Adaptive Threshold Selective Self-Attention for Chinese NER

Biao Hu[1], **Zhen Huang**[1*], **Minghao Hu**[2*], **Ziwen Zhang**[1], **Yong Dou**[1]
[1]College of Computer, National University of Defense Technology, Changsha, China
[2]Information Research Center of Military Science,
PLA Academy of Military Science, Beijing, China
{hubiao, huangzhen, ziwen, yongdou}@nudt.edu.cn
huminghao16@gmail.com

Abstract

Recently, Transformer has achieved great success in Chinese named entity recognition (NER) owing to its good parallelism and ability to model long-range dependencies, which utilizes self-attention to encode context. However, the fully connected way of self-attention may scatter the attention distribution and allow some irrelevant character information to be integrated, leading to entity boundaries being misidentified. In this paper, we propose a data-driven Adaptive Threshold Selective Self-Attention (ATSSA) mechanism that aims to dynamically select the most relevant characters to enhance the Transformer architecture for Chinese NER. In ATSSA, the attention score threshold of each query is automatically generated, and characters with attention score higher than the threshold are selected by the query while others are discarded, so as to address irrelevant attention integration. Experiments on four benchmark Chinese NER datasets show that the proposed ATSSA brings 1.68 average F1 score improvements to the baseline model and achieves state-of-the-art performance.

1 Introduction

Named Entity Recognition (NER) aims to identify named entities in the given text, including persons, locations, organizations, etc. It plays an important role in downstream natural language processing (NLP) tasks such as information retrieval (Chen et al., 2015), relation extraction (Miwa and Bansal, 2016) and question answering (Diefenbach et al., 2018). Compared with English NER, Chinese NER is more difficult since there is no natural delimiter between words in Chinese sentences.

To integrate word information related to each character and avoid error propagation of word segmentation, word-character lattice is first applied to Chinese NER in Lattice-LSTM (Zhang and Yang, 2018). However, the RNN-based model is hard to model long-range dependencies, and has poor computational efficiency. Recently, Transformer (Vaswani et al., 2017) is widely used in Chinese NER, which utilizes self-attention to encode context. Since vanilla Transformer lacks directionality, TENER (Yan et al., 2019) combines Transformer encoder with direction-aware position encoding to catch directions between characters. FLAT (Li et al., 2020) further exploits span relative position encoding to convert the lattice structure into a flat structure.

Figure 1: A comparison of selective self-attention and fully connected self-attention, where the attention weights are indicated by blue color. We only show the attention distribution of the character "上 (Shang)" in the head that focuses on entities.

Although Transformer-based models perform well on Chinese NER, they have a limitation that the fully connected self-attention drives queries to attend to all characters of the inputs, making irrelevant character-level attention integration, which leads to entity boundaries being misidentified. We argue that, in NER task, entities only depend on the most relevant characters, while others should be ignored. As shown in Figure 1, since the attention of the character "上 (Shang)" attends to all characters, irrelevant character-level information of "大 (Big)" and "众 (Zhong)" is integrated, which may lead the model to misidentify "上海大众 (Shanghai Volkswagen)" as an organization entity rather than "上海 (Shanghai)" as a geopolitical entity.

To avoid irrelevant attention integration, researchers have investigated sparse self-attention. Child et al. (2019) introduce local and strided patterns to select keys for queries. Since the sparsity pattern is fixed, it may lack transferability.

*Corresponding author

Proceedings of the 29th International Conference on Computational Linguistics, pages 1823–1833
October 12–17, 2022.

As alternatives, sparsemax (Martins and Astudillo, 2016) and entmax (Peters et al., 2019) collect most relevant keys via dedicated forward and backward algorithms, which incurs large computational costs. To improve efficiency, explicit sparse Transformer (Zhao et al., 2019) gathers a fixed number of keys with highest attention score through top-k selection at the cost of flexibility. However, none of above works have been applied to Chinese NER.

In this paper, we propose an Adaptive Threshold Selective Self-Attention (ATSSA) mechanism to avoid irrelevant character-level attention integration for Chinese NER. This mechanism is data-driven which assigns a various attention mass to each query in different heads parallelly. Specifically, we generate an attention score threshold for each query of input sequence via a matching matrix (Wang et al., 2018) automatically. Then, keys whose attention score exceeds the threshold are selected, while others are discarded. In the end, the softmax function only performs on those selected query-key pairs, thereby improving attention to relevant keys while avoiding irrelevant character-level attention integration. As shown in Figure 1, the character "上(Shang)" only attends to relevant characters in "上海(Shanghai)", while attention to irrelevant characters in "大众(Public)" is discarded.

We take the advantage of FLAT (Li et al., 2020) in lexicon fusion and apply ATSSA to it. Our code will be released at `https://github.com/hubiao20/atssa-ner`. The contributions of this paper can be summarized as:

- We propose Adaptive Threshold Selective Self-Attention to avoid irrelevant character-level attention integration for Chinese NER.
- We conduct experiments on four benchmark Chinese NER datasets, experimental results show that ATSSA brings 1.68 average F1 score improvements to the baseline model and achieves state-of-the-art performance.

