BiG-Transformer: Integrating Hierarchical Features for Transformer via Bipartite Graph

Xiaobo Shu, Mengge Xue, Yanzeng Li, Zhenyu Zhang, Tingwen Liu*
Institute of Information Engineering, Chinese Academy of Sciences. Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences. Beijing, China
{shuxiaobo, xuemengge, liyanzeng,zhangzhenyu1996, liutingwen}@iie.ac.cn

***Abstract*—Self-attention based models like Transformer have achieved great success on kinds of Natural Language Processing tasks. However, the traditional fixed fully-connected structure faces many challenges in practice, such as computing redundancy, fixed granularity, and inexplicable. In this paper, we present BiG-Transformer, which employs attention with bipartite-graph structure to replace the fully-connected self-attention mechanism in Transformer. Specifically, two parts of the graph are designed for integrating hierarchical semantic information, and two types of connection are proposed to fuse information from different positions. Experiments on four tasks show the BiG-Transformer achieves better performance compared to Transformer liked models and Recurrent Neural Networks.**

***Index Terms*—Attention, Transformer, Chinese, Multi-level Features**

I. Introduction

Traditionally, with the help of deep feature extractor, natural language sentences are provided with strong semantic representations. There are two mainly feature extractors in Natural Language Processing (NLP) area: Recurrent Neural Network (RNN) and Transformer. Undoubtedly, RNN-based models play an excellent performance in kinds of NLP tasks [1, 2, 3, 4]. They process all tokens in the sentence one by one based on the recurrent structure, which has been proved to be great success in learning context representation [5, 6, 7]. However, it is precisely the recurrent structure that limits their speed and causes gradient explode (vanish). Recently, a self-attention based model named Transformer [8], has become popular in various NLP applications, especially the language modeling [9] and machine translation [8]. Some recent works even suggest that Transformer could be regarded as an alternative of recurrent neural networks and convolutional neural networks in many NLP tasks, such as BERT [9], Transformer-XL [10] and Universal Transformer [11], because they are inherently bidirectional and efficient.

Although Transformer-based models have achieved great successes, there are still a lot of studies to work out their potentialities. For the first, the complexity can be further reduced by reforming their structure, since self-attention is a heavy mechanism with fully-connected interactions between any two tokens. This introduces too much structure information that may be not necessary, especially for two tokens that are not related in lexical, syntax or semantics. Meanwhile, it is really hard to interpret its action [12]. To overcome this disadvantage, Star-Transformer [13] replaces the fully-connected structure with a star-shaped topology, where every two non-adjacent nodes (tokens) are connected through a shared relay node. This modification greatly decreases the computation complexity of the self-attention module. Nevertheless, the star-shaped topology of star-transformer can not capture words semantic to some extent, because both of its radical connections and ring connection are not take the word boundary or the syntactic structure into consideration.

Second, they fail to integrate tokens of different granularity and extract rich semantic information. In fact, many East Asian languages, including Chinese, are written without explicit word boundary. Thus, some recent works factorize lexical into characters or subwords [14, 15], and integrate multi-level semantic information into tokens. For example, Zhang et al. [16] introduced Lattice-LSTM which adds a lexical gate to inject word boundary information via a lexical control gate of modified Long Short-Term Memory Network (LSTM). Xiao et al. [17] proposed Lattice-Transformer which aims to inject word boundary information into attention mechanism to explore effective word or subword representation in an automatic way during training. Unfortunately, they introduced a complex position relation into the self-attention mechanism which leads to one more step calculation and redundant embedding matrices to be trained at the same time.

In order to reduce unnecessary calculations in fully-connected self-attention and integrate multi-level information to enhance semantics. In this paper, we propose a tailor-made model called BiG-Transformer with a bipartite-graph topology. Fig. 1 gives an overview of our model.

Different from the standard self-attention mechanism that works on a fully-connected homogeneous graph where every node is a character (word), our BiG-Transformer works on a bipartite graph, where nodes are composed of two different granularity: characters and words. In BiG-Transformer, these two types of nodes are intrinsically related, because coarse-grained (word) information comes from fine-grained (character) information. It is obvious that the proposed BiG-Transformer is capable of fusing multi-granularity information without introducing new parameters (such as external word vectors). In this way, we transform the fully-connected homogeneous graph self-attention into a bipartite-connected heterogeneous structure. To reduce the amount of computa-

* Corresponding author: Tingwen Liu.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

