Position-Aware Self-Attention based Neural Sequence Labeling

Wei Wei, Zanbo Wang, Xianling Mao, Guangyou Zhou, Pan Zhou, and Sheng Jiang

Abstract—Sequence labeling is a fundamental task in natural language processing and has been widely studied. Recently, RNN-based sequence labeling models have increasingly gained attentions. Despite superior performance achieved by learning the long short-term (i.e., successive) dependencies, the way of sequentially processing inputs might limit the ability to capture the non-continuous relations over tokens within a sentence. To tackle the problem, we focus on how to effectively model successive and discrete dependencies of each token for enhancing the sequence labeling performance. Specifically, we propose an innovative attention-based model (called position-aware self-attention, i.e., PSA) as well as a well-designed self-attentional context fusion layer within a neural network architecture, to explore the positional information of an input sequence for capturing the latent relations among tokens. Extensive experiments demonstrate our proposed model outperforms the state-of-the-arts without any external knowledge, in terms of various metrics.

Index Terms—Sequence labeling, self-attention, discrete context dependency.

I. INTRODUCTION

Sequence labeling, named SL, is one of pattern recognition task in the filed of natural language processing (NLP) and machine learning (ML), which aims to assign a categorical label to each element of a sequence of observed values, such as part-of-speech (POS) tagging [1], chunking [2] and named entity recognition (NER) [3] and etc. It plays a pivotal role in natural language understanding (NLU) and significantly beneficial for a variety of downstream applications, e.g., syntactic parsing, relation extraction and entity coreference resolution and etc.

Conventional sequence labeling approaches are usually on the basis of classical machine learning technologies, such as Hidden Markov Models (HMM) [4] and Conditional Random Fields (CRF) [5], which heavily rely on hand-crafted features (e.g., with/without capitalized word) or language-specific resources (e.g., gazetteers), making it difficult to apply them to new language-related tasks or domains. With advances in deep learning, many research efforts have been dedicated to enhancing SL by automatically extracting features via different types of neural networks (NNs), where various characteristics of word information are encoded in distributed representations for inputs [6] and the sentence-level context representations are learned when end-to-end training.

Recently, Recurrent Neural Network (RNN) together with its variants, e.g., long short-term memory (LSTM) or gated recurrent unit (GRU), have shown great success in modeling sequential data [7]. Therefore, many researches have devoted to research on RNN based architectures for SL, such as Bi-LSTM-CNN [8], LSTM-CRF [9], [10], LSTM-CNN-CRF [11] and etc. Despite superior performance achieved, these models have limitations under the fact that RNNs recursively compose each word with its previous hidden state encoded with the entire history information, but the latent independent relations between each pair of words are not well managed. The sequential way to process the inputs only focuses on modeling the long-range successive context dependencies, while neglecting the discrete context pattern.

Discrete context dependency plays a significant role in sequence labeling tasks. Generally, for a given word, its label not only depends on its own semantic information and neighbor contexts, but may also rely on the separate word information within the same sequences, which would significantly affect the accuracy of labeling. Without loss of generality, we take the part-of-speech (POS) tagging task as example, as shown in Figure 1 the part-of-speech tag of word “Industries” in Sentence-1 primarily depends on word “it”, and thus should label with NNP, which refers to singular proper noun. However, if such discrete context dependency is not well modeled, “Industries” may tend to be labeled with plural proper noun (NNPS) mistakenly, since a word ending with “-s” and more so “-es” is more likely labeled with NNPS. Similar to Sentence-2, assigning the part-of-speech tag list item marker (LS) to word “B” should take account of word “A” and “C”, where these three constitute a list. Therefore, it is essential to selectivly choose the contexts that have strong impacts on the tag of the given word.

In the case study part of Section V, we present multiple error cases to help explain and prove that the RNN models may cause insufficient modelling of discrete context dependency.
Many works demonstrate that self-attention is capable of effectively improving the performance of several NLP tasks such as machine translation, reading comprehension and semantic role labeling. This inspires us to introduce self-attention to explicitly model position-aware contexts of a given sequence.

Although encoding absolute positions of the input sequence with attentions \[1\] has been proven the effectiveness, compared with injecting absolute position embedding into the initial representations, it is more intuitive to incorporate the positional information in a relative manner. Recently, Shaw et al. \[12\] present an alternative approach to take the account of the relative distance between sequence elements for representation. Nevertheless, their approaches only consider the relative position information independent of the sequence tokens while neglecting the interaction with the input representations. Hence, how to effectively exploit the position information with attentions for better modeling the context dependency is still an open problem.

In this paper, we propose a novel RNN neural architecture for sequence labeling tasks, which employs self-attention to implicitly encode position information to provide complementary context information on the basis of Bi-LSTM. Additionally, we further propose an extension of standard additive self-attention mechanism (named position-aware self-attention, PSA) to model the discrete context dependencies of the input sequence. Differ from previous works, PSA maintains a variable-length memory to explore position information in a more flexible manner for tackling the above mentioned problem. That is, it jointly exploits three different positional biases, i.e., self-disabled mask bias, distance-aware Gaussian bias and token-specific position bias, to induce the latent independent relations among tokens, which can effectively model the discrete context dependencies of given sequence. Additionally, we also develop a well-designed self-attentional context fusion layer with feature-wise gating mechanism to dynamically select useful information about discrete context dependency and also address the self-disabled mask bias problem. Specifically, it learns a parameter \(\lambda\) to adaptively combine the input and the output of the position-aware self-attention and then generate the context-aware representations of each token. The extensive experiments conducted on four classical benchmark datasets within the domain of sequence labeling, i.e., the CoNLL 2003 NER, the WSJ portion of the Penn Treebank POS tagging, the CoNLL 2000 chunking and the OntoNotes 5.0 English NER, demonstrate that our proposed model achieves a significant improvement over the state-of-the-arts. The main contributions of this work are as follows.

