Attention Is All You Need for Chinese Word Segmentation

Sufeng Duan[1,2,3], **Hai Zhao**[1,2,3]*

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]Key Laboratory of Shanghai Education Commission for Intelligent Interaction
and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
1140339019dsf@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

Abstract

Taking greedy decoding algorithm as it should be, this work focuses on further strengthening the model itself for Chinese word segmentation (CWS), which results in an even more fast and more accurate CWS model. Our model consists of an attention only stacked encoder and a light enough decoder for the greedy segmentation plus two highway connections for smoother training, in which the encoder is composed of a newly proposed Transformer variant, Gaussian-masked Directional (GD) Transformer, and a biaffine attention scorer. With the effective encoder design, our model only needs to take unigram features for scoring. Our model is evaluated on SIGHAN Bakeoff benchmark datasets. The experimental results show that with the highest segmentation speed, the proposed model achieves new state-of-the-art or comparable performance against strong baselines in terms of strict closed test setting.

1 Introduction

Chinese word segmentation (CWS) is the task of delimiting word boundaries in a sentence, as a basic and essential task for Chinese and many other East Asian languages which are written without explicit word delimiters, and thus different from alphabetical languages like English.

Learning from an annotated corpus with segmentation, the CWS task may be generally modeled as a decoder which performs segmentation based on a scoring module in terms of contextual feature based representations. Table 1 summarizes typical CWS models according to their decoding ways. Markov models such as (Ng and Low, 2004) and (Zheng et al., 2013) depend on the maximum entropy model or maximum entropy Markov model both with Viterbi decoding. Besides, conditional random field (CRF) or Semi-CRF for sequence labeling has been used for both traditional and neural models though with different representations (Peng et al., 2004; Andrew, 2006; Wang and Xu, 2017; Ma et al., 2018).

Recent neural CWS research have been concerned about the following three perspectives (Emerson, 2005).

Decoder. As CWS is a kind of structure learning task, the decoder module generally determines which type of detailed algorithm should be adopted for segmentation, also it may limit the capability of defining feature. As shown in Table 2, not all models can support the word-level features as CWS is a task to predict word boundary. Thus recent works focus on finding more general or flexible decoder design to make model learn the representation of segmentation more effective such as (Cai and Zhao, 2016; Cai et al., 2017).

Encoder. Practice in various natural language processing tasks has shown that effective representation is essential to the performance improvement. For such a module in neural models, it is more than an encoder now, which is regarded as the most improvement perspective against traditional models. Thus for better CWS, it is crucial to encode the input character, word or sentence into a distinguishable representation. Table 2 summarizes regular feature sets for typical CWS models including ours as well. The building blocks that encoders use include recurrent neural network (RNN) and convolutional neural network (CNN), and long short-term memory (LSTM) network.

External resources and pre-trained embedding. Using external resource such as pre-trained embeddings or language representation provides

*Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), Huawei-SJTU long term AI project, Cutting-edge Machine reading comprehension and language model.

	Traditional Models	**Neural Models**	**Decoding Algorithm**
Greedy Model	-	**Ours**	Greedy
Markov Model	(Ng and Low, 2004), (Low et al., 2005)	MMTNN: (Pei et al., 2014)	
		(Zheng et al., 2013), LSTM: (Chen et al., 2015)	Viterbi
Sequence Labeling Model	CRF: (Peng et al., 2004), semi-CRF: (Andrew, 2006), (Sun et al., 2009)	CNN+CRF:(Wang and Xu, 2017), BiLSTM+CRF:(Ma et al., 2018)	
General Graph Model	(Zhang and Clark, 2007)	LSTM+GCNN: (Cai and Zhao, 2016), LSTM+GCNN: (Cai et al., 2017) (Wang et al., 2019a)	Beam search

Table 1: The classification of Chinese word segmentation model.

Models		Characters	Words
character based	Ours	$c_0, c_1, \dots, c_i, c_{i+1}, \dots, c_n$	-
	(Zheng et al., 2013), ...	$c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}$	-
	(Chen et al., 2015)	$c_0, c_1, \dots, c_i, c_{i+1}, c_{i+2}$	-
word based	(Zhang and Clark, 2007), ...	c in w_{j-1}, w_j, w_{j+1}	w_{j-1}, w_j, w_{j+1}
	(Cai and Zhao, 2016; Cai et al., 2017)	$c_0, c_1, \dots, c_i$	$w_0, w_1, \dots, w_j$

Table 2: Feature windows of different models. $i(j)$ is the index of current character(word).

an alternative for performance improvement other than designing better models (Yang et al., 2017). SIGHAN Bakeoff therefore defines two types of evaluation settings, closed test limits all the data for learning not to be beyond the given training set, while open test does not take this limitation (Emerson, 2005). This work will focus on the closed test setting by finding a better model design for further CWS.

