Entity Enhanced BERT Pre-training for Chinese NER

Chen Jia[†‡*], **Yuefeng Shi**[‡*], **Qinrong Yang**[◇] and **Yue Zhang**[‡§]
[†]Fudan University, China
[‡]School of Engineering, Westlake University, China
[◇]Tuiwen Technology Inc., China
[§]Institute of Advanced Technology, Westlake Institute for Advanced Study, China
{jiachen,shiyuefeng,zhangyue}@westlake.edu.cn
yang_bigarm@hotmail.com

Abstract

Character-level BERT pre-trained in Chinese suffers a limitation of lacking lexicon information, which shows effectiveness for Chinese NER. To integrate the lexicon into pre-trained LMs for Chinese NER, we investigate a semi-supervised entity enhanced BERT pre-training method. In particular, we first extract an entity lexicon from the relevant raw text using a new-word discovery method. We then integrate the entity information into BERT using Char-Entity-Transformer, which augments the self-attention using a combination of character and entity representations. In addition, an entity classification task helps inject the entity information into model parameters in pre-training. The pre-trained models are used for NER fine-tuning. Experiments on a news dataset and two datasets annotated by ourselves for NER in long-text show that our method is highly effective and achieves the best results.

1 Introduction

As a fundamental task in information extraction, named entity recognition (NER) is useful for NLP tasks such as relation extraction (Zelenko et al., 2003), event detection (Kumaran and Allan, 2004) and machine translation (Babych and Hartley, 2003). We investigate Chinese NER (Gao et al., 2005), for which the state-of-the-art methods use a character-based neural encoder augmented with lexicon word information (Zhang and Yang, 2018; Gui et al., 2019a,b; Xue et al., 2019).

NER has been a challenging task due to the flexibility of named entities. There can be a large number of OOV named entities in the open domain, which poses challenges to supervised learning algorithms. In addition, named entities can be ambiguous. Take Figure 1 for example. The term "老妇人(the old lady)" literally means "older woman".

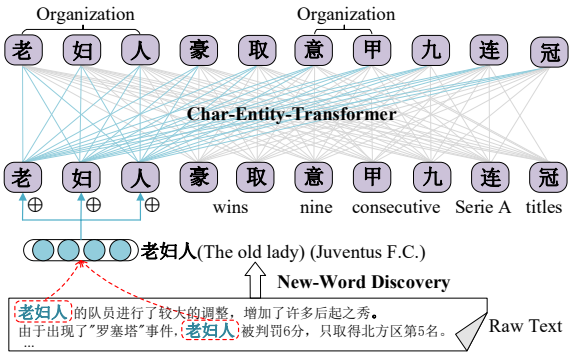


Figure 1: Entity enhanced pre-training for NER. "老妇人(The old lady)", the nickname of a football club Juventus F.C., is extracted by new-word discovery and integrated into the Transformer structure. After pre-training, the embedding of "老妇人(The old lady)" has the global information and correctly classifies itself as an ORG, which also helps recognize "意甲(Serie A)" as an ORG.

However, in the context of football news, it means the nickname of a football club Juventus F.C.. Thus entity lexicons that contain domain knowledge can be useful for the task (Radford et al., 2015; Xu et al., 2019).

Intuitively, such lexicons can be collected automatically from a set of documents that are relevant to the input text. For example, in the news domain, a set of news articles in the same domain and concurrent with the input text can contain highly relevant entities. In the finance domain, the financial report of a company over the years can serve as a context for collecting named entities when conducting NER for a current-year report. In the science domain, relevant articles can mention the same technological terms, which can facilitate recognition of the terms. In the literature domain, a full-length novel itself can serve as a context for mining entities.

There has been work exploiting lexicon knowledge for NER (Passos et al., 2014; Zhang and Yang,

[*]Equal contribution.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 6384–6396,
November 16–20, 2020.

2018). However, little has been done integrating entity information into BERT, which gives the state-of-the-art for Chinese NER. We consider enriching BERT (Devlin et al., 2019) with automatically extracted domain knowledge as mentioned above. In particular, We leverage the strength of new-word discovery on large documents by calculating point-wise mutual information to identify entities in the documents. Information over such entities is integrated into the BERT model by replacing the original self-attention modules (Vaswani et al., 2017) with a Char-Entity-Self-Attention mechanism, which captures the contextual similarities of characters and document-specific entities, and explicitly combines character hidden states with entity embeddings in each layer. The extended BERT model is then used for both LM pre-training and NER fine-tuning.

We investigate the effectiveness of this semi-supervised framework on three NER datasets, including a news dataset and two annotated datasets (novels and financial reports) by ourselves, which aims to evaluate NER for long-text. We make comparisons with two groups of state-of-the-art Chinese NER methods, including BERT and ERNIE (Sun et al., 2019a,b). For more reasonable comparison, we also complement both BERT and ERNIE with our entity dictionary and further pre-train on the same raw text as ours.

Results on the three datasets show that our method outperforms these methods and achieves the best results, which demonstrates the effectiveness of the proposed Char-Entity-Transformer structure for integrating entity information in LM pre-training for Chinese NER. To our knowledge, we are the first to investigate how to make use of the scale of the input document text for enhancing NER. Our code and NER datasets are released at `https://github.com/jiachenwestlake/Entity_BERT`.

2 Related Work

Chinese NER. Previous work has shown that character-based approaches perform better for Chinese NER than word-based approaches because of the freedom from Chinese word segmentation errors (He and Wang, 2008; Liu et al., 2010; Li et al., 2014). Lexicon features have been applied so that the external word-level information enhances NER training (Luo et al., 2015; Zhang and Yang, 2018; Gui et al., 2019a,b; Xue et al., 2019). However, these methods are supervised models, which cannot deal with a dataset with relatively little labeled data. We address this problem by using a semi-supervised method by using a pre-trained LM.

Pre-trained Language Models. Pre-trained language models have been applied as an integral component in modern NLP systems for effectively improving downstream tasks (Peters et al., 2018; Radford et al., 2019; Devlin et al., 2019; Yang et al., 2019; Liu et al., 2019b). Recently, there is an increasing interest to augment such contextualized representation with external knowledge (Zhang et al., 2019; Liu et al., 2019a; Peters et al., 2019). These methods focus on augmenting BERT by integrating KG embeddings such as TransE (Bordes et al., 2013). Different from the line of work, our model dynamically integrates document-specific entities without using any pre-trained entity embeddings. A more similar method is ERNIE (Sun et al., 2019a,b), which enhances BERT through knowledge integration. In particular, instead of masking individual subword tokens as BERT does, ERNIE is trained by masking full entities. The entity-level masking trick for ERNIE pre-training can be seen as an implicit way to integrate entity information through error backpropagation. In contrast, our method uses an explicit way to encode the entities to the Transformer structure.

3 Method

As shown in Figure 2, the overall architecture of our method can be viewed as a Transformer structure with multi-task learning. There are three output components, namely masked LM, entity classification and NER. With only the masked language model component, the model resembles BERT without the next sentence prediction task, and the entity classification task is added to enhance pre-training. While only NER outputs are yielded, the model is a sequence labeler for NER. We integrate entity-level information by extending the standard Transformer.