2 Related Work

Chinese NER with Lattice Structure Since lattice structure can provide rich lexical semantics and boundary information, lattice-based approaches have become the mainstream of Chinese NER. Lattice LSTM (Zhang and Yang, 2018) first proposed a lattice structure to encode all characters and matched words simultaneously. Gui et al. (2019a) combined CNN and rethinking mechanism to encode characters and potential words at different window sizes. Ma et al. (2020) generated the soft-lexicon feature by a static method, simplifying the structure of lattice. The backbone of these methods are RNNs, which are hard to model long-range dependencies. As a solution, models based on GNN (Gui et al., 2019b; Sui et al., 2019; Ding et al., 2019) are proposed. Recently, Transformer has achieved state-of-the-art performance on Chinese NER. FLAT (Li et al., 2020) integrated lattice structure via an ingenious span position encoding. Moreover, to capture fine-grained correlations in word-character spaces, DCSAN (Zhao et al., 2021) leveraged a cross-lattice attention to model dense interactions over lattice structure. However, the fully connected self-attention allows Transformer-based models to integrate some redundant information, which affects the performance on Chinese NER. Compared with traditional self-attention, our proposed ATSSA selectively activates keys and avoids incorrect character-level information fusion.

Selective Self-Attention In many NLP tasks, predictions of models only depend on a small part of the inputs. Many works based on Transformer employed sparse self-attention to discard attention to irrelevant keys. Some data-driven approaches, such as sparsemax (Martins and Astudillo, 2016) and entmax (Peters et al., 2019; Correia et al., 2019) selected keys iteratively, which have large computational costs. To keep computational efficiency, others (Raganato et al., 2020; Child et al., 2019) defined sparse patterns manually. However, the transferability of these methods has not been validated. Besides, Zhao et al. (2019) proposed explicit sparse Transformer which collects a fixed number of keys with highest attention score empirically. Unlike sparse self-attention based models, selective self-attention used an additional controller to select keys dynamically. ReSAN (Shen et al., 2018) used two RSS (Reinforced Sequence Sampling) modules to select keys that need to attend to. GA-Net (Xue et al., 2020) utilized an auxiliary network to generate binary gates to select elements for the backbone attention network. Inspired by the selective idea, we propose the adaptive threshold selective self-attention, which dynamically selects keys by an automatically generated attention threshold.

3 Background

Our proposed ATSSA is applied to FLAT (Li et al., 2020), thus, we briefly introduce the structure of

FLAT in this section. FLAT treats the characters and matched words of the input sequence as spans, and designs an ingenious relative position encoding of spans for lattice structure. Let $head[i]$ and $tail[i]$ denote the head and tail position of span x_i, four relative distances are used to indicate the relation between x_i and x_j:

$$\begin{aligned} d_{ij}^{hh} &= head[i] - head[j] \\ d_{ij}^{ht} &= head[i] - tail[j] \\ d_{ij}^{th} &= tail[i] - head[j] \\ d_{ij}^{tt} &= tail[i] - tail[j] \end{aligned} \quad (1)$$

Then, the span relative position encoding can be calculated by a non-linear transformation of the four distances:

$$R_{ij} = \mathrm{ReLU}(W_r[p_{d_{ij}^{hh}}; p_{d_{ij}^{ht}}; p_{d_{ij}^{th}}; p_{d_{ij}^{tt}}]) \quad (2)$$

where W_r is a learnable parameter, p_d is the sine and cosine functions explained as:

$$\begin{aligned} p_{(d,2i)} &= \sin(d/10000^{2i/d_{model}}) \\ p_{(d,2i+1)} &= \cos(d/10000^{2i/d_{model}}) \end{aligned} \quad (3)$$

where d is the distance acquired by Eq.(1) and i is the index of dimension, $d_{model} = N \times d_{head}$, N is the head number and d_{head} is the dimension of each head.

With the span relative position encoding, the dot-product attention can be calculated as:

$$Attn = \mathrm{softmax}\,(A)\,V \quad (4)$$

$$A_{ij} = Q_i K_j^T + Q_i R_{ij}^T + u K_j^T + v R_{ij}^T \quad (5)$$

$$Q, K, V = EW_q, EW_k, EW_v \quad (6)$$

where E is the embedding of each span. $W_q, W_k, W_v \in \mathbb{R}^{d \times d_{head}}$, $u, v \in \mathbb{R}^{d_{head}}$ are learnable parameters.

4 Adaptive Threshold Selective Self-Attention

Intuitively, keys with high attention score are more relevant than others to the correspondin query and attention should be centralized on these keys. However, in the vanilla Transformer, fully connected self-attention distributes attention to the entire context, even those irrelevant keys, which weakens the attention to critical ones. To this end, we propose the Adaptive Threshold Selective Self-Attention, in which keys with attention score lower than threshold of the corresponding query are discarded before the softmax operation.