Generally speaking, both the major difference between traditional and neural models, and what mostly distinguishes the neural models are about the way to represent input sentences, while the options of decoding algorithms are bounded to how to formalize the CWS into a structural learning task. As shown in Table 1, using Markov contextualized features, Markov models and CRF-based models are capable of using Viterbi decoders with polynomial time complexity. Furthermore, to accommodate more rich features means that the model has to take a deeper structural learning which also requires more complex decoding algorithms (Zhang and Clark, 2007; Cai and Zhao, 2016). However, for such a case, deterministic decoding algorithms may have an intractable complexity, thus it forces the model to use an approximate beam search strategy luckily with low-order polynomial time complexity $O(Mnb^2)$, where b is beam width,n is the sentence size, and M is a constant representing the model complexity. When the beam width b=1, the beam search will reduce to greedy algorithm with a much better time complexity $O(Mn)$.

To make the decoding practical, the beam width b has to be carefully tuned for a tradeoff between accuracy and efficiency: A larger b will make the learning and segmentation extremely slow, while a small b cannot sufficiently guarantee the segmentation performance. However, there has long been a unheeded observation that good enough representations can offer good enough segmentation even though only using a greedy segmentation algorithm. (Sproat and Emerson, 2003) create a topline evaluation by using only using vocabulary from test set to perform a greedy segmentation (maximum matching), which yields around 99% F-scores on all datasets. For neural models, (Cai et al., 2017) verify that if the representations are good enough, beam width 1 can still give state-of-the-art performance compared to their early model with a full beam search decoder in (Cai and Zhao, 2016). Therefore, undertaking a fixed greedy segmentation algorithm, this paper only focuses on more effective encoder design for even better representation.

Our model only consists of attention mechanisms as building blocks plus two highway connections via a virtual hidden layer for smooth training. Our model is simply stacked by a variant of Transformer encoder (Vaswani et al., 2017) and a biaffine attention scorer (Dozat and Manning, 2017). Empowered by the self-attention mechanism, the Transformer has been good at capturing long-range dependencies for input sentence. We propose Gaussian-masked Directional (GD) multi-head attention to facilitate the learning of localness, position and directional information for CWS, so that we have the proposed GD-Transformer.

With our further improved encoder, our model uses only simple unigram features to generate representation of sentences for scoring. Our model will be strictly evaluated on benchmark datasets from SIGHAN Bakeoff shared task in terms of closed test setting, and experimental results show that our model achieves new state-of-the-art.

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

- To especially enhance the representation of localness information and directional information, we propose a new Gaussian-masked Directional Transformer encoder.
- Motivated from a simple design idea, we present a new CWS model which is stacked with only attention blocks.
- With a powerful enough encoder, for the first time, we show that unigram (character) features plus greedy segmentation algorithm can support yielding strong performance instead of using diverse n-gram (character and word) features and highly complex decoding algorithms.

2 Related Work

(Xue, 2003) first formalize CWS as a sequence labeling task, considering CWS as a supervised learning from annotated corpus with human segmentation. (Peng et al., 2004) further adopt standard sequence labeling tool CRFs for CWS modeling, achieving new state-of-the-art. (Zhao et al., 2006b) show that different character tag sets can make essential impact for segmentation performance. (Zhao et al., 2006a) propose a CWS system developed for Bakeoff-2006 based on CRF, which is based on their proposed 6-tag set for character position tagging and achieved state-of-the-art performance at then. (Zhao and Kit, 2007) present a novel Character tagging based CRF framework which is capable of exploiting global information for performance enhancement.

Neural word segmentation has been widely used to minimize the efforts in feature engineering. (Zheng et al., 2013) first introduce the neural model into CWS with sliding-window based sequence labeling. (Chen et al., 2015) use LSTM to enhance the learning of long distance information.

However, introducing neural models themselves does not really introduce substantial performance improvement in terms of strict closed test of SIGHAN Bakeoff according to (Zhao et al., 2017). Most researchers actually seek help from joint learning, extra learning resources including dictionaries, pre-trained embedding, deeper information extracted from training set and so on. (1) For joint learning, (Lyu et al., 2016) explore a joint model that performs segmentation, POS-Tagging and chunking simultaneously. (Zhang et al., 2017) present a joint model to enhance the segmentation of Chinese microtext by performing CWS and informal word detection simultaneously. (2) For extra resources or clues, (Wang et al., 2019b) propose to incorporate unlabeled and partially-labeled data.