3.1 New-Word Discovery

In order to enhance a BERT LM with document-specific entities, we adopt an unsupervised method by Bouma (2009) to discover candidate entities automatically, which calculates the Mutual Information (MI) and Left and Right Entropy Measures between consecutive characters, respectively, and

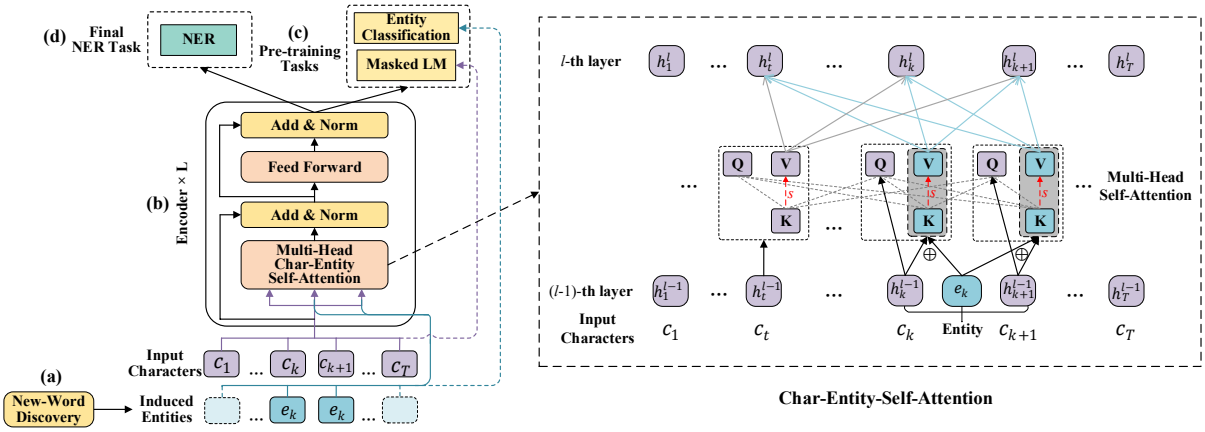


Figure 2: Overall model structure. The semi-supervised pre-training method has four main components: (a) New-word discovery based on information entropy; (b) A Char-Entity-Transformer component to enhance the character-level contextual representation with the acquired entities; (c) Output layers for masked language modeling and entity classification; (d) NER output layer.

adds these three values as the validity score of possible entities. The specific induction process is shown in Appendix A.

3.2 Char-Entity-Transformer

We construct models based on the Transformer structure of $\text{BERT}_{\text{BASE}}$ for Chinese (Devlin et al., 2019). In order to make use of the extracted entities, we extend the baseline Transformer to Char-Entity-Transformer, which consists of a stack of multi-head Char-Entity-Self-Attention blocks. We denote the hidden dimension of characters and the hidden dimension of new-words (entities) as H_c and H_e, respectively. L is the number of layers, and A is the number of self-attention heads.

Baseline Transformer. The Transformer encoder (Vaswani et al., 2017) is constructed with a stacked layer structure. Each layer consists of a multi-head self-attention sub-layer. In particular, given the hidden representation of a sequence $\{\mathbf{h}_1^{l-1}, ..., \mathbf{h}_T^{l-1}\}$ for the $(l-1)$-th layer and packed together as a matrix $\mathbf{h}^{l-1} \in \mathbb{R}^{T \times H_c}$, the self-attention function of the l-th layer is a linear transformation on the *Value* $\mathbf{V}^l$ space by means of *Query* $\mathbf{Q}^l$ and *Key* $\mathbf{K}^l$ mappings, represented as:

$$\{\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l\} = \{\mathbf{h}^{l-1}\mathbf{W}_q^l, \mathbf{h}^{l-1}\mathbf{W}_k^l, \mathbf{h}^{l-1}\mathbf{W}_v^l\}$$
$$\text{Atten}(\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l) = \text{softmax}\left\{\frac{\mathbf{Q}^l {\mathbf{K}^l}^\top}{\sqrt{d_k}}\right\}\mathbf{V}^l, \quad (1)$$

where d_k is the scaling factor and $\mathbf{W}_q^l, \mathbf{W}_k^l, \mathbf{W}_v^l \in \mathbb{R}^{H_c \times H_c}$ are trainable parameters of the l-th layer. The result of $\text{Atten}(\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l)$ is further fed to a feed-forward network sub-layer with layer normalization to obtain the final representation $\mathbf{h}^l$ of the l-th layer.

Algorithm 1 Maximum entity matching.

Input: Entity dictionary $\mathcal{E}_{ent}$; input character sequence $\boldsymbol{c} = \{c_1, \ldots, c_T\}$.
Output: Entity labeled sequence $\boldsymbol{e} = \{e_1, \ldots, e_T\}$.
Initialize: $i \leftarrow 1$; $\{e_1, ..., e_T\} \leftarrow \{0, ..., 0\}$.
# Maximum entity matching process
while $i \leq T-1$ **do**
  for $j \in \{i+1, \ldots, T\}$ **do**
    if $(\boldsymbol{c}_{i\sim j} \leftarrow \{c_i, \ldots, c_j\}) \in \mathcal{E}_{ent}$ **then**
      $\{e_i, \ldots, e_j\} \leftarrow \{\mathcal{E}_{ent}(\boldsymbol{c}_{i\sim j}), ..., \mathcal{E}_{ent}(\boldsymbol{c}_{i\sim j})\}$
      $k \leftarrow j$
    end if
    $i \leftarrow \max\{k+1,\ i+1\}$
  end for
end while

Char-Entity matching. Given a character sequence $\boldsymbol{c} = \{c_1, \ldots, c_T\}$ and an extracted entity dictionary $\mathcal{E}_{ent}$[1], we use the maximum entity matching algorithm to obtain the corresponding entity-labeled sequence $\boldsymbol{e} = \{e_1, ..., e_T\}$. In particular, we label each character with the index of the longest entity in $\mathcal{E}_{ent}$ that includes the character, and label characters with no entity matches with 0. The process is summarized in Algorithm 1.

Char-Entity-Self-Attention. The Char-Entity-Self-Attention structure is shown in Figure 2 (right). Following BERT (Devlin et al., 2019), given a character sequence $\boldsymbol{c} = \{c_1, \ldots, c_T\}$, the representation of the t-th ($t \in \{1, \ldots, T\}$) character in the input layer is the sum of character, segment and

[1]Entities extracted by new-word discovery in Sec 3.1.

position embeddings, represented as:

$$\mathbf{h}_t^1 = \mathbf{E}_c[c_t] + \mathbf{E}_s[0] + \mathbf{E}_p[t] \tag{2}$$

where $\mathbf{E}_c$, $\mathbf{E}_s$, $\mathbf{E}_p$ represent character embedding lookup table, segment embedding lookup table and position embedding lookup table, respectively. In particular, the segment index $s \in \{0, 1\}$ is used to distinguish the order of input sentences for the next sentence prediction task in BERT (Devlin et al., 2019), which is not included in our method. Thus we set the segment index s as a constant 0.

