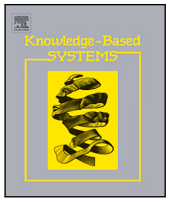
Knowledge-Based Systems 284 (2024) 111266

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



Exploiting global contextual information for document-level named entity recognition

Yiting Yu [a,1], Zanbo Wang [a,1], Wei Wei [a,*], Ruihan Zhang [a], Xian-Ling Mao [b], Shanshan Feng [c], Fei Wang [d], Zhiyong He [e], Sheng Jiang [a]

[a] Cognitive Computing and Intelligent Information Processing (CCIIP) Laboratory, School of Computer Science and Technology, Huazhong University of Science and Technology, China
[b] Department of Computer Science and Technology, Beijing Institute of Technology, China
[c] Centre for Frontier AI Research, IHPC, A*STAR, Singapore
[d] Institute of Computing Technology, Chinese Academy of Sciences, China
[e] School of Electronic Engineering, Naval University of Engineering, China

ARTICLE INFO

Keywords:
Named entity recognition
Global contextual information
Graph neural network
Epistemic uncertainty

ABSTRACT

Named entity recognition (NER, also known as entity chunking/extraction) is a fundamental sub-task of information extraction, which aims at identifying named entities from an unstructured text into pre-defined classes. Most of the existing works mainly focus on modeling local-context dependencies in a single sentence for entity type prediction. However, they may neglect the clues derived from other sentences within a document, and thus suffer from the sentence-level inherent ambiguity issue, which may make their performance drop to some extent. To this end, we propose a **G**lobal **C**ontext enhanced **Doc**ument-level NER (**GCDoc**) model for NER to fully exploit the global contextual information of a document in different levels, *i.e.,* word-level and sentence-level. Specifically, GCDoc constructs a document graph to capture the global dependencies of words for enriching the representations of each word in word-level. Then, it encodes the adjacent sentences for exploring the contexts across sentences to enhance the representation of the current sentence via the specially devised attention mechanism. Extensive experiments on two benchmark NER datasets (*i.e.,* CoNLL 2003 and Onenotes 5.0 English dataset) demonstrate the effectiveness of our proposed model, as compared to the competitive baselines.

1. Introduction

Named entity recognition aims to identify words or phrases as pre-defined categories, *e.g.,* persons, organizations, and locations. It is one of the fundamental subtasks in information extraction that benefits a variety of downstream applications, including relation extraction and knowledge graph completion [1,2]. With advancements in deep learning, much research [3–5] has been conducted to enhance NER systems by utilizing neural networks (NNs) to automatically extract features, yielding state-of-the-art performance.

Recent studies address named entity recognition as a sequence labeling task at the sentence level, which predicts the target named entity within a single sentence [6–8]. However, in fact, named entity mentions appearing repeatedly in a document may be naturally relevant to the current entity and thus can provide external context for entity understanding. For example, in Fig. 1(a), the surrounding sentences within the same document such as "6–0 win over", provide a supplemental clue that "VOLENDAM" represents a team, which needs to be tagged ORG rather than PER. In the case of Fig. 1(b), the baseline model fails to recognize "Rottweilers" (OOV) as a Miscellaneous (MISC) entity, while the previous sentence actually provides an obvious hint that it specifically refers to a pet name like "Doberman". By solely considering the individual sentence in which the named entity is located, there is a risk of encountering errors due to the limited context. This limitation aligns with the challenges faced by the baseline model. Therefore, in this paper, we propose to fully and effectively exploit the rich global contextual information to enhance the performance of document-level NER.

Indeed, there exist several attempts to utilize global information for the NER [5,9,10] task. However, indiscriminately incorporating surrounding information may meet with unrelated mentions. As shown in Fig. 1(c), it is important to note that not all instances of repeated entities mentioned in the document belong to the same categories. The

* Corresponding author.
E-mail address: weiw@hust.edu.cn (W. Wei).
[1] Both authors contributed equally to this research.

https://doi.org/10.1016/j.knosys.2023.111266
Received 30 July 2023; Received in revised form 29 November 2023; Accepted 1 December 2023
Available online 4 December 2023

Baseline	SOCCER - PSV HIT **VOLENDAM [S-PER]** FOR SIX.
Our model	SOCCER - PSV HIT **VOLENDAM [S-ORG]** FOR SIX.
Doc-sents	1. Brazilian striker Marcelo and Yugoslav midfielder Zeljko Petrovic each scored twice as Dutch first division leaders PSV Eindhoven romped to a 6-0 win over **Volendam [S-ORG]** on Saturday. 2. PSV, well on the way to their 14th league title, outgunned **Volendam [S-ORG]** in every department of the games.

(a)

Baseline	Today, **Rottweilers [O]** are on the way up," Rieck said.
Our model	Today, **Rottweilers [MISC]** are on the way up," Rieck said.
Neighbor sent	Speaking strictly of dogs, 15 years ago macho fad pet was a **Doberman [MISC]**.

(b)

Baseline	Midfielder Valentin Stefan and striker Viorel Ion of Otelul Galati and defender Liviu Ciobotariu of **National Bucharest [E-LOC]** are the newcomers for the European group eight clash in Macedonia on December 14.
Our model	... Viorel Ion of Otelul Galati and defender Liviu Ciobotariu of **National Bucharest [E-ORG]** are the newcomers for the European group ...
Doc-sents	1. Iordanescu said he had picked them because of their good performances in championship in which **National Bucharest [E-ORG]** are top ..... 2. **BUCHAREST [S-LOC]** 1996-12-06.

(c)

Fig. 1. Examples from the baseline and our model to illustrate our motivation.

occurrence of "BUCHAREST" in the second Doc-sents is considered noise in predicting the target entity type. Therefore, it becomes imperative to employ appropriate strategies to mitigate the interference caused by noise information.

To address the above issue, we propose a model named **GCDoc**, which fully exploits global contextual information for document-level NER from both the word-level and sentence-level perspectives. At the word-level, a simple but effective document graph is constructed to model the connection between words that reoccur in a document, and then the contextualized representation of each word is enriched. To avoid the interference of noise information, we further propose two strategies. We first apply epistemic uncertainty theory to filter out tokens whose representations are less reliable, thereby helping prune the document graph. Then we devise a selective auxiliary classifier for effectively capturing the importance of different neighboring nodes. At the sentence-level, we introduce a cross-sentence module that appropriately models a wider context. This involves utilizing three separate sentence encoders to encode the current sentence, as well as the preceding and succeeding sentences within a certain window size. The experimental results on two NER datasets demonstrate that our proposed method consistently outperforms the competitive baselines.

The main contributions of this paper can be summarized as follows.

- We propose to fully exploit the document-level context from the sentence-level and word-level by introducing the document graph and cross-sentence module.
- We devise a word-level graph constructed over recurrent tokens and develop two strategies to prune the graph, which captures the non-local dependencies existing among the recurrent tokens.
- We introduce a cross-sentence module, which encodes adjacent sentences and integrates cross-sentence representations through the utilization of attention and gating mechanisms.
- Extensive experimental results provide compelling evidence that our proposed model, GCDoc, outperforms competitive baselines on two NER benchmark datasets.

2. Related work

Named entity recognition is a long-standing study in the domain of natural language processing. According to the distinct decoder structures, the approaches can be categorized into three groups: tagging-based, span-based, and generation-based.

For tagging-based methods, named entity recognition is commonly regarded as a sequence labeling task, where the objective is to assign a specific tag from a predefined tagging scheme to each token in a given text. Traditional tagging-based models use classical machine learning methods, such as Hidden Markov Model (HMM) [11], Conditional Random Field (CRF) [12–14] and Support Vector Machine (SVM) [15]. Despite their good results, they heavily rely on handcrafted features and domain-specific knowledge. Consequently, numerous research endeavors have been undertaken to explore neural network-based methods for automatically learning feature representations for NER. Huang et al. [3] initially adopt a Bi-directional LSTM (Bi-LSTM) and the CRF model for NER to better capture long-range dependency features and thus achieve excellent performance. To encode character-level representations, several studies [4,6] expand upon the above model by using an additional LSTM/CNN layer. More Recently, the utilization of contextualized representations from pre-trained language models has been demonstrated to greatly enhance the performance [16–19].

