Leveraging Document-Level Label Consistency for Named Entity Recognition

Tao Gui[1*], **Jiacheng Ye**[1*], **Qi Zhang**[1], **Yaqian Zhou**[1], **Yeyun Gong**[2] and **Xuanjing Huang**[1]

[1]School of Computer Science, Fudan University, Shanghai, China

[2]Microsoft Research Asia

{tgui16, yejc19, qz, zhouyaqian, xjhuang}@fudan.edu.cn, yegong@microsoft.com

Abstract

Document-level label consistency is an effective indicator that different occurrences of a particular token sequence are very likely to have the same entity types. Previous work focused on better context representations and used the CRF for label decoding. However, CRF-based methods are inadequate for modeling document-level label consistency. This work introduces a novel two-stage label refinement approach to handle document-level label consistency, where a key-value memory network is first used to record draft labels predicted by the base model, and then a multi-channel Transformer makes refinements on these draft predictions based on the explicit co-occurrence relationship derived from the memory network. In addition, in order to mitigate the side effects of incorrect draft labels, Bayesian neural networks are used to indicate the labels with a high probability of being wrong, which can greatly assist in preventing the incorrect refinement of correct draft labels. The experimental results on three named entity recognition benchmarks demonstrated that the proposed method significantly outperformed the state-of-the-art methods.

1 Introduction

The task of named entity recognition (NER) involves determining entity boundaries and recognizing the categories of named entities, which is a fundamental task in the field of natural language processing (NLP). Because most neural NER models are implemented using bi-directional long short-term memory (BiLSTM) networks [Ma and Hovy, 2016; Lample *et al.*, 2016; Peters *et al.*, 2018], they have a limited ability to exploiting non-local and non-sequential dependencies such as co-references and identical mentions [Qian *et al.*, 2019].

To tackle the above challenges, several variations of LSTM have been proposed to incorporate sentence-level context information [Zhang *et al.*, 2018c; Liu *et al.*, 2019]. Recently, many of the existing methods have focused on the better incorporation of document-level context information [Hu *et al.*, 2019; Qian *et al.*, 2019], because different occurrences of a particular token sequence are very likely to have the same entity types within a document [Krishnan and Manning, 2006]. In particular, Luo *et al.* [2020] simultaneously utilized sentence-level and document-level representations. This hierarchical contextualized representation architecture enhanced NER with better global information modeling.

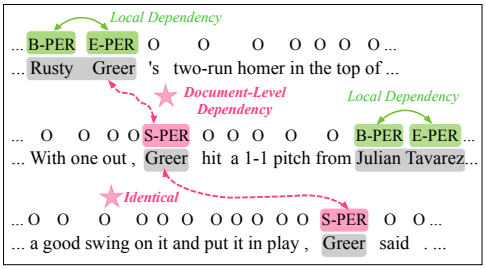

Figure 1: Example of label dependencies in NER task. Local label dependencies (green, solid) can be learned by CRF-based methods. Although much recent work has focused on document-level context representations, the CRF decoding methods were still limited to learning document-level label dependencies (pink, dashed).

However, the focus of the previous work was only on document-level context representations, with little work done to explicitly model the document-level label consistency between the same token sequences. As shown in Figure 1, we argue that using Conditional Random Fields (CRF) is not competent for modeling the document-level relationships of labels [Qian *et al.*, 2019; Luo *et al.*, 2020]. Recently, many methods have introduced label embeddings to manage longer ranges of dependencies [Zhang *et al.*, 2018b; Cui and Zhang, 2019]. However, these methods are still trapped in the sentence-level label dependencies.

This work handles document-level label consistency by introducing a novel two-stage label refinement networks that are stacked using the BiLSTM and multi-channel Transformer. In a document, all of the draft labels predicted by the BiLSTM networks are fed as inputs to the next layer (Transformer), so that document-level label dependencies can be considered. Specifically, at the first stage, the BiLSTM networks take a document as input, and yield all of the hidden state vectors, together with draft labels sentence by

[*]Both authors contributed equally

sentence. A key-value memory component [Miller *et al.*, 2016] is adopted, which memorizes all the hidden state vectors and their corresponding label embeddings of the entire document. At the second stage, a multi-channel Transformer performs attention over document-level label embeddings derived from the memory network to explicitly model the co-occurrence relationship, as well as sentence-level label embeddings to model the local label dependencies, and hidden state vectors to model the context representations. All of these features are fused to refine the draft labels in parallel, which can avoid the use of Viterbi decoding of the CRF for a faster prediction. In addition, in order to mitigate the side effects of incorrect draft labels, Bayesian neural networks [Gal and Ghahramani, 2016b] are used to indicate the draft labels with a high probability of being wrong, which can greatly assist in preventing the incorrect refinement of correct draft labels. The experimental results on three named entity recognition benchmarks demonstrated that the proposed method significantly outperformed the previous state-of-the-art methods.