Given the $(l-1)$-th layer character hidden sequence $\{\mathbf{h}_1^{l-1}, \ldots, \mathbf{h}_T^{l-1}\}$, the l-th layer *Query* matrix $\mathbf{Q}^l = \{\mathbf{q}_t^l\}_{t=1}^T \in \mathbb{R}^{T \times H_c}$ is computed as the baseline self-attention, but for the *Key* matrix $\mathbf{K}^l = \{\mathbf{k}_t^l\}_{t=1}^T \in \mathbb{R}^{T \times H_c}$ and the *Value* matrix $\mathbf{V}^l = \{\mathbf{v}_t^l\}_{t=1}^T \in \mathbb{R}^{T \times H_c}$, we compute the combination of the character hidden and its corresponding entity embedding as:

$$\begin{aligned}
\mathbf{q}_t^l &= {\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,q}^l; \\
\mathbf{k}_t^l &= \begin{cases} {\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,k}^l & \text{if } e_t = 0, \\ \frac{1}{2}\left({\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,k}^l + \mathbf{E}_{ent}^\top[e_t]\mathbf{W}_{e,k}^l\right) & \text{else;} \end{cases} \\
\mathbf{v}_t^l &= \begin{cases} {\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,v}^l & \text{if } e_t = 0, \\ \frac{1}{2}\left({\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,v}^l + \mathbf{E}_{ent}^\top[e_t]\mathbf{W}_{e,v}^l\right) & \text{else,} \end{cases}
\end{aligned} \tag{3}$$

where $\mathbf{W}_{h,q}^l, \mathbf{W}_{h,k}^l, \mathbf{W}_{h,v}^l \in \mathbb{R}^{H_c \times H_c}$ are trainable parameters of the l-th layer, and $\mathbf{W}_{e,k}^l, \mathbf{W}_{e,v}^l \in \mathbb{R}^{H_e \times H_c}$ are trainable parameters for the corresponding entities. $\mathbf{E}_{ent}$ is the entity embedding lookup table.

As shown in Eq. (3), if there is no corresponding entity for a character, the representation is equal to the baseline self-attention. To show how a character and its corresponding entity are encoded jointly, we denote a pack of entity embeddings $\{\mathbf{E}_{ent}[e_1], \ldots, \mathbf{E}_{ent}[e_T]\}$ as $\mathbf{e} \in \mathbb{R}^{T \times H_e}$. The attention score of the i-th character in the l-th layer $\mathbf{S}_i^l$ is computed as:

$$\begin{aligned}
\mathbf{S}_i^l &= \text{softmax}\left\{\frac{\mathbf{q}_i^l {\mathbf{K}^l}^\top}{\sqrt{d_k}}\right\} \\
&= \text{softmax}\left\{\frac{\mathbf{q}_i^l \left(\mathbf{h}^{l-1}\mathbf{W}_{h,k}^l + \mathbf{e}\mathbf{W}_{e,k}^l\right)^\top}{2\sqrt{d_k}}\right\} \\
&= \left\{\frac{\sqrt{s_t^c s_t^e}}{\sum_j \sqrt{s_j^c s_j^e}}\right\}_{t=1}^T \\
s.t. \quad s_t^c &= \exp\left(\frac{\mathbf{q}_i^l ({\mathbf{h}_t^{l-1}}^\top \mathbf{W}_{h,k}^l)^\top}{\sqrt{d_k}}\right); \\
s_t^e &= \exp\left(\frac{\mathbf{q}_i^l ({\mathbf{e}_t}^\top \mathbf{W}_{e,k}^l)^\top}{\sqrt{d_k}}\right),
\end{aligned} \tag{4}$$

where a char-to-char attention score s_t^c is computed equally to the baseline self-attention. A char-to-entity attention score s_t^e represents the similarity between a character and the corresponding entity.

Before normalization, the attention score of the i-th character and t-th character $\{\mathbf{S}_i^l\}_t$ is $\sqrt{s_t^c s_t^e}$, which is the geometric mean of s_t^c and s_t^e. This shows that the similarity between two characters by Char-Entity-Self-Attention is computed as a combination of the char-to-char geometric distance and the char-to-entity geometric distance.

Given the attention score $\mathbf{S}_i^l$, $\text{Atten}(\mathbf{q}_i^l, \mathbf{K}^l, \mathbf{V}^l)$ is computed as a weighted sum of the *Value* $\mathbf{V}^l$, which is a combination of character values and entity values.

$$\begin{aligned}
\text{Atten}(\mathbf{q}_i^l, \mathbf{K}^l, \mathbf{V}^l) &= \mathbf{S}_i^l \mathbf{V}^l \\
&= \mathbf{S}_i^l \frac{1}{2}\left(\mathbf{h}^{l-1}\mathbf{W}_{h,v}^l + \mathbf{e}\mathbf{W}_{e,v}^l\right)
\end{aligned} \tag{5}$$

3.3 Masked Language Modeling Task

Following Devlin et al. (2019), we use the masked LM (MLM) task for pre-training. In particular, given a character sequence $\boldsymbol{c} = \{c_1, \ldots, c_T\}$, we randomly select 15% of input characters and replace them with [MASK] tokens.

Formally, given the the hidden outputs of the last layer $\{\mathbf{h}_1^L, \ldots, \mathbf{h}_T^L\}$, for each masked character c_t in a character sequence, the prediction probability of MLM $p(c_t|\boldsymbol{c}_{<t} \cup \boldsymbol{c}_{>t})$ is computed as:

$$p(c_t|\boldsymbol{c}_{<t} \cup \boldsymbol{c}_{>t}) = \frac{\exp(\mathbf{E}_c^\top[c_t]\mathbf{h}_t^L + b_{c_t})}{\sum_{c \in \mathcal{V}} \exp(\mathbf{E}_c^\top[c]\mathbf{h}_t^L + b_c)}, \tag{6}$$

where $\mathbf{E}_c$ is the character embedding lookup table. $\mathcal{V}$ is the character vocabulary.

3.4 Entity Classification Task

In order to further enhance the coherence between characters and their corresponding entities, we propose an entity classification task, which predicts the specific entity that the current character belongs to. A theoretical explanation of this task is to maximize the mutual information $\mathcal{I}(e; c)$ between the character $c \sim p(c)$ and the corresponding entity $e \sim p(e)$, where $p(c)$ and $p(e)$ represent the probability distributions of c and e, respectively.

$$\begin{aligned}
\mathcal{I}(e; c) &= H(e) - H(e|c) \\
&= H(e) + \mathbb{E}_{c \sim p(c)}\left[\mathbb{E}_{e \sim p(e|c)}\left[\log p(e|c)\right]\right] \\
&= H(e) + \mathbb{E}_{c \sim p(c), e \sim p(e|c)}\left[\log p(e|c)\right],
\end{aligned} \tag{7}$$

where $H(e)$ indicates the entropy of $e \sim p(e)$, represented as $H(e) = -\mathbb{E}_{e \sim p(e)}[\log p(e)]$, which

is a constant corresponding to the frequency of entities in a document. Thus the maximization of the mutual information $\mathcal{I}(e;c)$ is equivalent to the maximization of the expectation of $\log p(e|c)$.