- We identify the problem of modeling discrete context dependencies in sequence labeling tasks.
- We propose a novel position-aware self-attention to incorporate three different positional factors for exploring the relative position information among tokens; and also develop a well-designed self-attentional context fusion with feature-wise gating mechanism to provide complementary context information on the basis of Bi-LSTM for better modeling the discrete context dependencies over tokens.
- Extensive experiments on part-of-speech (POS) tagging, named entity recognition (NER) and phrase chunking tasks verify the effectiveness of our proposed model.

Roadmap. The remaining of the paper is organized as follows. In Section \[II\] we review the related work, and in Section \[III\] we presents a background on sequence labeling tasks, as well as a Bi-LSTM-CRF baseline model, followed with the proposed position-aware self-attention mechanism and self-attentional context fusion layer in Section \[IV\]. Section \[V\] presents the quantitative results on benchmark datasets, also includes an in-depth analysis, case study and wraps up discussion over the obtained results. Finally, Section \[VI\] concludes the paper.

II. RELATED WORK

There exist three threads of related work regarding our proposed sequence labeling problem, namely, sequence labeling, self-attention and position based attention.

A. Sequence Labeling

Sequence labeling is a category of fundamental tasks in natural language processing (NLP), e.g., POS tagging, phrase chunking, named entity recognition (NER) and etc. Most of conventional high performance sequence labeling approaches are based on classical statistical machine learning models, such as HMM \[13\], CRFs \[5\], \[14\], Support Vector Machine (SVM) \[15\], Perceptron \[16\], and etc., where the well-designed features are required for training.

Although the great success has been achieved by the traditional supervised learning based methods, these approaches require a lot of engineering skill and domain expertise to design handcrafted features.

With the rise of deep learning, many research efforts have been conducted on neural network based approaches to automatically learning the feature representation for SL tasks. The pioneering work is firstly proposed by Collobert et al. \[6\] to extract context-aware features using a simple feed-forward neural network with a fixed-size window, and generate the final labeled sequence through a CRF layer, which yields good performance in POS tagging, chunking, NER and etc. However, such window-based methods essentially follow a hypothesis, according to which the tags of an input word mainly depend on its neighboring words, while neglecting the global long-range contexts.

Hence, several variants of bidirectional recurrent neural networks, e.g., Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU), are proposed to encode long-range dependency features for the representation learning of each word, and thus achieve excellent performances. Huang et al. \[17\] initially employ a Bi-LSTM model to encode contextual representations of each word and then adopt a CRF model to jointly decode. Subsequently, the proposed Bi-LSTM-CRF architecture is widely used for various sequence labeling tasks. Lample et al. \[3\] and Ma et al. \[11\] both extend such model with an additional LSTM/CNN layer to encode character-level representations. Liu et al. \[2\] conduct a multi-task learning for sequence labeling by incorporating a character-aware neural language
model. Zhang et al. [18] propose a multi-channel model to learn the tag dependency via a combination of word-level Bi-LSTM and tag LSTM. Besides, there also exist several Bi-GRU based sequence labeling models, e.g., [19]. However, these RNN-based architectures are poor in modeling discrete context dependencies. In contrary, our proposed model is based on the Bi-LSTM-CRF architecture with self-attention mechanism to model the discrete position-aware dependencies for addressing the sequence labeling problem.

B. Attention Mechanism

Self-Attention. Here, we mainly focus on reviewing self-attention based methods. Self-attention is a special case of the attention mechanism to flexibly capture both successive and discrete dependencies over a given sequence. Indeed, many studies have devoted to research on how to utilize self-attention mechanisms to improve the performance of several NLP tasks through aligning scores of different elements within a sequence, such as reading comprehension [20], textual entailment [21], sentiment analysis [21], machine translation [11], language understanding [22] and semantic role labeling [23]. Cheng et al. [20] extend the LSTM architecture with self-attention to enable adaptive memory usage during recurrence, which favors to several NLP tasks, ranging from sentiment analysis to natural language inference. Lin et al. [21] introduce a sentence embedding model with self-attention, in which a 2-dimensional matrix is utilized to represent the embedding and each row of the matrix attends on a different part of the sentence. The model is applied to author profiling, sentiment analysis and textual entailment, and yields a significant performance gain over other methods. Vaswani et al. [11] propose a RNN/CNN free self-attention network to construct a sequence-to-sequence (i.e., seq2seq) model and achieve the state-of-the-arts in the neural machine translation (NMT) task. Shen et al. [22] employ self-attention to encode sentences and achieve great inference quality on a wide range of NLP tasks.

However, the purposes of these studies are different from the current work and thus will not be discussed in detail. The most related work is proposed by Tan et al. [23], where they propose a deep neural architecture with self-attention mechanism for semantic role labeling task and achieves the excellent performance, which inspire us to follow this line to apply self-attention to sequence labeling tasks for better learning the word-level context features and modeling the discrete dependencies over a given sequence.

Position based Attention. Attention mechanism has strong ability to model dependencies among tokens, but it cannot effectively make full use of the position information of the sequence in its structure. Vaswani et al. [11] propose a transformer model solely based on attention mechanism that achieves excellent performance for Neural Machine Translation (NMT) tasks, and they also point out the problem of neglecting the position information within attention in the existing methods. As such, they consider to inject position information using timing signal approach to encode absolute position, and then embed it into the representation of the input sequence in pre-processing progress with attentions. Following the success of Transformer, several subsequent studies using the Transformer architecture with the same strategy are proposed [23]. Show et al. [12] extend the self-attention mechanism to take into account the representations of the relative distances among sequence elements, and yields the substantial improvements in NMT task. Similarly, Sperber et al. [24] model the relative position information by strictly limit the scope of self-attention within their neighboring representations, which favors to the long-sequence acoustic modeling. Nevertheless, these approaches solely take account of the absolute or relative position information independent of sequence tokens while neglecting its interactions with their input presentations. In contrast, our proposed position-aware self-attention model explore the positional information of the given sequence in a more flexible manner, i.e., mainly focusing on modeling of discrete context dependencies of that sequence.