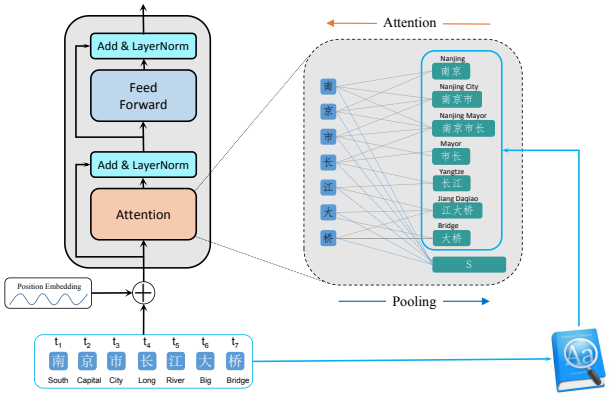

Fig. 1. The overall architecture of BiG-Transformer.

tion without losing semantics, two novel connection types are proposed to replace the fully-connection: Dynamic Local Connection and Fix Global Connection. Specifically, the Dynamic Local Connection preserves local semantic information between word and character, while the Fix Global Connection makes up for long-distance syntactic information.

We conduct extensive experiments on four widely-used NLP tasks including Text Classification (TC), Named Entity Recognition (NER), Machine Reading Comprehension (MRC) and Neural Machine Translation (NMT). The experimental results show the proposed BiG-Transformer model consistently outperforms than Star-Transformer and Lattice-Transformer overall tasks, which demonstrates the universality and effectiveness of our model. In addition, an ablation study indicates that the two types of connections are both indispensables.

II. Background

A. Self-Attention Mechanism

Self-attention mechanism, as one of the most representative variants of attention models, has attracted lots of interests due to its flexibility in parallel computation and long-term and short-term dependency modeling. Recently, many works have proved its effectiveness in kinds of NLP tasks, such as neural machine translation [8], question answering [5], and named entity recognition [18].

Formally, given an input sentence $X = \{x_1, \cdots, x_n\}$ with n tokens, the sequence is encoded into a set of distributed representations $\mathbf{H}^l = \{\mathbf{h}_1^l, \cdots, \mathbf{h}_n^l\}$ successively, where l is the layer number, and each hidden state in the l-th layer is constructed by attending to the states in the previous layer. The inputs of the first layer can be obtained from an embedding look-up table: $\mathbf{H}^0 = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$. To be more specific, the l-th layer representation $\mathbf{H}^l \in \mathbb{R}^{n \times d}$ is transferred into three different spaces as quires $\mathbf{Q}^l$, keys $\mathbf{K}^l$, and values $\mathbf{V}^l$:

$$\begin{bmatrix} \mathbf{Q}^l \\ \mathbf{K}^l \\ \mathbf{V}^l \end{bmatrix} = \mathbf{H}^l \begin{bmatrix} \mathbf{W}^Q \\ \mathbf{W}^K \\ \mathbf{W}^V \end{bmatrix} \tag{1}$$

where $\{\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V\} \in \mathbb{R}^{d \times d}$ are trainable transfer matrices and d is the dimension of hidden states. Next, we use $\mathrm{Att}(\cdot)$ to denote a attention model, which can be implemented as either additive attention or dot-product attention [6, 19]. In this paper, we use the dot-product attention, which achieves similar performance with additive counterpart but is faster and more space-efficient in practice:

$$\mathrm{Att}(\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l) = \mathrm{softmax}(\frac{\mathbf{Q}^l \mathbf{K}^{l\top}}{\sqrt{d}})\mathbf{V}^l. \tag{2}$$

where d is dimension of hidden state. Finally, we achieve the output vectors of the $(l+1)$-th layer (as well as the input vectors of the next layer):

$$\mathbf{H}^{l+1} = \sigma(\mathrm{Att}(\mathbf{Q}^l, \mathbf{K}^l, \mathbf{V}^l)). \tag{3}$$

where σ is some other functions, such as layer normalization, feed forward layer.

B. Motivation

Intuitively, the self-attention mechanism could be regarded as a fully-connected graph, in which each token is regarded as a node, every node is connected with each other and edges denote the semantic relationships between two nodes. In fact, for natural languages, words are naturally made up of characters, and there are plenty of inherent dependency relationships between word and character. This structure information has proven to be useful for word representation learning [15, 16]. To be more specific, some Asian languages like Chinese typically have no explicit boundary, and a character might have multiple connections with different words when using various word segmentation strategies. For the example shown in Fig. 1, in Chinese character "South" is an element of words "Nanjing", "Nanjing City" and "Nanjing Mayor" at the same time. It is obvious that words and characters can be divided into two independent parts to highlight the structure information between them. An intuitive way is to organize the character and word into a bipartite graph. By this means, the bipartite-graph is capable of capturing the potential relations among words and characters. Even for some alphabetical languages like English, there is still an open problem of selecting a proper subword vocabulary size, which determines the segmentation granularity for downstream tasks.