Considering the computational complexity due to the excessive number of candidate entities, we employ sampling softmax for output prediction (Jean et al., 2015). Formally, given the hidden outputs of last layer $\{\mathbf{h}_1^L, \ldots, \mathbf{h}_T^L\}$ and its corresponding entity labeled sequence $\boldsymbol{e} = \{e_1, \ldots, e_T\}$, we compute the probability of each character c_t (s.t. $e_t \neq 0$) aligning with its corresponding entity e_t as:

$$p(e_t|c_t) = \frac{1}{Z} \exp\left(\mathbf{E}_{ent}^\top[e_t]\mathbf{h}_t^L + b_{e_t}\right)$$
$$s.t. \quad Z = \sum_{e \in \{e_t \cup \mathcal{R}^-\}} \exp\left(\mathbf{E}_{ent}^\top[e]\mathbf{h}_t^L + b_e\right), \tag{8}$$

where $\mathcal{R}^-$ represents the randomly sampled negative set from the candidate entities of the current input document. $\mathbf{E}_{ent}$ is the entity embedding lookup table and b_e is the bias of entity e.

3.5 NER Task

Given the hidden outputs of the last layer $\{\mathbf{h}_1^L, \ldots, \mathbf{h}_T^L\}$, the output layer for NER is a linear classifier $f : \mathbb{R}^{H_c} \rightarrow \mathcal{Y}$, where $\mathcal{Y}$ is a $(m-1)$-simplex and m is the number of NER tags. The probability that the character c_t aligns with the k-th NER tag is computed using softmax:

$$p(k|c_t) = \frac{\exp(\mathbf{w}_k{}^\top \mathbf{h}_t^L + b_k)}{\sum_{j \in \{1,\ldots,m\}} \exp(\mathbf{w}_j{}^\top \mathbf{h}_t^L + b_j)}, \tag{9}$$

where $\mathbf{w}_k \in \mathbb{R}^{H_c}$ and b_k are trainable parameters specific to the k-th NER tag. We adopt the B-I-O tagging scheme for NER.

3.6 Training Procedure

Our model is initialized using a pre-trained BERT model[2], and the other parameters are randomly initialized. During training, we first pre-train an LM over all of the raw text to acquire the entity-enhanced model parameters and then fine-tune the parameters using the NER task.

Pre-training. Given raw text with induced entities $\mathcal{D}_{lm} = \{(\boldsymbol{c}^n, \boldsymbol{e}^n)\}_{n=1}^N$, where $\boldsymbol{c}^n$ is a character sequence and $\boldsymbol{e}^n$ is its corresponding entity sequence detected by Algorithm 1, we feed each training character sequence and its corresponding entities into the Char-Entity-Transformer to obtain last layer character hiddens.

We denote the masked subset of $\mathcal{D}_{lm}$ as $\mathcal{D}_{lm}^+ = \{(n,t)|\boldsymbol{c}_t^n = [\text{MASK}], \boldsymbol{c}^n \in \mathcal{D}_{lm}\}$, the loss of the masked LM task is:

$$\mathcal{L}_{MLM} = -\sum_{(n,t)\in\mathcal{D}_{lm}^+} \log p(\boldsymbol{c}_t^n | \boldsymbol{c}_{<t}^n \cup \boldsymbol{c}_{>t}^n) \tag{10}$$

We denote the entity prediction subset of $\mathcal{D}_{lm}$ as $\mathcal{D}_{lm}^e = \{(n,t)|\boldsymbol{e}_t^n \neq 0, \boldsymbol{c}^n \in \mathcal{D}_{lm}\}$, the loss of the entity classification task is:

$$\mathcal{L}_{ENC} = -\sum_{(n,t)\in\mathcal{D}_{lm}^e} \log p(\boldsymbol{e}_t^n | \boldsymbol{c}_t^n) \tag{11}$$

To jointly train the masked LM task and the entity classification task in pre-training, we minimize the overall loss:

$$\mathcal{L}_{LM} = \mathcal{L}_{MLM} + \mathcal{L}_{ENC} \tag{12}$$

Fine-tuning. Given an NER dataset $\mathcal{D}_{ner} = \{(\boldsymbol{c}^n, \boldsymbol{y}^n)\}_{n=1}^N$, we train the NER output layer and fine-tune both the pre-trained LM and entity embeddings by the NER loss:

$$\mathcal{L}_{NER} = -\sum_{n=1}^{N}\sum_{t=1}^{T} \log p(\boldsymbol{y}_t^n | \boldsymbol{c}_t^n) \tag{13}$$

The overall process of pre-training and fine-tuning is summarized in Algorithm 2.

Algorithm 2 Pre-training and fine-tuning.

Input: Raw text $\mathcal{D}_{lm}$, entity dict $\mathcal{E}_{ent}$, NER dataset $\mathcal{D}_{ner}$
Parameters: Entity embeddings $\mathbf{E}_{ent}$, Transformer layers $\mathbf{W}_T$, MLM output layer $\mathbf{W}_{MLM}^o$, entity classification output layer $\mathbf{W}_{ENC}^o$, NER output layer $\mathbf{W}_{NER}^o$.
Output: Target NER model

while LM pre-training stopping condition is not met **do**
  $\boldsymbol{x} \leftarrow \mathcal{D}_{lm}$; $\boldsymbol{e} \leftarrow$ Entity-Match$(\boldsymbol{x}; \mathcal{E}_{ent})$
  $\mathbf{h}^L \leftarrow$ Char-Entity-Transformer$(\boldsymbol{x}, \boldsymbol{e}; \mathbf{W}_T, \mathbf{E}_{ent})$
  $\mathcal{L}_{MLM} \leftarrow$ MLM$(\mathbf{h}^L, \boldsymbol{x}; \mathbf{W}_{MLM}^o)$ [MLM loss]
  $\mathcal{L}_{ENC} \leftarrow$ ENC$(\mathbf{h}^L, \boldsymbol{e}; \mathbf{W}_{ENC}^o)$ [Ent. class. loss]
  $\mathcal{L}_{LM} \leftarrow \mathcal{L}_{MLM} + \mathcal{L}_{ENC}$
  Update $\{\mathbf{W}_T, \mathbf{E}_{ent}, \mathbf{W}_{MLM}^o, \mathbf{W}_{ENC}^o\}$ by $\mathcal{L}_{LM}$
end while
while NER fine-tuning stopping condition is not met **do**
  $\{\boldsymbol{x}, \boldsymbol{y}\} \leftarrow \mathcal{D}_{ner}$; $\boldsymbol{e} \leftarrow$ Entity-Match$(\boldsymbol{x}; \mathcal{E}_{ent})$
  $\mathbf{h}^L \leftarrow$ Char-Entity-Transformer$(\boldsymbol{x}, \boldsymbol{e}; \mathbf{W}_T, \mathbf{E}_{ent})$
  $\mathcal{L}_{NER} \leftarrow$ NER$(\mathbf{h}^L, \boldsymbol{y}; \mathbf{W}_{NER}^o)$ [NER loss]
  Update $\{\mathbf{W}_T, \mathbf{E}_{ent}, \mathbf{W}_{NER}^o\}$ by $\mathcal{L}_{NER}$
end while

4 Experiments

We empirically verify the effectiveness of entity enhanced BERT pre-training on different NER datasets. In addition, we also investigate how different components in the model impact the performance of NER with different settings.

[2] https://github.com/google-research/bert, which is pre-trained on Chinese Wikipedia.