III. Preliminary

Typically, sequence labeling can be treated as a set of independent classification tasks, which makes the optimal label for each member and then the global best set of labels is chosen for the given sequence at once. Suppose we have a sequence (X) composed of n tokens, i.e., $x = [x_1, x_2, \ldots, x_n]^\top$, we aim to assign a tag to each member and output the corresponding globally best label sequence $y = [y_1, y_2, \ldots, y_n]^\top$. Many neural models are proposed for this task [3, 11]. By following the success of the state-of-the-art neural network architecture, we briefly describe a Bi-LSTM-CRF model for this task, which often consists of three major stages:

Distributed Representation, represents words in low dimensional real-valued dense vectors, where each dimension represents a latent feature. Besides pre-trained word embeddings for the basic input, several studies also incorporate character-level representations for exploiting useful intra-word information (e.g., prefix or suffix).

Context Encoder, captures the context dependencies and learns contextual representations for tag decoding. Traditional methods easily face the risk of gradient vanishing/exploding problem, and thus several variants of RNNs, e.g., LSTMs [23], are widely employed to be the context encoder architecture for different sequence labeling tasks, owing to their promising performance on handling such problems. Therefore, here we briefly illustrate a special case of LSTM-CRF model, i.e., Bi-directional LSTM-CRF, which incorporate past/future contexts from both directions (forward/backward) to generate the hidden states of each word, and then jointly concatenate them to represent the global information of the entire sequence.

However, the sequential way to process the inputs of RNNs might weaken the sensitivity of modeling discrete context dependencies, since it recursively compose each word with its forward/backward hidden state that encodes the entire history/future information. As such, the latent relationship between each pair of words is not well extracted, which is closely related to the final prediction task. To this end, in this paper we propose a self-attentional context fusion layer
to better capture the relations among tokens and help to model discrete context dependencies, via incorporating the complementary context information at different layers in our proposed neural architecture. We will detail it in the following sections, respectively.

**Tag Decoder.** employs a CRF layer to produce a sequence of tags corresponding to the input sequence. Typically, the correct label to each element of a given sequence often depends on the choices of nearby elements. As such, the correlations between labels of adjacent neighborhoods are usually considered for jointly decoding the best chain of labels for the entire sequence. Additionally, CRF model has been proven to be powerful in learning the strong dependencies across output labels, thus it is usually employed to make the optimal label for each element of the input sequence. Specifically, let \( Z = [\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_n]^\top \) be the output of context encoder of the given sequence \( X \), and thus the probability \( \Pr(\hat{y}|X) \) of generating the whole label sequence \( \hat{y} \) with regard to \( Z \) is calculated by CRF model \( \phi \).

\[
\Pr(\hat{y}|X) = \frac{\prod_{i=1}^{n} \phi(y_{j-1}, y_j, \hat{z}_j)}{\sum_{y' \in Y(Z)} \prod_{j=1}^{n} \phi(y'_{j-1}, y_j, \hat{z}_j)}, \tag{1}
\]

\[
\phi(y_j-1, y_j, \hat{z}_j) = \exp(W_{y_{j-1}, y_j} \hat{z}_j + b_{y_{j-1}, y_j}), \tag{2}
\]

where \( Y(Z) \) is the set of possible label sequences for \( Z \); \( W_{y_{j-1}, y_j} \) and \( b_{y_{j-1}, y_j} \) indicate the weighted matrix and bias parameters corresponding to the label pair \((y_{j-1}, y_j)\), respectively. Then, we employ a likelihood function \( L \) to minimize the negative log probability of the golden tag sequence for training,

\[
L = - \sum_{X \in \mathcal{X}, \hat{y} \in Y} \log p(\hat{y}|X), \tag{3}
\]

where \( \mathcal{X} \) denotes the set of training instances, and \( Y \) indicates the corresponding tag set.

**IV. PROPOSED APPROACH**

As aforementioned, RNN has limitations in modeling discrete context dependencies of the given sequence, thus in this paper we mainly focus on how to effectively model this kind of context dependencies during the context encoder stage within LSTM-CRF architecture (cf Section III). Therefore, we propose a new neural architecture for sequence labeling (shown in Figure 2), with a novel self-attentional context fusion layer that provides the complementary context information. Specifically, there are two context fusion layers are incorporated at different levels in our proposed architecture, i.e., the one is used for re-weighting the initial input (following the layer of distributed representations), and the other is added for re-weighting the output of word-level Bi-LSTM layer. The overall learning process of the proposed self-attentional context fusion network is illustrated in Algorithm 1. Besides, a well-designed position-aware self-attention mechanism with three different positional factors is also incorporated into the layer, which models the discrete context dependencies via exploring the relative position information of tokens in a flexible manner.

Next, we will elaborate our proposed sequence labeling model in detail. More concretely, Section IV-A will present the proposed position-aware self-attention mechanism, followed with the illustration of the proposed context fusion layer in Section IV-B.

**Algorithm 1 Learning Processes of Self-Attentional Context Fusion Network**

**Input:** The original token representations of sequence \( X = [x_1, x_2, \ldots, x_n] \)

**Output:** The context-aware representations of sequence \( \tilde{X} = [\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_n] \)

1: for \( i \in \{1 \ldots n\} \) do
2:     for \( j \in \{1 \ldots n\} \) do
3:         // Compute the alignment score of \( \hat{x}_i \) and \( \hat{x}_j \)
4:         Update \( f(\hat{x}_i, \hat{x}_j) \) based on Equation 4
5:         // Compute the positional bias
6:         Update \( \psi_{ij}(\hat{x}_i) \) based on Equation 4
7:         Update \( f(\hat{x}_i, \hat{x}_j) \) by adding \( \psi_{ij}(\hat{x}_i) \)
8:     end for
9:     for \( j \in \{1 \ldots n\} \) do
10:    \( a_i(j) = \text{Softmax}(f(\hat{x}_i, \hat{x}_j)) \)
11: end for
12: \( \hat{s}_i \leftarrow \text{Sum}(a_i(j) \odot \hat{x}_j) \)
13: \( \hat{s}_i \leftarrow \text{Fullyconnect}(\hat{s}_i) \)
14: // Fusion gate mechanism
15: Update \( \lambda \) based on Equation 13
16: Update \( \tilde{x}_i \) based on Equation 13
17: end for

**A. Position-Aware Self-Attention**

In this section, we present a novel position-aware self-attention for better inducing the importance of each token to a specified token within the same sequence.

