BiG-Transformer: Intergrate Hierarchical Feature 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 many NLP tasks. Originally fixed fully connected structure leads calculation redundant, granularity fixed and unexplainable. In this paper, we present BiG-Transformer, which replaces fully-connected self-attention in Transformer with attention of bipartite graph structure. Two parts of the graph are designed for fusing different levels of semantic information, two type connections for fusing information from different position. Experiments on four tasks show the BiG-Transformer outperformance compare to standard Transformer and Lattice based Transformer.**

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

I. Introduction

Traditionally, with the help of deep feature extractor, natural language sentence is provided with strong representation. There are two mainly feature extractor in neural language processing (NLP) area: Recurrent Neural Network and Transformer.

Recurrent Neural Network (RNN) based models play an excellent performance in NLP tasks [1]–[4]. RNN process each token in the sentence one by one. Recurrent structure helps them learn a better representation and achieved great success [5]–[7]. But this specific design limits their speed, and cause gradient explode(vanish). Recently, fully attention based models: Transformer [8], become popular in kinds of NLP applications, notably machine translation [8] and language modeling [9]. Some recent works also suggest that Transformer can be an alternative to recurrent neural networks (RNNs) and convolutional neural networks (CNNs) in many NLP tasks, such as GPT [10], BERT [9], Transformer-XL [11] and Universal Transformer [12], because the ability to bidirection and efficiency is inborn.

Although, RNN based models and Transformer liked models have achieved great success in kinds of NLP applications, 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. For example, self-attention is a heavy mechanism with fully-connected constructure. It will introduce more structure information but increase cost of calculation. It's not necessary to interact tokens one by one, and it's unexplainable action after pretrained on enormous free text corpus. To overcome this disadvantage, Star-Transformer [13] replaces the fully-connected structure with a star-shaped topology, in which every two non-adjacent nodes are connected through a shared relay node. This modification decrease the computation complex greatly in the self-attention module. However, a star-shaped topology in star-transformer lead it can not capture words semantic to some extent, because both of its radical connections and ring connection are not take word boundary or syntactic structure into consideration.

Second, many researches be conducted to extract rich semantic information by fusing different granularity token. Many East Asian languages, including Chinese are written without explicit word boundary. Thus, many researches of these languages are factorize lexical into character [14], or subword [15] level, then integrate multi-level semantic information into tokens. A well-known work is Lattice Model [16], [17], [18] introduce Lattice LSTM which add a lexical gate to inject word boundary via a lexical control gate of their modified LSTM. [17] proposes Lattice based Transformer which aim to inject word boundary information into low-level word representation to explore effective word or subword representation in an automatic way during training. But they introduced a complex position relation into attention mechanism which lead to one more step calculation and redundant embedding matrix be trained at the same time.

In order to integrate multi-level semantics and optimize the fully connected structure to reduce unnecessary calculations without losing semantics, in this paper, we propose a tailor-made model called BiG-Transformer. Fig. 1 given a roughly overview of our model.

The standard self-attention mechanism can be regarded as homogeneity graph, where both parts of the graph come from the same granular representation. In our model, we use two parts of bipartite graph to represent information of different granularities. At the same time, these two parts are intrinsically related, because coarse-grained(word) information comes from fine-grained(character) information. Avoid introducing new parameters (such as external word vectors) while fusing multi-granularity information in this way. Second, the original self attention connect characters between each other, and we don't think such a connection is necessary. To reduce the amount of computation without losing semantics, two new connection types were proposed: Dynamic Local Connection and Fix Global Connection. Dynamic Local Connection preserves local semantic information between word and character, and Fix Global Connection makes up for long distance syntactic

* Corresponding author: Tingwen Liu.

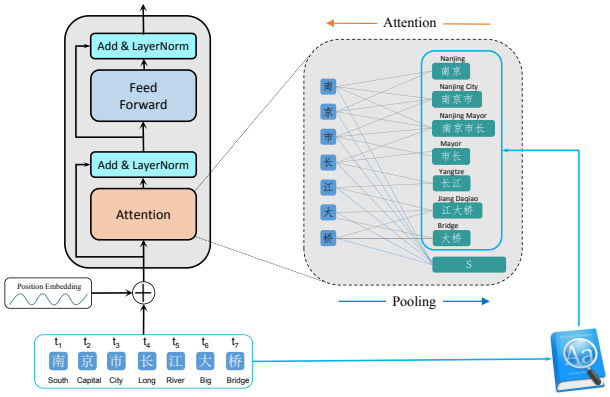

Fig. 1: The overall architecture of our model.

information. In this way, we transform the fully connected structure into a bipartite graph connection structure.