The main contributions of this paper can be summarized as follows: 1) novel two-stage label refinement networks are proposed, which can better model document-level label dependencies; 2) the proposed method can model label dependencies in parallel, which performs up to 5.48 times faster than state-of-the-art methods during the inference phase; 3) the use of Bayesian neural networks is proposed to estimate the uncertainty of the predictions and indicate potentially incorrect labels that should be refined; and 4) the experimental results across three NER datasets indicate that the proposed method significantly outperforms the start-of-the-art methods. Our codes are released at *Github*[1].

2 Related Work

2.1 Neural Named Entity Recognition

In recent years, many neural network-based methods have achieved competitive performances without massive hand-crafted feature engineering, including LSTM-based methods because of their advantages in modeling sequence data [Lample *et al.*, 2016; Ma and Hovy, 2016] and CNN-based methods because of their proficiency in parallel modeling [Strubell *et al.*, 2017]. In order to model non-local and non-sequential dependencies, sentence-level [Zhang *et al.*, 2018c; Liu *et al.*, 2019] and document-level contextualized information [Qian *et al.*, 2019; Luo *et al.*, 2020] has been adopted to eliminate the limitations of RNNs resulting from their sequential nature. In contrast to their work, which only modeled the dependencies between words, the method reported here also models the sentence-level and document-level dependencies between labels.

2.2 Label Dependency Modeling

Creating better models for label dependencies has always been the focus of sequence labeling tasks [Ye and Ling, 2018; Zhang *et al.*, 2018b]. In particular, the CRF layer is integrated with neural encoders to capture label transition patterns [Ma and Hovy, 2016]. Many of the recent methods have introduced label embeddings to manage longer ranges of dependencies [Zhang *et al.*, 2018b; Cui and Zhang, 2019]. However, these methods are still trapped in the sentence-level label dependencies. Inspired by Krishnan and Manning [2006], they proposed a two-stage approach for document-level label consistency, but it required slower two-layer CRFs and hand-crafted features. In contrast, the method reported here can model both the sentence-level label dependencies and document-label consistencies in parallel without hand-crafted features.

2.3 Bayesian Neural Networks

There are two main types of uncertainty in Bayesian modeling [Kendall and Gal, 2017]. Aleatoric uncertainty captures the noise inherent in the observations, and epistemic (model) uncertainty accounts for the uncertainty in the model parameters. This work focused on the use of epistemic uncertainty to indicate whether model predictions were likely to be incorrect, which could effectively prevent correct draft labels from being incorrectly refined.

Given the dataset $\mathcal{D}$ with training inputs $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ and their corresponding outputs $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$, Bayesian inference looks for the posterior distribution of the parameters given the dataset $p(\mathbf{W}|\mathcal{D})$. This makes it possible to predict an output for a new input point $\mathbf{x}^*$ by marginalizing over all of the possible parameters, as follows:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) = \int p(\mathbf{y}^*|\mathbf{W}, \mathbf{x}^*) p(\mathbf{W}|\mathcal{D}) \mathtt{d}\mathbf{W}. \quad (1)$$

Bayesian inference is intractable for many models because of the complex nonlinear structures and high dimension of the model parameters. Recent advances in variational inference introduced new techniques into the field. Among these, Monte Carlo dropout [Gal and Ghahramani, 2016a] requires minimum modification to the original model. It is possible to use the variational inference approach to find an approximation $q_\theta(\mathbf{W}^*)$ to the true posterior $p(\mathbf{W}|\mathcal{D})$ parameterized by a different set of weights θ, where the Kullback-Leibler (KL) divergence of the two distributions is minimized. The integral can be approximated as follows:

$$p(\mathbf{y}^*|\mathbf{x}^*, \mathcal{D}) \approx \sum_{j=1}^{T} p(\mathbf{y}^*|\mathbf{W}_j^*, \mathbf{x}^*) q_\theta(\mathbf{W}_j^*). \quad (2)$$

In contrast to non-Bayesian networks, at test time, dropouts are also activated. As a result, model uncertainty can be approximately evaluated by summarizing the variance of the model outputs from multiple forward passes.