Furthermore, one advantage of the standard self-attention mechanism is that each token of the sequence is global-aware since all tokens are calculated with each other based on the fully-connected structure. With that in mind, we think that a sentence node with the semantic representation of the input sentence is also global-aware. By interacting with the sentence node, each character node in the graph can capture long-distance information and have global awareness indirectly.

III. Bipartite-Graph Transformer

In this section, we introduce the primary architecture of our BiG-Transformer model firstly, then describe the main difference between our implementation details and that of standard Transformer.

A. Bipartite-Graph Construction

As shown in Fig. 1, for our bipartite-graph Transformer, the input sentence is converted into a bipartite-graph: one type of nodes are characters and the other type of nodes are words. We regard the sentence node as a special word node, by considering the whole sentence to be a very long word. To connect these character nodes and word nodes, we introduce two types of connections, namely dynamic local connection and fixed global connection.

1) Dynamic Local Connection: The dynamic local connections are designed for the character-to-word association. In BiG-Transformer, the edges between nodes represent the inclusion relationship from left to right, which means that an edge between character c_i and word w_j is established if c_i is an element of w_j. For the example in Fig. 1, the character "South" have edges with words "Nanjing", "Nanjing City" and "Nanjing Mayor". We use a dictionary prepared in advance to find all possible words appearing in the input sentence based on the string matching technique. Since words appearing in the input sentence are not known in advance, these connections between characters and words are dynamic.

2) Fixed Global Connection: Note that dynamic local connections focus on capturing local semantic information, because any character can only interact with its neighbors and adjacent characters appearing in the same words. As a result, some useful semantic information from long-distance characters is lost. To make up for this loss, we introduce a sentence node at the right part of our bipartite-graph, and all character nodes on the left are connected with the sentence node. In this way, the whole sentence is considered to be a long word, and any two characters are connected indirectly through this sentence node.

B. Implementation Details

Note that most of the implementations of our model are similar with that of the standard transformer. In this paper, we focus on the different modules between BiG-Transformer and standard Transformer.

1) Word and Sentence Representation: In order to keep the representations of all nodes in the same embedding space and avoid introducing more parameters, we utilize a simple and effective approach to get word and sentence representations from character representations. In detail, given a sequence of characters representations $\mathbf{H} \in \mathbb{R}^{n \times d}$ and word position tuples $\{(s_1, e_1), ..., (s_m, e_m)\}$, where s_i and e_i indicate the indexes of starting position and ending position of word i respectively (m is the number of all appearing words), we get the word representation as following:

$$\mathbf{O}_{i:} = \sum_{j=s_i}^{e_i} \mathbf{h}_j \tag{4}$$

where $\mathbf{O} \in \mathbb{R}^{m \times d}$ is the word representation matrix.

The representation of the sentence node indicates the global sentence-level semantic which is designed for learning long-distance information. We get the sentence representation with the average pooling operation as following:

$$\mathbf{s}^l = \text{Average}(\mathbf{H}^l)\mathbf{W}_s^l \tag{5}$$

where $\mathbf{W}_s^l$ is a learnable parameter matrix.

2) Multi-head Self-attention: Different from standard self-attention mechanism that calculates attention between any two characters, our BiG-Transformer only calculates the attention between an character and any word containing the character. In our model, given a sequence of character representations, and word set u_i contain the representations of all words containing the i-th character of the sequence, we utilize multi-head attention to capture more semantic information. The calculation of the word-to-character multi-head attentions are as following:

$$\text{MultiHeadAtt}(\mathbf{h}_i, \mathbf{U}_{i:}^l) = [\mathbf{r}_1 \oplus \cdots \oplus \mathbf{r}_p]\mathbf{W}_O \tag{6}$$

$$\mathbf{r}_j = \text{Att}(\mathbf{h}_i\mathbf{W}_j^Q, [\mathbf{U}_i \oplus \mathbf{s}]\mathbf{W}_j^K, \mathbf{H}\mathbf{W}_j^V), \ j \in [1, p] \tag{7}$$

$$\mathbf{U}_i = [u_i^1 \oplus \cdots \oplus u_i^{|u_i|}] \tag{8}$$

where $\oplus$ denotes the concatenation operation, p is the number of heads, $\mathbf{W}_j^Q, \mathbf{W}_j^K, \mathbf{W}_j^V$ and $\mathbf{W}^O$ are trainable matrices.

3) Position Embedding: Transformer-based models usually utilize position embedding to incorporate the sequence infor-

Algorithm 1 The Update of BiG-Transformer

Input: A sequence of characters embedding $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$, and word position tuple $\{(s_1, e_1), ..., (s_m, e_m)\}$.
Output: Hidden representation of each character: h_i^L.