However, most of the above approaches focus on the context within a sentence, resulting in diminished performance due to ambiguity inherent in such a context. To incorporate a wider range of contextual information beyond a single sentence, several approaches [5,9,10,20,21] have been carried out to utilize document-level information for NER. Luo et al. [20] introduce a document-level attention mechanism to allow the model to focus on tagging consistency across multiple instances of the same token in a document. Similarly, Zhang et al. [21] propose to retrieve supporting document-level context and dynamically calculate the attention of their contextual information. More recently, Luo et al. [5] propose to employ memory networks to memorize word representations during training, and Gui et al. [9] adopt similar memory networks to record document-level information and explicitly model the document-level label consistency of the same token sequences. Wang et al. [10] model the correlations for each token among all the related sentences in different documents through multi-task learning.

Meanwhile, span-based approaches [22–25] have been explored in NER, where all potential spans are enumerated and subjected to a span-level classification. For instance, Li et al. [22] reformulate NER as a machine reading comprehension (MRC) task and extract entities as the answer spans. Similarly, Shen et al. [23] introduce a two-stage identification process, utilizing a filter and regressor to generate span proposals, which are subsequently classified into their respective categories. To concurrently resolve unified NER, Li et al. [24] propose an innovative unified NER framework based on word-word relation classification.

Moreover, autoregressive generative named entity recognition (NER) methods aim to transform structured named entities into a sequential format. These approaches utilize sequence-to-sequence language models, such as BART [26] and T5 [27], to decode the named entities. [28,29] solve the NER task by a unified sequence-to-sequence (Seq2Seq) framework, which achieves promising performance and generalizability, eliminating the need for a specific tagging schema.

Although previous methods endeavor to exploit the global document-level representations with the attention operation, it is hard to measure the reliability of different types of contextual information and is thus far more limited in the ability to filter the noise information. Different from previous work, our proposed method fully exploits the global contextual information at both the word-level and sentence-level. Specifically, we use a more intuitive uncertainty-based pruning technique to prune the constructed word-level graph and introduce a selective auxiliary classifier to amend the edge weight, which enhances the comprehensibility of the process of aggregating the global contextual information across the sentences.

3. Preliminary

In this section, we first introduce the problem statement, and then describe a widely adopted and well-known method for NER, *i.e., Bi-LSTM-CRF*, which is also a baseline for our experiments. Next, we briefly present the theory of representing model uncertainty, which is used for pruning the document graph in our proposed method.

3.1. Problem formulation

Let $\mathbf{X}_D = \{x_i\}_m$ be a sequence composed of m tokens that represents a document D and $\mathbf{Y}_D = \{y_i\}_m$ be its corresponding sequence of tags over $\mathbf{X}_D$. Specifically, each sentence s_i in D is denoted by $s_i = \{x_j^i\}_{n_i}$, where n_i is the length of s_i and x_j^i indicates the jth token of s_i. Formally, given a document D, the problem of document-level name entity recognition (NER) is to learn a parameterized (θ) prediction function, *i.e.,* $f_\theta : \mathbf{X}_D \mapsto \mathbf{Y}_D$ from input tokens to NER labels over the entire document D.

3.2. Baseline model: Bi-LSTM-CRF

In the Bi-LSTM-CRF Model, an input sequence $\mathbf{X} = [x_1, x_2, \dots, x_n]$ is typically encoded into a sequence of low-dimensional distributed dense vectors, in which each element is formulated as a concatenation of pre-trained $\mathbf{w}_i$ and character-level $\mathbf{c}_i$ word embeddings $\mathbf{x}_i = [\mathbf{w}_i, \mathbf{c}_i]$. Then, a *context* encoder is employed (*i.e., Bi-directional LSTM*) to encode the embedded $\mathbf{X}$ into a sequence of hidden states $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]$ for capturing the local context dependencies within the sequence, namely,

$$\overrightarrow{\mathbf{h}}_i = \text{LSTM}(\overrightarrow{\mathbf{h}}_{i-1}, [\mathbf{w}_i; \mathbf{c}_i]) \tag{1}$$

$$\overleftarrow{\mathbf{h}}_i = \text{LSTM}(\overleftarrow{\mathbf{h}}_{i-1}, [\mathbf{w}_i; \mathbf{c}_i]) \tag{2}$$

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i, \overleftarrow{\mathbf{h}}_i], \tag{3}$$

where $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ are the hidden states incorporating past and future contexts of $\mathbf{x}_i$, respectively.

Subsequently, the Conditional Random Field (CRF [12]), a widely utilized tag decoder, is employed to jointly model and predict the final tags based on the output (*i.e.,* $\mathbf{H}$) of the *context* encoder. The CRF establishes a set of conditional probabilities, denoted as $\Pr(\mathbf{Y}|\mathbf{X})$, which represent the likelihood of different tag sequences Y given a specific input X.

$$\Pr(\mathbf{Y}|\mathbf{X}) = \frac{\prod_{y_j \in \mathbf{Y}} \phi(y_{j-1}, y_j, \mathbf{h}_j)}{\sum_{y'_j \in \mathbf{Y}', \mathbf{Y}' \in \mathbf{Y}(\mathbf{X})} \prod_{j=1}^{n} \phi(y'_{j-1}, y'_j, \mathbf{h}_j)} \tag{4}$$

$$\phi(y_{j-1}, y_j, \mathbf{h}_j) = \exp(\mathbf{W}_{y_{j-1}, y_j} \mathbf{h}_j + b_{y_{j-1}, y_j}), \tag{5}$$

where $\phi(.)$ represents the score function; $\mathbf{Y}(\mathbf{X})$ is the set of possible label sequences for $\mathbf{X}$; $\mathbf{W}_{y_{j-1}, y_j}$ and b_{y_{j-1}, y_j} represents the weight matrix and bias, respectively.

Then, a likelihood function $\mathcal{L}_m$ is employed, which aims to maximize the negative log probability of ground-truth sentences during the training process,

$$\mathcal{L}_m = -\sum_{\mathbf{X} \in \mathcal{X}; \mathbf{Y} \in \mathcal{Y}} \log \Pr(\mathbf{Y}|\mathbf{X}), \tag{6}$$

where $\mathcal{X}$ represents the collection of training instances and $\mathcal{Y}$ represents the corresponding tag set.

3.3. Representing model uncertainty

Bayesian modeling involves two primary categories of uncertainty [30]. Aleatoric uncertainty represents the intrinsic noise in the observations, while epistemic uncertainty takes into consideration the uncertainty in the model parameters. In this research, we adopt epistemic uncertainty to identify the tokens whose labels are likely to be incorrectly predicted by the model. This enables us to strategically prune the document graph, effectively mitigating the propagation of noise information.

Bayesian probability theory offers mathematical methodologies to estimate model uncertainty. However, it is worth mentioning that these approaches often involve high computational cost. Recently, there has been notable advancement in variational inference, which has introduced novel techniques to obtain improved approximations for Bayesian neural networks. Monte Carlo dropout [31] has emerged as a simple yet highly effective technique. It requires very minor changes to the original model and can be applied to any network architecture that includes dropout.

Given the dataset D with training inputs $\mathbf{X} = \{x_1, \dots, x_n\}$ and their corresponding labels $\mathbf{Y} = \{y_1, \dots, y_n\}$, the objective of Bayesian inference is to estimate $p(\mathbf{w}|D)$, *i.e.,* the posterior distribution of the model parameters $\mathbf{w}$ given the dataset D. Then the prediction for a new input x^* is obtained as follows:

$$p(y^*|x^*, D) = \int p(y^*|x^*, \mathbf{w}) p(\mathbf{w}|D) d\mathbf{w}. \tag{7}$$

Since the posterior distribution $p(\mathbf{w}|D)$ is intractable, Monte Carlo dropout adopts the variational inference approach to use $q_\theta(\mathbf{w})$. Then the minimization objective is the Kullback–Leibler (KL) divergence between the approximate posterior $q_\theta(\mathbf{w})$ and the true posterior $p(\mathbf{w}|D)$. With the Monte Carlo sampling strategy, the integral of Eq. (7) can be approximated as follows:

$$p(y^*|x^*, D) \approx \sum_{t=1}^{T} p(y^*|x^*, \mathbf{w}_t) q_\theta(\mathbf{w}_t), \tag{8}$$

where $\mathbf{w}_t \sim q_\theta(\mathbf{w})$ and T is the number of sampling times. In contrast to standard dropout which only works during the training stage, the Monte Carlo dropout is also activated during the testing phase. Then we can estimate the predictive uncertainty by aggregating the outcomes of stochastic forward iterations through the model and summarizing the predictive variance.

4. Proposed model

4.1. Overview

In this section, we present our proposed GCDoc model in detail, which is based on the Bi-LSTM-CRF framework (*rf.* Section 3.2). The overall model architecture is shown in Fig. 2.