Only a few researches are known for concentrating on strengthening the model itself. To accommodate more rich features through a more broadly structural modeling (Cai and Zhao, 2016) propose a neural framework that eliminates context windows and utilize complete segmentation history. (Wang and Xu, 2017) propose a character-based convolutional neural model to capture n-gram features automatically and an effective approach to incorporate word embeddings. (Cai et al., 2017) further improve the model in (Cai and Zhao, 2016) and show that a greedy segmenter can perform fast and accurately in terms of only presenting effective representations. This work follows this line of research by offering even strengthened model design from simple idea, including the least building block type for encoder (attention only), the least feature type for scoring (unigram only) and the least computational complexity for decoding (greedy segmentation).

The original Transformer encoder consists of a stack of **N** identical layers and each layer has one multi-head self-attention layer and one position-wise fully connected feed-forward layer (Vaswani et al., 2017). One residual connection is around two sub-layers and followed by layer normalization. Several variants are proposed to enhance ability of capturing the localness relationship. (Shaw et al., 2018) propose an effcient way to incorporate relative and absolute position representation. (Yang et al., 2018) cast localness modeling as a learnable Gaussian bias to enhance the ability of capturing useful local context. (Kim et al., 2020) propose a Transformer with Gaussian-weighted self-attention to improved speech-enhancement performance. (Zhang et al., 2020b) propose using syntax to guide the text modeling based on self-attention network sponsored Transformer-based encoder. Transformer based pre-trained language

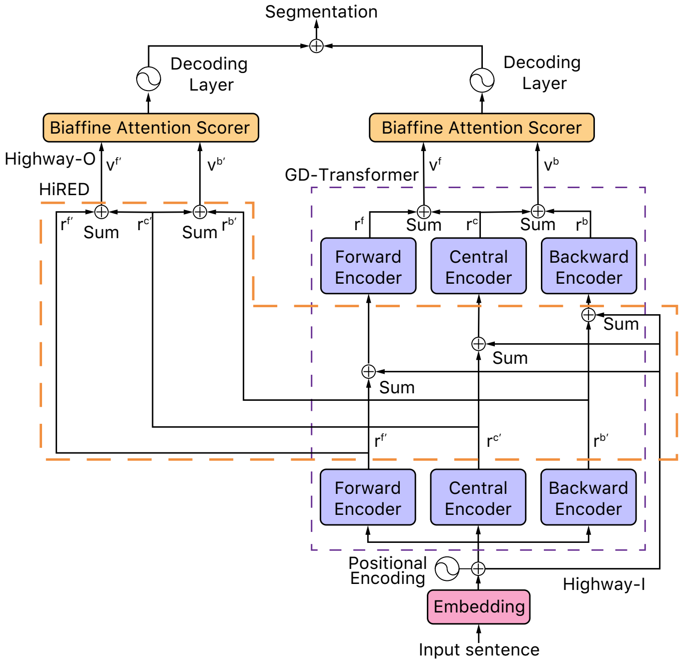


Figure 1: The architecture of our model.

models have become a standard performance enhancement means for various NLP tasks (Zhang et al., 2020a).

3 Models

Our model for CWS task is composed of an encoder to represent the input and a decoder based on the encoder to perform actual segmentation. Figure 1 is the architecture of our model. The model feeds sentence into encoder. Embedding captures the vector e of the input character sequences of c. The encoder maps vector sequences of e to two sequences of vector which are v^b and v^f as the representation of sentences. With v^b and v^f, the biaffine scorer scores each segmentation gaps which makes our decoder is as simple as one layer, using a threshold to directly and greedily predict every word boundaries of the input.

3.1 Gaussian-Masked Directional Transformer

The standard Transformer encoder consists of a stack of **N** identical layers and each has one multi-head self-attention layer and one position-wise fully connected feed-forward layer. One residual connection is around two sub-layers and followed by layer normalization (Vaswani et al., 2017).

The proposed Gaussian-masked Directional (GD) Transformer encoder adopts two key architecture revisions over the standard Transformer. (1) Our encoder includes three parallel directional encoding pipelines instead of only one bidirectional encoder in the original Transformer. (2) By replacing the standard multi-head self-attention with the proposed Gaussian-masked Directional (GD) multi-head self-attention which captures representations from different directions, the resulted encoder may gain better ability of capturing the localness information and position information for the importance of adjacent characters.

Encoder Stacks In CWS task, word boundary forms a gap between two adjacent characters and divides one sequence into two parts, one part in front of the gap and one part in the rear of it. The forward encoder and backward encoder are proposed to capture information of two directions which correspond to two parts divided by the gap. Assuming that one unidirectional encoder can capture information from one particular direction, we stack three parallel encoding modules, forward, backward and center encoders as shown in Figure 1.