Dataset			#Sentence	#Entity
News	Train		5.2K	10.8K
	Dev		0.6K	1.2K
	Test	GAM (Game)	0.3K	0.5K
		ENT (Entertainment)	48	0.1K
		LOT (Lottery)	0.1K	0.3K
		FIN (Finance)	0.3K	0.6K
		All	0.7K	1.5K
Novels	Train		6.7K	25.5K
	Dev		2.6K	10.3K
	Test	天荒神域 (*Story in Myth*)	0.8K	3.2K
		道破天穹 (*Taoist Story*)	0.9K	3.5K
		茅山诡术师 (*MS Wizards*)	0.9K	3.5K
		All	2.6K	10.2K
Financial Report Test			2.0K	4.1K

Table 1: Statistics of the three datasets.

Hyperparameter	Pre-train	Fine-tune
Epoch number	3	10
Max sentence length	180	-
Batch size	32	32
Entity sample number	5	-
Optimizer	Adam	Adam
Learning rate	$3e^{-5}$	$5e^{-5}$
Lr decay rate	0.01	-
Warmup proportion	0.1	-

Table 2: Hyperparameters.

4.1 Datasets

We conduct experiments on three datasets, including one public NER dataset, CLUENER-2020 (Xu et al., 2020), and two datasets annotated by ourselves, which are also contributions of this paper. The statistics of the datasets are listed in Table 1.

News dataset. We use the CLUENER-2020 (Xu et al., 2020) dataset. Compared with OntoNotes (Weischedel et al., 2012) and MSRA (Levow, 2006) datasets for Chinese news NER, CLUENER-2020 is constructed as a fine-grained Chinese NER dataset with 10 entity types, and its labeled sentences belong to different news domains rather than one domain. We randomly sample 5.2K, 0.6K and 0.7K sentences from the original CLUENER-2020 dataset as the training[3], dev and test sets, respectively. The corresponding raw text is taken from THUCNews (Sum et al., 2016) in four news domains[4], namely GAM (game), ENT (entertainment), LOT (lottery) and FIN (finance), with a total number of about 100M characters. The detailed entity statistics are shown in Appendix **B.1**.

Novel dataset. We select three Chinese Internet novels, titled "天荒神域(*Stories in Myth*)", "道破天穹(*Taoist Stories*)" and "茅山诡术师(*Maoshan Wizards*)", respectively, and manually label around 0.9K sentences for each novel as the development and test sets. We also label around 6.7K sentences from six other novels for the training set. Considering the literature genre, we annotate six types of entities. Besides, we use the original text of the nine novels with about 48M characters for pre-training. The details of annotation and entity statistics are shown in Appendix **B.2**.

Financial report dataset. We collect annual financial reports of 12 banks in China for five years and select about 2k sentences to annotate as the test set. The annotation rules follow the MSRA dataset (Levow, 2006), and the annotation process follows the novel dataset. In addition, we use the MSRA training and dev sets as our training and dev data. The unannotated annual reports of about 26M characters are used in LM pre-training. The detailed entity statistics are shown in Appendix **B.3**.

4.2 Experimental Settings

Model size. Our model is constructed using $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019), with the number of layers $L = 12$, the number of self-attention heads $A = 12$, the hidden size of characters $H_c = 768$ and the hidden size of entities $H_e = 64$. The total amount of non-embedding model parameters is about 86M. The total amount of non-embedding parameters of $\text{BERT}_{\text{BASE}}$ is about 85M. The entity integration module occupies only a small proportion in the whole model. Therefore, it has little impact on training efficiency.

Hyperparameters. For pre-training, we largely follow the default hyperparameters of BERT (Devlin et al., 2019). We use the Adam optimizer with an initial learning rate of $5e^{-5}$ and a maximum epoch number of 10 for fine-tuning. We list the details about pre-training and fine-tuning hyperparameters in Table 2.

Baselines. We compare our methods with three groups of state-of-the-art methods to Chinese NER.

BERT baselines. BERT (Devlin et al., 2019) directly fine-tunes a pre-trained Chinese BERT on

[3]In practice, a little manual labeling can be performed on each news domain separately for the best results. However, considering the expense of performing experiments to study the influence of training data scale, we use a single set of training data for all the news domains. This setting is also used for the novel dataset.

[4]The original CLUENER-2020 dataset has no domain divisions, but our method aims to leverage domain-specific entity information for NER. Thus we select some specific news domains according to raw text from THUCNews and construct an entity dictionary for each domain. We also released a smaller version of CLUENER-2020 with domain divisions.

Methods	News Dataset					Novel Dataset				Financial Report Dataset
	GAM	ENT	LOT	FIN	**All**	天荒神域 *St. in Myth*	道破天穹 *Taoist St.*	茅山诡术师 *MS Wizards*	**All**	
BILSTM	66.60	80.29	78.42	70.45	71.36	64.59	61.93	51.27	59.10	58.64
BILSTM+ENT	66.41	75.00	78.23	71.59	71.20	79.71	52.29	66.76	66.78	68.97
LATTICE	68.35	77.03	82.98	74.45	73.96	67.04	66.19	58.75	63.89	76.11
LATTICE (REENT)	63.93	73.38	75.89	71.43	69.62	69.95	30.62	38.88	47.60	66.67
ERNIE	69.36	80.84	**83.21**	**77.51**	75.73	73.66	76.52	68.48	72.83	82.99
ERNIE+FUR+ENT	67.92	86.52	78.29	76.66	74.59	78.51	76.45	72.55	75.78	83.48
BERT	68.67	80.14	77.36	76.88	74.22	75.50	76.68	68.58	73.50	82.76
BERT+FUR	69.22	78.79	81.34	77.30	75.14	74.17	76.06	69.60	73.22	82.68
BERT+FUR+ENT	62.37	85.71	75.79	70.29	69.59	80.11	76.36	72.48	76.23	74.37
Ours	**70.90**	**87.11**	82.73	77.18	**76.66**†	**82.33**	**77.70**	**73.08**	**77.58**†	**87.05**†

Table 3: Overall results on the three datasets. † indicates statistical significance with $p < 0.01$ by t-test.

NER. BERT+FUR uses the same raw text as ours to further pre-train the BERT with only the masked LM task. BERT+FUR+ENT uses the sum of character embeddings and the corresponding entity embeddings by the same entity matching algorithm as ours only in the input layer, and then further pre-trains BERT on the same raw text as ours.

ERNIE baselines. ERNIE[5] (Sun et al., 2019a,b) enhances BERT through knowledge integration using a entity-level masked LM task and more raw text from the Web resources, which achieves the currently best results on Chinese NER. ERNIE+FUR+ENT is a stronger baseline, which uses the same entity dictionary as ours for entity-level masking and further pre-trains ERNIE on the same raw text as ours.

LSTM baselines. We compare character-level BILSTM (Lample et al., 2016) and BILSTM+ENT, which concatenates the character embeddings and its corresponding entity embeddings as inputs. We also compare a gazetteer based method LATTICE (Zhang and Yang, 2018) and LATTICE (REENT), which replaces the word gazetteer of LATTICE with our entity dictionary for fair comparison. We use the same embeddings as (Zhang and Yang, 2018), which are pre-trained on Giga-Word[6] using Word2vec (Mikolov et al., 2013). The entity embeddings are randomly initialized and fine-tuned during training.

4.3 Overall Results

The overall F_1-scores are listed in Table 3.