Position modeling is benefit for optimizing the self-attention network, since self-attention cannot encode position information of tokens in sequence. Although the position information is implicitly encoded by LSTM in our neural architecture, the process of calculating alignment scores within self-attention is independent of the relative distance of tokens. To this end, here
we explore the positional information of an input sequence to extend self-attention model with a different and novel method, aiming to better model the discrete context dependencies of sequence. To be specific, we introduce three different positional factors, i.e., self-disabled mask bias $M_{ij}(\cdot)$, distance-aware Gaussian bias $G_{ij}(\cdot)$ and token-specific position bias $P_{ij}(\cdot)$, which are combined in a global positional bias function $\Psi_{ij}(\cdot)$, and added to the baseline self-attention. The three factors are combined by

$$\Psi_{ij}(\tilde{x}_i) = M_{ij}(\tilde{x}_i) + \alpha G_{ij}(\tilde{x}_i) + (1 - \alpha)P_{ij}(\tilde{x}_i),$$

where $\alpha$ is a trainable trade-off parameter that controls the contributions of different biases.

The self-disabled mask bias $M_{ij}(\cdot)$ disables the attention of each token to itself, for better measuring its dependency on other tokens. The distance-aware Gaussian bias $G_{ij}(\cdot)$ considers the information of relative distance by utilizing the form of Gaussian distribution, and explicitly affect the computation of attention weights. The token-specific position bias $P_{ij}(\cdot)$ further addresses the interactions between the representations of relative positions and the input presentations, thus explores the relative distance in a more flexible manner. The details of these three factors will be illustrated in the following sections.

More concretely, assume the token representations of sequence $X = [x_1, x_2, \ldots, x_n]^T$ with $x_i \in \mathbb{R}^d$. To measure the attention weight of each $x_i$ to a specified token $\tilde{x}_i$, a compatibility function $f(\tilde{x}_i, x_j)$ is employed to measure the pairwise similarity (i.e., the alignment score) of $\tilde{x}_i$ and $x_j$.

Many different self-attention mechanisms are proposed but are different in the compatibility function $f(\tilde{x}_i, x_j)$, here we adopt additive attention mechanism [20], which is implemented by a one-layer feed-forward neural network and is often superior to others in practice, which is computed by

$$f(\tilde{x}_i, x_j) = \hat{w}^T \sigma(W^{(1)}\tilde{x}_i + W^{(2)}x_j + \hat{b}),$$

where $\sigma(\cdot)$ is an activation function; $W^{(1)}, W^{(2)} \in \mathbb{R}^{d \times d}$ indet the weight matrices; $\hat{w} \in \mathbb{R}^d$ is a weight vector, and $\hat{b}$ denotes the bias vector.

For effectively encoding position information, we incorporate the proposed positional bias function $\Psi_{ij}(\cdot)$ to $f(\tilde{x}_i, x_j)$, and the position-aware self-attention is rewritten by

$$f(\tilde{x}_i, x_j) = w^T \sigma(W^{(1)}\tilde{x}_i + W^{(2)}x_j + \hat{b}) + \Psi_{ij}(x_j),$$

Then the alignment score is converted by a softmax function with the normalization of all the $n$ elements within $X$, i.e.,

$$a_i(j) = \frac{\exp(f(\tilde{x}_i, x_j))}{\sum_{j'} \exp(f(\tilde{x}_i, x_{j'}))}. \quad (7)$$

Finally, the output ($\hat{s}_i \in \mathbb{R}^d$) of the self-attention of $\tilde{x}_i$ is a weighted sum of representations of all tokens in $X$ according to the alignment scores, namely,

$$\hat{s}_i = \sum_{j=1}^n a_i(j) \odot \tilde{x}_j. \quad (8)$$

1) Self-Disabled Mask Bias: For a specific token $x_i$, the goal of our self-attentional model is to measure its dependency on other tokens in the same sequence and further capture discrete context information, thus it is benefit to prevent the interference of itself information when calculating alignment scores, through disabling the attention of each token to itself. As such, we adopt self-disabled mask [22] for self-attention, which is

$$M_{ij}(x_i) = \begin{cases} 0, & i \neq j, \\ -\infty, & i = j, \end{cases} \quad (9)$$

where $-\infty$ is used to neglect itself contribution in self-attention.

2) Distance-Aware Gaussian Bias: Self-attention mechanism models the global dependencies among input tokens regardless of their distance, while the relative position information is important for modeling the local context in sequence labeling tasks. Without loss of generality, we take POS tagging as an example, the POS tag of a word is more likely influenced by its neighbors, as compared with other long-distance words. In order to favor the modeling of short-range dependencies by self-attention, we take account of a distance-aware Gaussian bias to control the scope of local context of a specified token $x_i$, and by incorporating it into the compatibility function, we make the relative distance among tokens to explicitly affect the computation of their corresponding attention weights. The distance-aware Gaussian bias is defined as

$$G_{ij}(\tilde{x}_i) = \frac{-\left(i - j\right)^2}{2\epsilon^2}, \quad (10)$$

where $i, j$ indicates the order of $\tilde{x}_i$ and $\tilde{x}_j$; parameter $\epsilon$ refers to the standard deviation that is empirically set as $\epsilon = \frac{k}{2.3}$ and $k$ is a window size, which is set as 10 in our experiments.

