Named Entity Recognition in Multi-level Contexts

Yubo Chen Chuhan Wu Tao Qi Zhigang Yuan Yongfeng Huang
Department of Electronic Engineering & BNRist, Tsinghua University, Beijing 100084, China
{ybch14, wuchuhan15, taoqi.qt}@gmail.com
{zgyuan, yfhuang}@tsinghua.edu.cn

Abstract

Named entity recognition is a critical task in the natural language processing field. Most existing methods for this task can only exploit contextual information within a sentence. However, their performance on recognizing entities in limited or ambiguous sentence-level contexts is usually unsatisfactory. Fortunately, other sentences in the same document can provide supplementary document-level contexts to help recognize these entities. In addition, words themselves contain word-level contextual information since they usually have different preferences of entity type and relative position from named entities. In this paper, we propose a unified framework to incorporate multi-level contexts for named entity recognition. We use TagLM as our basic model to capture sentence-level contexts. To incorporate document-level contexts, we propose to capture interactions between sentences via a multi-head self attention network. To mine word-level contexts, we propose an auxiliary task to predict the type of each word to capture its type preference. We jointly train our model in entity recognition and the auxiliary classification task via multi-task learning. The experimental results on several benchmark datasets validate the effectiveness of our method.

1 Introduction

Named Entity Recognition (NER) is defined as automatically identifying and classifying named entities into specific categories (e.g., person, location, organization) in text. It is a critical task in Natural Language Processing (NLP) and a prerequisite for many downstream tasks, such as entity linking (Luo et al., 2015), relation extraction (Feldman and Rosenfeld, 2006) and question answering (Lee et al., 2006).

NER is usually modeled as a sentence-level sequence labeling task in previous work. For example, Lample et al. (2016) used long-short term memory (LSTM) (Gers et al., 2000) for capturing contextual word representations and conditional random fieid (CRF) (Lafferty et al., 2001) for jointly label decoding. In recent years, language models (LMs) were introduced to this task to learn better contextual representations of words (Peters et al., 2017, 2018; Devlin et al., 2019). However, these methods only consider the contexts within a sentence, which is insufficient.

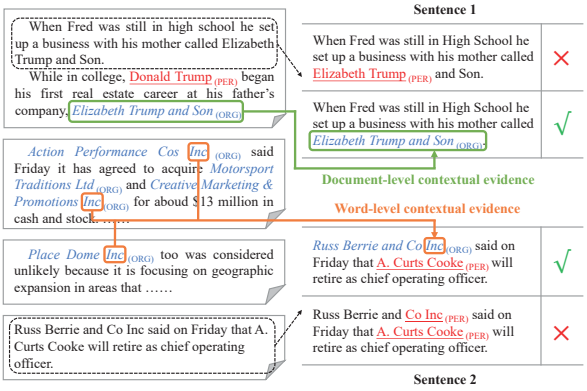

Figure 1: Examples of document- and word-level contextual evidence. Blue italic and red underlined entities are the names of organizations and persons respectively. Green and orange arrows indicate the document- and word-level contextual evidence respectively.

Our work is motivated by the observation that the contextual information beyond sentences can mitigate the negative effects of the ambiguous and limited sentence contexts. The sentences within a document are highly related, and the interactions between them can provide **document-level** contextual information. For example, in Figure 1, sentence 1 is ambiguous because it can be either his mother called *Elizabeth Trump* or a business called *Elizabeth Trump and Son*. But another sentence in this document explicitly mentions *Elizabeth Trump and Son* as a company's name and solves the ambiguity. Besides, words themselves contain prefer-

Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, pages 181–190
December 4 - 7, 2020.

ences of entity type and relative position from the entities, and the preferences provide **word-level** contextual information. For instance, the sentence 2 in Figure 1 has limited contexts, and the word *said* can easily mislead the classification of the type of *Co Inc*. However, the multiple mentions of *Inc* in other sentences indicate its preference to appear as the last word of organizations. Thus, these preferences of words have the potential to help recognize entity types more correctly.

In this paper, we propose a unified framework for NER to incorporate multi-level contexts. Our framework is based on TagLM (Peters et al., 2017), which captures morphological and sentence-level contextual information with two-layer bidirectional gated recurrent units (BiGRUs) (Chung et al., 2014). We apply the neural attention mechanism (Bahdanau et al., 2014) to the hidden states of TagLM's bottom BiGRU to learn sentence representations, and contextualize them with a sentence-level BiGRU. To mine document-level contexts, we propose to apply the multi-head self attention mechanism (Vaswani et al., 2017) to the sentence-level BiGRU's hidden states to capture the relations between sentences. To fuse the document-level context, we combine the output document representations of the self attention module with the corresponding sentence's bottom hidden states and feed them into TagLM's top BiGRU. Besides, to mine word-level contextual information, we propose an auxiliary word classifier to predict the probability distributions of word labels because the label distributions describe the type and position preferences of words. The auxiliary word classification task is jointly trained with our NER model via multi-task learning. We concatenate the top BiGRU's output representations with the output probability vectors of the word classifier to fuse the word-level context and feed them into a CRF for sequence decoding.