$\mathbf{h}_1^0, ..., \mathbf{h}_n^0 \leftarrow \mathbf{x}_1, ..., \mathbf{x}_n$.
$\mathbf{s}^0 \leftarrow \text{average}(\mathbf{H}^0)\mathbf{W}_\text{s}^0$
for l from 1 to L **do**
  // word representation.
  for j from 1 to m **do**
    $\mathbf{o}_j^l = \text{sum}(\mathbf{H}_{s_j:e_j}^{l-1})$
  end for
  // multi-head self-attention.
  for i from 1 to n **do**
    $\mathbf{U}_{i:}^{l-1}$ = NULL
    for j from 1 to m **do**
      if i between s_j and e_j **then**
        $\mathbf{U}_{i:}^{l-1} = \text{Concatenate}(\mathbf{U}_{i:}^{l-1}, \mathbf{o}_j^{l-1})$
      end if
    end for
    $\mathbf{U}_{i:}^{l-1} = \text{Concatenate}([\mathbf{U}_{i:}^{l-1}, \mathbf{s}^{l-1}])$
    $\mathbf{h}_i^l = \text{MultiHeadAtt}(\mathbf{h}_i^{l-1}, \mathbf{U}_{i:}^{l-1})$
  end for
  $\mathbf{H}^l = \text{LayerNorm}(\mathbf{H}^{l-1})$
  // global representation.
  $\mathbf{s}^l = \text{Average}(\mathbf{H}^l)\mathbf{W}_s^l$
end for

mation. In this study, we also adopt the same strategy with standard transformer:

$$p(j, 2i) = \sin(j/100000^{2i/d}) \\ p(j, 2i+1) = \cos(j/100000^{2i/d}) \tag{9}$$

where j is the position of tokens, i is index of position embedding and d is the model dimension. As a result, the position embedding is added to corresponding token embedding as their initial representation.

4) Update of BiG-Transformer: The overall update of our model is shown in Algorithm 1. Formally, let $\mathbf{H^l} = \{h_1^l, \cdots, h_n^l\}$ to be the hidden state representation of a sequence characters at l-th layer. We initialize $\mathbf{H}^0$ with $\mathbf{E} = \{\mathbf{e}_1, \cdots, \mathbf{e}_n\}$, which is a character embedding layer.

Next, in order to update the hidden state representations, we get word representations based on Formula 3, the attention mechanism integrates the lexical information into characters, and the long-term distance information is learned by interacting with global sentence representation $\mathbf{s}$.

C. Why BiG-Transformer Works?

Character-based self-attention mechanisms have been proved effective in many NLP tasks. However, these character-based methods still have their disadvantages: it is difficult to explain intuitively why they work, because the character can neither reflect the syntactic relationship nor the true semantics of the sentence. For the attention of a word, an intuitive approach is that the word should focus on two aspects. One is the word that could form the word locally, and the second is the whole semantic relationship. In addition, a word may be contained in multiple words. When calculating the attention mechanism, the final expression of the word strengthens the representation of overlapping words between multiple words. We think this is also the reason why it can stand out in NER.

IV. Experiments

A. Tasks and Datasets

To demonstrate the effectiveness of our BiG-Transformer model, we conduct lots of experiments on ten open-world datasets of four mainstream NLP tasks:

- **Named Entity Recognition (NER):** Given a sentence, the task is to classify the named entity label in each position. We adopt four widely used datasets for this task: Weibo-NER [20, 21], OntoNotes, MSRA[22], and Chinese Resume[21].
- **Text Classification:** Given a sentence or a pair of sentences, the task is to predict its label. For this task, we take four datasets. ChnSentiCorp[1] and Sina Weibo[2] are sentiment classification (SC) datasets, THUCNews is a document-level text classification (DC) dataset, BQ Corpus[23] is a dataset of the sentence pair matching (SPM) task which predicts the relation label for a given pair of sentence.
- **Machine Reading Comprehension (MRC):** DRCD [24] is an open domain traditional Chinese machine reading comprehension dataset. Given a paragraph and a question, the MRC task is to predict the answer span which appeared in the paragraph.
- **Neural Machine Translation (NMT):** We introduce the NMT task to make a comparison with Lattice-Transformer. Following Lattice-Transformer, we chose the NIST 2005 dataset as development set and use the NIST 2002, 2003, 2004, 2006, and 2008 datasets as test sets.

For the statistical details of our datasets, please refers to Table I.