The central encoder is to capture information from both directions, which is with the same architecture as the original Transformer. Standard scaled dot-product attention matrix is calculated by dotting query Q with all keys K. For the forward encoder, we forcibly set all values inside the attention matrix representing the character pair relation after the concerned character as 0 so that the encoder can focus on the forward characters. For the backward encoder, we take the similar matrix value setting operations.

The encoder respectively outputs one forward and one backward representations for each position, and then both are fused with the representation given by the center encoder to form the updated forward and backward representations, respectively.

$$v^b = r^b + r^c, \; v^f = r^f + r^c,$$

where v^b and v^f represent the backward and forward representation, respectively, r^b, r^c and r^f are representations from backward encoder, center encoder and forward encoder, respectively.

Gaussian-Masked Directional Multi-Head Attention Similar as scaled dot-product attention in the original Transformer (Vaswani et al., 2017), our proposed Gaussian-masked directional attention can be described as a function to map queries and key-value pairs to the representation of input. Here queries, keys and values are all vectors. Standard scaled dot-product attention is calculated by dotting query Q with all keys K, dividing each values by $\sqrt{d_k}$, where $\sqrt{d_k}$ is the dimension of keys, and apply a softmax function to generate the weights in the attention:

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \quad (1)$$

Different from scaled dot-product attention, Gaussian-masked directional attention expects to pay attention to the adjacent characters of each positions and cast the localness relationship between characters as a fix Gaussian weight for attention. We assume that the Gaussian weight only relies on the distance between characters.

Firstly we introduce the Gaussian weight matrix G=(g_{ij}) which presents the localness relationship between each two characters:

$$g_{ij} = \Phi(dis_{ij}) = \sqrt{\frac{2}{\sigma^2\pi}} \int_{-\infty}^{-dis_{ij}} exp(-\frac{x^2}{2\sigma^2})dx \quad (2)$$

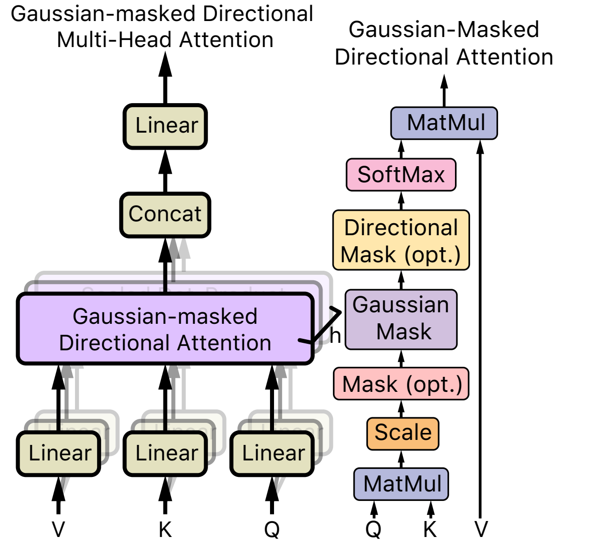
where g_{ij} is the Gaussian weight between character i and j, dis_{ij} is the distance between character i and j, $\Phi(x)$ is the cumulative distribution function of Gaussian, σ is the standard deviation of Gaussian function and it is a hyperparameter in our method. Eq. (2) ensures the Gaussian weight equals 1 when dis_{ij} is 0. The larger distance between characteristics, the smaller the weight is, which lets one character affect its neighbors more than those non-neighbors.

To combine the Gaussian weight to the self-attention, we produce the Hadamard product of Gaussian weight matrix G and the score matrix produced by QK^T

$$AG(Q,K,V) = softmax(\frac{QK^T * G}{\sqrt{d_k}})V \quad (3)$$

where AG as the Gaussian-masked attention ensures that adjacent characters have a stronger relationship than those non-neighbored ones.

The scaled dot-product attention models the relationship between two characters without regard to their distances in one sequence. For CWS task, the weight between adjacent characters should be more important while it is hard for self-attention to achieve the effect explicitly because the self-attention cannot get the order of sentences directly. The Gaussian-masked attention adjusts the weight between characters and their adjacent character to a larger value which stands for the effect of adjacent characters.


Gaussian-masked Directional Multi-Head Attention
Linear
Concat
Gaussian-masked Directional Attention
h
Linear
Linear
Linear
V
K
Q
Gaussian-Masked Directional Attention
MatMul
SoftMax
Directional Mask (opt.)
Gaussian Mask
Mask (opt.)
Scale
MatMul
Q
K
V

(a) The architecture of Gaussian-masked directional multi-head attention.

(b) The Gaussian-masked directional attention.

Figure 2: Illustration of Gaussian-masked directional multi-head attention.