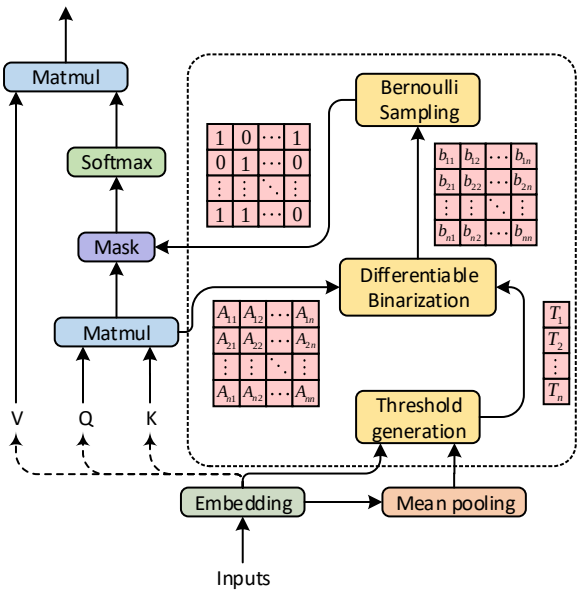


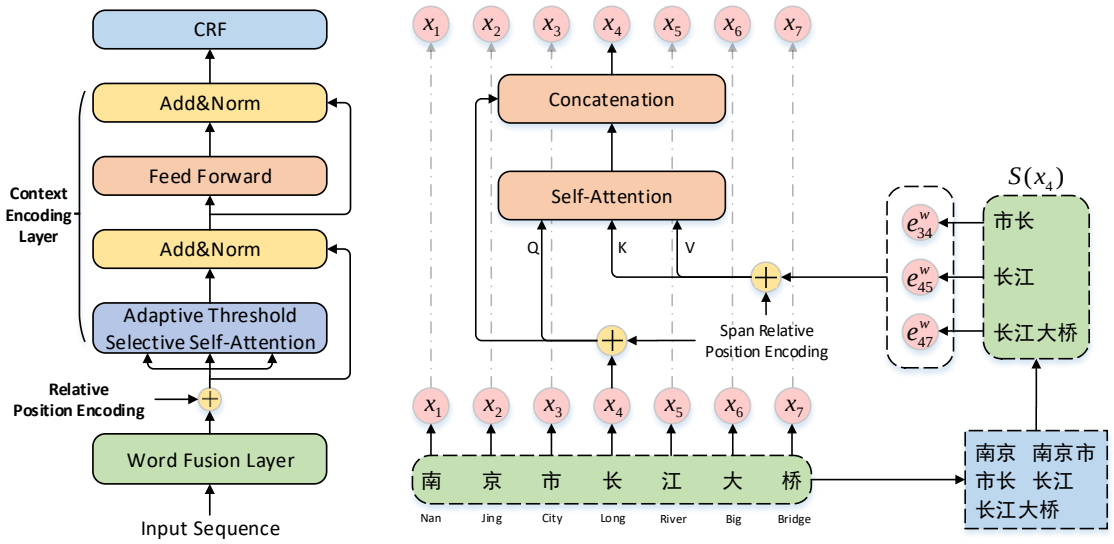
Figure 2: Structure of the proposed adaptive threshold selective self-attention. The dotted box area illustrates the main difference between ATSSA and fully connected self-attention. Threshold is generated by a matching matrix and used for differentiable binarization. Then, the binary matrix is sampled by Bernoulli distribution and exploited to select keys for each query.

4.1 Structure of ATSSA

In this section, we introduce the proposed ATSSA in details, as shown in Figure 2. The goal of the mechanism is to dynamically collect a subset of keys to attend to through an adaptive obtained threshold. Given an input sequence $\mathrm{x} = [x_1, x_2, \cdots, x_n]$, the attention score matrix A can be calculated by dot production of Q and K. Different with the self-attention in vanilla Transformer, the proposed mechanism additionally utilizes an automatically generated threshold $\mathrm{T} = [T_1, T_2, \cdots, T_n]$ to selectively activate elements in A. For each element, $A_{ij} \geq T_i$ means A_{ij} is activated, that is, key K_j is selected by query Q_i.

The threshold acts as an controller to determine whether the information of keys should flow to the target, which is the crux of our proposed mechanism. We argue that each query has a various attention score threshold since its relevance to the input sequence varies. To generate the threshold, we introduce a matching matrix (Wang et al., 2018). The matrix models the correlation between x_i and the entire sequence, and threshold of query Q_i is generated by applying a linear projection to it:

$$T_i = W_t[x_i; \mathrm{mp}(x); x_i \odot \mathrm{mp}(x); x_i - \mathrm{mp}(x)] \quad (7)$$



(a) Overall structure (b) Word Fusion Layer

Figure 3: (a) The overall structure of the proposed model. Word information is fused to generate the representation of each character by a word fusion layer. Context is then encoded with the ATSSA based Transformer. (b) Illustration of the word fusion layer, in which word representations only depend on matched words.

where W_t is a learnable parameter, $\odot$ denotes the element-wise product, and $\mathrm{mp}(\cdot)$ denotes the mean pooling operation.

With threshold selection, keys with attention score above or equal to the threshold are kept, while others are dropped. In the process of model training, we discover that the threshold of some queries is higher than the attention score of all keys in the initial stage, leading to the training error of the model. To prevent the loss of critical information, top-k (Zhao et al., 2019) is introduced to force the model to keep at least k keys with largest attention score. Therefore, the threshold of query Q_i can be calculated as:

$$\tilde{T}_i = \min(topk_i, T_i) \tag{8}$$

where $topk_i$ is the k-th largest value at row i in A.

The essence of threshold selection is to perform a binarization operation on attention score matrix A. We employ a binary matrix z to indicate whether the element in matrix A is selected, where $z_{ij} = 1$ means that K_j is selected by query Q_i whereas $z_{ij} = 0$ indicates that K_j is discarded by query Q_i. Since the binary function is not differentiable, we use an approximate step function (Liao et al., 2020) for binarization:

$$b_{ij} = \frac{1}{1 + e^{-\alpha(A_{ij} - \tilde{T}_i)}} \tag{9}$$

where α is amplification factor larger than 1. The approximate step function is similar to the binary function and has differentiability, which enables the model to be optimized during training. b_{ij} is a value in interval $[0, 1]$, not the 0 or 1 we need. To address this issue, we use it to parameterize a Bernoulli distribution, and then, z_{ij} is a sample generated from the Bernoulli distribution:

$$z_{ij} \sim \mathrm{Bernoulli}(b_{ij}) \tag{10}$$

As z is sampled from Bernoulli distribution, our attention strategy can be defined as:

$$Attn = \mathrm{softmax}(A')V \tag{11}$$

$$A'_{ij} = \begin{cases} A_{ij}, & z_{ij} = 1 \\ -\infty, & z_{ij} = 0 \end{cases} \tag{12}$$

4.2 Chinese NER with ATSSA

To adapt ATSSA for Chinese NER, we fine-turn the structure of FLAT. As shown in Figure 3, the context encoder is divided into two layers, consisting of a word-level word fusion layer and a character-level context encoding layer.