B. Experimental Settings

In order to make a fair comparison, for each dataset, we keep the same hyper-parameters (such as maximum length, warm-up steps), and tune the initial learning rate from 1e-3 to 1e-5. We run the same experiments for three times and report the average results to ensure reliability. And the best learning rate is determined by selecting the best development set performance. The details of parameter setting of our model on these ten datasets is also shown in Table I.

C. Results of Text Classification

The text classification task includes three types of datasets in this paper: Sentiment Classification (ChnSentiCorp, Sina Weibo), Sentence Pair Matching (BQ Corpus), Document Classification (THUCNews). For all experiments of text classification, we perform average pooling operation over the character representations as the final text representation and feed it into the softmax classifier. Results of text classification are shown in Table II.

For word2vec-based initialization approach, our model outperforms all Transformer-based models on four datasets. This demonstrates the effectiveness of our proposed attention mechanism. It is worthy to note that LSTM gets better results compared with Transformer and Star-Transformer but is worse than BiG-Transformer on ChnSenti, Weibo and THUCNews datasets. This implies that attention-based model still needs better structure to tap its potential.

Our model beats all other models in ChnSenti dataset, and achieves 0.50 points of improvement compared with the best baseline (LSTM). ChnSenti is a small corpus which shows that our model could extract feature better from a small dataset. Our model also performs well on Weibo dataset which is the largest one, but is slightly lower than LSTM on BQ which is sentence pair matching dataset. We speculate that spatial information may be important for the matching task.

When initializing the embedding layer with Chinese BERT[3], our model still outperforms all baselines including pure BERT model with a fully-connected classify layer. This proves that word-character based attention is more helpful to capture the semantic character-based model. We think these results agree

[1]https://github.com/pengming617/bert_classification

[2]https://github.com/SophonPlus/ChineseNlpCorpus/

[3]https://github.com/google-research/bert

TABLE I

AN OVERVIEW OF EXPERIMENTAL DATASETS AND THE HYPER-PARAMETERS IN OUR EXPERIMENTS, "HEAD" INDICATES THE NUMBER OF HEADS IN THE MULTI-HEAD ATTENTION. PARAMETERS IN TEXT CLASSIFICATION TASK ARE DIVIDED BY "/", WHICH DENOTES THAT BERT-BASED AND WORD2VEC-BASED WORD EMBEDDING INITIALIZATION METHODS.

Dataset	TASK	MAXLEN	BATCH	LR	HEAD	EPOCH	TRAIN	DEV	TEST
Weibo-NER	NER	-	16	1e-5	8	15	1,350	270	270
MSRA	NER	-	12	1e-5	8	15	46,306	4,361	4,361
Resume	NER	-	12	3e-5	8	5	3,821	463	477
Ontonotes	NER	-	16	1e-5	8	15	15,717	4,298	4,345
ChnSenti	SC	256	16/128	3e-5/7e-4	6	3/20	9,601	1,201	1,201
BQ	SPM	128	64/128	1e-5/7e-4	6	3/20	100,001	10,001	10,001
THUCNews	DC	512	16/128	1e-5/7e-4	6	3/20	50,001	5,000	10,001
Weibo	SC	128	16/128	2e-5/7e-4	6	3/20	100,001	9,989	10,001
DRCD	MRC	512	16/10	3e-5	6	-/2	26,932	3,524	3,485
NIST	NMT	-	-/64	1.0	8	-/2	100,0000	1,082	-

TABLE II

RESULTS ON THE TEXT CLASSIFICATION DATASETS. BOLD MARKS HIGHEST NUMBER AMONG ALL MODELS. THERE ARE TWO WAYS TO INITIATE THE WORD EMBEDDING LAYER: WORD2VEC (TOP) AND BERT (BOTTOM).

	ChnSenti		Weibo		BQ		THUCNews	
Model	dev	test	dev	test	dev	test	dev	test
LSTM (Word2vec)	**89.58**	88.17	**94.92**	95.38	**68.82**	**67.19**	77.81	79.21
Transformer (Word2vec)	87.33	86.17	94.73	95.19	67.49	65.24	72.41	72.90
Star-Transformer (Word2vec)	84.25	81.83	94.66	95.04	65.85	65.34	77.23	78.77
BiG-Transformer (Word2vec)	88.83	**89.50**	94.51	**95.93**	67.06	66.02	**79.20**	**80.71**
BERT	94.30	94.42	97.38	97.31	83.76	80.08	97.40	94.70
LSTM (BERT)	94.42	94.22	96.38	95.17	83.90	82.54	96.84	97.18
Transformer (BERT)	93.01	93.23	59.60	95.17	85.05	83.47	97.26	96.99
Star-Transformer (BERT)	93.08	93.58	94.71	97.63	**85.54**	83.34	95.88	96.65
BiG-Transformer (BERT)	**94.90**	**94.92**	**97.52**	**98.04**	84.40	**83.69**	**98.01**	**97.55**

with recent works of span-based pre-trained language models [25, 26].