Comparison with BERT baselines. BERT+FUR achieves a slightly better result than BERT on the news dataset **All** (75.14% F_1 v.s. 74.22% F_1), but similar results on the novel dataset **All** and the financial report dataset. This shows that simply further pre-training BERT on document-specific raw text can hardly improve the performances. After using a naive method to integrate entity information, BERT+FUR+ENT achieves significantly better results on the novel dataset **All** (76.23% F_1 v.s. 73.22% F_1) compared to BERT+FUR, but lower F_1 on the news and the financial report datasets, which shows that this naive method cannot effectively benefit from the entities of arbitrary text genre.

Compared with BERT, **Ours** achieves more significantly better results on the novel dataset and the fiancial report dataset than the news dataset (at least over 4% F_1 v.s. 2.4% F_1), indicating the effectiveness of **Ours** for long-text genre. Compared with all of the BERT baselines, **Ours** achieves significant improvement (over at least 1.5% F_1 on the news dataset **All**, over 1.3% F_1 on the novel dataset **All** and over 4% F_1 on the financial report dataset), which shows that the Char-Entity-Transformer structure effectively integrates the document-specific entities extracted by new-word discovery and benefits for Chinese NER.

Comparison with the state-of-the-art. We make comparisons with ERNIE baselines. Even though ERNIE uses more raw text and entity information from the Web resources for pre-training, **Ours** outperforms ERNIE significantly (about 1% F_1 on the news dataset **All**, over 4% F_1 on both the novel dataset **All** and the financial report dataset), which shows the importance of document-specific entities for pre-training.

Using the same entity dictionary as **Ours** to further pre-train ERNIE on the same raw text as **Ours**, ERNIE+FUR+ENT achieves better results on the novel dataset and the financial report dataset

[5] https://github.com/PaddlePaddle/ERNIE/tree/repro

[6] https://catalog.ldc.upenn.edu/LDC2011T13

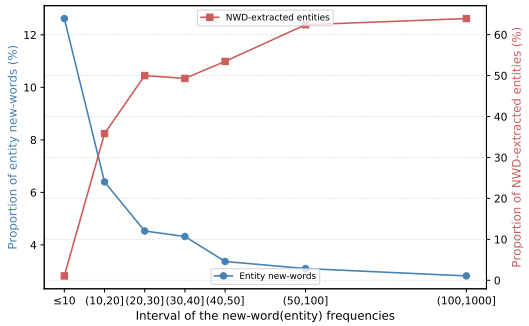


Figure 3: Performances of new-word discovery against word frequency on the news dataset. We ignore the interval >1000, because it occupies less than 5% new-words or entities.

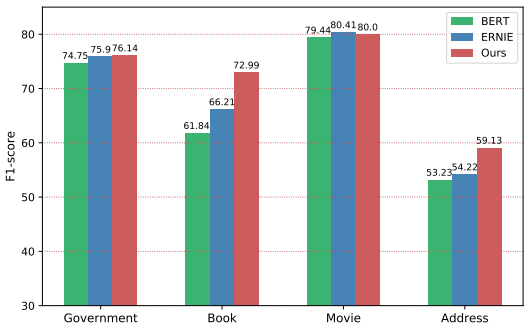


Figure 4: Detailed comparison on the news dataset.

Method	P	R	F_1	ΔF_1
FINAL	**81.19**	**74.27**	**77.58**	-
NO-ENT-CLASS	79.24	72.65	75.80	−1.78
HALF-RAW	78.37	72.75	75.46	−2.12
NO-PRETRAIN	78.17	72.51	75.23	−2.35
HALF-ENT	74.53	67.29	70.72	−6.86
N-GRAMS	73.92	63.48	68.30	−9.28
OPEN-DOMAIN	70.23	61.86	65.78	−11.80

Table 4: Ablation study on the novel dataset.

than ERNIE, but suffers a decrease on the news dataset **All**, which shows that integrating document-specific entity dictionary benefits ERNIE for Chinese NER in long-text genre. Compared with ERNIE+FUR+ENT, **Ours** achieves significant improvements, which shows that our explicit method of integrating entity information by the Char-Entity-Transformer structure is more effective than entity-level masking for Chinese NER.

Finally, BERT and ERNIE outperform the LSTM baselines on all of the three datasets, indicating the effectiveness of LM pre-training for Chinese NER.

4.4 Analysis

MI-based new-word discovery. Figure 3 illustrates the relationships between new-words extracted by the MI-based new-word discovery (NWD) and the named entities with the scope of the news dataset.

On the one hand, within the scope of the news dataset, the proportion of entities extracted by the MI-based NWD is relatively higher when they are more frequently appearing *n*-grams in the raw text (overall 31.04% of the named entities are extracted by the NWD), as shown by the red line in Figure 3. On the other hand, within the *n*-grams in the news dataset, new-words with lower frequencies extracted by the MI-based NWD are more likely to be named entities (overall 3.86% of new words within the news dataset are named entities), as shown by the blue line in Figure 3.

Fine-grained comparison. In order to study the performances of our method on different entity types, we make fine-grained comparisons on the news dataset, which has plenty of entity types in different news domains. Figure 4 illustrates F_1-scores of several typical entity types, including GOV (government), BOO (book), MOV (movie) and ADD (address), for fine-grained comparison on the news dataset with BERT and ERNIE. The trends are consistent with the overall results. The full table is shown in Appendix **C**.

Ablation study. As shown in Table 4, we use two groups of ablation study to investigate the effect of entity information.

(1) *Entity prediction task.* We consider (i) NO-ENT-CLASS, which does not use the entity classification task in pre-training; and (ii) NO-PRETRAIN, which does not use entity enhanced pre-training. Results of these methods suffer significantly decreases compared to **FINAL**, which shows that pre-training, especially with the entity classification task, plays an important role in integrating the entity information. In addition, we also explore the effect of raw text quantity. The result of (iii) HALF-RAW shows that a larger amount of the raw text is helpful.

(2) *Entity dictionary.* We consider (i) HALF-ENT, which uses 50% randomly selected entities from the original entity dictionary; (ii) *N*-GRAMS, which uses randomly selected *n*-grams from the raw text; (iii) OPEN-DOMAIN, which uses an open-domain dictionary from Jieba[7]. The results of these methods decrease significantly (at least over 6% F_1) compared to **FINAL**, which shows that document-

[7] http://github.com/fxsjy/jieba

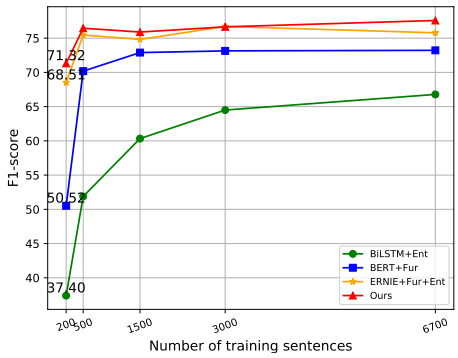


Figure 5: Influence of the amount of training data.