Token Representation. Given a document composed of a set of tokens, *i.e.,* $\mathbf{X}_D = [x_1, \dots, x_m]$, the embedding layer is formulated as a concatenation of three different types of embeddings for obtaining the *word*-level (*i.e.,* pre-trained word embedding $\mathbf{w}_i$ and character-level embedding $\mathbf{c}_i$) and the *sentence*-level (*i.e.,* $\mathbf{s}_i$) simultaneously,

$$\mathbf{x}_i = [\mathbf{w}_i; \mathbf{c}_i; \mathbf{s}_i]. \tag{9}$$

Generally, the adjacent sentences may be naturally topically relevant to the current sentence within the same document, and thus we propose a novel cross-sentence contextual embedding module (bottom part in Fig. 2, *rf.* Section 4.2). This module aims to encode global-level contextual information at the sentence-level, thereby enhancing the representation of tokens within the current sentence.

Context Encoder. Next, the concatenated token embeddings $\mathbf{x}_i = [\mathbf{w}_i; \mathbf{c}_i; \mathbf{s}_i]$ (instead of $[\mathbf{w}_i; \mathbf{c}_i]$) is fed into a Bi-directional LSTM model to capture the local context dependencies within the sequence, and then we obtain a sequence hidden states $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n]$.

To enrich the contextual representation of each token, we develop a document graph module based on gated graph neural network to capture the non-local dependencies between tokens across sentences, namely,

$$\hat{\mathbf{h}}_i = GGNN(\mathbf{h}_i), \tag{10}$$

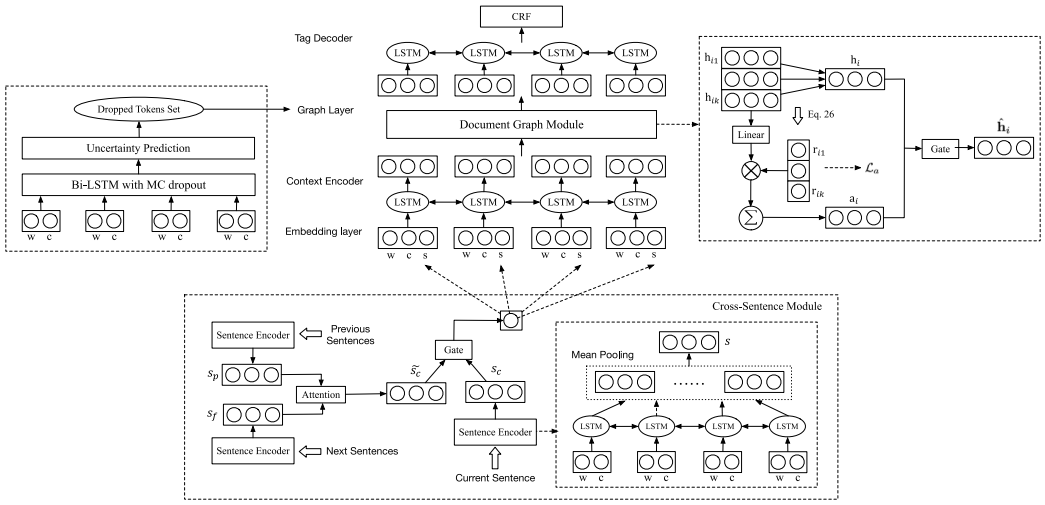


Fig. 2. The architecture of our proposed NER model. The upper center side shows the overall architecture, including the embedding layer, context encoder, graph layer, and tag decoder. We show the uncertainty prediction sub-module on the upper left side, and on the upper right side, we show the details of the document graph module. Note that the vectors $\mathbf{h}_{i1} \ldots \mathbf{h}_{ik}$ represents the neighbor nodes' representations of node v_i. At the bottom, we present the cross-sentence module that generates the enhanced sentence-level representation.

where $GGNN(.)$ is the output of the graph neural network for learning the contextualized representations of tokens.

Then, the output of Eq. (10) can be fed into the tag decoder for training via Eqs. (4), (5), and (6).

4.2. Cross-sentence contextual embedding module

In general, adjacent sentences within the same document often exhibit inherent topical relevance to the current sentence. Hence, we introduce an innovative cross-sentence module that considers neighboring sentences as supplementary background at the sentence-level so as to enhance the current sentence's representation. Consistent with [32], we only consider the contextual information across sentences within a range of k.

Specifically, we use the k sentences before/after the current sentence to enhance the current sentence's representation. Formally, for the representation of the current sentence $\mathbf{s}_c$, the preceding sentence's representation is denoted as $\mathbf{s}_p$ and the following sentence's representation is denoted as $\mathbf{s}_f$. We first adopt a compatibility function $f(\mathbf{s}_i, \mathbf{s}_j)$ to measure the relevance of such two auxiliary representations to the current sentence's representation $\mathbf{s}_c$, and then calculate a weighted sum of $\mathbf{s}_p$ and $\mathbf{s}_f$.

$$\tilde{\mathbf{s}}_c = a_p \mathbf{s}_p + a_f \mathbf{s}_f \tag{11}$$

$$a_p = \text{softmax}(f(\mathbf{s}_c, \mathbf{s}_p)) \tag{12}$$

$$a_f = \text{softmax}(f(\mathbf{s}_c, \mathbf{s}_f)), \tag{13}$$

$$f(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{w}^\top \sigma\left(\mathbf{W}_{\mathbf{s}}^{(\mathbf{i})} \mathbf{s}_i + \mathbf{W}_{\mathbf{s}}^{(\mathbf{j})} \mathbf{s}_j + \mathbf{b}\right), \tag{14}$$

where a_p and a_f are the learned attentions of the past and the future context information to the current sentence; $\sigma(\cdot)$ is an activation function; $\mathbf{W}_{\mathbf{s}}^{(\mathbf{i})}$, $\mathbf{W}_{\mathbf{s}}^{(\mathbf{j})} \in \mathbb{R}^{d_s \times d_s}$ are the learnable weight matrices; $\mathbf{w}^\top \in \mathbb{R}^{d_s}$ denotes a weight vector, while $\mathbf{b}$ represents the bias vector.

Finally, we employ a gating mechanism to determine the extent to which cross-sentence context information ($\tilde{\mathbf{s}}_c$) should be fused with the initial sentence representation $\mathbf{s}_c$.

$$\mathbf{s}'_c = \lambda \odot \mathbf{s}_c + (1 - \lambda) \odot \tilde{\mathbf{s}}_c, \tag{15}$$

where $\odot$ represents element-wise multiplication, and λ is the trade-off parameter, which is calculated by

$$\lambda = \text{sigmoid}\left(\mathbf{W}_{\mathbf{g}}^{(\mathbf{3})} \tanh(\mathbf{W}_{\mathbf{g}}^{(\mathbf{1})} \tilde{\mathbf{s}}_c + \mathbf{W}_{\mathbf{g}}^{(\mathbf{2})} \mathbf{s}_c)\right), \tag{16}$$

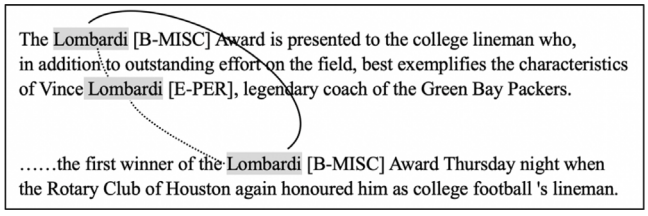


Fig. 3. Illustration for constructing the document graph. Words appearing repeatedly in a document are connected, and for clarity, we only adopt the word "Lombardi" as an example. Aside from the edge (solid line) connecting two nodes belonging to the same entity type, there are edges (dashed lines) that may introduce noise.

where $\mathbf{W}_{\mathbf{g}}^{(\mathbf{1})}, \mathbf{W}_{\mathbf{g}}^{(\mathbf{2})}, \mathbf{W}_{\mathbf{g}}^{(\mathbf{3})} \in \mathbb{R}^{d_s \times d_s}$ are trainable weight matrices.

The enhanced sentence-level representation $\mathbf{s}'_c \in \mathbb{R}^{d_s}$ is then concatenated with the word-level embeddings of tokens (*rf.* Eq. (9)) and fed into the context encoder. Note that the sentence representation $\mathbf{s}$ is computed by mean pooling over all hidden states $\mathbf{H}_s$ generated by the Bi-LSTM model according to Eqs. (1)–(3), that is,

$$\mathbf{s} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{h}_i^s, \tag{17}$$

where n is the length of the sequence.