The main contributions of this paper are:

- We propose to fuse multi-level contexts for the NER task with a unified framework.
- We propose to exploit the document-level context by capturing the interactions between sentences within a document with the multi-head self attention mechanism.
- We propose to mine the word-level context with an auxiliary word classification task to learn the words' preferences of entity type and relative position from the entities.
- We conduct experiments on several benchmark datasets, and the results validate the effectiveness of our method.

2 Related Work

In traditional NER methods, contexts are usually modeled via hand-crafted features. For example, Passos et al. (2014) trained phrase vectors in their lexicon-infused skip-gram model. Lin and Wu (2009) used a linear chain CRF and added phrase cluster features extracted from the web data. However, these methods require heavy feature engineering, which necessities massive domain knowledge. In addition, these methods cannot make full use of contextual information within texts.

In recent years, many neural networks were applied to the NER task. Collobert et al. (2011) first adopted CNNs to learn word representations. Recently, BiLSTM was widely used for long distance context modeling (Chiu and Nichols, 2016; Lample et al., 2016; Ma and Hovy, 2016). Additionally, Chiu and Nichols (2016) employed CNNs to capture morphological word representations; Lample et al. (2016) utilized CRF to model the dependencies between adjacent tags; Ma and Hovy (2016) proposed LSTM-CNNs-CRF model to combine the strengths of these components. Besides, Strubell et al. (2017) proposed iterated-dilated CNNs for higher efficiency than BiLSTM and better capacity with large context than vanilla CNNs. Recent work proved that the context-sensitive representations captured by language models are useful in NER systems. Peters et al. (2017) proposed TagLM model and introduced LM embeddings in this task. Afterwards, ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) were proposed for better contextual representations. However, these methods focused only on the context within a sentence, so their performance is substantially hurt by the ambiguity and limitation of sentence context.

To combine contexts beyond sentences, several methods were proposed to mine document-level information, such as logical rules (Mikheev et al., 1999), global attention (Xu et al., 2018; Zhang et al., 2018; Hu et al., 2020) and memory mechanisms (Gui et al., 2020). But these methods ignored the sequential characteristics of the sentences within a document, which may be sub-optimal. We observe that contextual associations between sentences in a document have the potential of improving the NER performance. Moreover, the words' preferences of entity type and relative position from the entities

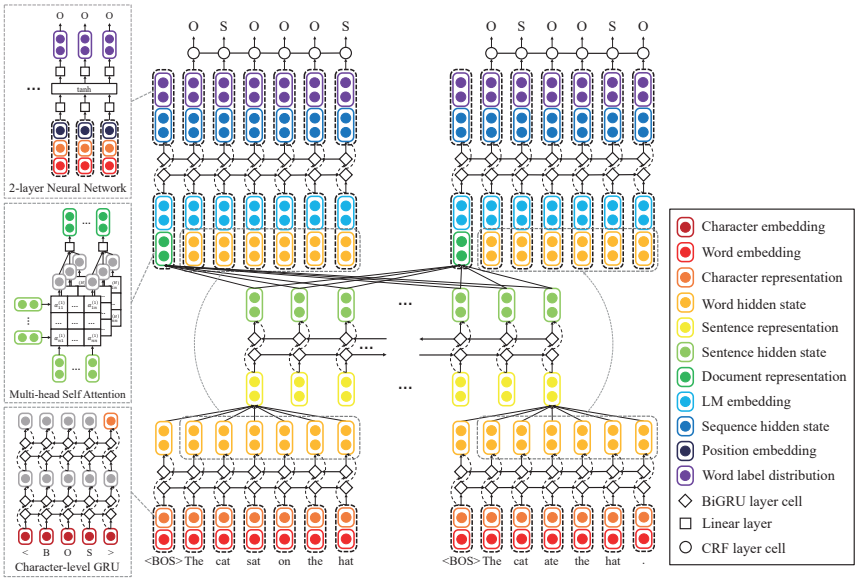

Figure 2: Overview of our multi-level context framework. The character representation is captured with a two-layer BiGRU. The document representation is captured with the multi-head self attention mechanism. The word label distribution is predicted by a two-layer neural network.

contain word-level contextual information, which is ignored by most previous work.

Based on these observations, we propose a unified framework to combine multi-level contexts in this paper. Our framework is based on the TagLM model, which captures sentence-level context with two stacked BiGRUs and models tag dependencies with CRF. To exploit the document-level context, we propose to capture the interactions between sentences within a document with multi-head self attention mechanism (Vaswani et al., 2017). Besides, to mine the word-level context, we propose an auxiliary word classification task to encode the words' type and position preferences. We train our model in the NER and the auxiliary task via multi-task learning. We conduct experiments on several benchmark datasets, and the results demonstrate the effectiveness of multi-level contexts.

3 Our Approach

In this section, we will introduce our approach in detail. The overall framework of our approach is shown in Figure 2. We will first briefly introduce the basic model in our approach, then introduce how to incorporate document- and word-level contexts into our model.