Word Fusion Layer For the input sequence x and a given dictionary D, all potential words in x can be identified by matching the sequence with D. Each matched word $w_{jk}(j \neq k)$ starts with the j-th character and ends with the k-th character. Position tags are employed to construct the word set S corresponding to the character x_i:

$$S(x_i) = \{w_{jk} | \forall w_{jk} \in D, j \leq i \leq k\} \tag{13}$$

Take the sequence given in Figure 3(b) as an example, the word set corresponding to the character "长 (Long)" is represented as [市长 (Mayor), 长江 (Yangtze River), 长江大桥 (Yangtze River Bridge)]. For characters with no matched word, word sets corresponding to them are empty, and the word representations of these characters are None.

As the word sets constructed, a character-word cross attention is exploited to generate word representations. Let $head[\cdot]$ and $tail[\cdot]$ denote the head and tail position of the matched word w_{jk}, the relative distances in FLAT can be modified as:

$$\begin{aligned} d_{i,jk}^{hh} &= i - head[w_{jk}] \\ d_{i,jk}^{tt} &= i - tail[w_{jk}] \end{aligned} \tag{14}$$

where $w_{jk} \in S(x_i)$. The relative position encoding of the characters and matched words can be calculated by a simple linear transformation of the two distances:

$$R_{i,jk} = W_r[p_{d_{i,jk}^{hh}}; p_{d_{i,jk}^{tt}}] \tag{15}$$

where W_r is a learnable parameter.

Given characters as queries, words as keys and values, word representation x_i^w of each character can be obtained by self-attention. Finally, x_i^w is concatenated with character x_i as the input of context encoding layer: $x_i = [x_i; x_i^w]$.

Context Encoding Layer In this layer, self-attention based on relative position encoding is introduced. For the input sequence x fused with word information, relative position encoding of each query-key pair can be indicated as $R_{ij} = p_{d_{ij}}$, where $d_{ij} = i - j$, i and j denote the position of query and key respectively. Then, the attention matrix A is generated by Eq.(5). Applying our proposed ATSSA to A, attention of each query is centralized on those critical keys. The following calculation is the same with FLAT.

4.3 Training and Decoding

Since the binary matrix z is sampled by Bernoulli distribution, model cannot be backpropagated during training. Gumbel-Softmax (Jang et al., 2017) provides a reparameterization solution. In Eq.(10), b_{ij} denotes the probability that $z_{ij} = 1$, let $b_{ij}^{(1)} = b_{ij}$ and $b_{ij}^{(0)} = 1 - b_{ij}$, z_{ij} can be expressed as:

$$z_{ij} = \arg\max_k b_{ij}^{(k)}, k = 0, 1 \tag{16}$$

The argmax operation is not differentiable, therefore, we substitute Gumbel-Softmax distribution for Bernoulli distribution to acquire z_{ij} in the training stage. The Gumbel-Softmax distribution makes a softmax approximation to b_{ij} with the following continuous and differentiable calculation:

$$\tilde{b}_{ij}^{(k)} = \frac{\exp((\log(b_{ij}^{(k)}) + g_k)/\tau)}{\sum_{l=0}^{1} \exp((\log(b_{ij}^{(k)}) + g_l)/\tau)} \tag{17}$$

where g_l is a random sample from $\mathrm{Gumbel}(0, 1)$, and τ is a hyperparameter called temperature.

To make use of the dependencies between labels, CRF (Lafferty et al., 2001) is used to predict entity labels. Given the label sequence $y = [y_1, y_2, \cdots, y_n]$ and output $H = [h_1, h_2, \cdots, h_n]$ of the fine-turned model, the probability of the ground-truth label sequence can be calculated as:

$$p(y|\mathrm{x}) = \frac{\exp(\sum_{i=1}^{n} \varphi(y_{i-1}, y_i, \mathrm{x}))}{\sum_{y' \in Y(\mathrm{x})} \exp(\sum_{i=1}^{n} \varphi(y'_{i-1}, y'_i, \mathrm{x}))} \tag{18}$$

where $Y(\mathrm{x})$ is the set of all arbitrary label sequences, $\varphi(y_{i-1}, y_i, \mathrm{x}) = W_{(y_{i-1}, y_i)} h_i + b_{(y_{i-1}, y_i)}$, $W_{(y_{i-1}, y_i)}$ and $b_{(y_{i-1}, y_i)}$ are parameters specific to y_{i-1} and y_i. Therefore, loss function is defined as:

$$\mathcal{L} = -\sum_{i=1}^{n} \log\left(p\left(y_i|\mathrm{x}\right)\right) + \frac{\lambda \|z\|_1}{L} \tag{19}$$

The first term in loss function is negative log-likelihood loss, and the second term is a l_1 norm regularizer over z. λ is a trade-off between the two terms, and L is the length of input sequence.

5 Experiments

Experiments are carried out on Chinese NER datasets across different domains. F1-score (F1) is exploited to evaluate the performance of the model. All experiments are conducted on a single Nvidia Titan RTX GPU.