3) Token-specific Position Bias: Gaussian bias only takes into account the information of relative distance among tokens, however the way a relative distance affects the distribution of attention might not be the same for different focused tokens, and the discrete context dependencies within the sequence also have much diversity. As such, modeling of the relative distance should be further explored in a more flexible manner for addressing the interactions between the representations of relative positions and the input presentations.

Inspired by Shaw’s work [12], we learn a relative position representation matrix $R \in \mathbb{R}^{r \times d}$ and inject the position information into the attention score. Here $d$ denotes the representation dim and $r$ is a nonnegative value that reflects the maximum margin between two different tokens. In other words, the relative distance between two tokens would be clipped to $r$ if it is greater than the threshold, following the essential hypothesis that the precise relative position information is not useful while beyond a certain distance. And its value is equal to the window size $w$. Specifically, a scalar $P_{ij}(\tilde{x}_i)$ is composed of two term, which are parameterized as follows,

$$P_{ij}(\tilde{x}_i) = \begin{cases} \tilde{x}_i^\top R_r + (\hat{v}^\top R_r + \hat{b}), & |i - j| > r, \\ \tilde{x}_i^\top R_{|i-j|} + (\hat{v}^\top R_{|i-j|} + \hat{b}), & |i - j| \leq r. \end{cases} \quad (11)$$

Note that in the remainder it has the similar meaning when the context is clear and discriminative.
product of \( \tilde{x}_i \) and \(|i-j|\)-th (or \( r \)-th) element of \( \mathbf{R} \), which represent the content-dependent positional information. And the second term that transforms the corresponding representation of relative position to a scalar score, can be regarded as a general global position bias. Note that the relative position representation matrix \( \mathbf{R} \) is also trainable which is optimized during training along with other parameters.

**B. Self-Attentional Context Fusion Layer**

The success of neural networks stems from their highly flexible non-linear transformations.

Attention mechanism utilizes a weighted sum to generate the output vectors, which limits its representational ability. To further enhance the power of feature extraction of the attentional layer, we take account of employing two fully connected layers to transform the outputs of the attention module, which is formally computed by

\[
\tilde{s}_i = \tanh(W^{(z1)} \tanh(W^{(a1)} \tilde{s}_i + \tilde{b}_1)), \quad (12)
\]

where \( W^{(z1)}, W^{(a1)} \in \mathbb{R}^{d \times d} \) are trainable matrices; and \( \tilde{s}_i \) denotes the output of the self-attention of \( \tilde{x}_i \) (cf Eq 8).

As we introduce a self-disabled mask (cf Section IV-A) to disable the attention of each token to itself, the output of the proposed self-attention layer is insufficient for learning context-aware representation. As such, we propose a feature-wise fusion gate mechanism to adaptively combine the feature of each token with its context. Hence, the final context-aware representation of \( \tilde{x}_i \) and the output of the fully connected layers \( \tilde{s}_i \), namely

\[
\lambda = \text{sigmoid}(W^{(\lambda)} \tanh(W^{(\tilde{l})} \tilde{x}_i + W^{(\tilde{f})} \tilde{s}_i)) \quad (13)
\]

\[
\tilde{x}_i = \lambda \odot \tilde{x}_i + (1 - \lambda) \odot \tilde{s}_i \quad (14)
\]

where \( W^{(\tilde{l})}, W^{(\tilde{f})}, W^{(\lambda)} \in \mathbb{R}^{d \times d} \) are trainable weight matrices of the fusion gate. Note that the learned parameter \( \lambda \) is a vector that has the same dimension with \( \tilde{s}_i \), because different features of \( \tilde{s}_i \) can contain different information of discrete context dependency. Hence the designed fusion gate is able to dynamically select useful information from the self-attention layer in a fine-grained manner.

### V. Experiments

#### A. Data Sets

We use four benchmark sequence labeling datasets for evaluation, i.e., CoNLL 2003 NER dataset (CoNLL03 NER), the Wall Street Journal portion of Penn Treebank dataset (WSJ), CoNLL 2000 chunking dataset (CoNLL00 chunking) and OntoNotes 5.0 English NER datasets (OntoNotes 5.0). The details about such corpora are shown in Table I.

- **CoNLL03 NER** is a collection of news wire articles from the Reuters corpus, which includes four different types of named entities: PER, LOC, ORG, and MISC. We use the standard dataset split [6] and follow BIOES tagging scheme (B, I, O, E, S).

- **WSJ** contains 25 sections and classifies each word into 45 different types of POS tags. Here, we also adopt a standard data split method used in [13], namely, sections 0-18 as training data, 19-21 as development data, and sections 22-24 as test data.

- **CoNLL00 chunking** uses sections 15-18 from the Wall Street Journal corpus for training and section 20 for testing. It defines 11 syntactic chunk types (e.g., NP, VP, ADJP) in addition to other. Following previous works [27], we randomly sampled 1000 sentences from the training set as development data.

- **OntoNotes 5.0** is much larger than CoNLL 2003 NER dataset, and consists of text from a wide variety of sources (broadcast conversation, newswire, magazine, Web text, etc.). It is tagged with eighteen entity types (PERSON, ORG, GPE, LAW, etc.). Following previous works [8], we adopt the portion of the dataset with gold-standard named entity annotations, in which the New Testament portion is excluded.

#### B. Experimental Setting

We use LSTM to learn character-level representation of words, and together with the pre-trained word embedding contributes to the distributed representation for input. Then we initialize word embedding with 100-dimensional GloVe [28] and randomly initialize 30-dimensional character embedding. Fine-tuning strategy is adopted that we modify initial word embedding during gradient updates of the neural network model by back-propagating gradients.
random initialization and report both average and max results of our proposed model as well as our re-implemented Bi-LSTM-CRF baseline. The comparison methods used in this work are the state-of-arts in recent years that usually compared in many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.