4.3. Gated graph neural network-based context encoder

As mentioned, it is intuitive to assume that the entity mentions appearing repeatedly in a document are more likely to be of the same entity category. To this end, we devise a simple but effective word-level graph layer, which is built over the recurrent tokens to capture the non-local dependencies among them, as the local context of a token in a sentence may be ambiguous or limited.

Graph Construction. Given a document $\mathbf{X}_D = \{x_i\}_m$, a recurrent graph is defined as an undirected graph $\mathcal{G}_D = (\mathcal{V}_D, \mathcal{E}_D)$, where each node $v_i \in \mathcal{V}_D$ denotes a token and each edge $e_{ij} = (v_i, v_j) \in \mathcal{E}_D$ is a connection of every two same case-insensitive tokens (*rf.* Fig. 3).

Node Representation Learning. We adopt the graph neural network to generate the local permutation-invariant aggregation on the neighborhood of a node in a graph, and thus capture the non-local context information of node representations.

Specifically, for each node v_i, the information propagation process between different nodes can be formalized as:

$$\mathbf{a}_i = \mathrm{ReLU}\left(\frac{1}{|\mathcal{N}(i)|}\sum_{j\in\mathcal{N}(i)}\mathbf{W}_a\mathbf{h}_j + \mathbf{b}_a\right), \tag{18}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of v_i, which does not include v_i itself; $\mathbf{W}_a \in \mathbb{R}^{d\times d}$ indicates the weight matrices; $\mathbf{b}_a \in \mathbb{R}^d$ denotes the bias vector; and $\mathbf{h}_j$ is the hidden state of the neighbor node $v_j \in \mathcal{N}(i)$.

Through Eq. (18), we can easily aggregate the neighbors' information to enrich the context representation of node v_i via the constrained edges ($e_{ij} \in \mathcal{E}_D$). Inspired by the gated graph neural network, the enhanced representation $\mathbf{a}_i$ is combined with its initial embedding ($\mathbf{h}_i$) via a gating mechanism, which is employed for deciding the amount of aggregated context information to be incorporated, namely,

$$\hat{\mathbf{h}}_i = (1-\mathbf{z}_i)\odot\mathbf{h}_i + \mathbf{z}_i\odot\tilde{\mathbf{h}}_i \tag{19}$$

$$\tilde{\mathbf{h}}_i = \tanh(\mathbf{W}_o\mathbf{a}_i + \mathbf{U}_o(\mathbf{r}_i\odot\mathbf{h}_i)) \tag{20}$$

$$\mathbf{z}_i = \sigma(\mathbf{W}_z\mathbf{a}_i + \mathbf{U}_z\mathbf{h}_i) \tag{21}$$

$$\mathbf{r}_i = \sigma(\mathbf{W}_r\mathbf{a}_i + \mathbf{U}_r\mathbf{h}_i), \tag{22}$$

where $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W}_o, \mathbf{U}_z, \mathbf{U}_r, \mathbf{U}_o \in \mathbb{R}^{d\times d}$ are learnable weight matrices, $\sigma(\cdot)$ denotes the logistic sigmoid function and $\odot$ represents element-wise multiplication. $\mathbf{z}_i$ and $\mathbf{r}_i$ are the update gate and reset gate, respectively. The output $\hat{\mathbf{h}}_i$ derived from Eq. (19) is the ultimate output of our document graph module (*rf.* Eq. (10)), which will be subsequently fed into the tag decoder to generate the tag sequence.

However, directly aggregating the contextual information may contain noise since not all recurrent tokens can provide reliable context information for disambiguating others. Therefore, we propose two strategies to mitigate the inclusion of noise information in the document graph module. First, we apply epistemic uncertainty to identify the tokens whose labels are likely to be incorrectly predicted, thereby pruning the document graph. Then we adopt a selective auxiliary classifier for distinguishing the entity categories of two different nodes, thus guiding the process of calculating the edge weight.

(1) Document Graph Pruning. It is hypothesized that the tokens whose labels are prone to be accurately predicted by the model exhibit more trustworthy contextual representations, which will be more helpful for disambiguating other tokens in the graph module. Thus, we employ Monte Carlo dropout to estimate the model's uncertainty toward the prediction, so as to recognize whether the model's predictions are probable to be correct or not. Specifically, we obtain the uncertainty value of each token through an independent submodule. Then we set a threshold $\mathcal{T}$ and put tokens whose uncertainty value is greater than the threshold into a set *Drop*. Since the context information of these tokens should be less reliable, we prune the graph module by ignoring the impact of these tokens on their neighboring nodes. Thus Eq. (18) is rewritten as follows,

$$\mathbf{a}_i = \mathrm{ReLU}\left(\frac{1}{Num(i)}\sum_{j\in\mathcal{N}(i)\&j\notin Drop}\mathbf{W}_a\mathbf{h}_j + \mathbf{b}_a\right), \tag{23}$$

where $Num(i)$ represents the number of neighboring nodes connected to node i that actually participate in the operation.

As for the model uncertainty prediction, we apply the Monte Carlo dropout and adopt an independent submodule that predicts labels of each token by a simple architecture composed of a Bi-LSTM and dense layer. The forward pass of the Bi-LSTM is executed T times using the same inputs, employing the approximated posterior $q_\theta(\mathbf{w})$ as described in Eq. (8). Then given the output representation $\mathbf{h}_i$ of the Bi-LSTM layer, the probability distribution of the ith word's label can be acquired by a fully connected layer and a final softmax function,

$$\mathbf{p}_i \approx \sum_{t=1}^{T}\mathrm{softmax}(\mathbf{W}^\top\mathbf{h}_i|\mathbf{w}_t), \tag{24}$$

where $\mathbf{w}_t \sim q_\theta(\mathbf{w})$, $\mathbf{W} \in \mathbb{R}^{d\times C}$, and C is the number of all possible labels. Then, the uncertainty value of the token can be represented by the uncertainty of its corresponding probability vector $\mathbf{p}_i$, which can be summarized using the entropy of the probability vector:

$$u_i = -\sum_{c=1}^{C} p_c \log p_c. \tag{25}$$

(2) Edge Weight Calculation. Since not all of the repeated entities mentioned in the document belong to the same categories (*rf.* Fig. 3), we devise a selective auxiliary classifier to distinguish the entity categories of two different nodes, which takes the representation of two nodes as input and outputs a score that denotes the likelihood of these nodes belonging to the same entity category. Formally, the score r_{ij} is formulated as follows:

$$r_{ij} = \sigma(\mathbf{W}_c[\mathbf{h}_i;\mathbf{h}_j] + b_c), \tag{26}$$

where $\mathbf{W}_c \in \mathbb{R}^{2d}$ is a learnable weight matrix, b_c represents the bias and $\sigma(\cdot)$ denotes the logistic sigmoid function.

Specifically, the loss function of the selective auxiliary classifier is defined as:

$$\mathcal{L}_a = -\sum_{i=1}^{m}\sum_{j\in\mathcal{N}(i)}[y_{ij}\log(r_{ij}) + (1-y_{ij})\log(1-r_{ij})], \tag{27}$$

where y_{ij} is the ground truth label whose value is assigned 1 when nodes v_i and v_j belong to the same entity category, and 0 otherwise.

As a result, Eq. (23) can be transformed via the computed relation scores as follows:

$$\mathbf{a}_i = \mathrm{ReLU}\left(\frac{1}{Num(i)}\sum_{j\in\mathcal{N}(i)\&j\notin Drop} r_{ij}\left(\mathbf{W}_a\mathbf{h}_j + \mathbf{b}_a\right)\right). \tag{28}$$

After obtaining the enhanced representation $\mathbf{a}_i$ that encodes the neighbors' information, Eqs. (19)–(22) are applied to generate the final output of our document graph module $\hat{\mathbf{h}}_i$.

The auxiliary classifier can be regarded as a specialized regularization term that explicitly integrates supervisory information into the calculation of edge weights, which facilitates the model in effectively selecting useful information during the neighbor aggregation process. The final loss is the combination of loss $\mathcal{L}_m$ (Eq. (6)) and loss $\mathcal{L}_a$ (Eq. (27)),

$$Loss = \mathcal{L}_m + \theta\mathcal{L}_a, \tag{29}$$

where θ is a hyperparameter.

5. Experiments

5.1. DataSets

For evaluation, we utilize two benchmark Named Entity Recognition datasets, namely the CoNLL 2003 dataset (CoNLL03) and the OntoNotes 5.0 English datasets (OntoNotes 5.0). And both datasets are evaluated at the document-level.