5.1 Experimental Setup

Datasets We conduct experiments on four datsets, including Weibo NER (Peng and Dredze, 2015; He and Sun, 2017), Resume NER (Zhang and Yang, 2018), OntoNotes (Weischedel et al., 2011), and MSRA (Levow, 2006). Weibo NER is drawn from Sina Weibo[1], and Resume NER is collected from Sina Finance[2], while OntoNotes and MSRA are in news domain. Statistics of the above datasets are shown in Table 1.

[1]https://www.weibo.com/

[2]https://finance.sina.com.cn/stock/

Datasets	Type	Train	Dev	Test
Weibo	Sentence	1.4k	0.27k	0.27k
	Char	73.8k	14.5k	14.8k
Resume	Sentence	3.8k	0.46k	0.48k
	Char	124.1k	13.9k	15.1k
OntoNotes	Sentence	15.7k	4.3k	4.3k
	Char	491.9k	200.5k	208.1k
MSRA	Sentence	46.4k	-	4.4k
	Char	2169.9k	-	172.6k

Table 1: Statistics of datasets

Hyperparameter	Range
learning rate	[1e-3,8e-4,6e-4]
-decay	0.05
head	[4,8,12]
head dimension	[8,10,12]
FFN size	[4,6,8]×head×head dimension
warmup	[1,5,10](epoch)

Table 2: Searching range of hyperparameters.

Baseline We take TENER (Yan et al., 2019) and FLAT (Li et al., 2020) as baseline models, which encode context with Transformer encoder.

Implementation Details We use the pre-trained character embedding and bigram embedding trained with word2vec (Mikolov et al., 2013) over automatically segmented Chinese Giga-Word[3]. The BERT in the experiments is BERT-wwm (Cui et al., 2021), and the pre-trained word embedding is released by (Li et al., 2018) that contains about 1.29 million words. The above four embeddings have sizes of 50, 50, 768 and 300 respectively, all of which are fine-tuned during training. For hyperparameter configurations, batch size is set to 8 and SGD with 0.9 momentum is used to optimize the model. To avoid overfitting, dropout is applied to embeddings with a rate of 0.5. When calculating selective self-attention, each query is forced to keep at least 3 keys, i.e. the value of k in top-k is set to 3. The amplification factor α in approximate step function, temperature τ in Gumbel-Softmax and λ in loss fuction are set to 50, 1 and 4×10^{-6} respectively. We use random search to find the optimal values of other hyperparameters, and their ranges are shown in Table 2.

5.2 Overall Performance

We compare our proposed model with the baseline models and other state-of-the-art word-character lattice based methods, results are reported in Table 3. Our model outperforms TENER by 7.39 average F1 score on four datasets, and for FLAT+BERT, the value is 1.68. In particular, the proposed ATSSA brings improvements of 3.98 and 1.49 on Weibo and OntoNotes respectively. Compared with SoftLexicon (Ma et al., 2020) and DCSAN (Zhao et al., 2021), which statically integrates word-level information, and MECT (Wu et al., 2021) with glyph information, our model is still competitive. The results above indicate the effectiveness of ATSSA, and suggest that ATSSA can better encode context at character-level.

Model	Weibo	Resume	Ontonotes	MSRA
Lattice-LSTM[1]	58.79	94.46	73.88	93.18
SoftLexicon*[2]	70.50	96.11	82.81	95.42
MECT*[3]	70.43	95.98	82.57	96.24
DCSAN*[4]	71.27	96.67	-	96.41
TENER[5]	58.39	95.25	72.82	93.01
FLAT*[6]	68.55	95.86	81.82	96.09
FLAT*+ATSSA	**72.53**	**96.73**	**83.31**	**96.45**

Table 3: Results(F1) on four datasets. * denotes the models equipped with BERT. Zhang and Yang (2018)[1], Ma et al. (2020)[2], Wu et al. (2021)[3], Zhao et al. (2021)[4], Yan et al. (2019)[5], Li et al. (2020)[6].

Models	Weibo	Resume	Ontonotes	MSRA
FLAT*+ATSSA	**72.53**	**96.73**	**83.31**	**96.45**
-Selective SA	70.02	95.52	82.13	96.02
-top-k	68.47	96.19	82.60	95.37
-AF	68.67	95.89	82.65	95.51
-DT	68.53	95.70	82.63	95.68

Table 4: An ablation study of the proposed model. SA stands for Self-Attention, and AF and DT indicates amplification factor and dynamical threshold respectively.

5.3 Ablation Study

To investigate the effectiveness of the main components of our proposed ATSSA , we conduct an ablation study on all four datasets. The results are reported in Table 4.

(1) We propose an adaptive threshold selective self-attention to avoid irrelevant character-level attention integration. To investigate the contribution of this mechanism, we replace selective self-attention with global self-attention. The average F1 score on all four datasets is reduced by 1.29, especially by 2.51 on Weibo test set. The decline in performance verifies the significance of our proposed adaptive threshold selective self-attention.

(2) When calculating selective self-attention, top-k is introduced to force the query to reserve atten-

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

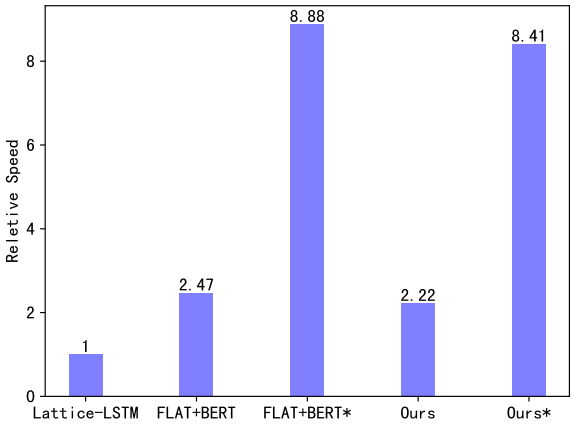


Figure 4: Relative inference speed of each model compared with Lattice-LSTM. The default batch size is 1, * denotes the model is run in 16 batch size.