3 Document-Level Label Consistency NER

This work proposes a two-stage label refinement framework for document-level label consistency NER (DocL-NER), which first uses a key-value memory network to record the draft labels predicted by the base model, and then applies a multi-channel Transformer to makes refinements on the draft predictions based on the explicit co-occurrence relationship

[1]https://github.com/jiacheng-ye/DocL-NER

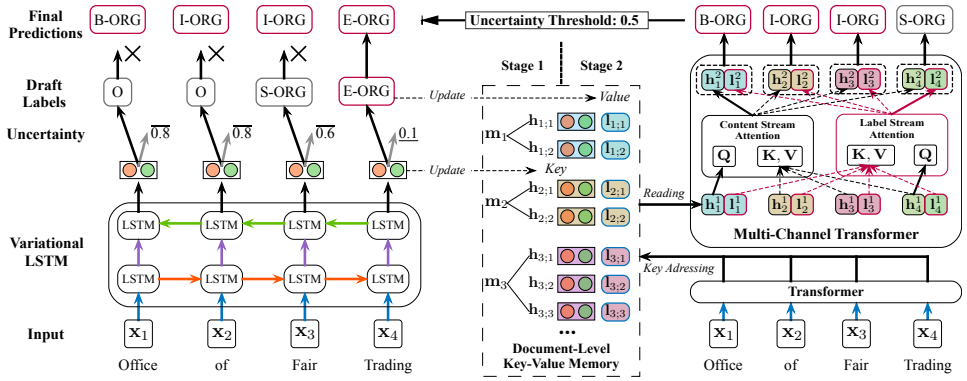

Figure 2: Architecture of DocL-NER. The refinement only works on draft labels with uncertainty greater than the threshold. For example, the threshold is set to 0.5 in the figure, and the final predictions are those labels in red blocks.

derived from the memory network. In order to make a fair comparison with the general LSTM-based methods and to mitigate the side effects of incorrectly refining correct draft labels, in this work, variational LSTMs as special Bayesian neural networks are used to encode sentences and determine the labels with a high probability of being wrong. The proposed model is shown in Figure 2.

3.1 Key-Value Memory for Draft Label Recording

At the first stage, a variational LSTM (VLSTM) [Gal and Ghahramani, 2016b] is adopted as the base model for the draft label and uncertainty predictions. To store document-level information in preparation for subsequent label consistency modeling, a key-value memory network is used to record all of the hidden state vectors and corresponding label embeddings in a document.

Word and Label Representation Following Lample *et al.* [2016], character information was used to enhance the word representation. Given a sequence of words $s = \{w_1, w_2, \ldots, w_n\}$, the product of the one-hot encoded vector with an embedding matrix then gives a word embedding: $\mathbf{w}_i = \mathbf{e}^w(w_i)$, where $\mathbf{e}^w$ denotes a word embedding lookup table. Each word is made up of a sequence of characters, and CNNs are adopted for character-level encoding $\mathbf{c}_i$. Then, a word is represented by concatenating its word embedding and character-level encoding: $\mathbf{x}_i = [\mathbf{w}_i; \mathbf{c}_i]$. All of the word representations make up an embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d_w}$, where d_w is the embedding dimensionality of $\mathbf{x}$, and V is the number of words in the vocabulary.

Given the label set $L = \{l_1, \ldots, l_N\}$, each label l_k is represented by $\mathbf{l}_k = \mathbf{e}^l(l_k) \in \mathbb{R}^{d_l}$, where $\mathbf{e}^l$ denotes a label embedding lookup table. Label embeddings are randomly initialized and tuned during training.

Draft Label and Uncertainty Prediction After obtaining a sequence of word representations $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, VLSTM is adopted to generate the contextual features and uncertainty for each word. The forward and backward hidden states are concatenated for the final representation: $\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \in \mathbb{R}^{d_h}$. The forward pass of the VLSTM is run T times with the same inputs by using the approximated posterior $q_\theta(\mathbf{W}^*)$ in Equation 2 to obtain the draft labels as follows:

$$\mathbf{p}_i \approx \sum_{t=1}^{T} \texttt{Softmax}(\mathbf{l}_j^\top W_l \mathbf{h}_i | \mathbf{W}_t^*), \tag{3}$$

where W_l is a $d_l \times d_h$ matrix and $\texttt{Softmax}(z_{i;j}) = exp(z_{i;j}) / \sum_j exp(z_{i;j})$. Similar to a classic sequence labeling model, the model applies $l_i^* = \text{argmax}(\mathbf{p}_i)$ to obtain the draft label. Then, the uncertainty of this probability vector $\mathbf{p}_i$ can be summarized using the entropy of the probability vector:

$$u_i = H(\mathbf{p}_i) = -\sum_{c=1}^{C} p_c \log p_c. \tag{4}$$

In this way, it is possible to obtain the draft labels $L^* = \{l_1^*, l_2^*, \ldots, l_n^*\}$ coupled with the corresponding epistemic uncertainties $U = \{u_1, u_2, \ldots, u_n\}$ for each input sentence.