D. Results of Machine Reading Comprehension

In this section, we compare BiG-transformer with state-of-the-art baselines to show the natural language understandability on MRC task. We use BERT as the initial embedding layer for all models, and predict the start/end label for each position by passing the output of model into a fully-connected layer. Table V shows the experimental results.

For all three baselines, BiG-Transformer obtains better performance on both EM and F1 scores. Transformer liked models initialized by BERT still outperform LSTM, this conclusion keeps consistent to we mentioned in section IV-F. Compare within Transformer liked models, we obtain close results between standard Transformer and Star-Transformer, but lower than BiG-Transformer, we think that the model is effective but unable to learn the words level semantic information to improve character-based start/end classification.

E. Results of Machine Translation

To make a comparison with Lattice-Transformer [17], we introduce machine translation experiments in which task is used to demonstrate the effectiveness of Lattice-Transformer in [17]. All baseline results are from [17]. From Table III, we can find that our model outperforms Transformer and Lattice-Transformer by 0.88 and 0.30 BLEU in the overall performance. These results give a piece of strong evidence to show our model's effectiveness.

F. Results of Named Entity Recognition

To verify the ability of our model in sequence labeling, we choose the classical NER task. We follow Lattice-LSTM [16], and adopt four frequently-used datasets to show the effectiveness of BiG-Transformer [4].

From Table IV, we can find that BiG-Transformer achieves state-of-the-art performance on all four datasets. This leads to the same conclusion with Lattice-LSTM: lexical information is really important for named entity recognition task. Comparing to Transformer and Star-Transformer, our BiG-Transformer model beats both of them on no matter large (MSRA, Ontonotes) or small datasets (Weibo-NER, Resume), it follows that our model helps to learn a better character representation benefit by bipartite-graph structure. Interestingly, Transformer achieves better performance than LSTM on all four datasets.

[4]Transformer-based models initialized by word2vec are poor for NER task, and there is no any official results reported, so we omit the results here.

TABLE III
EVALUATION OF TRANSLATION PERFORMANCE ON NIST ZH-EN DATASET. RNN, LATTICE-RNN AND LATTICE-TRANSFORMER RESULTS ARE CITED FROM [17]. WE HIGHLIGHT THE HIGHEST BLEU SCORE IN BOLD FOR EACH SET.

System	Input	MT05	MT02	MT03	MT04	MT06	MT08	ALL
RNN	PKU	31.42	34.68	33.08	35.32	31.61	23.58	31.76
	CTB	31.38	34.95	32.85	35.44	31.75	23.33	31.78
	MSR	29.92	34.49	32.06	35.10	31.23	23.12	31.35
Lattice-RNN	Lattice	32.40	35.75	34.32	36.50	32.77	24.84	32.95
Transformer	PKU	41.67	43.61	41.62	43.66	40.25	31.62	40.24
	CTB	41.87	43.72	42.11	43.58	40.41	31.76	40.35
	MSR	41.17	43.11	41.38	43.60	39.67	31.02	39.87
Lattice-Transformer	Lattice	42.65	**44.14**	42.24	**44.81**	41.37	32.98	40.93
BiG-Transformer	Matching	**43.35**	43.84	**44.10**	44.11	**42.56**	**33.78**	**41.23**

TABLE IV
RESULTS ON THE NER DATASETS, BOLD MARKS THE HIGHEST NUMBER AMONG ALL MODELS. WE DON'T REPORT TRANSFORMER, STAR-TRANSFORMER AND BIG-TRANSFORMER'S RESULTS WHICH INITIATING THE EMBEDDING LAYER USING CONTEXT-FREE EMBEDDING, BECAUSE THERE ARE NO RESULTS REPORTED.