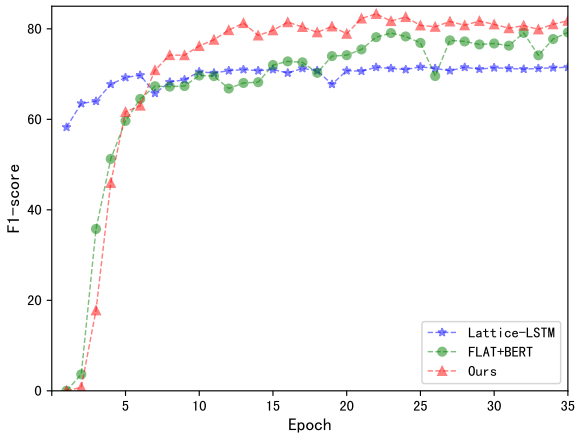


Figure 5: Convergence speed on OntoNotes dataset

Model	Complexity
Vanilla Transformer[1]	$O(n^2)$
Explicit Sparse Transformer[2]	$O(n^2 + nlogn)$
ATSSA	$O(n^2 + 2n + nlogn)$

Table 5: Computational complexity of different methods. Vaswani et al. (2017)[1], Zhao et al. (2019)[2].

tion to a fixed number of keys with largest attention scores. After removing top-k, we discover that the stability of the model training is affected, and F1 scores obtained on the test set are decreased.

(3) The approximate step function is a variant of the sigmoid function, in which an amplification factor makes it more similar to a binary function. Since the use of Bernoulli sampling, the selection of keys with attention score close to the threshold becomes random without the amplification factor. After removing amplification factor, the degradation in performance on all four datasets indicates the importance of amplification factor in approximate step function.

(4) In the computer vision domain, most existing threshold based methods set a fixed threshold, which makes the approach inflexible. Our proposed adaptive threshold selective self-attention assigns a various threshold to each query of all heads in the self-attention that enables the query to select keys dynamically. We empirically replace the dynamic threshold with a fixed threshold 0, and observe that the performance of the model degenerates to be close to FLAT.

5.4 Analysis in Efficiency

Compared with self-attention in vanilla Transformer, the proposed ATSSA has an additional threshold computation, its computational complexity is $O(n^2 + 2n + nlogn)$. The dot products of queries and keys in self-attention correspond to the first term. The threshold calculation corresponds to the second term, that is, the dot product and subtraction of each input character and mean pooling of the entire sequence. The sorting operation in top-k corresponds to the third term. Since the additional term $O(2n + nlogn)$ is overshadowed by the dominant term $O(n^2)$, as shown in Table 4, our proposed ATSSA is slightly slower than FLAT in computational efficiency.

To verify the computational efficiency of ATSSA, we compare the inference speed of Lattice-LSTM, FLAT and our model on Weibo test set, as shown in Figure 4. Lattice-LSTM cannot run in parallel due to the use of directed acyclic graph. As we can see, even with an additional threshold generation operation, our model is only about 5 percent slower in inference speed than FLAT.

To further explore the convergence speed of ATSSA, we conduct experiments on OntoNotes dataset. Figure 5 illustrates the F1 scores of Lattice-LSTM, FLAT and our model relative to the number of training iterations. We observe that the performance of our model is lower than Lattice-LSTM and FLAT in the initial stages of model training. As the number of iterations increases, our model converges faster than FLAT. This is likely because our model is less disturbed by irrelevant character information than FLAT.

5.5 Influence of Hyperparameters

In our proposed ATSSA, k in top-k operation and amplification factor α are two important hyperparameters that affect the performance of Chinese

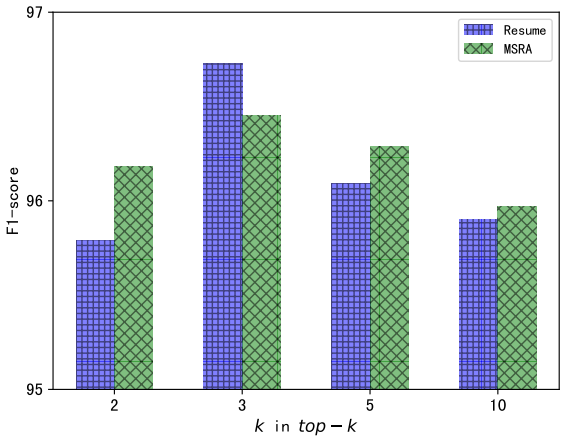


Figure 6: Influence of k value

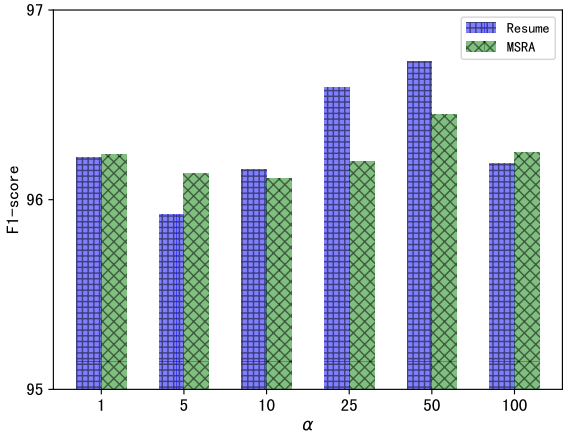


Figure 7: Influence of α value

NER. We conduct experiments on Resume and MSRA datasets to explore the influences by assigning k and α with various values. The results are summarized in Figure 6 and Figure 7.