Sentence	我们可以运用花旗中国的全球经验和知识。 We can leverage Citi China 's global experience and knowledge. 核战的点子已经被《辐射》系列拿去用了。 The idea of nuclear warfare has been used for the *Radiation* series.
BERT	我们可以运用 花旗 COM 中国的全球经验和知识。 核战的点子已经被 《辐射》 GAM 系列拿去用了。
ERNIE	我们可以运用 花旗中国 COM 的全球经验和知识。 核战的点子已经被 《辐射》 GAM 系列拿去用了。
Ours	我们可以运用 花旗中国 COM 的全球经验和知识。 核战的点子已经被 《辐射》 MOV 系列拿去用了。

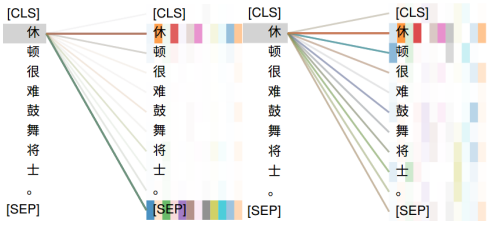
Table 5: Examples from the news test set. Green (Yellow) represents correct (incorrect) entities.

specific entity dictionary benefits the performance, and the new-word discovery method is effective for collecting entity dictionary.

The amount of NER training data. To compare performances of different models under different numbers of labeled training sentences, we randomly select different numbers of training sentences for training on the novel dataset.

As shown in Figure 5, in nearly unsupervised settings, **Ours** gives the largest improvements (33.92% F_1 over BILSTM+ENT, 20.80% F_1 over BERT+FUR and 2.81% F_1 over ERNIE+FUR+ENT). With only 500 training sentences, **Ours** achieves competitive result, which shows the effectiveness of our LM pre-training method for the few-shot setting.

Case study. Table 5 shows a case study on the news dataset. "花旗中国(Citi China)" is a COM (company) and "《辐射》(*Radiation*)" is a MOV (movie). Since the text genre and entities in the news are so different from Wikipedia, BERT does not recognize the company name "花旗中国(Citi China)" and misclassifies "《辐射》(*Radiation*)" as a GAM (game). Benefiting from integrating entity information into LM pre-training, both ERNIE and **Ours** recognize "花旗中国(Citi China)".



(a) BERT. (b) Ours.

Figure 6: Visualization of last layer attention scores. We use an example in the news dataset, "休顿很难鼓舞将士。(It is difficult for Hughton to encourage team members.)".

Ours uses document-specific entities to pre-train on raw news text. So with the global information, **Ours** also classifies "《辐射》(*Radiation*)" accurately as a MOV.

Visualization. Figure 6 uses BertViz (Vig, 2019) to visualize the last-layer attention patterns of "休(Hugh)" in a news example. BERT only has a higher attention score to itself, while **Ours** has relatively higher attention scores to all the tokens in the current entity "休顿(Hughton)", especially for the first attention head (in blue). This shows that **Ours** enables entity information to enhance the contextual representation.

5 Conclusion

We investigated an entity enhanced BERT pre-training method for Chinese NER. Results on a news dataset and two long-text NER datasets show that it is highly effective to explicitly integrate the document-specific entities into BERT pre-training with a Char-Entity-Transformer structure, and our method outperforms the state-of-the-art methods for Chinese NER.

Acknowledgments

We thank the anonymous reviewers for their helpful comments and suggestions. Chen Jia proposes the method. Chen Jia and Yuefeng Shi jointly conduct the experiments and write the paper. Qinrong Yang on behalf of Tuiwen Technology Inc. provides the novel dataset and runs new-word discovery. Yue Zhang is the corresponding author. This research is funded by the National Natural Science Foundation of China (NSFC No.61976180) and a grant from Tuiwen Technology Inc.

References

Bogdan Babych and Anthony Hartley. 2003. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Gerlof Bouma. 2009. Normalized (pointwise) mutual information in collocation extraction. *GSCL*, pages 31–40.

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) (NAACL-HLT)*, pages 4171–4186.

Jianfeng Gao, Mu Li, Chang-Ning Huang, and Andi Wu. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. 2019a. Cnn-based chinese ner with lexicon rethinking. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (AAAI)*, pages 4982–4988. AAAI Press.

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

Jingzhou He and Houfeng Wang. 2008. Chinese named entity recognition and word segmentation based on character. In *Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing*.

Hung Huu Hoang, Su Nam Kim, and Min-Yen Kan. 2009. A re-examination of lexical association measures. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 31–39.

Sebastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) (ACL-IJCNLP)*, pages 1–10.

Giridhar Kumaran and James Allan. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 297–304.

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 (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.

Haibo Li, Masato Hagiwara, Qi Li, and Heng Ji. 2014. Comparison of the impact of word segmentation on name tagging for chinese and japanese. In *LREC-2014*, pages 2532–2536.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2019a. K-bert: Enabling language representation with knowledge graph. *arXiv preprint arXiv:1909.07606*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhangxun Liu, Conghui Zhu, and Tiejun Zhao. 2010. Chinese named entity recognition with a sequence labeling approach: based on characters, or based on words? In *International Conference on Intelligent Computing*, pages 634–640.

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 879–888.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochem Med (Zagreb)*, 22(3):276–282.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3111–3119.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 78–86.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers) (NAACL-HLT)*, pages 2227–2237.

Matthew E Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9.

Will Radford, Xavier Carreras, and James Henderson. 2015. Named entity recognition with document-specific kb tag gazetteers. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 512–517.

Maosong Sum, Jingyang Li, Zhipeng Guo, Yu Zhao, Yabin Zheng, Xiance Si, and Zhiyuan Liu. 2016. Thuctc: an efficient chinese text classifier. *GitHub Repository*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019a. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Hao Tian, Hua Wu, and Haifeng Wang. 2019b. Ernie 2.0: A continual pre-training framework for language understanding. *arXiv preprint arXiv:1907.12412*.

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 *Advances in Neural Information Processing Systems*, pages 5998–6008.

Jesse Vig. 2019. A multiscale visualization of attention in the transformer model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 37–42.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Nianwen Xue, Martha Palmer, Jena D Hwang, Claire Bonial, et al. 2012. Ontonotes release 5.0.

Kai Xu, Zhenguo Yang, Peipei Kang, Qi Wang, and Wenyin Liu. 2019. Document-level attention-based bilstm-crf incorporating disease dictionary for disease named entity recognition. *Computers in biology and medicine*, 108:122–132.

Liang Xu, Qianqian Dong, Cong Yu, Yin Tian, Weitang Liu, Lu Li, and Xuanwei Zhang. 2020. Cluener2020: Fine-grained name entity recognition for chinese. *arXiv preprint arXiv:2001.04351*.

Mengge Xue, Bowen Yu, Tingwen Liu, Bin Wang, Erli Meng, and Quangang Li. 2019. Porous lattice-based transformer encoder for chinese ner. *arXiv preprint arXiv:1911.02733*.

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 (NeurIPS)*, pages 5754–5764.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research (JMLR)*, 3(Feb):1083–1106.

Yue Zhang and Jie Yang. 2018. Chinese ner using lattice lstm. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (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 *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1441–1451.

A New-Word Discovery

First, we calculate Mutual Information using this formula:

$$\mathrm{MI}\left(\boldsymbol{x},\boldsymbol{y}\right)=\log_2\frac{p(\boldsymbol{x}\oplus\boldsymbol{y})}{p(\boldsymbol{x})p(\boldsymbol{y})}, \tag{14}$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ represent two continual characters or words. $\oplus$ represents string concatenation. The notation $p(\cdot)$ represents the probability of a string occurs. Higher MI indicates that two sub-strings are more likely to form a new phrase.