3.1 Baseline NER model

We choose TagLM (Peters et al., 2017) as our basic model. TagLM first captures character-level information of words because named entities usually have specific morphological patterns. For example, *China* refers to the country in most cases, while *china* mostly refers to porcelains. Therefore, given a sentence of words $w_1, w_2, \ldots, w_n$, TagLM learns morphological information with a two-layer BiGRU, as shown in Figure 2. It takes the character embeddings (whose dimension denoted as d_{ce}) as input, and the last output hidden state is adopted as character representation $\mathbf{c}_k$. Then we concatenate $\mathbf{c}_k$ with a word embedding $\mathbf{w}_k$ to construct context-independent representation $\mathbf{x}_k$ for each word:

$$\begin{aligned} \mathbf{c}_k &= \text{BiGRU}(w_k; \theta_c) \in \mathbb{R}^{d_{ch}} \\ \mathbf{w}_k &= E(w_k; \theta_w) \in \mathbb{R}^{d_{we}} \\ \mathbf{x}_k &= [\mathbf{c}_k; \mathbf{w}_k] \in \mathbb{R}^{d_{we}+d_{ch}} \end{aligned} \tag{1}$$

The word embedding $\mathbf{w}_k$ is obtained by looking up a pre-trained embedding matrix θ_w, which is fine-tuned during training (Collobert et al., 2011).

To learn context-sensitive word representations, TagLM applies two layers of BiGRUs on $[\mathbf{x}_{1:n}]$. Then the pre-trained LM embeddings are concatenated with the hidden states of the bottom BiGRU. We denote the output of the bottom and the top BiGRU as $\mathbf{h}_k^{word} \in \mathbb{R}^{d_{sh}}$ and $\mathbf{h}_k^{seq} \in \mathbb{R}^{d_{sqh}}$:

$$\begin{aligned} \mathbf{h}_k^{word} &= \text{BiGRU}(\mathbf{x}_k), \\ \mathbf{h}_k^{seq} &= \text{BiGRU}([\mathbf{h}_k^{word}; \mathbf{LM}_k]). \end{aligned} \tag{2}$$

Finally, we feed $[\mathbf{h}_{1:n}^{seq}]$ into a linear-chain CRF to model the correlations between labels in neighbor-

hoods and jointly decode the best label sequence. The probabilistic model for linear CRF defines a family of conditional probability $p(\boldsymbol{y}|\mathbf{z};\boldsymbol{\theta})$ over all possible label sequences $\boldsymbol{y}$ given $\mathbf{z}$:

$$p(\boldsymbol{y}|\mathbf{z};\boldsymbol{\theta}) = \frac{\prod_{i=1}^{n}\psi_i(y_{i-1},y_i,\mathbf{z})}{\sum_{y'\in\mathcal{Y}(\mathbf{z})}\prod_{i=1}^{n}\psi_i(y'_{i-1},y'_i,\mathbf{z})} \quad (3)$$

where $\psi_i(y',y,\mathbf{z}) = \exp(\mathbf{W}_{y',y}^{\top}\mathbf{z}_i + \mathbf{b}_{y',y})$ are potential functions, and $\mathbf{W}_{y',y}$, $\mathbf{b}_{y',y}$ are parameters of the CRF. Following Lafferty et al. (2001) and Collobert et al. (2011), we utilize the sentence CRF loss for training, which is formulated as the negative log-likelihood:

$$L_{CRF} = -\sum_{i}\log p(\boldsymbol{y}|\mathbf{z};\boldsymbol{\theta}) \quad (4)$$

We compute the likelihood using the forward-backward algorithm at the training phase, and use the Viterbi algorithm to find the most likely label sequence at the test phase.

3.2 Document-level Context

Sentences within a document are highly correlated, and these correlations provide contextual information at the document level. For example, in the document "Jason Little is a rugby union player. Little won 75 caps as captain", the second sentence is ambiguous because it can also mean "Hardly any person won 75 caps as captain". In this case, the first sentence in this document explicitly mentions *Jason Little* as a player. The interaction between the two sentences helps to solve this ambiguity. Therefore, we capture and fuse the document-level context as follows.

To capture the document-level context, we first obtain the context-independent sentence representations. Since each word in a sentence has different importance (e.g. *a* contributes less information than *player* in "Jason Little is a rugby union player.") , we apply the neural attention mechanism (Bahdanau et al., 2014) to filter the uninformative words and learn better sentence representations. Then we contextualize these representations with a sentence-level BiGRU. Formally,

$$\begin{aligned}\alpha_k &= \text{softmax}(\mathbf{u}_w^{\top}\cdot\tanh(\mathbf{W}_a\mathbf{h}_k^{word}+\mathbf{b}_a))\\ \mathbf{s}_i &= \sum_{k=1}^{n}\alpha_k\mathbf{h}_{ik}^{word}\\ \mathbf{h}_i^{sen} &= \text{BiGRU}(\mathbf{s}_i)\end{aligned} \quad (5)$$

where $\mathbf{W}_a \in \mathbb{R}^{d_{na}\times d_{wh}}, \mathbf{b}_a \in \mathbb{R}^{d_{na}}, \mathbf{u}_w \in \mathbb{R}^{d_{na}}$ are the parameters of the neural attention module.