C. Evaluation Results and Analysis

1) Over Performance: This experiment is to evaluate the effectiveness of sequence labeling on different datasets by our approach. Specifically, we report standard F1-score for CoNLL 2003 NER, CoNLL 2000 chunking and OntoNotes NER tasks, and accuracy for POS tagging task on WSJ. In order to enhance the fairness of the comparisons and verify the solidity of our improvement, we rerun 5 times with different

| Index & Model | Type | Value (± std) |
|---------------|------|---------------|
| Collobert et al., 2011 [6] | reported | 91.26 ± 0.21 |
| Yang et al., 2017 [19] | reported | 91.33 ± 0.08 |
| Peters et al., 2017 [22] | reported | 93.09 ± 0.12 |

In many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.

| Index & Model | Type | Value (± std) |
|---------------|------|---------------|
| Collobert et al., 2011 [6] | reported | 91.26 ± 0.21 |
| Yang et al., 2017 [19] | reported | 91.33 ± 0.08 |
| Peters et al., 2017 [22] | reported | 93.09 ± 0.12 |

In many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.

| Index & Model | Type | Value (± std) |
|---------------|------|---------------|
| Collobert et al., 2011 [6] | reported | 91.26 ± 0.21 |
| Yang et al., 2017 [19] | reported | 91.33 ± 0.08 |
| Peters et al., 2017 [22] | reported | 93.09 ± 0.12 |

In many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.

| Index & Model | Type | Value (± std) |
|---------------|------|---------------|
| Collobert et al., 2011 [6] | reported | 91.26 ± 0.21 |
| Yang et al., 2017 [19] | reported | 91.33 ± 0.08 |
| Peters et al., 2017 [22] | reported | 93.09 ± 0.12 |

In many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.

| Index & Model | Type | Value (± std) |
|---------------|------|---------------|
| Collobert et al., 2011 [6] | reported | 91.26 ± 0.21 |
| Yang et al., 2017 [19] | reported | 91.33 ± 0.08 |
| Peters et al., 2017 [22] | reported | 93.09 ± 0.12 |

In many previous work. The results for these four tasks are given in Table II, Table III, Table IV, and Table V, respectively.
Note we do not compare all of models listed in Table II and Table [V] as such methods (with *) utilize external knowledge excluding in the setting of training set, like character type and lexicon features [8], shared information learned from other tasks [19], other language models pre-trained from large unlabeled corpus [36].

Specifically, among the models listed in these tables, Collobert et al. [6] employ a simple feed-forward neural network with a fixed-size window for context feature extraction, and adopt CRF method for jointly label decoding; Huang et al. [17] introduce a Bi-LSTM-CRF model and outperform [6] by 0.51% and 0.14% on the dataset of CoNLL03 NER and CoNLL00 chunking, respectively, since Bi-LSTM has good characteristics in modeling sequential data and can better capture contextual information than window-based feed-forward neural network; Lample et al. [3] utilize the same architecture as baseline and further apply a LSTM layer to extract character level features of words, which outperform [17] by 0.84% for CoNLL03 NER task; Similarly, Ma and Hovy [11] achieve a significant improvement of 1.11% over [17] on CoNLL03 NER dataset by equipping the Bi-LSTM-CRF model with a CNN layer to obtain character-level representations of words, which indicates the importance of exploiting useful intra-word information, and their proposed model also becomes a popular baseline for most subsequent work in this field; Zhang et al. [34] propose a method called Multi-Order BiLSTM which combines low order and high order LSTMs together in order to learn more tag dependencies, and this method outperforms [17] by 0.6% and 0.55% on the dataset of CoNLL03 NER and CoNLL00 chunking, however, it yields a worse performance than [11]; Zhang et al. [18] propose a multi-channel model that performs better than [11] with a slight improvement of 0.01% and 0.04% on CoNLL03 NER and WSJ dataset, which takes the long range tag dependencies into consideration by incorporating a tag LSTM in their model; Liu et al. [2] incorporate character-aware neural language models into the Bi-LSTM-CRF model and outperform [1] by 0.02% on CoNLL03 NER task, but fail to achieve a better performance for POS tagging.

Note the results show that our proposed model outperforms Bi-LSTM-CRF model by 0.32%, 0.08%, 0.17% and 0.48% for the dataset of CoNLL03 NER, WSJ POS tagging, CoNLL00 chunking and OntoNotes 5.0, respectively, which could be viewed as significant improvements in the filed of sequence labeling. Even compared with the top-performance popular baseline [11], our model achieves a much better result for both NER and POS tagging tasks than other top-conference work in recent two years [18], with an improvement of 0.12% and 0.04%, respectively. Besides, the std (Standard Deviation) value of our model is smaller than the one of Bi-LSTM-CRF, which demonstrates our proposed method is more robust. We also observe that our model consistently outperforms all these baselines for different tasks. Because such models mostly adopt Bi-LSTM as their context encoder architecture, which cannot directly induce the relations among two words, and thus omit modeling part of context dependency especially some discrete patterns. By proposing a novel position-aware self-attention and incorporating self-attentional context fusion layer into the neural architecture, our proposed model is capable of extracting the sufficient latent relationship among words, thus can provide the complementary context information on the basis of Bi-LSTM.

2) Ablation Study: In this section, we run experiments on the CoNLL 2003 NER dataset to dissect the relative impact of each modeling decision by ablation studies.

For better understanding the effectiveness of our proposed position-aware self-attention in our model, we evaluate the performance of various position modeling strategies. Training process is performed 5 times, and then the average F1-scores are reported in Table VI. Note that Model 3 is our final proposed architecture. Model 1 remains the same as Model 3 except that it minus $\Psi_{ij}(\hat{x}_i)$ in Eq [6] which suggests there exists no position information within self-attention. Model 2 applies an absolute position encoding before context encoder layer on the basis of Model 1, which is the position modeling strategy adopted by Vaswani et al. [11] in the Transformer model. Comparing Model 1 with Model 3, we can see that after removing the proposed positional bias $\Psi_{ij}(\hat{x}_i)$ the performance decreases a lot, indicating that our proposed flexible extension of the self-attention achieves a significant improvement since it effectively explores the positional information of an input sequence. But Model 2 with absolute position encoding yields worse performance than Model 1. We conjecture that it is because the absolute position embedding might weaken model’s ability to fusion context features in our architecture.