- **CoNLL03** [33] comprises a collection of news wire articles extracted from the Reuters corpus. This dataset encompasses four types of named entities, namely persons (PER), locations (LOC), organizations (ORG), and miscellaneous (MISC). To maintain consistency and facilitate evaluation, we adopt the standard dataset split [34]. Furthermore, we adhere to the BIOES tagging scheme, wherein each token is labeled as B (beginning), I (inside), E (end), S (single-word entity), or O (non-entity).
- **OntoNotes 5.0** [35] contains 76,714 sentences from various sources (magazine, telephone conversation, newswire, and so on). The dataset is labeled with eighteen different entity kinds, including persons (PERSON), organizations (ORG), geographical entity (GPE), and law (LAW). Following [36], we adopt the portion of the dataset with gold-standard named entity annotations, in which the New Testaments portion is excluded.

Table 1
Comparison of overall performance on CoNLL 2003 NER task.

Category	Index & Model	F_1-score	
		Type	Value (±std[a])
Sentence-level	Lample et al., 2016 [4]	Reported	90.94
	Ma et al., 2016 [6]	Reported	91.21
	Yang et al., 2017 [40]	Reported	91.26
	Liu et al., 2018 [41]	Reported	91.24 ± 0.12
	Ye and Ling, 2018 [42]	Reported	91.38
Document-level	Zhang et al., 2018 [21]	Reported	91.64
	Qian et al., 2019 [43]	Reported	91.74
	Luo et al., 2020 [5]	Reported	91.96 ± 0.03
	Gui et al., 2020 [9]	Reported	92.13
	Bi-LSTM-CRF[b] [4]	Avg	91.01 ± 0.21
		Max	91.27
	GCDoc*	Avg	92.22 ± 0.02
		Max	**92.26**
+ Language Models/External knowledge			
Sentence-level	Peters et al., 2018 [18]	Reported	92.22 ± 0.10
	Devlin et al., 2018 [19]	Reported	92.80
	Shen et al., 2022 [44]	Reported	92.87
	Li et al., 2022 [24]	Reported	93.07
Document-level	Chen et al., 2020 [45]	Reported	92.68
	Luo et al., 2020 [5]	Reported	93.37 ± 0.04
	Gui et al., 2020 [9]	Reported	93.05
	Yan et al., 2021 [28]	Reported	93.24
	GCDoc + $BERT_{LARGE}$	Avg	93.40 ± 0.02
		Max	**93.42**

* Indicates statistical significance on the test dataset against Bi-LSTM-CRF by a paired t-test with $p < 0.01$.

[a] *std* means Standard Deviation.

[b] Here we re-implement the classical Bi-LSTM-CRF model by employing the identical model configuration and optimization approach utilized in our proposed model.

Table 2
Comparison of overall performance on OntoNotes 5.0 English datasets.

Category	Index & Model	F_1-score	
		Type	Value (±std)
Sentence-level	Chiu and Nichols, 2016 [36]	Reported	86.28 ± 0.26
	Strubell et al., 2017 [46]	Reported	86.84 ± 0.19
	Li et al., 2017 [47]	Reported	87.21
	Chen et al., 2019 [7]	Reported	87.67 ± 0.17
Document-level	Qian et al., 2019 [43]	Reported	87.43
	Luo et al., 2020 [5]	Reported	87.98 ± 0.05
	Bi-LSTM-CRF [4]	Avg	87.64 ± 0.23
		Max	87.80
	GCDoc*	Avg	88.32 ± 0.04
		Max	**88.35**
+ Language models/External knowledge			
Sentence-level	Clark et al., 2018 [48]	Reported	88.81 ± 0.09
	Liu et al., 2019 [49]	Reported	89.94 ± 0.16
	Jie and Lu, 2019 [50]	Reported	89.88
Document-level	Luo et al., 2020 [5]	Reported	90.30
	Yan et al., 2021 [28]	Reported	90.38
	GCDoc+$BERT_{LARGE}$	Avg	90.49 ± 0.09
		Max	**90.56**

* Indicates statistical significance on the test dataset against Bi-LSTM-CRF by a paired t-test with $p < 0.01$.

Table 3
Comparison of precision, recall and F_1-score between GCDoc and Bi-LSTM-CRF on CoNLL03 and OntoNotes 5.0 datasets.

Model	CoNLL03			OntoNotes 5.0		
	Precision	Recall	F_1-score	Precision	Recall	F_1-score
Bi-LSTM-CRF	90.96	91.07	91.01	87.58	87.70	87.64
GCDoc	92.12	92.33	92.22	87.90	88.74	88.32
Δ	+1.16	+1.26	+1.21	+0.32	+1.04	+0.68

Table 4
Results on CoNLL03 and OntoNotes 5.0 datasets.
Source: The results are reported from [25].

Model	CoNLL03	OntoNotes 5.0
Generation-based		
Yan et al., 2021 [28]	93.05	90.27
Lu et al., 2022 [29]	92.99	–
Span-based		
Li et al., 2020 [22]	93.04	90.02
Shen et al., 2021 [23]	92.94	–
Li et al., 2022 [24]	93.07	90.50
Shen et al., 2023 [25]	92.78	90.66
GCDoc (Ours)	**93.42**	**90.56**

5.2. Implementation details

Initialization. The word embedding is initialized using a 300-dimensional GloVe [37], whereas the character embedding is randomly initialized with a dimensionality of 30. We adopt a fine-tuning strategy and modify the initial word embedding during the training phase. All weight matrices are initialized by Glorot Initialization [38] and the bias parameters are initialized with 0.

Optimization. We optimize the model parameters using mini-batch stochastic gradient descent (SGD) with a batch size of 20 and a learning rate of 0.01. The L_2 regularization parameter is $1e{-}8$. We apply a dropout rate of 0.5 and set the clip norm threshold to 5. Early stopping [39] is also employed during the training process of our model.

Network Structure. The hidden state sizes of the character-level and word-level Bi-LSTM layers are configured as 50 and 200, respectively. Both of these layers are maintained at a fixed depth of 1. Additionally, the hidden dim of the sentence-level representation is set to 300 and the window size k in the cross-sentence module is 2. The threshold $\mathcal{T}$ in the uncertainty prediction submodule is set to 0.5. The hyperparameter θ in Eq. (29) is set to 0.1.

Training Time. Our model is implemented using the PyTorch library and trained on a single GeForce GTX 1080 Ti GPU. The model training is finished in around 1.6 h on the CoNLL03 dataset and approximately 6.7 h on the OntoNotes 5.0 dataset.

5.3. Evaluation results and analysis

5.3.1. Overall performance

In this section, we present an overall comparison of our proposed model with other competitive baselines on two NER datasets. The standard F_1-score is employed as the evaluation metric. Furthermore, we conduct three iterations of our model with distinct random initializations and report both the average and maximum results.

Tables 1 and 2 present the comparison results with the existing state-of-the-art approaches on the two benchmark datasets, from which we can observe that our model surpasses previous approaches on both datasets.

On the CoNLL 2003 dataset, we compare our model with the state-of-the-art models, including the models that only use the local context in a single sentence [4,6,18,19,24,40–42,44] and the models that use the global information to enhance the representation [5,9,21,28,43,45]. We find that the model leveraging document-level information outperforms the model solely relying on information contained within a single sentence. To ensure fair comparisons with models that employ pre-trained language models or external knowledge, we also incorporate the BERT language model [19] into our framework. We can observe that our model achieves 93.40% with BERT in terms of CoNLL03

Table 5
Ablation study of overall architecture on the CoNLL03 dataset.

No	Model	CoNLL03
1	Base model*	91.01 ± 0.21
2	+Cross-sentence module*	91.53 ± 0.09
3	+Doc graph	92.00 ± 0.07
4	+Doc graph & Pruning	92.10 ± 0.03
5	+Doc graph & Pruning & Edge Weight	92.22 ± 0.02
6	+Document graph module*	91.86 ± 0.02
7	+ALL	92.22 ± 0.02

* Indicates statistical significance on the test dataset against GCDoc by a paired t-test with $p < 0.01$. Note that in this table, $*$ measures the drop in performance.

datasets, demonstrating superior performance compared to the baseline models, regardless of whether pre-trained language models are employed.

On OntoNotes 5.0 English dataset, we compare our model with both sentence-level models [7,36,46–50] and document-level models [5,28,43]. As shown in Table 2, our model shows an advantage on this dataset, which surpasses previous baseline methods substantially at 88.32 (+0.68) without BERT and 90.49 (+0.11) with BERT.

To further illustrate the proposed module compared with the base model Bi-LSTM-CRF, we list the detailed comparison in Table 3, from which we can observe that our proposed model consistently and significantly outperforms Bi-LSTM-CRF under all metrics. The above phenomenon demonstrates that our proposed model is capable of leveraging the document-level contextual information to enhance NER tasks, even without the additional help of external knowledge.