The top-k operation is used to prevent the loss of critical information at the initial stage of training, it is a modification of the threshold. According to Figure 6, our model achieves the best performance when k is set to 3, which is due to the large proportion of two- and three-character words in Chinese. When k is smaller than 3, some critical information may be lost at context encoding layer, and when k is larger than 3, too much irrelevant character-level information is integrated.

The amplification factor α is utilized to control the similarity between approximate step function and binary function. The larger it is, the higher the similarity. As shown in Figure 7, our model performs best when α is 50. This is because when α is small, the selection or discarding of keys with attention score around the threshold has high uncertainty due to the Bernoulli sampling. When α becomes large, the gradient of the approximate step function (except the neighborhood of point $A_{ij} = T_i$) tends to 0, which brings difficulties to the optimization of the model.

In addition, since the Resume dataset has a relatively smaller scale, the model is more sensitive to the above two hyperparameters on Resume dataset than on MSRA dataset.

5.6 Test of Significance

For a fair comparison with the baseline, we implement a significance test on all four datasets. We randomly select ten various seeds and conduct experiments on our model and FLAT[4]. Then, a paired T-test is performed on each dataset. The p-values obtained on Weibo, Resume, Ontonotes and MSRA datasets are 0.0058, 0.0013, 0.0008 and 0.0161 respectively. Each of them is less than 0.02, which verifies the effectiveness of our proposed ATSSA.

5.7 Case Study

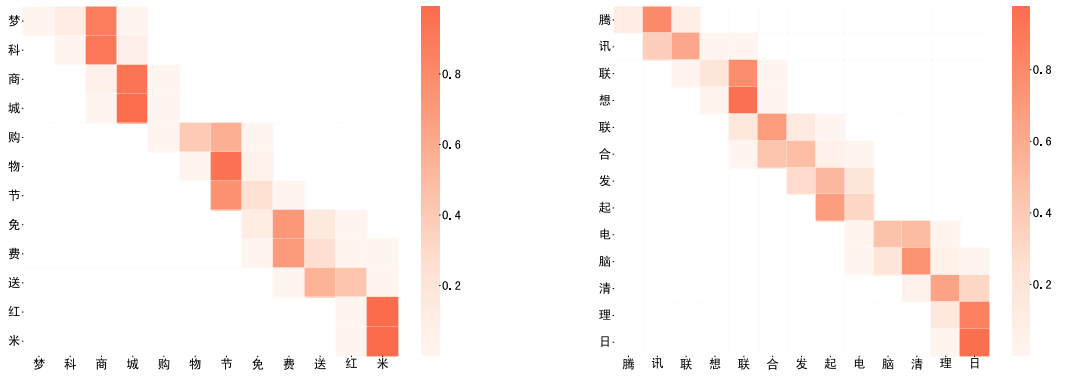
To verify that our proposed ATSSA can better recognize entity boundaries than the fully connected self-attention, we analyze two examples from Weibo test set, as shown in Figure 8. In the first case, due to the use of fully connected self-attention, the two characters "梦(Meng)" and "科(Ke)" as queries attend to the characters in "购物节(Shopping Festival)", which leads to the integration of the character information of "购物节(Shopping Festival)". As a result, FLAT misidentifies the "购物节(Shopping Festival)" as a part of the organization entity "梦科商城购物节(Mengke Mall Shopping Festival)". Our proposed ATSSA uses a threshold to discard characters in "购物节(Shopping Festival)" and correctly identify "梦科商城(Mengke Mall)" as an organization entity. In the second case, the fully connected self-attention assigns attention to each character in "腾讯(Tencent)" and "联想(Lenovo)" and FLAT misidentifies "腾讯联想(Tencent and Lenovo)" as a complete entity. However, ATSSA blocks the interactions between them via the threshold and identifies "腾讯(Tencent)" and "联想(Lenovo)" as two separate organization entity entities. These results show that the adaptive threshold can effectively filter irrelevant character information and help the

[4]https://github.com/LeeSureman/Flat-Lattice-Transformer

Case 1	梦科商城购物节免费送红米											
	Mengke Mall shopping festival gives Redmi away for free											
Gold Labels	梦	科	商	城	购	物	节	免	费	送	红	米
	B-ORGI-ORGI-ORGI-ORG	0	0	0	0	0	0	0	0			
FLAT+BERT	梦	科	商	城	购	物	节	免	费	送	红	米
	B-ORGI-ORGI-ORGI-ORGI-ORGI-ORGI-ORG	0	0	0	0	0						
Ours	梦	科	商	城	购	物	节	免	费	送	红	米
	B-ORGI-ORGI-ORGI-ORG	0	0	0	0	0	0	0	0			

Case 2	腾讯联想联合发起电脑清理日											
	Tencent and Lenovo co-sponsored Computer Cleanup Day											
Gold Labels	腾	讯	联	想	联	合	发	起	电	脑	清	理	日
	B-ORGI-ORGB-ORGI-ORG	0	0	0	0	0	0	0	0	0	0	0	
FLAT+BERT	腾	讯	联	想	联	合	发	起	电	脑	清	理	日
	B-ORGI-ORGB-ORGI-ORG	0	0	0	0	0	0	0	0	0	0	0	
Ours	腾	讯	联	想	联	合	发	起	电	脑	清	理	日
	B-ORGI-ORGB-ORGI-ORG	0	0	0	0	0	0	0	0	0	0	0	

Figure 8: Examples of Weibo test set, where blue colors represent the correct labels and red colors represent the wrong labels.