For forward and backward encoder, the self-attention sub-layer needs to use a triangular matrix mask to let the self-attention focus on different weights:

$$g_{ij}^f = \begin{cases} g_{ij}, & pos_j \leq pos_i, \\ -\infty, & others. \end{cases}$$
$$g_{ij}^b = \begin{cases} g_{ij}, & pos_i \leq pos_j, \\ -\infty, & others. \end{cases} \quad (4)$$

where pos_i is the position of character c_i. The triangular matrix for forward and backward encode are:

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 1 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

Similar as (Vaswani et al., 2017), we use multi-head attention to capture information from different dimension positions as Figure 2(a) and get Gaussian-masked directional multi-head attention GMH as follows,

$$GMH(Q,K,V) = Concat(head_1, ..., head_h)W_m,$$
$$head_i = AG(QW_i^q, KW_i^k, VW_i^v) \quad (5)$$

where $W_i^q, W_i^k, W_i^v \in \mathbb{R}^{d_k \times d_h}$ is the parameter matrices to generate heads, W_m is a parameter matrices of $\mathbb{R}^{d_k \times d_k}$ to generate the attention, d_k and

d_h are dimensions of model and one head, respectively.

3.2 Biaffine Attention Scorer

Our model straightforwardly predicts gap between two adjacent characters as word boundary or not. In detail, we set a label value 1 to indicate word boundary, and 0 means no word boundary. Such a gap labeling task thus requires information of the two adjacent characters. In the meantime, the relationship between adjacent characters can be represented as the gap label.

Biaffine attention scorer is used to label the gap (Dozat and Manning, 2017; Li et al., 2018; Cai et al., 2018; Zhou and Zhao, 2019; He et al., 2019). The distribution of labels in a labeling task is often uneven. Biaffine attention uses bias terms to alleviate the burden of the fixed bias term and get the prior probability which makes it different from bilinear attention. The distribution of the gap is uneven that is similar as other labeling task, which makes biaffine available for our task.

Biaffine attention scorer labels the target depending on information of independent unit and the joint information of two units. In biaffine attention, the score s_{ij} of characters c_i and c_j $(i < j)$ is calculated by:

$$\begin{aligned} s_{ij} &= BiaffinalScorer(v_i^f, v_j^b) \\ &= (v_i^f)^T W v_j^b + U(v_i^f \oplus v_j^b) + b \end{aligned} \tag{6}$$

where v_i^f and v_i^b represent respectively the forward and backward information of c_j, W, U and b are all learnable parameters. W is a matrix with shape $(d_i \times N \times d_j)$ and U is a $(N \times (d_i + d_j))$ matrix where d_i is the dimension of vector v_i^f and N is the number of labels.

In our model, the biaffine scorer uses both the forward and backward character information on either side of the gap to distinguish the position of characters. Figure 3 is an example of gap labeling. The bidirectional scoring ensures that the boundaries of words can be determined by adjacent characters with different directional information. The score vector of the gap is formed by the probability of being a boundary of word. Further, the model generates all boundaries using activation function in a greedy decoding way.

3.3 Highway Connections via Hidden Layer

To smooth the training and fully exploit representations from hidden states, we additionally introduce two Highway connections (Srivastava et al., 2015) via a virtual hidden layer which is called Hidden Representations for Early Decoding (HiRED) in the middle of the Transformer encoder. In our model design, we always put the HiRED layer in the central position among all layers of the encoder, thus the HiRED layer divides each directional encoder (forward, backward or center) pipelines into two parts (front and rear) as shown in Figure 1.

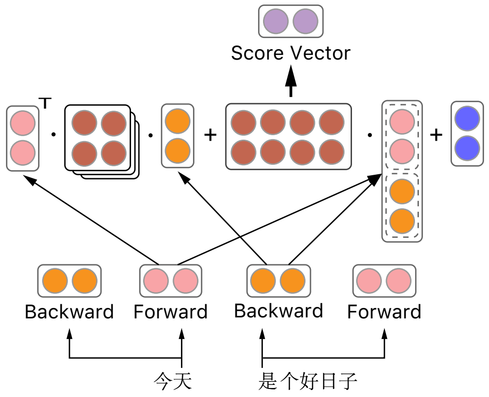


Figure 3: An example of biaffine scorer labeling the gap. The biaffine attention scorer only uses the forward information of front character and the backward information of character to label the gap.