	Ontonotes			Resume			Weibo-NER			MSRA		
Model	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
LSTM	74.36	69.43	71.81	92.97	90.80	91.87	50.55	60.11	56.75	94.53	94.29	94.41
Lattice-LSTM	76.35	71.56	73.88	61.08	47.22	53.26	52.71	53.92	53.13	94.81	94.11	94.46
LR-CNN	76.40	72.60	74.45	94.50	92.93	93.71	65.06	50.00	54.43	95.37	94.84	95.11
BERT-Tagger	78.01	80.35	79.16	94.43	93.86	94.14	67.12	66.88	67.33	96.12	95.45	95.78
Lattice-LSTM (BERT)	**79.79**	79.41	79.60	93.57	92.79	93.18	61.08	47.22	53.26	95.79	95.03	95.41
LR-CNN (BERT)	79.41	80.32	79.86	**94.68**	94.03	94.35	64.11	**67.77**	65.89	95.68	**96.44**	96.06
Transformer (BERT)	76.46	81.41	78.86	92.00	93.41	92.70	64.50	66.41	65.44	95.20	96.13	95.67
Star-Transformer (BERT)	77.37	81.52	79.39	92.89	94.01	93.45	65.73	63.24	64.46	94.20	94.66	94.43
BiG-Transformer (BERT)	78.34	**82.43**	**80.33**	94.47	**94.30**	**94.38**	**70.05**	67.13	**68.56**	**96.50**	96.32	**96.41**

TABLE V
RESULTS ON THE DRCD DATASET, BOLD MARKS THE HIGHEST NUMBER AMONG ALL MODELS.

Model	EM	F1
LSTM (BERT)	91.13	85.02
Transformer (BERT)	91.31	85.28
Star-Transformer (BERT)	91.22	85.25
BiG-Transformer (BERT)	**91.77**	**86.23**

TABLE VI
ABLATION STUDY RESULTS ON CHISENTI (TC), ONTONOTES (NER) AND DRCD (MRC) DATASETS.

Model	ChiSenti Acc	Ontonotes F1	DRCD F1
BiG-Transformer	89.50	80.33	86.23
w/o global	87.21	72.21	82.11
w/o local	80.38	50.10	73.01
r. radical	83.56	77.33	85.16

We conjecture that this is because BERT has compatibility with Transformer instead of the LSTM model.

G. Ablation Study

In this section, we perform an ablation study to verify the effectiveness of the dynamic local connection and fixed global connection. Results are shown in Table VI.

We test three variants of our model in three datasets: ChnSenti (Text Classification), Ontonotes (NER) and DRCD (MRC). For the first one, we remove dynamic connection and keep the global connection in the bipartite graph. As shown in the first line of Table VI, without the dynamic local connection, performance on three datasets decline, which demonstrates that our dynamic connection between different grains is capable of fusing lexical level information into character representation. For the second, we remove the global connection to prove the effectiveness of long-distance semantic our model learned, as the results indicated, discarding global connection will make the performance decrease sharply, because model losses the ability to model long-distance dependencies. Third, we replace the dynamic connection with the adjacent connection. comparing to our model, it performs worse on three of the tasks, which give the evidence of useful of dynamic connection. Therefore, both the dynamic local attention and fixed global connections are necessary and helpful to BiG-Transformer.

V. Related Work

Our model can be divided into two main novel parts: Word-Character level Compositionality and Global-Level Compositionality. In view of this, we roughly review the related works via two parts.

A. Word-Level Compositionality

As the word is the fundamental unit of semantic expression, the most intuitive approach to represent a word to vector [27], but to deal with the Out-Of-Vocabulary problem and learn more semantic relation between words, subword level models are proposed. For example, [28] uses a character LSTM to solve obtaining competitive results for the NER task. [29, 30] pre-train the subword enrich word representation to improve its performance on kinds of NLP tasks. [16] design a novel lattice LSTM representation for mixed characters and lexicon words by adding a word-level gate. SA-LSTM [31] incorporates segment information into word-level attention, and is capable of learning relational expressions. Many works integrating word-level information and character-level information have been found to achieve good performance. [28] hierarchy structure by incorporating BiLSTM-based character embeddings. Tree Transformer [32] adds an extra constraint to attention heads of the bidirectional Transformer encoder in order to encourage the attention heads to follow tree structures.

B. Global-Level Compositionality

For integrating global-level semantic information into words, prior work can be divided into three parts: recurrent-based, CNN-based and attention-based. Hierarchically stacked CNN layers [33, 34] allow better interactions between non-local components in a sentence via incremental levels of abstraction. S-LSTM [35] uses a global sentence-level node to assemble and back-distribute local information in the recurrent state transition process, suffering less information loss compared to pooling. Recently, attention-based models achieve great success because of its bi-direction and effectiveness. Transformer [8] learns the dependencies between words in a sentence based entirely on self-attention without any recurrent or convolutional layers. Star-transformer replaces the fully-connected self-attention structure with a star-shaped topology, in which every two non-adjacent nodes are connected through a shared relay node.

VI. CONCLUSION

In this paper, we propose a novel word-character attention mechanism to replace the original fully-connected self-attention mechanism in the Transformer. Experiments base on four tasks demonstrated the effectiveness. In other languages, there are segment operations divide the lexical level word into character or n-grams. So in future work, we will extend our model to subword or character base approach in other languages.