Next, we propose to capture the interactions between sentences with the multi-head self attention mechanism (Vaswani et al., 2017). In most existing attention mechanisms, a sentence's attention weight is only based on its representation, and the relationships between sentences cannot be modeled. Self attention is an effective way to capture the interactions between sentences. Besides, a sentence may interact with multiple sentences. For example, in the document "LeBron James is a basketball player for the Lakers. In 2016 James won the championship of NBA. In 2018 he signed with the Lakers", the first sentence interacts with the remaining two sentences simultaneously because they jointly mention *James* and *Lakers* respectively. Thus, we propose to apply the multi-head self attention mechanism to learn better representations of sentences by modeling their relationship with multiple sentences. We first project the sentence hidden states into the h-th sub-space, and calculate the attention weights in this sub-space:

$$\begin{aligned}&[Q_j^{(h)};K_j^{(h)};V_j^{(h)}] = [W_Q^{(h)};W_K^{(h)};W_V^{(h)}]\mathbf{h}_j^{sen}\\ &z_{ij}^{(h)} = {Q_i^{(h)}}^{\top}K_j^{(h)},\quad \beta_{ij}^{(h)} = \frac{\exp\left(z_{ij}^{(h)}\right)}{\sum_j\exp\left(z_{ij}^{(h)}\right)}\end{aligned} \quad (6)$$

Then we calculate the sub-representation $\mathbf{y}_i^{(h)}$ for the i-th sentence by weighted summing the $V_j^{(h)}$. Finally, these sub-representations are concatenated and projected, resulting in the final representation $\mathbf{d}_i$ for the i-th sentence. We denote the number of heads as H and the sub-space dimension of each head as d_{sa}, then we have:

$$\begin{aligned}\mathbf{y}_i^{(h)} &= \sum_j\beta_{ij}^{(h)}V_j^{(h)}\\ \mathbf{d}_i &= W_O[\mathbf{y}_i^{(1)};\ldots;\mathbf{y}_i^{(h)};\ldots;\mathbf{y}_i^{(H)}]\end{aligned} \quad (7)$$

where $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{d_{sa}\times d_{sh}}, W_O \in \mathbb{R}^{d_{sh}\times Hd_{sa}}$ are projection matrices. $\mathbf{d}_i$ combines representations of all sentences within this document, thus is regarded as the document representation for the i-th sentence.

To fuse the document-level context, we first add a special token <BOS> (denoted as w_{i0}) at the beginning of the sentence $w_{i1},\ldots,w_{in}$, and feed the sentence into TagLM's bottom BiGRU to compute $[\mathbf{h}_{i0}^{word},\mathbf{h}_{i1}^{word},\ldots,\mathbf{h}_{in}^{word}]$. Next we compute the document representation $\mathbf{d}_i$ and replace $\mathbf{h}_{i0}^{word}$ with it (requires $d_{wh} = d_{sh}$). Then we feed them into

the top BiGRU. The input of the top BiGRU contains document- and sentence-level contextual representations simultaneously. Thus its output hidden states act as the fusion of the two contexts.

3.3 Word-level Context

In natural language, words themselves have different preferences on different entity types and relative positions from the entities. These preferences provide word-level contextual information for the NER task. For example, in the sentence "With only one match before New Year, Real will spend Christmas ahead of others", the type of the entity *Real* is uncertain because the context of the sentence is inadequate. However, *Real* prefers to appear as the first word of organizations (e.g. *Real Madrid*, *Real Betis* are football clubs). This preference helps to ensure the entity type of *Real*. Thus we learn and incorporate the word-level context as follows.

To learn the word-level context, we encode the preferences with the probability distributions of word labels, because the label of a word indicates its entity type and relative position from the entities (e.g., *B-ORG* means the first word of an organization). To learn the distributions automatically, we propose an auxiliary word classification task and employ a two-layer neural network as the classifier. The classifier's input consists of the morphological representation $\mathbf{c}_k$ and the word embedding $\mathbf{w}_k$. Besides, we add a position embedding $\mathbf{p}_k$ to represent the relative position information:

$$\begin{aligned} \mathbf{p}_k &= E(k;\theta_p) \in \mathbb{R}^{d_{pe}} \\ \mathbf{x}'_k &= [\mathbf{c}_k;\mathbf{w}_k;\mathbf{p}_k] \in \mathbb{R}^{d_{we}+d_{ch}+d_{pe}} \end{aligned} \tag{8}$$

where $\mathbf{p}_k$ is obtained by looking up a randomly-initialized embedding matrix and tuned during training. Then $\mathbf{x}'_k$ is fed into the two-layer classifier to predict label distribution:

$$\begin{aligned} \mathbf{m}_k &= \tanh(\mathbf{W}_{c_1}\mathbf{x}'_k + \mathbf{b}_{c_1}) \\ \mathbf{p}_k^{label} &= \text{softmax}(\mathbf{W}_{c_2}\mathbf{m}_k + \mathbf{b}_{c_2}) \end{aligned} \tag{9}$$

where $\mathbf{W}_{c_1} \in \mathbb{R}^{d_{lch}\times(d_{we}+d_{ch}+d_{pe})}$, $\mathbf{b}_{c_1} \in \mathbb{R}^{d_{lch}}$, $\mathbf{W}_{c_2} \in \mathbb{R}^{|C|\times d_{lch}}$, $\mathbf{b}_{c_2} \in \mathbb{R}^{|C|}$ are the parameters of the classifier (the number of all labels denoted as $|C|$). During training, we use $\mathbf{p}_k^{label}$ to compute the loss function for word classification, which is formulated as cross-entropy loss:

$$L_{WC}(\theta) = -\sum_{k=1}^{n} \log p_k^{label}(y_k|\theta). \tag{10}$$

To incorporate the word-level context, we concatenate $\mathbf{p}_{ik}^{label}$ with the original CRF input $\mathbf{h}_{ik}^{seq}$ to enrich word representations with the label distributions (Seyler et al., 2018). The CRF takes the enhanced word representations as input and decodes the best label sequence. Our framework is jointly trained on the original NER and the auxiliary classification task via multi-task learning:

$$L(\theta) = L_{CRF}(\theta) + \lambda L_{WC}(\theta), \tag{11}$$

where λ is the weight of word classification loss.

4 Experiments

4.1 Datasets and Evaluation Metrics

We evaluate our approach on the CoNLL-2002, CoNLL-2003, and Wikigold NER datasets. The Wikigold dataset contains annotations for English (denoted as **WIKI**). The CoNLL-2002 dataset contains annotations for Dutch (denoted as **NLD**)[1]. The CoNLL-2003 dataset contains annotations for English and German (denoted as **ENG** and **DEU** respectively). All datasets are manually tagged with four different entity types (*LOC*, *PER*, *ORG*, *MISC*). The CoNLL datasets have standard train, development, and test sets. Since the Wikigold dataset doesn't have standard separation, we randomly split the data into the three sets and perform all experiments on the same separation. Table 1 shows the number of documents and sentences of the datasets. We report the official micro-averaged F_1 scores on all the datasets.

Dataset	Train	Dev.	Test
WIKI	101 (1,227)	22 (402)	22 (212)
NLD	287 (16,093)	74 (2,969)	119 (5,314)
DEU	533 (12,705)	201 (3,068)	155 (3,160)
ENG	946 (14,987)	216 (3,466)	231 (3,684)

Table 1: Numbers of documents (and sentences) in datasets statistics.

4.2 Experimental Settings

In our experiments, we use the BIOES labeling scheme for output tags, which was proven to outperform other options in previous work (Ratinov and Roth, 2009). Under this tagging scheme, the number of labels $|C| = 17$ ([*B,I,E,S*]×

[1]The CoNLL-2002 dataset contains Dutch and Spanish data. But the Spanish data lacks the marks of doucument boundaries. Thus we only conduct experiments on the Dutch data.

Hyper-parameter	Value
Word **e**mbedding dim. (d_{we})	50/300
Character **e**mbedding dim. (d_{ce})	25
Position **e**mbedding dim. (d_{pe})	30
Character **h**idden state dim. (d_{ch})	80
Word **h**idden state dim. (d_{wh})	300
Sentence **h**idden state dim. (d_{sh})	300
Sequence **h**idden state dim. (d_{sqh})	300
Neural **a**ttention subspace dim. (d_{na})	100
Self **a**ttention subspace dim. (d_{sa})	60
Label **c**lassifier **h**idden dim. (d_{lch})	64
Number of heads (H)	5
Weight of L_{WC} (λ)	0.1

Table 2: Hyper-parameters of our model.

[*LOC,PER,ORG,MISC*] + *O*). For English datasets, we use the 50-dimensional Senna word embeddings (Collobert et al., 2011) and pre-process the text by lower-casing the words and replacing all digits with 0 (Chiu and Nichols, 2016; Peters et al., 2017). For Dutch and German datasets, we use the pre-trained 300-dimensional word2vec embeddings (Mikolov et al., 2013), which are trained on the Wikipedia dumps[2]. We adopt ELMo (Peters et al., 2018; Che et al., 2018) as the pre-trained LM embeddings[3]. The hyper-parameters of our model are shown in Table 2. For regularization, we add 25% dropout (Srivastava et al., 2014) to the input of all BiGRUs, but not to the recurrent connections.

Following Peters et al. (2017), we use the Adam optimizer (Kingma and Ba, 2014) with gradient norms clipped at 5.0. We fine-tune the pre-trained word embeddings and ELMo model parameters. We train our model with a constant learning rate of $\gamma = 0.001$ for 20 epochs. Then we start a simple learning rate decay schedule: divide γ by ten, train for 5 epochs, divide γ by ten, train for 5 epochs again and stop. We train the model's parameters on the train set and tune the hyper-parameters on the development set. Then we compute F_1 score on the test set at the epoch with the highest development performance. Following previous work (Chiu and Nichols, 2016; Peters et al., 2017), we train our model for multiple times with different random seeds and report the mean of F_1.