For the highway connection specifications, the first connection (called Highway-I) respectively feeds the input embedding to the rear pipelines of the three directional encoders by adding into the embeddings from HiRED layer. Suppose that three front directional encoders respectively give encoding output, $r^{f'}$, $r^{c'}$ and $r^{b'}$. Then the corresponding three rear directional encoders will receive input as $e + r^{f'}$, $e + r^{c'}$ and $e + r^{b'}$. To feed the second connection (called Highway-O), we perform the same summing as the main encoder output,

$v^{b'} = r^{b'} + r^{c'}$, $v^{f'} = r^{f'} + r^{c'}$,

then let $v^{f'}$ and $v^{b'}$ as the HiRED output go through another same biaffine scorer and a decoder as that of the main encoder. The two decoder layers together give a sum loss for the entire model.

Biaffine attentin scorer makes it possible to generate a segmentation by using output of HiRED with little cost during training. With this segmentation, we add representation of characters which belong to the same word together and get a new vector, which plays a similar role as a word embedding. This vector will be fed to encoder layer behind HiRED directly. The operations in HiRED layer can also be viewed as one attention. It makes the model focus on adjacent characters which may be likely in one word.

3.4 Training Objective

The training target of our model is to let the biaffine attention scorer approach the the gold score vector according to the gold segmentation. We adopt cross entropy (CE) loss for training,

$$q_i^j = -s_{i,i+1}^j + \log(\exp(s_{i,i+1}^0) + \exp(s_{i,i+1}^1)),$$

$$CE = \frac{1}{l}\sum_{i=1}^{l}(q_i^1 p + q_i^0(1-p))$$

where q_i^j is the log-probability of the i-th gap labeled as $j \in \{1,0\}$. Here 1 indicates word boundary and 0 means not. $s_{i,i+1}^j$ is the biaffine score of i-th gap labeled as j. p is the ground-truth probability which can only be 0 or 1. l is the number of gaps in one input sentence.

	PKU	**MSR**
Sentences	19,056	86,924
Max length (Character)	1019	581
Max length (Word)	659	338
Word Types	55,303	88,119
Words	1,109,947	2,368,391
Character Types	4,698	5,167
Characters	1,826,448	4,050,469
	AS	**CITYU**
Sentences	708,953	53,019
Max length (Character)	188	350
Max length (Word)	211	85
Word Types	141,340	69,085
Words	5,449,698	1,455,629
Character Types	6,117	4,923
Characters	8,368,050	2,403,355

Table 3: Statistics of SIGHAN Bakeoff 2005 datasets.

	Parameters
dimension of hidden vector	256
number of layer	6
dimension of FF	1024
dropout	0.1
warmup	8000
number of head	4
batch size	4096

Table 4: Hyperparameters.

4 Experiments

4.1 Experimental Settings

Data Our models are trained and evaluated on benchmark datasets from SIGHAN Bakeoff 2005 (Emerson, 2005) which has four datasets, PKU, MSR, AS and CITYU. Table 3 shows the statistics of train data. F-score is to evaluate the performance.

Embedding Initialization Our model only adopts unigram features, so we only train character embeddings. On closed test, we use embeddings initialized randomly. On open test, our character embeddings are pre-trained on Chinese Wikipedia corpus by word2vec (Mikolov et al., 2013) toolkit. The corpus for pre-training is converted to simplified Chinese[1] and trivially segmented into characters.

Hyperparameters Our hyperparameter settings are in Table 4. All the settings are tuned on development sets[2]. We set the standard deviation of Gaussian function in Eq. (2) to 2. Each training batch contains sentences with at most 4096 tokens.

Optimizer To train our model, we use the Adam (Kingma and Ba, 2015) optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The learning rate schedule is the same as (Vaswani et al., 2017):

$lr = d^{-0.5} \cdot \min(step^{-0.5}, step \cdot warmup_{step}^{-1.5})$
where d is the dimension of embeddings, $step$ is the step number of training and $warmup_{step}$ is the step number of warmup. When the number of step is smaller than the step of warmup, the learning rate increases linearly and then decreases.

Hardware and Implements Our models are trained on a single CPU (Intel i7-5960X) and an nVidia 1080 Ti GPU, in terms of an implementation using Pytorch 1.0[3].

4.2 Results

Tables 5 compares recent models and ours in terms of closed test setting, showing that our model achieves new state-of-the-art and outperforms all the other models in MSR and AS. In the meantime, our model can achieve state-of-the-art efficiency.

Our models are also compared to the latest neural models in terms of open test setting in which any external resources, especially pre-trained embeddings or language models are allowedly used. Table 6 shows that our models get comparable results in AS and MSR though unremarkable ones in CITYU and PKU.

However, it is well known that comparing models accurately is hard for open test setting. Though

[1] OpenCC is used to transfer data from traditional Chinese to simplified Chinese, available at https://github.com/BYVoid/OpenCC.

[2] Following conventions, the last 10% sentences of training corpus are used as development set.