VII. ACKNOWLEDGEMENTS

This work was supported by the National Key Research and Development Program of China (No.2016YFB0801003) and the Strategic Priority Research Program of Chinese Academy of Sciences (No.XDC02040400).

References

[1] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[2] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[3] S. Li, W. Li, C. Cook, C. Zhu, and Y. Gao, "Independently recurrent neural network (indrnn): Building a longer and deeper rnn," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5457–5466.

[4] Y. Shen, S. Tan, A. Sordoni, and A. Courville, "Ordered neurons: Integrating tree structures into recurrent neural networks," *arXiv preprint arXiv:1810.09536*, 2018.

[5] D. Chen, A. Fisch, J. Weston, and A. Bordes, "Reading wikipedia to answer open-domain questions," *arXiv preprint arXiv:1704.00051*, 2017.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[7] D. Misra, J. Langford, and Y. Artzi, "Mapping instructions and visual observations to actions with reinforcement learning," *arXiv preprint arXiv:1704.08795*, 2017.

[8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] Z. Dai, Z. Yang, Y. Yang, W. W. Cohen, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-xl: Attentive language models beyond a fixed-length context," *arXiv preprint arXiv:1901.02860*, 2019.

[11] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv preprint arXiv:1807.03819*, 2018.

[12] P. M. Htut, J. Phang, S. Bordia, and S. R. Bowman, "Do attention heads in bert track syntactic dependencies?" *arXiv preprint arXiv:1911.12246*, 2019.

[13] Q. Guo, X. Qiu, P. Liu, Y. Shao, X. Xue, and Z. Zhang, "Star-transformer," *arXiv preprint arXiv:1902.09113*, 2019.

[14] W. Ling, I. Trancoso, C. Dyer, and A. W. Black, "Character-based neural machine translation," *arXiv preprint arXiv:1511.04586*, 2015.

[15] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.
[16] Y. Zhang and J. Yang, "Chinese ner using lattice lstm," *arXiv preprint arXiv:1805.02023*, 2018.
[17] F. Xiao, J. Li, H. Zhao, R. Wang, and K. Chen, "Lattice-based transformer encoder for neural machine translation," *arXiv preprint arXiv:1906.01282*, 2019.
[18] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI Blog*, vol. 1, no. 8, 2019.
[19] D. Britz, A. Goldie, M.-T. Luong, and Q. Le, "Massive exploration of neural machine translation architectures," *arXiv preprint arXiv:1703.03906*, 2017.
[20] N. Peng and M. Dredze, "Named entity recognition for chinese social media with jointly trained embeddings," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 548–554.
[21] H. He and X. Sun, "A unified model for cross-domain and semi-supervised named entity recognition in chinese social media," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[22] S. Zhang, Y. Qin, W.-J. Hou, and X. Wang, "Word segmentation and named entity recognition for sighan bakeoff3," in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 158–161.
[23] J. Chen, Q. Chen, X. Liu, H. Yang, D. Lu, and B. Tang, "The bq corpus: A large-scale domain-specific chinese corpus for sentence semantic equivalence identification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4946–4951.
[24] C. C. Shao, T. Liu, Y. Lai, Y. Tseng, and S. Tsai, "Drcd: a chinese machine reading comprehension dataset," *arXiv preprint arXiv:1806.00920*, 2018.
[25] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "Spanbert: Improving pre-training by representing and predicting spans," *arXiv preprint arXiv:1907.10529*, 2019.
[26] Y. Cui, W. Che, T. Liu, B. Qin, Z. Yang, S. Wang, and G. Hu, "Pre-training with whole word masking for chinese bert," *arXiv preprint arXiv:1906.08101*, 2019.
[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.
[28] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.
[29] M. E. Peters, W. Ammar, C. Bhagavatula, and R. Power, "Semi-supervised sequence tagging with bidirectional language models," *arXiv preprint arXiv:1705.00108*, 2017.
[30] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.
[31] B. Yu, Z. Zhang, T. Liu, B. Wang, S. Li, and Q. Li, "Beyond word attention: using segment attention in neural relation extraction," in *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 5401–5407.
[32] Y.-S. Wang, H.-Y. Lee, and Y.-N. Chen, "Tree transformer: Integrating tree structures into self-attention," *arXiv preprint arXiv:1909.06639*, 2019.
[33] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 933–941.
[34] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *arXiv preprint arXiv:1703.04826*, 2017.
[35] Y. Zhang, Q. Liu, and L. Song, "Sentence-state lstm for text representation," *arXiv preprint arXiv:1805.02474*, 2018.