4.3 Performance Evaluation

To demonstrate the effectiveness of our method, we compare our experimental results on the CoNLL-2002 and CoNLL-2003 datasets with previously published state-of-the-art models: Ando and Zhang (2005) proposed a structural learning algorithm for semi-supervised NER; Qi et al. (2009) proposed Word-Class Distribution Learning (WCDL) method; Nothman et al. (2013) introduced Wikipedia articles as extra knowledge; Gillick et al. (2015) proposed a byte-level model for multilingual NER; Lample et al. (2016) proposed BiLSTM-CRF model; Yang et al. (2017) applied transfer learning mechanism for NER; Peters et al. (2018) proposed ELMo embeddings; Clark et al. (2018) proposed Cross-View Training (CVT) method; Devlin et al. (2019) proposed BERT representations; Liu et al. (2019) introduced external gazetters to this task; Akbik et al. (2018) proposed contextual character language model and achieved the state-of-the-art performance; Zhang et al. (2018) and Hu et al. (2020) utilized global attention to mine document-level information; Gui et al. (2020) used memory mechanism to capture document-level label consistency. Table 3 shows the comparison results, from which we can observe that the incorporation of multi-level contexts brings 0.47%,

Method	ENG	DEU	NLD
Ando et al. (2005)	89.31	75.27	–
Qi et al. (2009)	88.69	75.72	–
Nothman et al. (2013)	85.2	66.5	78.6
Gillick et al. (2015)	86.50	76.22	82.84
Lample et al. (2016)	90.94	78.76	81.74
Yang et al. (2017)	91.26	–	85.19
Peters et al. (2018)	92.22	–	–
Clark et al. (2018)	92.6	–	–
Akbik et al. (2018)	**93.09**	**88.32**	–
Devlin et al. (2019)	92.8	–	–
Liu et al. (2019)	92.75	–	–
Zhang et al. (2018)	91.81	79.21	87.40
Hu et al. (2020)	91.92	–	–
Gui et al. (2020)	93.05	–	–
TagLM (Peters et al., 2017)	91.93	–	–
TagLM+ELMo (baseline)	92.21†	77.83†	88.05†
Our model	**92.68***	**78.87***	**88.93***

Table 3: Comparison results of F_1 score on the CoNLL-2002 and CoNLL-2003 test sets. $\dagger$ denotes the results of our implementation. $*$ denotes statistically significant improvements over the baseline model with $p < 0.01$ under a t-test.

[2]https://github.com/Kyubyong/wordvectors

[3]We also conduct experiments with TagLM+BERT$_{\text{BASE}}$ with released parameters. Due to the limitation of GPU memory, we didn't fine-tune BERT. The dev and test set F_1 scores are **95.03±0.22** and **91.64±0.18** respectively. Our results have a surprisingly huge gap between the reported scores (we refer readers to Section 4.3 and 5.4 of Devlin et al. (2019)).

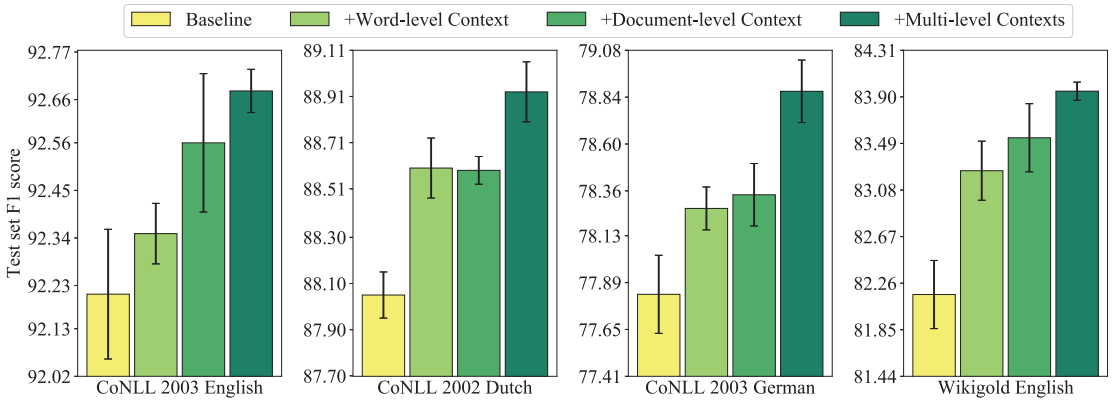

Figure 3: An ablation study of our framework. We compare the mean of test set F_1 score under the four settings on the four datasets. The bars indicate the standard deviation of F_1 score.

1.04%, and 0.88% absolute F_1 score improvement on the English, German and Dutch dataset respectively compared with our baseline model. In addition, our model outperforms most of the previous sentence- and document-level methods on the three languages. The improvements demonstrate the effectiveness of our framework, which fully exploits the document and word-level contexts and combines the multi-level contexts. With the assistance of multi-level contexts, our model can capture more contextual information beyond sentences and recognize entities more correctly.

4.4 Ablation Study

To study the contribution of the document- and word-level context respectively, we conduct experiments on two settings: only incorporating the word-level context and the document-level context, and compare the F_1 score with our model. Figure 3 shows the results, from which we have the following observations: (1) The document- and word-level contexts both bring improvements on the four datasets. It indicates the utility of these contexts respectively. The document-level context contains interactions between sentences within a document. The word-level context contains words' type and position preferences. Either of the contexts can help alleviate the effects of limited or ambiguous sentence context. (2) The multi-level contexts method improves the F_1 score over the other two settings on all the datasets. It validates the effectiveness of the fusion of multi-level contexts. Our framework can exploit and fuse the contexts at the document and word level simultaneously. With the assistance of more extra contextual information from the document and word level, our method performs better than the other two settings of combining only one context.