Document-Level Representation We introduce a key-value memory networks [Miller *et al.*, 2016] to simultaneously record the context representations (key) and corresponding label embeddings (value) for each word. Specifically, we create a document-level memory matrix $\mathbf{M} = \{\mathbf{m}_{w_1}, \ldots, \mathbf{m}_{w_m}\}$, in which the same words under different context would occupy many different slots and form a quired subset $\mathbf{m}_{w_i}$. The memory slots in one subset $\mathbf{m}_{w_i}$ are defined as pairs of vectors $(k_{i;1}, v_{i;1}), \ldots, (k_{i;m}, v_{i;m})$. In every single slot j, the key represents a certain hidden state vector $\mathbf{h}_{i;j}$, and the value is the corresponding label embedding $\mathbf{l}_{i;j}$.

3.2 Multi-Channel Transformer for Refinement

At the second stage, the draft predictions are refined based on the explicit co-occurrence relationship derived from the key-value memory network.

With the constructed memory matrix, a key addressing and value reading mechanism is designed to access the document-level context representations and label consistencies. During addressing, each slot of the corresponding subset is assigned a relevance probability by comparing the word to each key:

$$p_{h_{i;j}} = \texttt{softmax}(\mathbf{x}_i^\top W_h \mathbf{h}_{i;j}), \tag{5}$$

where W_l is a $d_w \times d_h$ matrix. We hope that the model can refer to the semantics and corresponding labels of other

sentences in the document when predicting the label of a word. Hence, in the reading step, the keys and values of the memory are read by taking their weighted sum using the addressing probabilities, and the document-level representations and label embeddings are returned:

$$\mathbf{h}_i^1 = \sum_j p_{h_{i;j}} \mathbf{h}_{i;j}^1; \quad \mathbf{l}_i^1 = \sum_j p_{h_{i;j}} \mathbf{l}_{i;j}^1. \tag{6}$$

We further use the multi-channel Transformer [Vaswani *et al.*, 2017] incorporating relative position encoding [Dai *et al.*, 2019] to model sentence-level dependencies in parallel. We propose to reparameterize the relative position encoding as follows:

$$\begin{aligned} \mathbf{A}_{i,j}^{h2h} &= \mathbf{h}_i^{1\top} \mathbf{W}_{qh}^\top \mathbf{W}_{kh} \mathbf{h}_j^1 + \mathbf{h}_i^{1\top} \mathbf{W}_{qh}^\top \mathbf{W}_{kR} \mathbf{R}_{i-j} \\ &\quad + \mathbf{u}_h^\top \mathbf{W}_{kh} \mathbf{h}_j^1 + \mathbf{v}_h^\top \mathbf{W}_{kR} \mathbf{R}_{i-j} \\ \mathbf{A}_{i,m}^{h2l} &= \mathbf{h}_i^{1\top} \mathbf{W}_{ql}^\top \mathbf{W}_{kl} \mathbf{l}_m^1 + \mathbf{h}_i^{1\top} \mathbf{W}_{ql}^\top \mathbf{W}_{kR} \mathbf{R}_{i-m} \\ &\quad + \mathbf{u}_l^\top \mathbf{W}_{kl} \mathbf{l}_m^1 + \mathbf{v}_l^\top \mathbf{W}_{kR} \mathbf{R}_{i-m}, \end{aligned} \tag{7}$$

where $\mathbf{A}_{i,j}^{h2h}$ and $\mathbf{A}_{i,m}^{h2l}$ denotes the attention from $\mathbf{h}_i$ to $\mathbf{h}_j$ and $\mathbf{h}_i$ to $\mathbf{l}_m$, respectively. $\mathbf{R}_{i-j}$ is the encoding of the relative distance between position i and j, and $\mathbf{R}$ is a sinusoid matrix like that in [Dai *et al.*, 2019]. $\varphi = \{\mathbf{W}, \mathbf{u}, \text{ and } \mathbf{v}\}$ are learnable parameters.