[3] Code is available at: https://github.com/akibcmi/SAMS

Models	PKU			MSR			AS			CITYU		
	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)	F_1	Tr. (hours)	Test (sec.)
(Chen et al., 2015)	**95.7**	58	105	96.4	117	120	-	-	-	-	-	-
(Cai and Zhao, 2016)	95.2	48	95	96.4	96	105	-	-	-	-	-	-
(Cai et al., 2017)	95.4	**3**	25	97.0	**6**	30	95.2	-	-	95.4	-	-
(Zhou et al., 2017)	95.0	-	-	97.2	-	-	-	-	-	-	-	-
(Ma et al., 2018)	95.4	-	-	97.5	-	-	95.5	-	-	95.7	-	-
(Wang et al., 2019a)	**95.7**	-	-	97.4	-	-	95.6	-	-	**95.9**	-	-
Our results	95.5	33	**4**	**97.6**	15	**4**	**95.7**	**67**	**10**	95.4	**17**	**1.5**

Table 5: Results on SIGHAN Bakeoff datasets in closed test. - indicates there is no reported result in the corresponding paper. (Tr.: Training).

external strengths like pre-trained embeddings or models can indeed improve the performance, it is difficult to determine which factor exactly makes such a contribution, the model itself, the resource or the better using of the resource. In terms of closed test setting, that is also the reason why this work keeps focusing on improvement of the model design itself.

	PKU	**MSR**	**AS**	**CITYU**
(Cai et al., 2017)	95.8	97.1	95.3	95.6
(Chen et al., 2017)	94.3	96.0	94.6	95.6
(Wang and Xu, 2017)	95.7	97.3	-	-
(Zhou et al., 2017)	96.0	97.8	-	-
(Ma et al., 2018)	96.1	**98.1**	96.2	97.2
(Wang et al., 2019a)	96.1	97.5	-	-
(Huang et al., 2019)	**96.6**	97.9	**96.6**	**97.6**
Our Method	95.5	97.7	95.7	96.4

Table 6: F1 scores in open test.

Compared with other LSTM models, our model performs better in AS and MSR than in CITYU and PKU. We attribute the performance difference to the impact of dataset sizes. Namely, the larger size is, the better model performs. For small corpus, the model tends to be overfitting.

Table 5 also shows the decoding time in different datasets. Our model finishes the segmentation with the least decoding time in all four datasets, thanks to the architecture of model which only takes attention mechanism as basic block, only adopts unigram features and a greedy decoding strategy from the very beginning.

4.3 Ablation Studies

This subsection presents ablation studies on MSR and PKU datasets to verify the benefits of each individual component in our model[4].

Gaussian-masked Directional Transformer. Table 7 gives the result of model with different Gaussian-masked directional self-attention. The third column and the fifth column are the difference of performance between GD-Transformer and other models. The results show that our full model GD-Transformer significantly outperforms the original Transformer by a large performance margin. Removing either Gaussian mask or directional mask will put negative impact over the performance of our model, which shows that both masks are indispensably necessary for our model performance.

	PKU		**MSR**	
GD-Transformer	95.4		97.6	
-Gaussian mask	94.6	-0.8	97.1	-0.5
-Directional mask	95.1	-0.3	97.4	-0.2
Transformer	94.1	-1.3	96.5	-1.1

Table 7: F1 scores on models removing different components from GD-Transformer.

Highway Connections. Table 8 gives the results of our model respectively removing the highway connections and the related HiRED layer part, which shows that each highway takes its contribution to the overall performance. However, the comparison shows that introducing all the components makes our model training much faster.

Directional Encoder. Table 9 gives the results of our models respectively removing the forward, center and backward encoders, which impacts performance of our model and shows that directional encoder and undirectional encoders are all indispensable for our model. The third column and the fifth column are the difference of performance between our full model and our models removing one encoder.

[4] Following (Cai et al., 2017), we show the results on the respective test set for either dataset, as SIGHAN Bakeoff did not provide official development sets.

Models	PKU		MSR	
	F_1	Training (hours)	F_1	Training (hours)
Our full model	95.5	33	97.6	15
-Highway-I	95.2	60	97.5	96
-Highway-O	95.3	45	97.4	102
-both highways	95.1	80	97.5	105

Table 8: F1 scores and training time on models related to highway connections and HiRED layer.

	PKU		**MSR**	
Our full model	95.5		97.6	
-Forward encoder	95.3	-0.2	97.4	-0.1
-Center encoder	95.3	-0.2	97.5	-0.1
-Backward encoder	95.4	-0.1	97.5	-0.2

Table 9: F1 scores of results on model removing different encoder from model.