In addition to the above tagging-based methods, recent advanced methods have tried to use different decoder structures. Therefore, we also compare our proposed method with the following baselines: (1) Generation-based methods [28,29]; (2) Span-based methods [22–25]. The experimental results presented in Table 4 reveal that our proposed method exhibits competitive performance in the field of named entity recognition (NER). It achieves comparable or even superior results when compared with state-of-the-art models on both NER datasets.

5.3.2. Ablation study

To better understand the effectiveness of two key components in our model, *i.e.,* the document graph module and cross-sentence module, we conduct ablation tests where one of the two components is individually adopted. The results are reported in Table 5.

The experimental results show that by adding either of the two modules, the model's performance on both datasets is significantly improved. Specifically, the adoption of the document graph module alone yields a substantial gain of 0.85% over the baseline. Furthermore, the integration of the cross-sentence module results in the best performance, affirming the effectiveness of our proposed approach in leveraging document-level contextual information at both the word and sentence levels.

In addition, in order to analyze the working mechanism of the document graph module in our proposed model, we incrementally add corresponding strategies to the base model only with the cross-sentence module. When simply incorporating a basic document graph, the model's performance gains a 0.47% improvement, which highlights the significance of capturing the word-level connections within a document in exploring valuable global contextual information. However, indiscriminately incorporating all information towards the target words would be more detrimental than beneficial. Thus, when adding the document graph pruning strategy, the performance of the model is further enhanced since less reliable tokens on their neighboring nodes are filtered out. Finally, when we add the designed selective auxiliary classifier to explicitly refine the edge weight of the document-level graph, there is a subsequent boost in performance, which proves the effectiveness of the proposed method in selecting useful information in the process of aggregating neighbors.

Table 6
Experimental results for ablating additional context in cross-sentence Module.

Previous sentences	Next sentences	F_1-score ± std
×	×	91.92 ± 0.14
✓	×	92.04 ± 0.13
×	✓	91.96 ± 0.18
✓	✓	92.22 ± 0.02

Apart from the above experiments, we also study the effectiveness of previous and next sentences in enhancing the semantic information of the current single sentence in our proposed model. In this experiment, one of the two parts is excluded from the model each time, and the corresponding results are presented in Table 6. In the first baseline, we only encode the current sentence and utilize it as the sentence-level representation. We find that including either previous or next sentences leads to enhancements compared to the base model. This observation confirms the effectiveness of incorporating a broader context and encoding neighboring sentences to improve the performance of the NER task.

5.3.3. Complexity analysis

In this section, we present a thorough analysis of the computational complexity of the proposed GCDoc model, with the goal of facilitating a more realistic comparison with the baseline model. In comparison to the Bi-LSTM-CRF model, GCDoc introduces two additional modules: the cross-sentence module and the document graph module. Therefore, we first analyze the complexity of these two modules individually and then compare the overall complexity of GCDoc with Bi-LSTM-CRF. Notably, we use the sign "n" to represent the length of the sentence, and the sign "d" denotes the dimensionality of the sentence embedding.

The cross-sentence module includes a sentence encoder based on Bi-LSTM (*rf.* Eqs. (1)–(3), (17)) and an attention mechanism for aggregating cross-sentence context (*rf.* Eqs. (11)–(16)). The time complexity of the two can be expressed as $O(nd^2)$ and $O(d^2)$ respectively. Therefore, the overall time complexity of the cross-sentence module is $O(nd^2)$.

The document graph module. The computational complexity of this module mainly comes from two parts: one is to use GNN to aggregate the neighbors' information of each node (*rf.* Eqs. (26), (28)), and the other is to combine the enhanced representation with the initial embedding via a gating mechanism (*rf.* Eqs. (19)–(22)). Note that the size of each node's neighbors varies and some may be very large. Therefore, we sample a fixed-size (*i.e., p*) neighborhood of each node as the receptive field during the data preprocessing stage, which can facilitate parallel computing in batches and improve efficiency. In this way, the computational complexity of aggregating the neighbors' information can be expressed as $O(npd^2)$. And the gating mechanism for combining representations requires $O(nd^2)$ computation complexity. Therefore, the overall time complexity of the document graph module is $O(npd^2)$.

Regarding the Bi-LSTM-CRF model, its time complexity can be represented as $O(nd^2)$. With the addition of the aforementioned two modules, the cross-sentence module and the document graph module, GCDoc expands its overall time complexity to $O(npd^2)$. However, it is worth noting that the variable "p" in this context is considered a constant, with a value of 5 in our experiments. As a result, the time complexity of GCDoc remains $O(nd^2)$. Notably, despite the enhancements in performance achieved by our model on both datasets, this does not come at the expense of increased computational complexity when compared to the Bi-LSTM-CRF model.

5.3.4. Case study

In this section, we provide a comprehensive analysis of the predictions obtained from our proposed model. Fig. 4(a) illustrates four instances in which our model predicts accurately but the Bi-LSTM-CRF

Case 1	
Baseline	**Charlton [S-ORG]** managed Ireland for 93 matches, during which time they lost only 17 times in almost 10 years until he resigned in December 1995.
Our model	**Charlton [S-PER]** managed Ireland for 93 matches, during which time they. lost only 17 times in almost 10 years until he resigned in December 1995.
Doc-sents	1. That is why this is so emotional a night for me, **Charlton [S-PER]** said. 2. SOCCER-ENGLISHMAN **CHARLTON [S-PER]** IS MADE AN HONORARY IRISHMAN.
Case 2	
Baseline	A mix-up in the **Barcelona [S-LOC]** defence let Croatian international Suker in midway through the first half, and Montenegrin striker Mijatovic made it 2-0 after fine work. by Clarence Seedorf just after the break.
Our model	A mix-up in the **Barcelona [S-ORG]** defence let Croatian international Suker in midway through the first half, and Montenegrin striker Mijatovic made it 2-0 after fine work. by Clarence Seedorf just after the break.
Doc-sents	1. Real Madrid 's Balkan strike force of Davor Suker and Predrag Mijatovic shot their side to a 2-0 win over **Barcelona [S-ORG]** in Spain 's old firm game. 2. The result leaves Real on 38 points after 16 games, four ahead of **Barcelona [S-ORG]**.
Case 3	
Baseline	A Barrick has teamed up with a construction company in the **Citra Group of Suharto [E-ORG]** 's eldest daughter, Siti Hardianti Rukmana ……
Our model	A Barrick has teamed up with a construction company in the Citra Group of **Suharto [S-PER]** 's eldest daughter, Siti Hardianti Rukmana ……
Doc-sents	Bre-X has a partnership deal with PT Panutan Duta of the Panutan Group run by President **Suharto [S-PER]** 's eldest son, Sigit Harjojudanto ……
Case 4	
Baseline	Viorel Ion of Otelul Galati and defender Liviu Ciobotariu of **National Bucharest [E-LOC]** are the newcomers for the European group eight clash in Macedonia.
Our model	Viorel Ion of Otelul Galati and defender Liviu Ciobotariu of **National Bucharest [E-ORG]** are the newcomers for the European group eight clash in Macedonia.
Doc-sents	1. Iordanescu said he had picked them because of their good performances in championship in which **National Bucharest [E-ORG]** are top and Otelul Galati third." 2. **BUCHAREST [S-LOC]** 1996-12-06. 3. League title-holders **Steaua Bucharest [E-ORG]**, who finished bottom of their Champions ' League group in the European Cup, have only two players.

(a)

Case 1	
Reference	..... attack-oriented Perugia led by in-form Croat striker **Milan [B-PER] Rapajic [B-PER]** and the experienced Fausto Pizzi.
Baseline	..... attack-oriented Perugia led by in-form Croat striker **Milan [B-PER] Rapajic [B-PER]** and the experienced Fausto Pizzi.
Our model	..... attack-oriented Perugia led by in-form Croat striker **Milan [B-ORG] Rapajic [B-ORG]** and the experienced Fausto Pizzi.
Doc-sents	1. Good news for **Milan [S-ORG]** is that Udinese 's German striker Oliver Bierhoff is out through injury. 2. Liberian striker George Weah makes a welcome return for **Milan [S-ORG]** alongside Roberto Baggio, with Montenegrin Dejan Savicevic in midfield.
Case 2	
Reference	5. **Switzerland [B-ORG] I [E-ORG]** (Reto Goetschi, Guido Acklin) 1:45.98
Baseline	5. **Switzerland [S-LOC]** I (Reto Goetschi, Guido Acklin) 1:45.98
Our model	5. **Switzerland [S-LOC]** I (Reto Goetschi, Guido Acklin) 1:45.98

(b)

Fig. 4. Case study of the Bi-LSTM-CRF and our proposed model. (a) presents the predictions of both models and (b) shows the incorrect predictions of GCDoc.

model does not. All examples are selected from the CoNLL 2003 test datasets. We also list the sentences in the same document that contain words mistakenly tagged by the baseline model (*Doc-sents* in the figure) to better understand the influence of utilizing document-level context in our model.