Then we calculate the Left and Right Entropy Measures to distinguish the independence and boundary of candidate multi-word expressions (Hoang et al., 2009):

$$\begin{aligned}\mathrm{E}_L(\boldsymbol{w})&=-\sum_{a\in\mathcal{A}}\left\{p\left(a\oplus\boldsymbol{w}|\boldsymbol{w}\right)\log_2 p\left(a\oplus\boldsymbol{w}|\boldsymbol{w}\right)\right\};\\ \mathrm{E}_R(\boldsymbol{w})&=-\sum_{b\in\mathcal{B}}\left\{p\left(\boldsymbol{w}\oplus b|\boldsymbol{w}\right)\log_2 p\left(\boldsymbol{w}\oplus b|\boldsymbol{w}\right)\right\},\end{aligned} \tag{15}$$

Dataset		#Entity										
		GAM	POS	MOV	NAM	ORG	SCE	COM	GOV	BOO	ADD	**All**
Train		1,613	1,748	428	1,316	663	1,748	174	540	1,764	773	10,767
Dev		212	218	55	153	53	168	24	84	184	86	1,237
Test	GAM	226	47	38	34	38	28	1	42	47	37	538
	ENT	1	39	4	17	2	1	1	59	6	9	139
	LOT	-	42	3	23	1	156	-	-	26	11	262
	FIN	-	108	25	132	53	26	12	-	150	50	556
	All	227	236	70	206	94	211	14	101	229	107	1,495

Table 6: Entity statistics of the news dataset. We use the gray scale to represent the proportion of different entities in the test sets of four domains, respectively.

Dataset		#Entity						
		PER	LOC	ORG	TIT	WEA	KUN	**All**
Novel Train		11.8K	2.4K	3.2K	4.1K	2.5K	1.6K	25.5K
Novel Dev		4.8K	0.8K	0.9K	2.4K	1.1K	0.3K	10.3K
Novel Test	天荒神域 (*Stories in Myth*)	1,481	215	454	729	225	60	3.2K
	道破天穹 (*Taoist Stories*)	1,709	231	146	806	412	153	3.5K
	茅山诡术师 (*Maoshan Wizards*)	1,538	333	236	838	421	163	3.5K
	All	4.7K	0.8K	0.8K	2.4K	1.1K	0.4K	10.2K
Financial report Test		0.4K	0.7K	2.9K	-	-	-	4.1K

Table 7: Entity statistics of the novel dataset and the financial report dataset. We use the gray scale to represent the proportion of different entities in four test sets, respectively.

where E_L and E_R represent the left and right entropy, respectively. $\boldsymbol{w}$ represents an N-gram substring. $\mathcal{A}$ and $\mathcal{B}$ are the sets of words that appear to the left or right of $\boldsymbol{w}$, respectively.

Finally, we add the three values MI, E_L and E_R as the validity score of possible new entities, remove the common words based on an open-domain dictionary from Jieba[8], and save the top 50% of the remaining words as the potential input document-specific entity dictionary.

B Details of the Datasets

B.1 News Dataset

Entity statistics. As listed in Table 6, the fine-grained news dataset consists of 10 entity types, including GAM (game), POS (position), MOV (movie), NAM (name), ORG (organization), SCE (scene), COM (company), GOV (government), BOO (book) and ADD (address). The four test domains have obvious different distributions of entity types, which are visualized by the gray scale of color in Table 6.

B.2 Novel Dataset

Data collection. We construct our corpus from a professional Chinese novel reading site named Babel Novel[9]. Unlike news, the novel dataset covers a mixture of literary style including historical novels, and martial arts novels in the genre of fantasy, mystery, romance, military, etc. Therefore, unique characteristics of this dataset such as novel-specific types of named entities present challenges for NER.

Annotation. Considering the literature genre, we annotate three more entity types other than PER (person), LOC (location) and ORG (organization) in MSRA (Levow, 2006), namely (i) TIT (title), which represents the appellation or nickname of a person, such as "冥界之主(Load of Underworld)" and "无极剑圣(Sward Master)"; (ii) WEA (weapon), which represents weapons or objects with special-purpose (e.g. "天龙战戟(Dragon Spear)" and "星辰法杖(Stardust Wand)"); and (iii) KUN (kongfu), which represents the name of martial arts such as "太极(Tai Chi)" and "忍术(Ninjutsu)". The annotation work is undertaken by five undergraduate students and two experts. All of the annotators have read the whole novels before annotation, which aims to prevent the labeling inconsistent problem. In terms of annotation progress, each sentence is first annotated by at least two students, and then the experts select the examples with inconsistent annotations and modify the mistakes. The inter-annotator agreement exceeded a Cohen's kappa value (McHugh, 2012) of 0.915 on the novel dataset.

[8] http://github.com/fxsjy/jieba
[9] https://babelnovel.com/

Methods	GAM	POS	MOV	NAM	ORG	SCE	COM	GOV	BOO	ADD	All
BILSTM	82.00	66.00	77.06	71.88	73.13	66.67	73.83	65.33	55.24	55.5	71.36
BILSTM+ENT	81.70	66.85	70.90	74.35	73.89	71.42	73.14	63.10	55.74	53.26	71.20
LATTICE	82.05	70.73	75.65	78.93	78.12	68.97	78.34	74.75	57.14	61.17	73.96
LATTICE (REENT)	81.48	65.63	72.64	71.49	74.75	53.33	77.97	67.74	56.00	53.20	69.62
ERNIE	81.51	72.35	80.41	83.74	73.50	66.67	78.35	75.90	66.21	54.22	75.73
ERNIE+FUR+ENT	82.47	71.43	81.73	82.87	69.28	48.78	77.52	68.69	70.15	55.79	74.59
BERT	78.85	76.21	79.44	83.20	71.33	64.86	73.57	74.75	61.84	53.23	74.22
BERT+FUR	79.92	73.87	73.63	82.40	73.45	52.63	78.75	72.92	71.64	55.90	75.14
BERT+FUR+ENT	76.77	66.36	74.88	81.70	67.86	57.89	68.55	61.06	58.97	50.00	69.59
Ours	84.30	72.93	80.00	83.10	73.57	61.54	78.04	76.14	72.99	59.13	76.66

Table 8: Fine-grained comparisons on the news dataset.

Entity statistics. The statistics for the above six entity types are listed in Table 7. We can see that the entity distributions on the three test novels are similar with only a few differences, which are because of the differences in the topics of novels.

B.3 Financial Report Dataset

Annotation. The annotation process is similar to that of the novel dataset. The inter-annotator agreement exceeded a Cohen's kappa value (McHugh, 2012) of 0.923 on the financial report dataset.

Entity statistics. The detailed statistics for the financial report dataset are listed in Table 7.

C Fine-grained Comparison

The total results of fine-grained comparisons on the news dataset are listed in Table 8. The news dataset has a total of 10 entity types, including GAM (game), POS (position), MOV (movie), NAM (name), ORG (organization), SCE (scene), COM (company), GOV (government), BOO (book) and ADD (address).