4.5 Analysis

4.5.1 How to fuse the document-level context?

In this experiment, we propose four alternative ways to fuse document-level contextual representation $\mathbf{d}_i$ with sentence-level contextual representations $\mathbf{h}_i^{word}$ or $\mathbf{h}_i^{seq}$ (Equation 2):

- *Concatenate* $\mathbf{h}_{ik}^{word}$ with $\mathbf{d}_i$;
- *Add* $\mathbf{h}_{ik}^{word}$ to $\mathbf{d}_i$;
- *Initialize* $\mathbf{h}_{i(-1)}^{seq}$ with $\mathbf{d}_i$;
- *Replace* $\mathbf{h}_{i0}^{word}$ with $\mathbf{d}_i$.

Table 4 shows the comparison result on the CoNLL-2003 English test set. The first two options essentially translate $\mathbf{h}_{ik}^{word}$ in the vector space, because they enhance $\mathbf{h}_{ik}^{word}$ with the same $\mathbf{d}_i$ for all words. Therefore they cannot fully combine the contexts. To distinguish between the latter two options, we need to focus on the internal calculation of GRU: $\mathbf{h}_t = (1 - z_t)\mathbf{n}_t + z_t\mathbf{h}_{t-1}, \mathbf{n}_t = \tanh(\mathbf{W}_{in}\mathbf{x}_t + \mathbf{b}_{in} + r_t(\mathbf{W}_{hn}\mathbf{h}_{t-1} + \mathbf{b}_{hn}))$. GRU uses non-linearly transformed $\mathbf{x}_t$ and raw $\mathbf{h}_t$ to calculate hidden states. We speculate that the non-linear transformation on $\mathbf{d}_i$ aligns it to the same space as $\mathbf{h}_{ik}^{word}$ and produces better performance.

4.5.2 How to fuse the word-level context?

In this experiment, we compare three ways of fusing word-level contextual representations $\mathbf{p}_i^{label}$ with the sentence-level context:

- Concatenate the input $\mathbf{x}_k$ with $\mathbf{p}_{ik}^{label}$;
- Concatenate $\mathbf{h}_{ik}^{word}$ with $\mathbf{p}_{ik}^{label}$;
- Concatenate $\mathbf{h}_{ik}^{seq}$ with $\mathbf{p}_{ik}^{label}$.

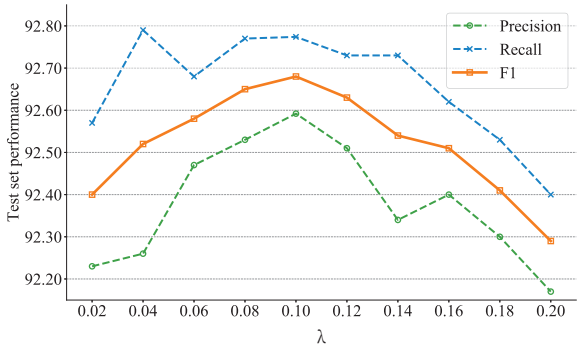

Figure 4: The CoNLL 2003 English test set performance of our model with different λ.

Table 5 shows the comparison results. The first two options use BiGRU to encode the label distributions but perform worse than the last one using CRF. We speculate that CRF is more suitable to encode the distributions of word label than BiGRU because there exist strong connections between two adjacent words' label distributions intuitively.

4.5.3 Which attention mechanism to use at document level?

In this part, we compare three choices of attention mechanism: the *multi-head self attention*, *self attention*, and the most-popular *neural attention* mechanism. Table 6 shows the comparison results. We can observe that the self attention mechanism outperforms neural attention because it can capture interactions between sentences in the document. In contrast, the neural attention mechanism only learns the sentence's weight based on its representation, thus fails to capture the interactions. Furthermore, multi-head self attention performs better than self attention because it can capture a sentence's interactions with multiple sentences.

4.5.4 How to choose the weight λ of the auxiliary task ?

We conduct experiments on different weights λ to investigate its influence and illustrate the result in Figure 4. We speculate that λ controls the proportion of the word-level context in all contexts. When λ changes, the balance of the contexts is broken, and the performance is affected. Besides, λ controls the learning rate of the word label classifier's parameters. Its increase and decrease will hurt the accuracy of the label classification.

Document-level fusion method	$F_1 \pm$ **std**
Concatenate $\mathbf{h}_{ik}^{word}$ with $\mathbf{d}_i$	92.36±0.08
Add $\mathbf{h}_{ik}^{word}$ to $\mathbf{d}_i$	92.43±0.05
Initialize $\mathbf{h}_{i(-1)}^{seq}$ with $\mathbf{d}_i$	92.42±0.10
Replace $\mathbf{h}_{i0}^{word}$ with $\mathbf{d}_i$	**92.68±0.09**

Table 4: Comparison of different ways of fusing the document-level context on CoNLL 2003 test set.