The proposed relative positional encoding can be used for the multi-channel Transformer architecture. The computational procedure for one layer with a single attention head can be summarized as follows:

$$\begin{aligned} &\mathbf{V}_h = \mathbf{H}^1 \mathbf{W}_h, \mathbf{a}_h = \texttt{Softmax}(\mathbf{A}^{h2h}) \mathbf{V}_h, \mathbf{H}^1 = \{\mathbf{h}_1^1, \ldots, \mathbf{h}_n^1\} \\ &\mathbf{V}_l = \mathbf{L}^1 \mathbf{W}_l, \mathbf{a}_l = \texttt{Softmax}(\mathbf{A}^{h2l}) \mathbf{V}_l, \mathbf{L}^1 = \{\mathbf{l}_1^1, \ldots, \mathbf{l}_n^1\} \\ &\mathbf{H}^2 = \texttt{FeedForward}(\texttt{LayerNorm}(\texttt{Linear}(\mathbf{a}_h) + \mathbf{H}^1)) \\ &\mathbf{L}^2 = \texttt{FeedForward}(\texttt{LayerNorm}(\texttt{Linear}(\mathbf{a}_l) + \mathbf{L}^1)). \end{aligned} \tag{8}$$

3.3 Training and Decoding with Uncertainty

There are two networks to be optimized: the VLSTM for draft labels and uncertainty estimation, and the multi-channel Transformer for label refinement. The ultimate training goal is to minimize the total loss function on the two models: $\mathcal{L}_{total} = \mathcal{L}_1(\theta, r) + \mathcal{L}_2(\varphi)$.

The VLSTM performs approximate variational inference, where we use a simple Bernoulli distribution (dropout) $q_\theta(\mathbf{W}^*)$ in a tractable family, which minimizes the Kullback-Leibler (KL) divergence to the true model posterior $p(\mathbf{W}|\mathcal{D})$. The minimization objective is given as follows:

$$\mathcal{L}_1(\theta, r) = -\frac{1}{N} \sum_{i=1}^{N} \log p(y_i | \mathbf{W}_j^*) + \frac{1-r}{2N} \parallel \theta \parallel^2, \tag{9}$$

where N is the number of data points, and r is the dropout probability to sample $\mathbf{W}_j^* \sim q_\theta(\mathbf{W}^*)$.

For the multi-channel Transformer, we use the concatenation of $\mathbf{H}^2$ and $\mathbf{L}^2$ for the final prediction $\hat{l}_i$. In particular, we can optimize the model using the cross entropy loss as follows:

$$\mathcal{L}_2(\varphi) = -\sum_{i=1}^{N} l_i \log \hat{l}_i, \tag{10}$$

where l_i is the one-hot vector of the true label.

When the training is complete, we can obtain the draft labels $L^* = \{l_1^*, l_2^*, \ldots, l_n^*\}$ coupled with corresponding uncertainties $U = \{u_1, u_2, \ldots, u_n\}$ from the VLSTM, and refined labels $\hat{L} = \{\hat{l}_1, \hat{l}_2, \ldots, \hat{l}_n\}$ from the multi-channel Transformer. We find that the average uncertainty value of incorrect draft labels is 29 times larger than that of correct draft labels in CoNLL2003. Hence,we can use uncertainty to indicate the labels with a high probability of being wrong. To prevent the correct labels from being incorrectly modified, we set an uncertainty threshold Γ to distinguish which labels should be used, i.e., refined labels are used when $u_i > \Gamma$ and vice versa (as an example, given $u_1 > \Gamma$, $u_2 \leq \Gamma$, and $u_n > \Gamma$, decoding labels will become $\{\hat{l}_1, l_2^*, \ldots, \hat{l}_n\}$).

4 Experimental Setup

In this section, we describe the three datasets used in our experiments. The applied baseline methods are then discussed in detail for comparison, along with the hyper-parameter configuration of the proposed model (DocL-NER).

4.1 Datasets

We conduct experiments on three benchmark NER datasets. We follow the standard split of each corpora. Statistics are listed in Table 1.

CoNLL2003. The shared task of the CoNLL2003 dataset [Tjong Kim Sang and De Meulder, 2003] for named entity recognition was collected from Reuters Corpus. This dataset divides name entities into four different types: persons (PER), locations (LOC), organizations (ORG), and miscellaneous(MISC). The original files use *-DOCSTART-* as document separator and we keep this to conduct our document-level experiments.

OntoNotes 5.0. The English NER dataset OntoNotes 5.0 [Weischedel *et al.*, 2013] is a large corpus consisting of various sources (newswire, broadcast, telephone speech, etc.). It is tagged with eighteen entity types (DATE, TIME, ORDINAL, etc.), *Part number* was regraded as document indicator in our experiments.

CHEMDNER. The CHEMDNER corpus consists of 10,000 PubMed abstracts published in the top journals from various chemistry-related disciplines, which contains a total of 84,355 chemical entity mentions labeled manually by expert chemistry literature curators [Krallinger *et al.*, 2015]. The corpus is tagged with only one chemical entity type. We regard each abstract as a document in our experiments.

