.98**	81.25	82.11
CoFEE-BERT	80.27	80.64	80.46
CoFEE-MRC	82.5	**82.78**	**82.64**
E-commerce			
Model	P	R	F1
BiLSTM-CRF (Huang et al., 2015)	71.1	76.1	73.6
BERT (Devlin et al., 2019)	77.06	**80.65**	78.81
MRC-NER (Li et al., 2020)	79.47	78.3	78.88
CoFEE-BERT	79.13	80.34	**79.73**
CoFEE-MRC	**80.26**	78.88	79.56
Twitter			
Model	P	R	F1
BiLSTM-CRF (Huang et al., 2015)	–	–	65.32
BERT (Devlin et al., 2019)	69.83	69.35	69.59
MRC-NER (Li et al., 2020)	72.06	70.83	71.44
CoFEE-BERT	75.17	71.17	73.11
CoFEE-MRC	**75.89**	**71.93**	**73.86**

Table 2: Model performance (%) for supervised NER on three benchmark datasets. Bold marks highest number among all models.

5.4 Hyper-parameter settings

We use the BertAdam as our optimizer, all of the models are implemented under PyTorch using a single NVIDIA Tesla V100 GPU, we use ”bertbase-chinese” and ”bert-base-cased” as our pretrained models for Chinese and English language, the number of parameters is same to these pretrained models in addition to two binary classifier. For each training stage, we vary the learning rate from $1e-6$ to $1e-4$. In NEE stage, we select the best trade-off from 0.1 to 0.5 with an incremental 0.1. In FET stage, we choose the number of clusters K from $\{K-2, K-1, K, K+1, K+2\}$ if we set K as the categories of fine-grained entity. For all these hyper-parameters, we select the best according to the F1-score on the dev sets.

5.5 Evaluation

Following the evaluation metrics in previous work (Li et al., 2020), we apply the entity-level (exact entity match) standard micro Precision (P), Recall (R), and F1 score to evaluate the results.

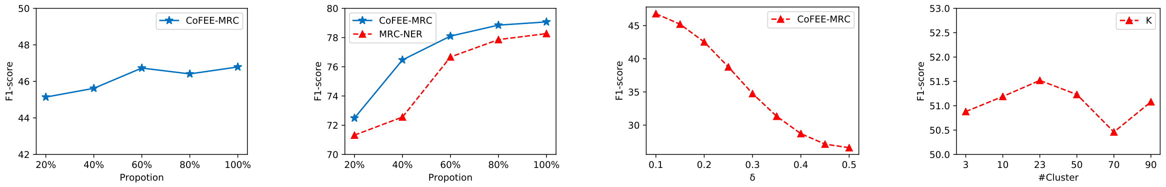
5.6 Overall Performance

Table 2 contains results for models tuned on human-labeled NER data. We can observe that our CoFEE-MRC pre-training performs remarkably better than MRC-NER, establishing an impressive new state-of-the-art for supervised NER on OntoNotes and Twitter of 82.64% and 73.86%, respectively. CoFEE-BERT also significantly improves the performance compared with BERT and achieves a new SOTA for supervised NER on E-commerce of 79.73%, which confirms the model-agnostic property of our CoFEE pre-training framework. Please note that the results of MRC-NER on OntoNotes have a few concerns need to be addressed. MRC-NER set the max sentence length as 77, which is far less than the true maximum length of the dataset. While in our method, we promise that the maximum length is more than 100.

Chinese OntoNotes 4.0			
Model	P	R	F1
Matching	28.29	40.95	33.46
MRC-NER (Li et al., 2020)	44.85	33.06	38.06
CoFEE-MRC	**48.01**	**41.22**	**44.36**
-FET	48.0	39.32	43.23
-FET-ESI	48.19	30.64	40.3
E-commerce			
Model	P	R	F1
Matching	38.94	38.34	39.14
MRC-NER (Li et al., 2020)	54.84	22.78	32.19
CoFEE-MRC	50.27	**53.22**	**51.7**
-FET	52.42	42.88	47.17
-FET-ESI	**55.03**	38.03	44.98
Twitter			
Model	P	R	F1
Matching	28.29	24.58	26.30
MRC-NER (Li et al., 2020)	52.07	45.59	48.62
CoFEE-MRC	**56.44**	**52.81**	**54.56**
-FET	54.92	51.35	53.07
-FET-ESI	56.06	47.28	51.3

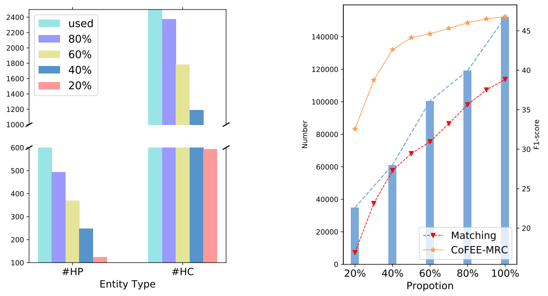
Table 3: Model performance (%) for weakly supervised NER on three benchmark datasets. Bold marks highest number among all models.