(a) Attention distribution of case 1.

(b) Attention distribution of case 2.

Figure 9: Selective self-attention distributions of the two examples in the head that focus on entities.

model to correctly identify entity boundaries.

Furthermore, we visualize the selective self-attention distributions of these two examples, as shown in Figure 9. In multi-head self-attention, different heads play some specific roles (Voita et al., 2019). For entity recognition, our analysis is based on the head that focuses on entities. In the figure, the vertical axis indicates queries and the horizontal axis represents keys. We observe that in Figure 9(a), the attention of queries in "梦科(Mengke)" is only assigned to keys in "梦科商城(Mengke Mall)", and attention of queries in "购物节(Shoping Festival)" is assigned to other keys, which provide support for the correct identification of the entity "梦科商城(Mengke Mall)". In Figure9(b), even though queries in "腾讯(Tencent)" attend to the key "联(Link)", queries in "联想(Lenovo)" only attend to the keys followed by "腾讯(Tencent)", which separates "腾讯(Tencent)" and "联想(Lenovo)" as two entities. We also observe that each query attends to a different number of keys, which demonstrates the flexibility of the proposed ATSSA.

6 Conclusion

In this paper, we propose an adaptive threshold-selective self-attention mechanism to dynamically select keys for queries in parallel to enhance the architecture of Transformer, which avoids the integration of irrelevant character-level information when encoding context. This data-driven mechanism maintains computational efficiency without losing flexibility. Based on FLAT, we apply it to Chinese NER and achieve state-of-the-art performance on four benchmark Chinese NER datasets. In future work, we will adapt ATSSA to different kinds of NLP tasks, such as text classification and natural language inference.

Acknowledgement

We thank the anonymous reviewers for their helpful comments. This work was supported by the National Natural Science Foundation of China (No. 62006243).

References

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *ACL*, pages 167–176.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *CoRR*, abs/1904.10509.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively sparse transformers. In

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 2174–2184.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese BERT. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:3504–3514.

Dennis Diefenbach, Vanessa López, Kamal Deep Singh, and Pierre Maret. 2018. Core techniques of question answering systems over knowledge bases: a survey. *Knowl. Inf. Syst.*, 55(3):529–569.

Ruixue Ding, Pengjun Xie, Xiaoyan Zhang, Wei Lu, Linlin Li, and Luo Si. 2019. A neural multi-digraph model for chinese NER with gazetteers. In *ACL*, pages 1462–1467.

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

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019b. A lexicon-based graph neural network for chinese NER. In *EMNLP-IJCNLP*, pages 1040–1050.

Hangfeng He and Xu Sun. 2017. F-score driven max margin neural network for named entity recognition in chinese social media. In *EACL*, pages 713–718.

Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *5th International Conference on Learning Representations*.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289.

Gina-Anne Levow. 2006. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth Workshop on Chinese Language Processing*, pages 108–117.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. 2018. Analogical reasoning on chinese morphological and semantic relations. In *ACL*, pages 138–143.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER using flat-lattice transformer. In *ACL*, pages 6836–6842.

Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. 2020. Real-time scene text detection with differentiable binarization. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 11474–11481.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in chinese NER. In *ACL*, pages 5951–5960.

André F. T. Martins and Ramón Fernandez Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *Proceedings of the 33nd International Conference on Machine Learning*, pages 1614–1623.

Tomás Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*, pages 1105–1116.

Nanyun Peng and Mark Dredze. 2015. Named entity recognition for chinese social media with jointly trained embeddings. In *ACL*, pages 548–554.

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1504–1519.

Alessandro Raganato, Yves Scherrer, and Jörg Tiedemann. 2020. Fixed encoder self-attention patterns in transformer-based machine translation. In *Findings of the Association for Computational Linguistics*, pages 556–568.

Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Sen Wang, and Chengqi Zhang. 2018. Reinforced self-attention network: a hybrid of hard and soft attention for sequence modeling. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, pages 4345–4352.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for chinese named entity recognition via collaborative graph network. In *EMNLP-IJCNLP*, pages 3828–3838.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5797–5808.

Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerry Tesauro, Bowen Zhou, and Jing Jiang. 2018. R^3: Reinforced ranker-reader for open-domain question

answering. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence,*, pages 5981–5988.

Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, and Robert Belvin. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Shuang Wu, Xiaoning Song, and Zhen-Hua Feng. 2021. MECT: multi-metadata embedding based cross-transformer for chinese named entity recognition. In *ACL/IJCNLP*, pages 1529–1539.

Lanqing Xue, Xiaopeng Li, and Nevin L. Zhang. 2020. Not all attention is needed: Gated attention network for sequence data. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, pages 6550–6557.

Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. 2019. TENER: adapting transformer encoder for named entity recognition. *CoRR*, abs/1911.04474.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *ACL*, pages 1554–1564.

Guangxiang Zhao, Junyang Lin, Zhiyuan Zhang, Xuancheng Ren, Qi Su, and Xu Sun. 2019. Explicit sparse transformer: Concentrated attention through explicit selection. *CoRR*, abs/1912.11637.

Shan Zhao, Minghao Hu, Zhiping Cai, Haiwen Chen, and Fang Liu. 2021. Dynamic modeling cross- and self-lattice attention network for chinese NER. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 14515–14523.