4.2 Comparison Methods

BiLSTM-CRF Ma and Hovy [2016] utilizes a CRF layer on the top of the BiLSTM to model the interaction between two successive labels [Lample *et al.*, 2016] instead of making independent labeling decisions for each output. In the decoding stage, the Viterbi algorithm is used to find the highest scored label sequence over the entire word sequence.

GraphIE GraphIE [Qian *et al.*, 2019] utilizes a co-occurrence graph to incorporate document-level contextual information and CRF to model sentence-level label dependency. This methods achieves great performances

Dataset	Type	Train	Dev	Test
CoNLL2003	#doc	946	216	231
	#sent	14,041	3,250	3,453
OntoNotes	#doc	2,483	319	322
	#sent	59,924	8,528	8,262
CHEMDNER	#doc	3,500	3,500	3,000
	#sent	30,802	30,807	26,435

Table 1: Statistics of CoNLL2003, OntoNotes and CHEMDNER datasets.

in many information extraction tasks, including textual, social media and visual information extraction.

Hier-NER Hier-NER [Luo *et al.*, 2020] introduces a corpus-level memory mechanism to utilize the global contextual information of a token. Moreover, a sentence-level encoder is used to enhance the sentence representation learned from an independent BiLSTM. A CRF layer is used as the final decoder in Hier-NER. With two-level hierarchical representations, Hier-NER established state-of-the-art results on several NER tasks.

4.3 Hyper-Parameter Settings

Following [Ma and Hovy, 2016; Qian *et al.*, 2019; Luo *et al.*, 2020], we use the same 100-dimensional Glove embedding[2] [Pennington *et al.*, 2014] as word embedding for the CoNLL2003 and OntoNotes datasets. For CHEMDNER, we use 50-dimensional pretrained word2vec [Mikolov *et al.*, 2013] embedding, which is the same as [Qian *et al.*, 2019]. For CharCNN, we use 32-dimensional character embeddings and 32 filters of width 3 for CoNLL2003 and OntoNotes, and 128-dimensional character embeddings and 128 filters of width 2 to 4 for CHEMDNER. For the first stage, we use 1 layer of VLSTM with 200 dimensions for CoNLL2003 and 2 layers for the other datasets. For the second stage, we use 3 layers and 4 layers of Transformer for CoNLL2003, OntoNotes and CHEMDNER, respectively. Dropout is set to 0.5, and the number of samples is set to 32 for all the datasets. The standard entity-level F_1 score is used as evaluation metric. For all the experiments, we use the BIOES tag scheme instead of standard BIO2, as previous studies have reported a meaningful improvement with this scheme [Ma and Hovy, 2016; Lample *et al.*, 2016]. The computations for a single model are run on a GeForce GTX 1080Ti GPU.

5 Results and Analysis

In this section, we detail the performances of the proposed and baseline models. We present the results of a series of experiments to demonstrate the effectiveness of the proposed model.

5.1 Method Comparison

Table 2 lists the results of the proposed DocL-NER and previous approaches on the CoNLL2003, OntoNotes and CHEMDNER datasets. DocL-NER surpasses the state-of-the-art methods on all the three datasets. The models incorporating document-level context information (GraphIE and Hier-NER) can outperform those without. Because of the novel design of our document-level refinement networks, DocL-NER can utilize not only document-level context information, but also document-level label consistency. Hence, DocL-NER obtains significant improvement compared with the BiLSTM-CRF, GraphIE and Hier-NER models. Moreover, DocL-NER also outperforms the models that exploit additional task-specific resources or annotated corpora [Chiu and Nichols, 2016].

Models	CoNLL2003	OntoNotes	CHENDNER
[Chiu and Nichols, 2016]† ‡	91.62	86.34	-
[Lample *et al.*, 2016]	90.94	-	-
[Strubell *et al.*, 2017]	90.54	86.84	-
[Zhang *et al.*, 2018b]	91.22	-	-
[Ye and Ling, 2018]	91.26	-	-
[Zhang *et al.*, 2018a]	91.43	-	-
[Chen *et al.*, 2019]	91.44	87.67	-
[Cui and Zhang, 2019]	-	88.16	-
BiLSTM-CRF	91.21	86.99	89.45
GraphIE	91.74	87.43*	89.71
Hier-NER	91.96	87.98	89.53*
DocL-NER	**92.13**	**88.49**	**90.72**

Table 2: Results on CoNLL2003 test set. † refers to adopting external task-specific resources. ‡ refers to models trained on both training and development set. * are results using official released codes.

Models	$\mathbf{F_1}$
BERT-base [Devlin *et al.*, 2019]	91.82*
BERT-base + DocL-NER	**92.92**
ELMo [Peters *et al.*, 2018]	92.64*
ELMo + DocL-NER	**93.05**

Table 3: Results on CoNLL2003 test set by integrating Language Models. * refers to results rerun using fastNLP[3].