Table 3 reports the results of our models against to baselines under the weakly supervised setting. We can find that: **1)** Gazetteer Matching performs quite poorly and the capability of this method is strongly influenced by the size of the gazetteers. For OntoNotes, the coverage of the large scale gazetteer is almost 40%, but also its huge size causes the low precision. For Twitter, the recall value is about 14% due to its limited gazetteers. **2)** If we directly use MRC-NER to perform weakly supervised NER task with gazetteer labeled data, the model achieves a degree of improvement but is still inaccurate due to the distantly labeled data. **3)** CoFEE-MRC achieves the state-of-the-art F1 score on all three benchmarks, which confirms the validity of our proposed CoFEE pre-training framework. **4)** FET pre-training task brings performance improvements, which verifies the effective-



(a) (b) (c) (d)

Figure 2: (a) Impact of pre-training data size on the weakly supervised setting; and (b) Impact of fine-tuning data size on the supervised setting; and (c) Impact of picking rate δ; and (d) Impact of cluster size K.



(a) The number of named entities in gazetteers.

(b) The number of training sentences and the model performance.

Figure 3: Statistic information and model performance with gazetteers of different sizes on the Weibo dev set.

ness of the introducing fine-grained named entity knowledge. **5)** ESI pre-training further improves the performance, which demonstrates the necessity to warm-up the pre-trained language model using general-typed named entity knowledge.

6 Analysis

6.1 Impact of Data Size

We analyze the influence of reducing the amount of pre-training data and fine-tuning data. The results on the dev set of E-commerce are shown in Figure 2(a) and 2(b), respectively. From Figure 2(a), we can observe that increasing the size of the pre-training data will improve the performance generally, but the improvement tends to flatten out with $60\% \sim 80\%$ data. We suppose that this is because of the number of unique patterns, the influence of the training data size has its local minimum and maximum critical point. From Figure 2(b), we see that knowledge enhanced pre-training is more effective for low-resource cases, where there is a larger gap in performance between our CoFEE-MRC and MRC-NER. Besides, the performance of CoFEE pre-training is more stable as data scale changes. This further demonstrates that our CoFEE pre-training framework can significantly reduce human efforts to create NER taggers.

6.2 Impact of Picking Rate

We then evaluate the influence of the value and variation of our picking rate δ. From Figure 2(c), we can see that setting a lower picking rate to recall more named entities can indeed improve a great performance for the model and gives the highest result with $\delta_0 = 0.1$.

6.3 Impact of Gazetteer Size

We further explore the change of the training data and performance when we use gazetteers of different sizes. In particular, we used 20%, 40%, 60%, 80% and 100% of the original gazetteers to construct pre-training corpora. Statistical information of each resultant gazetteer is illustrated in Figure 3(a), and the model performance on the E-commerce dev set with these gazetteers is demonstrated in Figure 3(b). We can observe that increasing the size of gazetteers will generally improve the performance of our proposed CoFEE-MRC model and the performance growths in line with the performance of "Matching", indicating that in addition to the gazetteer size, matching degree also has a crucial influence on the model performance.

6.4 Impact of Cluster Size

The proposed CoFEE framework does require a cluster size K as the scope for pseudo labels. One may wonder whether the choice of K has a significant influence on the final results. In this subsection, we vary K from 4 to 90 and report the F1 score of CoFEE-MRC on the E-commerce dev set. As shown in Figure 2(d), the best performance is obtained when K is exactly set as the number of fine-grained entity types described in the queries (23), indicating that our CoFEE pre-training can leverage this information as useful prior knowledge. Thanks to the self-supervised learning schema, when we very from 3 to 90, the model achieves stable F1 score and is not sensitive to the choice of K. The results also further indicate the applicability of the proposed framework when being applied to a new kind of named entity where the number of fine-grained entity types is not available in advance. We can safely assign a larger value than needed and the model is still robust.

7 Conclusion

We investigated coarse-to-fine entity knowledge enhanced pre-training for named entity recognition, which integrates three kinds of entity knowledge with different granularity levels. Though conceptually simple, our framework is highly effective and easy to implement. On three popular NER benchmarks, we found consistent improvements over both state-of-the-art supervised and weakly-supervised methods. Further analysis verifies the necessity of utilizing NER knowledge for pre-training models.

Acknowledgements

We would like to thank the anonymous reviewers for their insightful comments and suggestions. This research is supported by the National Key Research and Development Program of China (grant No.2016YFB0801003) and the Strategic Priority Research Program of Chinese Academy of Sciences (grant No.XDC02040400).

References

Ritter Alan, Clark Sam, Etzioni Oren, and et al. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *ACL*, pages 1524–C1534.