In the first case, the token "Charlton" is initially classified as an organization (ORG) by the baseline model, despite its correct categorization as a person's name (PER). The contextual cues within the provided *Doc-sents*, such as "said" in the first sentence and "ENGLISHMAN" in the second sentence, strongly indicate that "Charlton" should be correctly recognized as a person (PER). Nevertheless, our model successfully employs the rich contextual information available at the document level to accurately identify "Charlton" as a person (PER). In the second case, "Barcelona" is a polysemous word that can either represent a city in Spain or an organization as a football club name. Without obvious ORG-related context information, "Barcelona" is mistakenly labeled as S-LOC by the baseline model. However, our model successfully labels it as S-ORG by utilizing the useful sentences in the document. A similar situation is shown in the third case. We can infer from "s eldest daughter" that the word "Suharto" represents the name of a person in the sentence, rather than a part of the organization denoted as the "Citra Group of Suharto". And our model assigns the correct label S-PER to it with the help of another sentence in the same document but the baseline model fails.

In the fourth case, we show a situation where the document-level context information contains noise. Specifically, the baseline model incorrectly assigns the location (LOC) label to the word "Bucharest", despite its affiliation with the organization "National Bucharest". However, our model successfully identifies the crucial cues within the first and third sentences and correctly predicts the entity category, demonstrating the effectiveness of our model in mitigating the impact of noise and maintaining accurate entity classification.

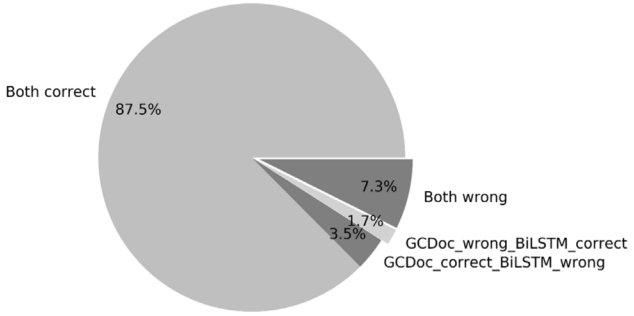


Fig. 5. Statistics of prediction of GCDoc and Bi-LSTM-CRF baseline on the sampled test set.

To enhance the credibility of our analysis, we also sample 20% of the CoNLL 2003 test set and carry out human assessment based on the sampled test set outputs. Based on the model's predictions, we can divide the sampled test set into four categories. The specific results are shown in Fig. 5.

For the cases where the prediction of GCDoc is correct but the prediction of the baseline is wrong, most of them contain limited or ambiguous context information within the sentence, which is similar to the cases shown in Fig. 4(a). However, our proposed model successfully improves the NER performance by leveraging global contextual information.

For the cases where the prediction of the baseline is accurate while GCDoc's prediction is incorrect, most of them belong to a situation where this entity appears multiple times in the document, but its entity category is different from the others. For instance, in the first case of Fig. 4(b), the word "Milan" is mentioned six times throughout the

Table 7
Comparison of performance on different types of entities between our proposed model and Bi-LSTM-CRF baseline on the CoNLL03 dataset.

Model	PER	LOC	ORG	MISC
Bi-LSTM-CRF	95.18	94.12	88.44	80.91
GCDoc	97.34	93.65	90.79	81.91
Δ	+2.16	−0.47	+2.35	+1.00

document, with five occurrences referring to the organizational name of the football club AC Milan. However, "Milan Rapajic" is indeed a person's name. In this situation, the document-level global information introduced by our model may be disruptive to the current sentence. One possible solution is to enhance the denoising strategy by adding a specific threshold for computing the edge weight. In cases where the edge weight is below this threshold, the information from that edge can be disregarded.

For the cases where both models predict incorrectly, most of them have limited context information within the sentence. And thus there is no document-level information that can be used to assist prediction. For example, in the second case of Fig. 4(b), the word "Switzerland" possesses a distinct connotation from its commonly known usage, which is challenging for the models to identify it correctly.

5.3.5. Performance on different types of entities

In this section, We further compare the performance of our model and Bi-LSTM-CRF baseline with respect to different types of entities. Table 7 shows the corresponding recall scores of the two models with regard to different types of entities on the CoNLL03 dataset.

In comparison to the Bi-LSTM-CRF model, the GCDoc model exhibits a minor decrease (0.47%) in recall score for the LOC entity type. We analyze the error cases and find that most of our prediction errors for LOC-type entities belong to the same situation as the second case in Fig. 4(b). It is postulated that the observed decline in performance for the LOC-type entity may be attributed to the disparity in dataset distribution. However, GCDoc achieves a substantial improvement of over 2% specifically for the PER and ORG entity types. We attribute this enhancement to the inclusion of the document graph module in GCDoc, which effectively integrates relevant global information for entities. It is noteworthy that there exist more scenarios where entity tokens of these two types appear repeatedly in the same document in the CoNLL03 test dataset.

5.3.6. Sensitivity analysis

Impact of uncertainty threshold. To study the influence of uncertainty threshold, we vary $\mathcal{T}$ in range of {0.1, 0.3, 0.5, 0.7, 1.0}. From the results shown in Fig. 6, we can find that: a too large value of $\mathcal{T}$ will cause poor performance, which related information may be filtered out due to the high $\mathcal{T}$. And generally, a $\mathcal{T}$ equals 0.5 could lead to a satisfactory performance.

Impact of selective auxiliary loss weight θ. The overall selective auxiliary loss defined in Eq. (29), can be considered as a specific form of regularization term. This parameter controls the influence of explicitly incorporating supervision information into the calculation of edge weight. To analyze the influence of coefficient θ, we vary θ in {0, 0.1, 0.2, 0.3, 0.4, 0.5} and report the results in Fig. 7. The experimental results indicate that an appropriate value of θ can enhance the model's ability to select relevant information during the neighbor aggregation process. Notably, the optimal performance is achieved when θ is set to 0.1.

6. Conclusions

This paper proposes a model that exploits document-level contextual information for NER at both the word-level and sentence-level.

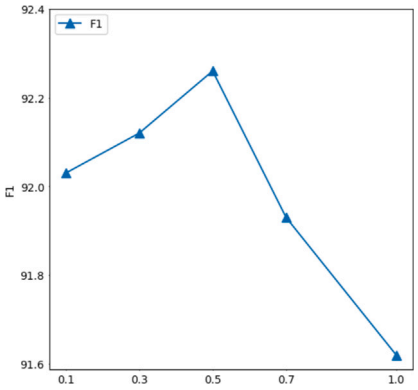


Fig. 6. Performance of our model with different uncertainty threshold on the CoNLL2003 NER task.

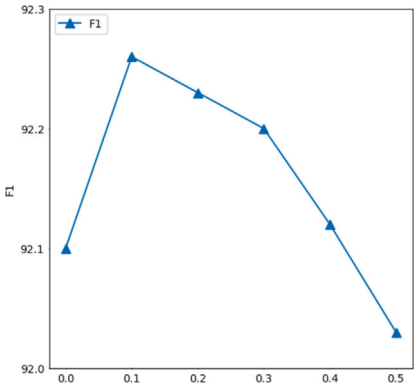


Fig. 7. Performance of our model with different loss weight θ on the CoNLL2003 NER task.

Specifically, we first construct a document graph to model the connection between words that reoccur in a document and further propose two strategies to avoid introducing noise information in the document graph module. In addition, a cross-sentence module is also designed to encode adjacent sentences to enrich the context of the current sentence. Extensive experiments conducted on two benchmark NER datasets (CoNLL 2003 and Ontonotes 5.0 English dataset) demonstrate the effectiveness of the proposed method.

CRediT authorship contribution statement

Yiting Yu: Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Zanbo Wang:** Writing – original draft, Visualization, Validation, Methodology, Investigation, Formal analysis, Data curation. **Wei Wei:** Writing – review & editing, Supervision. **Ruihan Zhang:** Visualization, Data curation. **Xian-Ling Mao:** Writing – review & editing. **Shanshan Feng:** Supervision. **Fei Wang:** Supervision. **Zhiyong He:** Supervision. **Sheng Jiang:** Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant No. 62276110, No. 62172039 and in part by the fund of the Joint Laboratory of HUST and Pingan Property & Casualty Research (HPL). The authors would also like to thank the anonymous reviewers for their comments on improving the quality of this paper.