We also use the pretrained models (BERT, ELMo) to replace the default embeddings. The results are shown in Table 3. Although BERT and ELMo can model much more powerful context representations, their inability to model document-level label consistency still leads to insufficient performance. We find that by adding our method on the BERT and ELMo embeddings, the F_1 score on the CoNLL2003 dataset further improves 1.1% and 0.41%, respectively.

5.2 Efficiency Advantage

Table 4 shows a comparison of training and inference speeds on CoNLL2003 dataset, as well as accuracy on co-occurrence entities. In term of speed, DocL-NER outperforms baseline models by a large margin, because we use the multi-layer Transformer to fully exploit the GPU's parallelism instead of CRF. In term of advantage on co-occurrence entities, For the baseline models, they use CRF as decoder. In contrast, our method performs attention over document-level label embeddings derived from the memory network to explicitly model the co-occurrence relationship. Hence, our model can achieve better performance on the co-occurrence entity recognition.

[2]http://nlp.stanford.edu/projects/glove/

[3]https://github.com/fastnlp/fastNLP

Models	Train	Inference	Co-Acc
Hier-NER	1.00x	1.00x	92.74
GraphIE	1.78x	2.86x	93.05
BiLSTM-CRF	2.22x	4.76x	91.33
DocL-NER	**2.64x**	**5.48x**	**93.36**

Table 4: Speed and Co-Acc comparison on CoNLL2003 datasets. Co-Acc refers to the accuracy of co-occurrence tokens.

Models	CoNLL2003	OntoNotes	CHEMDNER
DocL-NER	**92.13**	**88.49**	**90.72**
- document-level label	91.81	88.20	90.36
- document-level context	91.46	88.00	90.12
- both	91.36	87.76	89.83
- sentence-level label	91.63	88.05	90.21
+ CRF	92.05	88.28	90.69
- refinement stage	90.83	87.09	89.43

Table 5: Ablation study of DocL-NER.

5.3 Ablation Study

To study the contribution of each component in DocL-NER, we conducted ablation experiments on the three datasets and display results in Table 5. In the first part, the results show that the model's performance is degraded when the document-level label information is removed, indicating that the previous methods that only incorporate document-level contextual information are insufficient to model the dependency between labels. More notably, we discover that results can be improved more when using document-level label and document-level context together than simply adding up the separate increases, which verifies the effectiveness of the proposed model. Since we also use the Transformer to model sentence-level label dependencies instead of CRF, we further investigate its advantage. As shown in the second part of Table 5, our method outperforms CRF since we utilize Transformer with relative position embedding to capture long-term label dependency, while CRF only considers the neighboring label dependencies. Moreover, an accompanying additional benefit is a faster training and decoding speed, as we have mentioned earlier. Furthermore, when we remove the second stage, the performance drops by 1.30%, 1.31%, and 1.29% for CoNLL2003, OntoNotes, and CHEMDNER, respectively, indicating that our refinement method is useful and can yield significant improvements.

5.4 Hyper-parameters Investigation

Memory Size The left part of figure 3 illustrates the performance of our model with respect to the maximum size of queried subset m for each word. The maximum size 10 is derived from the observation that 97.47% of the tokens belonging to an entity appeared less than 10 times in an document for the development set of CoNLL2003, with 91.65% for OntoNotes and 81.78% for CHEMDNER, respectively. We find that incorporating the first co-occurrence key-value information of an document can provide a performance improvement of at least 0.5% for all the three datasets. Note that different from Luo *et al.* [2020], we don't randomly select memories for a unique word. We let the memories in the original order as they appear in the document, since we believe that the contextual information for the place where an entity first appears in an document should be more adequate, and would be more useful in the future predictions.

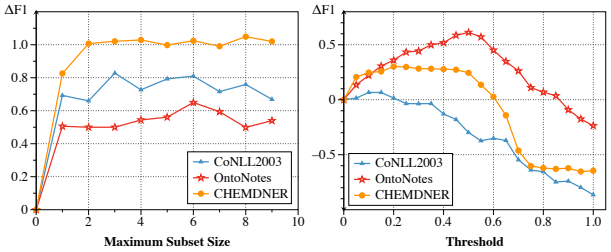

Figure 3: F_1 score with respect to the maximum size of queried subset m and threshold Γ. The results are evaluated on the development sets. ΔF_1 represents the F_1 scores at different steps minus the initial results.