Bogdan Babych and Anthony Hartley. 2003. Improving Machine Translation Quality with Automatic Named Entity Recognition. In *EAMT workshop*, pages 1–8.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-Resource Name Tagging Learned with Weakly Labeled Data. In *EMNLP-IJCNLP*, pages 261–270.

Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. 2009. Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, 20(3):542–542.

Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural Language Processing almost from Scratch. *Journal of Machine Learning Research*, pages 2493–2537.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL*, pages 4171–4186.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A Neural Multi-digraph Model for Chinese NER with Gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1462–1467.

Seyler Dominic, Dembelova Tatiana, Del Corro Luciano, Hoffart Johannes, and Weikum Gerhard. 2018. A Study of The Importance of External Knowledge in The Named Entity Recognition Task. In *ACL*, pages 241–246.

Dayne Freitag. 2004. Trained Named Entity Recognition Using Distributional Clusters. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 262–269.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019a. CNN-Based Chinese NER with Lexicon Rethinking. In *IJCAI*, pages 4982–4988.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019b. A Lexicon-Based Graph Neural Network for Chinese NER. In *EMNLP*, pages 1039–1049.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. In *EMNLP-IJCNLP*, pages 3509–3514.

Zhiheng Huang, Wei Liang Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *arXiv: Computation and Language*.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-Domain NER using Cross-Domain Language Modeling. In *ACL*, pages 2464–2474.

Yoav Levine, Barak Lenz, Or Dagan, Dan Padnos, Or Sharir, Shai Shalevshwartz, Amnon Shashua, and Yoav Shoham. 2020. SenseBERT: Driving Some Sense into BERT. In *ACL*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A Unified MRC Framework for Named Entity Recognition. *ACL*.

Bill Yuchen Lin and Wei Lu. 2018. Neural Adaptation Layers for Cross-domain Named Entity Recognition. In *EMNLP*, pages 2012–2022.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2015. Effects of Semantic Features on Machine Learning-Based Drug Name Recognition Systems: Word Embeddings vs. Manually Constructed Dictionaries. *Information*, 6(4):848–865.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019. Towards Improving Neural Named Entity Recognition with Gazetteers. In *ACL*, pages 5301–5307.

Xuezhe Ma and uard Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *ACL*, pages 1064–1074.

Jian Ni, Georgiana Dinu, and Radu Florian. 2017. Weakly Supervised Cross-Lingual Named Entity Recognition via Effective Annotation and Representation Projection. In *ACL*, pages 1470–1480.

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, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL*, pages 2227–2237.

Zhang Qi, Fu Jinlan, Liu Xiaoyu, and Huang Xuanjing. 2018. Adaptive Co-attention Network for Named Entity Recognition in Tweets. In *AAAI*, pages 5674–5681.

Weischedel Ralph, Pradhan Sameer, Ramshaw Lance, Palmer Martha, Xue Nianwen, Marcus Mitchell, Taylor Ann, Greenberg Craig, Hovy Eduard, Belvin Robert, and et al. 2011. Ontonotes release 4.0. In *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium.*

Cícero dos Santos and Victor Guimarães. 2015. Boosting Named Entity Recognition with Neural Character Embeddings. In *Proceedings of the Fifth Named Entity Workshop*, pages 25–33.

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*, pages 3821–3831.

Hao Tian, Can Gao, Xinyan Xiao, Hao Liu, Bolei He, Hua Wu, Haifeng Wang, and Feng Wu. 2020. SKEP: Sentiment Knowledge Enhanced Pretraining for Sentiment Analysis. *arXiv preprint arXiv:2005.05635.*

Jiateng Xie, Zhilin Yang, Graham Neubig, and Jaime Smith, Noah A. andCarbonell. 2018. Neural Cross-Lingual Named Entity Recognition with Minimal Resources. In *EMNLP*, pages 369–379.

Mengge Xue, Weiming Cai, Jinsong Su, Linfeng Song, Yubin Ge, Yubao Liu, and Bin Wang. 2019a. Neural Collective Entity Linking Based on Recurrent Random Walk Network Learning. In *IJCAI*, pages 5327–5333.

Mengge Xue, Bowen Yu, Tingwen Liu, Erli Meng, and Bin Wang. 2019b. Porous Lattice Transformer Encoder for Chiense NER. *arXiv: Computation and Language.*

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly Supervised NER with Partial Annotation Learning and Reinforcement Learning. In *COLING*, pages 2159–2169.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized Autoregressive Pretraining for Language Understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer Learning for Sequence Tagging with Hierarchical Recurrent Networks. In *ICLR*.

Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. Beyond Word Attention: Using Segment Attention in Neural Relation Extraction. In *IJCAI*, pages 5401–5407.

Yue Zhang and Jie Yang. 2018. Chinese NER Using Lattice LSTM. In *ACL*, pages 1554–1564.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. In *ACL*, pages 1441–1451.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual Adversarial Neural Transfer for Low-Resource Named Entity Recognition. In *ACL*, pages 3461–3471.