We conducted experiments on four widely-used tasks: Text Classification, Named Entity Recognition, Question Answering and Neural Machine Translation. The proposed model consistently improves performance over the strong Star-Transformer and Lattice Transformer baseline on all tasks, demonstrating its universality and effectiveness. An additional ablation study indicates that two connection type in Big-Transformer are indispensable.

II. Background

A. Self-Attention Mechanism

Self-attention mechanism, as one of the most representative variants of attention model, 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], named entity recognition [10] and etc.

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 input of first layer can be obtained by word embeddings with a 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 transfer 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 matrix and d is the dimension of hidden states. Next, we use $\mathrm{Att}(\cdot)$ to denote a attention model, which can be also implemented as either additive attention or dot-product attention [6], [19]. In this study, we use the dot-product attention, which achieves similar performance with additive counterpart while is much faster and more space-efficient in practice:

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

B. Motivation

Intuitively, the self-attention mechanism could be regard as a fully-connected graph, in which each token is regarded as a node, every node is connected to each other and edges denotes the semantic relationships between two nodes. In fact, for the natural language, words are naturally made up of characters, and there are plenty of inherent dependency relationships between them. These structure information has proved useful for word representation learning []. To be more specific, some asian languages like Chinese typically has no explicit boundary, and a character might has more than one association with different words when using various word segmentation strategies. Look at the example shown in Figure 1, the character "nan" may be a part of words "nan jing", "nan jing shi" or "nan jing shi zhang". It is obvious that words and characters can be divided into two independent parts to highlight the structure information between words and characters, thus an intuitive thinking 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 issue about select a proper subword vocabulary size, which determine the segmentation granularities for downstream tasks.

Furthermore, one advantage of the original self-attention mechanism is that each token in the sequence is global-aware, since all tokens are calculated with each other based on the fully-connected structure. With that in mind, sentence nodes representing the global semantics of sentences also help characters to have global awareness, thus it is very necessary for each character node in the graph to capture long-distance information by interacting with the sentence nodes.

III. Model

In this study, we propose BiG-Transformer, which aim to better extract lexical feature from characters. In this section, we will introduce our model at first, then make detail implementation of the main difference to standard Transformer.

A. Architecture

As shown in Figure 1, the architecture of our model can be simply regarded as a pseudo bipartite-graph, in which one part of the bipartite-graph is character-level representation, while another is word-level representation, and the global information is injected into character nodes by a sentence representation. To connect two parts of pseudo bipartite-graph, we will introduce two type connection: Dynamic Local Connection and Fixed Global Connection.

a) Dynamic Local Connection: The dynamic local connection is 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 is established if a character in the left part is one element of a word in the right part. We use a dictionary to match all possible words contained in a sentence. Every sentence is composed by different words, so for different sentences, the calculation between two parts of the pseudo bipartite-graph is dynamic.

b) Fixed Global Connection: Dynamic local connection focuses on localness information, because all of characters compose words with their neighbors and adjacent characters are usually linked to the same word. Obviously, it results in the semantic loss of long-distance characters. To solve this problem, we insert a global representation node at the right part of the pseudo bipartite-graph. All character nodes on the left are linked to the global sentence representation node.

B. Implementation

Note that the implementation of our model is similar to the standard transformer. Here we focus on the different implementations of these two models.

a) Word and Global Representation: In order to avoid introducing more parameters and keep word and character in the same space, we utilize a simple and effectiveness approach. In detail, given a sequence of character representations $\mathbf{H} \in \mathbb{R}^{n \times d}$, word position tuples: $\{(s_0, e_0), \cdots, (s_m, e_m)\}$, s_i and e_i indicate the index of the word begin position and end position respectively, m is the count of all possible words. we get the word representation following:

$$\mathbf{O}_{i:} = \sum_{j=s_i}^{e_i} \mathbf{h}_j \tag{3}$$

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

The global representation indicates sentence level semantic which is designed for learning long distance information. We employ average pooling as global representation:

$$\mathbf{s}^l = \text{average}(\mathbf{H}^{\text{l}})\mathbf{W}_{\mathbf{s}}^{\mathbf{l}} \tag{4}$$