5 Conclusion

For Chinese word segmentation, upholding the belief that a better representation is all we need and thus taking a greedy decoder for fast segmentation as the basis, we only focus on the encoder design and propose an attention mechanism only based CWS model. Our model uses the proposed GD-Transformer encoder to take sequence input and biaffine attention scorer to directly predict the word boundaries. To improve the ability of capturing the localness and directional information, Gaussian-masked directional multi-head attention in the GD-Transformer replaces the standard self-attention in the original Transformer. With powerful enough encoding ability, our model only needs unigram features for scoring instead of various n-gram features in previous work. Our model is evaluated on standard benchmark SIGHAN Bakeoff datasets, which shows not only our model performs segmentation faster than any previous models but also gives new higher or comparable segmentation performance against previous state-of-the-art models.

References

Galen Andrew. 2006. A hybrid Markov/semi-Markov conditional random field for sequence segmentation. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 465–472, Sydney, Australia. Association for Computational Linguistics.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for Chinese. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615, Vancouver, Canada. Association for Computational Linguistics.

Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic over syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2753–2765, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal. Association for Computational Linguistics.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203, Vancouver, Canada. Association for Computational Linguistics.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Shexia He, Zuchao Li, and Hai Zhao. 2019. Syntax-aware multilingual semantic role labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5350–5359, Hong Kong, China. Association for Computational Linguistics.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2019. Toward Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning. *CoRR*, abs/1903.04190.

Jaeyoung Kim, Mostafa El-Khamy, and Jungwon Lee. 2020. T-GSA: transformer with gaussian-weighted self-attention for speech enhancement. In

2020 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2020, Barcelona, Spain, May 4-8, 2020, pages 6649–6653. IEEE.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zuchao Li, Shexia He, Zhuosheng Zhang, and Hai Zhao. 2018. Joint learning of POS and dependencies for multilingual Universal Dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 65–73, Brussels, Belgium. Association for Computational Linguistics.

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Chen Lyu, Yue Zhang, and Donghong Ji. 2016. Joint word segmentation, pos-tagging and syntactic chunking. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pages 3007–3014.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese word segmentation with bi-LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium. Association for Computational Linguistics.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 277–284, Barcelona, Spain. Association for Computational Linguistics.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for Chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland. Association for Computational Linguistics.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 562–568, Geneva, Switzerland. COLING.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.

Richard Sproat and Thomas Emerson. 2003. The first international Chinese word segmentation Bakeoff. In *The Second SIGHAN Workshop on Chinese Language Processing*, page 133–143.

Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.

Xu Sun, Yaozhong Zhang, Takuya Matsuzaki, Yoshimasa Tsuruoka, and Jun'ichi Tsujii. 2009. A discriminative latent variable Chinese segmenter with hybrid word/character information. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 56–64, Boulder, Colorado. Association for Computational Linguistics.

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 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for Chinese word segmentation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 163–172, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019a. Unsupervised learning helps supervised neural word segmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019.*, pages 7200–7207.

Xiaobin Wang, Deng Cai, Linlin Li, Guangwei Xu, Hai Zhao, and Luo Si. 2019b. Unsupervised learning helps supervised neural word segmentation. In *AAAI*.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48.

Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. Modeling localness for self-attention networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. Association for Computational Linguistics.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada. Association for Computational Linguistics.

Meishan Zhang, Guohong Fu, and Nan Yu. 2017. Segmenting Chinese microtext: Joint informal-word detection and segmentation with neural networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 4228–4234.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847, Prague, Czech Republic. Association for Computational Linguistics.

Zhuosheng Zhang, Yuwei Wu, Hai Zhao, Zuchao Li, Shuailiang Zhang, Xi Zhou, and Xiang Zhou. 2020a. Semantics-aware BERT for language understanding. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9628–9635. AAAI Press.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020b. Sg-net: Syntax-guided machine reading comprehension. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 9636–9643. AAAI Press.

Hai Zhao, Deng Cai, Huang Changning, and Chunyu Kit. 2017. Chinese word segmentation, another decade review (2007-2017). In *The Frontier of Empirical and Corpus Linguistics*. China Social Sciences Press.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006a. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165, Sydney, Australia. Association for Computational Linguistics.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006b. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pages 87–94, Huazhong Normal University, Wuhan, China. Tsinghua University Press.

Hai Zhao and Chunyu Kit. 2007. Incorporating global information into supervised learning for Chinese word segmentation. In *In 10th Conference of the Pacific Association for Computational Linguistics*, pages 66–74.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2017. Word-context character embeddings for Chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766, Copenhagen, Denmark. Association for Computational Linguistics.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2396–2408, Florence, Italy. Association for Computational Linguistics.