In order to better understand the working mechanism of our proposed position-aware self-attention, we further analysis the influence of three different positional factors incorporated in it. One of the three factors is removed from proposed positional bias function (Eq 4) each time and the results are shown in Table VII. We can clearly see that the final proposed model including all three factors achieves the best performance and ablating any one bias contributes to a worse score. It

| No | Model | F1-score±std |
|----|-------|--------------|
| 1  | w/o $\Psi_{ij}(\hat{x}_i)$ in Eq [6] | 91.15±0.12 |
| 2  | add position encoding | 91.05±0.19 |
| 3  | Our model | 91.33±0.08 |

| M_{ij}(x_i) | P_{ij}(x_i) | G_{ij}(x_i) | F1-score±std |
|-------------|-------------|-------------|--------------|
| √           | √           | √           | 91.12±0.21   |
| √           | ×           | √           | 91.07±0.05   |
| √           | √           | ×           | 91.19±0.24   |
| √           | √           | √           | 91.33±0.08   |

| First layer | Second layer | F1-score±std |
|-------------|--------------|--------------|
| ×           | ×            | 91.01±0.21   |
| ×           | √            | 91.13±0.17   |
| √           | ×            | 91.27±0.05   |
| √           | √            | 91.33±0.08   |
Table IX: Experimental results for adjusting the architecture of the proposed model (SAN denotes the proposed self-attentional context fusion network).

| Num of layers | Position modeling strategy | F1-score ± std |
|---------------|-----------------------------|----------------|
| 1             | Absolute position embedding | 88.7 ± 0.23    |
| 2             | Absolute position embedding | 88.79 ± 0.39   |
| 2             | Proposed positional bias    | 90.7 ± 0.15    |
| 3             | Absolute position embedding | 88.57 ± 0.23   |
| 3             | Proposed positional bias    | 90.6 ± 0.12    |

Table X: Experimental results of the Transformer model.

![Fig. 4: Performances on Different Lengths.](image)

3) Performances on Different Lengths: We further analyze the performance of different models with respect to the different length of sentences. In Figure 4, we compare Bi-LSTM-CRF baseline and our proposed model on different sentence lengths. For NER and chunking, our model significantly outperforms Bi-LSTM-CRF on short sentences (sentence length less than 5 on CoNLL2003, length less than 10 on CoNLL2000 and OntoNotes), which indicates that the improvement of the proposed model on short sentences is much larger than those on long sentences. The discrete context dependencies with short distances in a sequence are captured very well by our proposed model but simply neglected by

![Graph showing performance on different lengths.](image)

The Transformer model [11] which is based on self-attention mechanism has been proven to have strong capabilities for feature extraction. We evaluate the transformer with different numbers of layers on CoNLL03 NER task, and the result is given in Table IX. In the experiment we adopt the transformer as the context encoder architecture and remain the distributed representations and tag decoder part of our model. We conjecture that it’s because the transformer model may be sensitive to the hyper-parameters for different sequence labeling tasks, since there are lots of hyper-parameters like dimension of keys/queries/values, dimension of attention model, dimension of inner-layer, number of heads and etc. As for the setting of this experiment, the parameters are set to 64, 512, 1024 and 8, respectively. However, it’s obvious that changing the position modeling strategy leads to a great improvement to the results, which further demonstrate the effectiveness of our proposed method to explore position information for sequence labeling tasks.
This paper proposes an innovative neural architecture for sequence labeling tasks, in which a self-attentional context fusion layer is designed and incorporated to better model discrete and discontinuous context patterns of sequence. The strengths of our work are that we identify the problem of modeling discrete context dependencies in sequence labeling tasks, and a position-aware self-attention is proposed to induce the latent independent relations among tokens over the input method for comparison. Table X] shows four cases that our model predicts correctly but Bi-LSTM-CRF doesn’t. For better comparison, we visualize the alignment score by heat-maps of words that baseline model fails to predict their labels correctly.

In the first case, the POS tag of “lower” should be tagged with adjective comparative (RBR), while Bi-LSTM-CRF recognizes it as adverb comparative (JJR). It’s obvious that the tag of “lower” is dependent on the 3rd word “open”, where an adverb is associated with a verb, and the 4th word “sharply” is a direct modifier of it. Figure 6(a) shows that for word “lower” it pays more attention on “opened” and “sharply”, while less on other words. Similar situation is shown in the second case, where our model assigns correct POS tag to “higher” which depends largely on its previous word “moved” but Bi-LSTM-CRF fails. Regarding the third case, our model succeeds in assigning verb past participle (VBN) to word “hit” by considering “been” and “hard” while Bi-LSTM-CRF makes a wrong decision. The consistent conclusion is also reflected in Figure 6(c) that “been” and “hard” obtain large attention from the focus word “hit”. And our model predict the POS tag of “upset” correctly in the fourth case which can be speculated from the common phrase “have been done by”.

Our analysis suggests if the choice of assigning label to a specified token \( x_i \) depends on several other words, they will receive a large amount of attention scores from \( x_i \), which also provides a high level interpretability for our self-attentional model.

VI. CONCLUSIONS

This paper proposes a innovative neural architecture for sequence labeling tasks, in which a self-attentional context fusion layer is designed and incorporated to better model discrete and discontinuous context patterns of sequence. The strengths of our work are that we identify the problem of modeling discrete context dependencies in sequence labeling tasks, and a position-aware self-attention is proposed to induce the latent independent relations among tokens over the input

---

**Fig. 5:** Performance of our model with various window sizes.

**Fig. 6:** Heatmaps of four cases.

---

Bi-LSTM-CRF. For POS tagging, performances of the two models on different sentence lengths are relatively comparable, while in the range of (20;40) our model performs slightly better than Bi-LSTM-CRF.