References

[1] E. Cambria, B. White, Jumping NLP curves: A review of natural language processing research [review article], IEEE Comput. Intell. Mag. 9 (2) (2014) 48–57, http://dx.doi.org/10.1109/MCI.2014.2307227.

[2] T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing [review article], IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75, http://dx.doi.org/10.1109/MCI.2018.2840738.

[3] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, Comput. Sci. (2015).

[4] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: NAACL, 2016, pp. 260–270.

[5] Y. Luo, F. Xiao, H. Zhao, Hierarchical contextualized representation for named entity recognition, in: AAAI, 2020, pp. 8441–8448.

[6] X. Ma, E. Hovy, End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF, in: ACL, 2016, pp. 1064–1074.

[7] H. Chen, Z. Lin, G. Ding, J. Lou, Y. Zhang, B. Karlsson, GRN: Gated relation network to enhance convolutional neural network for named entity recognition, in: AAAI, 2019, pp. 6236–6243.

[8] W. Wei, Z. Wang, X. Mao, G. Zhou, P. Zhou, S. Jiang, Position-aware self-attention based neural sequence labeling, Pattern Recognit. 110 (2021) 107636, http://dx.doi.org/10.1016/j.patcog.2020.107636, URL https://www.sciencedirect.com/science/article/pii/S0031320320304398.

[9] T. Gui, J. Ye, Q. Zhang, Y. Zhou, X. Huang, Leveraging document-level label consistency for named entity recognition, in: IJCAI, 2020, pp. 3976–3982.

[10] D. Wang, H. Fan, J. Liu, Learning with joint cross-document information via multi-task learning for named entity recognition, Inform. Sci. 579 (2021) 454–467, http://dx.doi.org/10.1016/j.ins.2021.08.015, URL https://www.sciencedirect.com/science/article/pii/S002002552100815X.

[11] L.R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. IEEE 77 (2) (1989) 257–286.

[12] J. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: ICML, 2001, pp. 282–289.

[13] A. McCallum, W. Li, Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons, in: CoNLL@NAACL, 2003, pp. 188–191.

[14] A.L.F. Han, D.F. Wong, L.S. Chao, Chinese named entity recognition with conditional random fields in the light of Chinese characteristics, in: IIS, 2013, pp. 57–68.

[15] T. Kudoh, Y. Matsumoto, Use of support vector learning for chunk identification, in: CoNLL, 2000, pp. 142–144.

[16] R. Li, D. Li, J. Yang, F. Xiang, H. Ren, S. Jiang, L. Zhang, Joint extraction of entities and relations via an entity correlated attention neural model, Inform. Sci. 581 (2021) 179–193, http://dx.doi.org/10.1016/j.ins.2021.09.028, URL https://www.sciencedirect.com/science/article/pii/S0020025521009592.

[17] H. Zhang, X. Wang, J. Liu, L. Zhang, L. Ji, Chinese named entity recognition method for the finance domain based on enhanced features and pretrained language models, Inform. Sci. 625 (2023) 385–400, http://dx.doi.org/10.1016/j.ins.2022.12.049, URL https://www.sciencedirect.com/science/article/pii/S0020025522015444.

[18] M.E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: NAACL, 2018, pp. 2227–2237.

[19] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: NAACL, 2019, pp. 4171–4186.

[20] L. Luo, Z. Yang, P. Yang, Y. Zhang, L. Wang, H. Lin, J. Wang, An attention-based BiLSTM-CRF approach to document-level chemical named entity recognition, Bioinformatics (8) (2017) 8.

[21] B. Zhang, S. Whitehead, L. Huang, H. Ji, Global attention for name tagging, in: CoNLL, 2018, pp. 86–96.

[22] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, J. Li, A unified MRC framework for named entity recognition, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 5849–5859.

[23] Y. Shen, X. Ma, Z. Tan, S. Zhang, W. Wang, W. Lu, Locate and label: A two-stage identifier for nested named entity recognition, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 2782–2794.

[24] J. Li, H. Fei, J. Liu, S. Wu, M. Zhang, C. Teng, D. Ji, F. Li, Unified named entity recognition as word-word relation classification, in: AAAI, Vol. 36, 2022, pp. 10965–10973.

[25] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, Y. Zhuang, DiffusionNER: Boundary diffusion for named entity recognition, 2023, arXiv e-prints, arXiv–2305.

[26] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7871–7880, http://dx.doi.org/10.18653/v1/2020.acl-main.703, URL https://aclanthology.org/2020.acl-main.703.

[27] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (1) (2020) 5485–5551.

[28] H. Yan, T. Gui, J. Dai, Q. Guo, Z. Zhang, X. Qiu, A unified generative framework for various NER subtasks, in: ACL, Association for Computational Linguistics, Online, 2021, pp. 5808–5822, http://dx.doi.org/10.18653/v1/2021.acl-long.451, URL https://aclanthology.org/2021.acl-long.451.

[29] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, H. Wu, Unified structure generation for universal information extraction, Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 5755–5772.

[30] A. Kendall, Y. Gal, What uncertainties do we need in bayesian deep learning for computer vision? Adv. Neural Inf. Process. Syst. 30 (2017).

[31] Y. Gal, Z. Ghahramani, Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: ICML, 2016, pp. 1050–1059.

[32] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: ICLR, 2013, pp. 1–12.

[33] E.F.T.K. Sang, F.D. Meulder, Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition, in: NAACL-HLT, 2003, pp. 142–147.

[34] R. Collobert, K. Kavukcuoglu, J. Weston, L. Bottou, P. Kuksa, M. Karlen, Natural language processing (almost) from scratch, JMLR 12 (1) (2011) 2493–2537.

[35] S. Pradhan, A. Moschitti, N. Xue, H.T. Ng, A. Björkelund, O. Uryupina, Y. Zhang, Z. Zhong, Towards robust linguistic analysis using OntoNotes, in: Proceedings of the Seventeenth Conference on Computational Natural Language Learning, 2013, pp. 143–152.

[36] J.P. Chiu, E. Nichols, Named entity recognition with bidirectional LSTM-CNNs, TACL 4 (2016) 357–370.

[37] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.

[38] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: AISTATS, 2010, pp. 249–256.

[39] R. Caruana, S. Lawrence, C. Giles, Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping, in: NIPS, 2000, pp. 402–408.

[40] Z. Yang, R. Salakhutdinov, W.W. Cohen, Transfer learning for sequence tagging with hierarchical recurrent networks, in: ICLR, 2017, pp. 1–10.

[41] L. Liu, J. Shang, X. Ren, F.F. Xu, H. Gui, J. Peng, J. Han, Empower sequence labeling with task-aware neural language model, in: AAAI, 2018, pp. 5253–5260.

[42] Z.-X. Ye, Z.-H. Ling, Hybrid semi-markov crf for neural sequence labeling, in: ACL, 2018, pp. 235–240.

[43] Y. Qian, E. Santus, Z. Jin, J. Guo, R. Barzilay, Graphie: A graph-based framework for information extraction, in: NAACL, 2019, pp. 751–761.

[44] Y. Shen, X. Wang, Z. Tan, G. Xu, P. Xie, F. Huang, W. Lu, Y. Zhuang, Parallel instance query network for named entity recognition, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022, pp. 947–961.

[45] Y. Chen, C. Wu, T. Qi, Z. Yuan, Y. Huang, Named entity recognition in multi-level contexts, in: AACL/IJCNLP, 2020, pp. 181–190.

[46] E. Strubell, P. Verga, D. Belanger, A. McCallum, Fast and accurate entity recognition with iterated dilated convolutions, in: EMNLP, 2017, pp. 2670–2680.

[47] P.-H. Li, R.-P. Dong, Y.-S. Wang, J.-C. Chou, W.-Y. Ma, Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks, in: EMNLP, 2017, pp. 2664–2669.

[48] K. Clark, M.-T. Luong, C.D. Manning, Q.V. Le, Semi-supervised sequence modeling with cross-view training, in: EMNLP, 2018, pp. 1914–1925.

[49] T. Liu, J.-G. Yao, C.-Y. Lin, Towards improving neural named entity recognition with gazetteers, in: ACL, 2019, pp. 5301–5307.

[50] Z. Jie, W. Lu, Dependency-guided LSTM-CRF for named entity recognition, in: EMNLP, 2019, pp. 3860–3870.