Uncertainty Threshold To investigate the influence of uncertainty threshold Γ, we analyze the performance with different uncertainty thresholds on the three datasets, as shown in Figure 3. $\Gamma = 0$ denotes that the model use all of the refined labels as final predictions. As the threshold gets larger, the performance of DocL-NER improves by reducing the negative effects on correct draft labels. However, when Γ is too large, the model mainly utilizes draft labels from the first stage as final predictions, which leads to performance degradation. These results verify our motivation that a reasonable uncertainty threshold can avoid side effects on correct draft labels.

6 Conclusion

In this work, we introduce a novel two-stage label refinement approach to handle document-level label consistency. A key-value memory network is used to record the context representations and draft labels. A multi-channel Transformer explicitly models document-level word and label dependencies based on the information derived from the memory network. In order to mitigate the side effects of incorrect draft labels, we use Bayesian neural networks to indicate the labels with a high probability of being wrong. In addition, the proposed method can model the relationship between labels in parallel for faster inference. The experimental results on three named entity recognition benchmarks demonstrated that the proposed method significantly outperformed the previous state-of-the-art methods.

Acknowledgements

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by China National Key R&D Program (No. 2018YFB1005104, 2018YFC0831105, 2017YFB1002104), National Natural Science Foundation of China (No. 61751201, 61976056, 61532011), Shanghai Municipal Science and Technology Major Project (No.2018SHZDZX01), Science and Technology Commission of Shanghai Municipality Grant (No.18DZ1201000, 16JC1420401, 17JC1420200).

References

[Chen *et al.*, 2019] Hui Chen, Zijia Lin, Guiguang Ding, Jian-Guang Lou, Yusen Zhang, and Börje F. Karlsson. Grn: Gated relation network to enhance convolutional neural network for named entity recognition. In *AAAI*, 2019.

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

[Cui and Zhang, 2019] Leyang Cui and Yue Zhang. Hierarchically-refined label attention network for sequence labeling. In *EMNLP-IJCNLP*, 2019.

[Dai *et al.*, 2019] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *ACL*, 2019.

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

[Gal and Ghahramani, 2016a] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *ICML*, pages 1050–1059, 2016.

[Gal and Ghahramani, 2016b] Yarin Gal and Zoubin Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *NeurIPS*, 2016.

[Hu *et al.*, 2019] Anwen Hu, Zhicheng Dou, Jian-Yun Nie, and Ji-Rong Wen. Leveraging multi-token entities in document-level named entity recognition. In *AAAI*, 2019.

[Kendall and Gal, 2017] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, pages 5574–5584, 2017.

[Krallinger *et al.*, 2015] Martin Krallinger, Florian Leitner, Obdulia Rabal, Miguel Vazquez, Julen Oyarzabal, and Alfonso Valencia. Chemdner: The drugs and chemical names extraction challenge. *Journal of cheminformatics*, 7(1):S1, 2015.

[Krishnan and Manning, 2006] Vijay Krishnan and Christopher D Manning. An effective two-stage model for exploiting non-local dependencies in named entity recognition. In *ACL*, pages 1121–1128, 2006.

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

[Liu *et al.*, 2019] Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. Gcdt: A global context enhanced deep transition architecture for sequence labeling. *arXiv preprint arXiv:1906.02437*, 2019.

[Luo *et al.*, 2020] Ying Luo, Fengshun Xiao, and Hai Zhao. Hierarchical contextualized representation for named entity recognition. In *AAAI*, 2020.

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

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NeurIPS*. 2013.

[Miller *et al.*, 2016] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. In *EMNLP*, pages 1400–1409, 2016.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014.

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

[Qian *et al.*, 2019] Yujie Qian, Enrico Santus, Zhijing Jin, Jiang Guo, and Regina Barzilay. Graphie: A graph-based framework for information extraction. In *NAACL-HIT*, pages 751–761, 2019.

[Strubell *et al.*, 2017] Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. Fast and accurate entity recognition with iterated dilated convolutions. In *EMNLP*, 2017.

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

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

[Weischedel *et al.*, 2013] Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, et al. Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23, 2013.

[Ye and Ling, 2018] Zhixiu Ye and Zhen-Hua Ling. Hybrid semi-markov crf for neural sequence labeling. In *ACL*, pages 235–240, 2018.

[Zhang *et al.*, 2018a] Boliang Zhang, Spencer Whitehead, Lifu Huang, and Heng Ji. Global attention for name tagging. *CoNLL*, 2018.

[Zhang *et al.*, 2018b] Yuan Zhang, Hongshen Chen, Yihong Zhao, Qun Liu, and Dawei Yin. Learning tag dependencies for sequence tagging. In *IJCAI*, pages 4581–4587, 2018.

[Zhang *et al.*, 2018c] Yue Zhang, Qi Liu, and Linfeng Song. Sentence-state lstm for text representation. In *ACL*, 2018.