Fusion method	$F_1 \pm$ **std**
Concatenate $\mathbf{p}_{ik}^{label}$ with $\mathbf{x}_k$	91.99±0.14
Concatenate $\mathbf{p}_{ik}^{label}$ with $\mathbf{h}_{ik}^{word}$	92.33±0.11
Concatenate $\mathbf{p}_{ik}^{label}$ with $\mathbf{h}_{ik}^{seq}$	**92.68±0.09**

Table 5: Comparison of different ways of fusing the word-level context on CoNLL 2003 test set.

Attention mechanism	$F_1 \pm$ **std**
Neural attention	92.49±0.10
Self attention	92.52±0.09
Multi-head self attention	**92.68±0.09**

Table 6: Comparison of different attention mechanisms at document level on CoNLL 2003 test set.

Case #1	Label	*LITTLE* TO MISS CAMPESE FAREWELL
	TagLM	*LITTLE* TO MISS CAMPESE FAREWELL
	Ours	*LITTLE* TO MISS CAMPESE FAREWELL
	D-lvl	Centre Jason **Little** will miss ...
Case #2	Label	... play at the *Melbourne Cricket Ground*.
	TagLM	... play at the *Melbourne Cricket Ground*.
	Ours	... play at the *Melbourne Cricket Ground*.
	W-lvl	1. ... the Sydney Cricket **Ground** ... 2. ... the Melbourne Cricket **Ground** ...

Table 7: Comparison between the baseline and our method on two cases. Blue, red and orange entities indicate the names of organizations, persons and locations. The bold words are word-level (W-lvl) or document-level (D-lvl) supporting contextual evidence.

4.6 Case Study

Table 7 shows the comparison of the baseline and our model on two example sentences. In the first case, the ambiguity of *LITTLE* disturbs the baseline model. Our model finds another explicit mention *Jason Little* as a person (centre) in this document and correctly identifies this entity. In the second case, the *Melbourne Cricket Ground* (location) is wrongly classified as organization, because one can either *play* at a team or *play* at a stadium. Our model notices the two other mentions of *Ground*, both of which appears as the last word of location, and corrects the erroneous entity type. The examples prove that our model can mine contextual information outside sentences and recognize

entities more correctly than the baseline model.

5 Conclusion

In this paper, we propose a unified structure to incorporate multi-level contexts for the NER task. We use TagLM as our baseline model to capture the sentence-level context. To incorporate the document-level context, we propose to learn relationships between sentences within a document with the multi-head self attention mechanism. Besides, to mine word-level contextual information, we propose an auxiliary task to predict the word type to capture its type preferences. Our model is jointly trained on the NER and auxiliary tasks through multi-task learning. We evaluate our model on several benchmark datasets, and the experimental results prove the effectiveness of our method.

Acknowledgement

This work was supported by the National Key Research and Development Program of China under Grant number 2018YFB2101501, the National Natural Science Foundation of China under Grant numbers U1936208, U1936216.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *JMLR*, 6(Nov):1817–1853.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. In *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 55–64.

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

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*.

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

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

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

Ronen Feldman and Benjamin Rosenfeld. 2006. Boosting unsupervised relation extraction by using ner. In *EMNLP*, pages 473–481.

Felix A Gers, Jürgen Schmidhuber, and Fred Cummins. 2000. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471.

Dan Gillick, Cliff Brunk, Oriol Vinyals, and Amarnag Subramanya. 2015. Multilingual language processing from bytes. *arXiv preprint arXiv:1512.00103*.

Tao Gui, Jiacheng Ye, Qi Zhang, Yaqian Zhou, Yeyun Gong, and Xuanjing Huang. 2020. Leveraging document-level label consistency for named entity recognition. In *IJCAI*, pages 3976–3982.

Anwen Hu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. 2020. Leveraging multi-token entities in document-level named entity recognition. In *AAAI*, pages 7961–7968.

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

John D Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, pages 282–289.

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

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*, pages 581–587. Springer.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *ACL-AFNLP*, pages 1030–1038. Association for Computational Linguistics.

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

Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. 2015. Joint entity recognition and disambiguation. In *EMNLP*, pages 879–888.

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

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *EACL*, pages 1–8.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from wikipedia. *Artificial Intelligence*, 194:151–175.

Alexandre Passos, Vineet Kumar, and Andrew McCallum. 2014. Lexicon infused phrase embeddings for named entity resolution. In *CoNLL*, pages 78–86.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL-HLT*, volume 1, pages 2227–2237.

Yanjun Qi, Ronan Collobert, Pavel Kuksa, Koray Kavukcuoglu, and Jason Weston. 2009. Combining labeled and unlabeled data with word-class distribution learning. In *CIKM*, pages 1737–1740. ACM.

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

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *ACL*, volume 2, pages 241–246.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*, pages 2670–2680.

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

Guohai Xu, Chengyu Wang, and Xiaofeng He. 2018. Improving clinical named entity recognition with global neural attention. In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pages 264–279. Springer.

Zhilin Yang, Ruslan Salakhutdinov, and William W Cohen. 2017. Transfer learning for sequence tagging with hierarchical recurrent networks. *arXiv preprint arXiv:1703.06345*.

Boliang Zhang, Spencer Whitehead, Lifu Huang, and Heng Ji. 2018. Global attention for name tagging. In *CoNLL*, pages 86–96.