4) Impact of Window Size: The window size \( k \) (cf Section V-A2) is clearly a hyperparameter which must be optimized for, thus we investigate the influence of the value of \( k \) on the CoNLL2003 NER task. We also rerun 5 times with different random initialization and report the average score, which is consistent with our other experiments in this paper. The plot in Figure 5 shows that when assigning the value of \( k \) to 10 we do outperform other models substantially. And with other window sizes (except 2) our model performs relatively well and is superior to the Bi-LSTM-CRF baseline (91.01%), which also suggests the effectiveness of our proposed distance-aware Gaussian bias to favor the local context dependencies of sequence.

5) Efficiency: We implement our model based on the PyTorch library. Models have been trained on one GeForce GTX 1080 GPU, with training time recorded in Table X. In terms of efficiency, our model only introduces a small number of parameters in two self-attention layer, which may not have a very large impact on efficiency. And it can be drawn from Table X that the training speed of our model is only 13% lower than the baseline, but bring a significant improvement in the performance.

6) Qualitative Analysis: The weight \( \lambda \) in fusion gate mechanism actually indicates the balance between the feature of each token and its context representation obtained by self-attention. If \( \lambda \) is bigger than 0.5, the final contextual representation relies more on its own feature, otherwise, the context representation play a more important role. As shown in Table X, \( \lambda \) varies along the sentence, showing the effectiveness of both feature. Besides, we can observe that for most tokens, \( \lambda_1 \) is smaller than \( \lambda_2 \), which indicates that the first self-attentional layer before Bi-LSTM incorporates more useful contextual information.

7) Case Study: In this section, we present an in-depth analysis of results given by our proposed approach for better understanding the influence of self-attention mechanism in our proposed model. Without loss of generality, we take POS tagging as the task and Bi-LSTM-CRF as the comparison
The market opened sharply lower, with the Nikkei average down nearly 600 after 20 minutes.

The dollar also moved higher in Tokyo.

But the rally was confined to the stocks, which had been hard hit during Friday’s selling frenzy.

Spending patterns in newspapers have been upset by shifts in ownership and general hardships.

TABLE XI: Examples of the predictions of Bi-LSTM-CRF baseline and our model.

| Model       | F1-score±std | speed  | time  |
|-------------|--------------|--------|-------|
| Bi-LSTM-CRF | 91.01±0.21   | 23 iter/s | 1.6 h |
| Our Model   | 91.33±0.08   | 20 iter/s | 1.9 h |

TABLE XII: Training speed, training time and performance of Bi-LSTM-CRF baseline and our proposed model on CoNLL 2003 NER task. N iter/s means processing N iterations per second.

TABLE XIII: Qualitative analysis of learned parameter λ (cf Eq [14]). λ1 and λ2 denote λ in the first and second self-attentional context fusion layer, respectively. Since λ is a multi-dimensional vector, here we take its average in various dimensions to facilitate the observation.

sequence via three different bias, which can effectively model the context dependencies of given sequence according to the relative distance among tokens. Experimental results on part-of-speech (POS) tagging, named entity recognition (NER) and phrase chunking tasks demonstrate the effectiveness of our proposed model which achieves state-of-the-art performance. Furthermore, our analysis reveals the effects of each modeling decision from different perspectives. The way we model the discrete context dependencies of sentences in sequence labeling tasks can also inspire other researchers in the field to innovate from this perspective. Despite the good performance, our work still has weaknesses, which is reflected in the limited improvement of our model for longer sequences. The main reason is that the second positional bias that we introduced tends to let self-attention learn the influence of neighboring words in the sequence. In the future, we plan to further apply our neural architecture to data from other domains such as social media and empower more sequence labeling tasks. Additionally, we also plan to employ our model to other sequence learning tasks besides sequence labeling, such as event extraction and neural machine translation. More recently, pretrained language models from huge corpus are widely adopted to enhance the representation of words. We will in the future explore integrating language modeling into this architecture to further boosting performance.

REFERENCES

[1] X. Ma and E. Hovy, “End-to-end sequence labeling via bi-directional lstm-cnn-crf,” in ACL, 2016.
[2] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, “Empower sequence labeling with task-aware neural language model,” in AAAI, 2018.
[3] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, “Neural architectures for named entity recognition,” in NAACL, 2016.
[4] L. E. Baum and T. Petrie, “Statistical inference for probabilistic functions of finite state markov chains;” The annals of mathematical statistics, pp. 1554–1563, 1966.
[5] J. Lafferty, A. McCallum, and F. C. Pereira, “Conditional random fields: Probabilistic models for segmenting and labeling sequence data,” in ICML, 2001.
[6] R. Collobert, K. Kavukcuoglu, J. Weston, L. Bottou, P. Kuksa, and M. Karlen, “Natural language processing (almost) from scratch,” Journal of Machine Learning Research, vol. 12, no. 1, pp. 2493–2537, 2011.
[7] S. Youssi, S.-A. Berrani, and C. Garcia, “Contribution of recurrent connectionist language models in improving lstm-based arabic text recognition in videos,” Pattern Recognition, vol. 64, pp. 245–254, 2017.
[8] J. P. Chiu and E. Nichols, “Named entity recognition with bidirectional lstm-cnns;” TACL, vol. 4, pp. 357–370, 2016.
[9] H.-g. Lee, G. Park, and H. Kim, “Effective integration of morphological analysis and named entity recognition based on a recurrent neural network,” Pattern Recognition Letters, vol. 112, pp. 361–365, 2018.
[10] S. Kwon, Y. Ko, and J. Seo, “Effective vector representation for the korean named-entity recognition,” Pattern Recognition Letters, vol. 117, pp. 52–57, 2019.
[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in NIPS, 2017, pp. 5998–6008.
[12] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” in NAACL, 2018.
[13] L. R. Rahier, “A tutorial on hidden markov models and selected applications in speech recognition;” Proceedings of the IEEE, vol. 77, no. 2, pp. 257–286, 1989.
[14] A. McCallum and W. Li, “Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons,” in CoNLL, 2003.