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

b) Multi-head Self-attention: The standard self-attention mechanism aligns the characters. In our model, given a sequence of character representations, and words set u_i contain the representation of words which make up by i-th character in the sequence. In order to capture more semantic information, we utilize multi-head attention with p heads. we calculate the word-to-character multi-head attention as follows:

$$\text{MultiHeadAtt}(\mathbf{h}_{\text{i}}, \mathbf{H}) = [\mathbf{r_1} \oplus \cdots \oplus \mathbf{r_p}]\mathbf{W}_{\text{O}} \tag{5}$$

$$\mathbf{r}_j = \text{Att}_{\text{j}}(\mathbf{h}_{\text{i}}\mathbf{W}_{\text{j}}^{\text{Q}}, [\mathbf{U}_{\text{i}} \oplus \mathbf{s}]\mathbf{W}_{\text{j}}^{\text{K}}, \mathbf{H}\mathbf{W}_{\text{j}}^{\text{V}}),\ \text{j} \in [1, \text{p}] \tag{6}$$

$$\mathbf{U}_i = [u_i^0 \oplus \cdots \oplus u_i^{|u_i|}] \tag{7}$$

where $\oplus$ denotes the concatenation operation, $\mathbf{W}^Q$, $\mathbf{W}^K$, $\mathbf{W}^V$ and $\mathbf{W}^O$ are trainable transfer matrix.

Algorithm 1 The Update of Our Model

Input: A sequence of characters embedding $X = \{x_1, \cdots, x_n\}$, word position tuple $\{(s_0, e_0), \cdots, (s_m, e_m)\}$.
Output: Representation of each character h_i^L.

$\mathbf{h}_1^0, \cdots, \mathbf{h}_n^0 \leftarrow \mathbf{x}_1, \cdots, \mathbf{x}_n$.
$\mathbf{s}^0 \leftarrow average(\mathbf{H}^0)\mathbf{W}_s^0$
for l from 1 to L **do**
  // word representation.
  for j from 0 to m **do**
    $\mathbf{v}_j^l = sum(\mathbf{H}_{s_j:e_j})$
  end for
  // multi-head self attention.
  for i from 1 to n **do**
    $\mathbf{O}^{l-1} = [\mathbf{V}^{l-1}, \mathbf{s}^{l-1}]$
    $\mathbf{h}_i^l = MultiHeadAtt(\mathbf{h}_i^{l-1}, \mathbf{O}^{l-1}, \mathbf{h}_i^{l-1})$
  end for
  $\mathbf{H}^l = LayerNorm(\mathbf{H}^{l-1})$
  // global representation.
  $\mathbf{s}^l = average(\mathbf{H}^l)\mathbf{W}_s^l$
end for

c) Position Embedding: Transformer-based models usually utilize position embedding to incorporate the sequence information. In this study, we also adopt the same strategy with standard transformer:

$$p(j, 2i) = \sin(j/100000^{2i/d}) \tag{8}$$

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

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.

d) Overall: The overall update algorithm of our model is shown in the 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 updating 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 does it works

Character based self-attention mechanism has been proved effectiveness in many NLP tasks. However, these character-based methods still have its disadvantages: they are different to explain intuitively why it works, 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 can form the word locally, and the second is the whole semantic relationship . In addition, a word is 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. Experiment

A. Tasks and Datasets

To demonstrate the effectiveness of our model, we evaluate our model for four mainly tasks which contains ten datasets total:

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

For detail data statistic, please see Table I.

B. Setting

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

C. Text Classification

The text classification task include three types corpus in this paper: Sentiment Classification(ChnSentiCorp, Sina Weibo), Sentence Pair Matching(BQ Corpus), Document Classification(THUCNews). For all experiments of text classification, we pooling the sequence vector in sentence, and feed them into as the final representation and feed it into the softmax classifier. Results are shown in Table II

For word2vec based word embedding initializaion approach, our model outperforms all of them on 4 datasets within Transformer based models, this demonstrates the effectiveness of our proposed attention mechanism. It is worthy to notice that LSTM gets better results compared with Transformer and Star-Transformer but lower than our on ChnSenti, Weibo and THUCNews datasets, this shows that attention based model still need better structure to tap its potential. Our model beats all other model in ChnSenti datast, achieves 1.33 points to best baseline(LSTM). ChnSenti is a small corpus which shows that our model could extract feature better from a small dataset. Our model also perform well Weibo dataset which is the largest one, but slightly lower than LSTM on BQ which is sentence pair matching dataset, we speculate that is by reason of spatial information is more important in matching task.

When initialize the embedding layer with Chinese Bert[3], Our model still outperform 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 char based model. We think these results agree with recently works of span based pre-trained language models [25], [26].

D. Machine Reading Comprehension

In this section, we compare our model with baselines to show the nature language understand ability on MRC task. Table V shows the results.

For all three baseline, we obtain better performance on both EM and F1 scores. Transformer liked models initialized by Bert still outperform LSTM, this conclusion keep consistent to we mentioned in section IV-F. Compare within Transformer liked models, we got close results between standard Transformer and Star-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. Machine Translation

For a fair comparison to Lattice-Transformer, we introduce machine translation experiments. All baseline results are from [17]. From Table III, we see that our model outperform Transformer and Lattice Transformer by 0.88 and 0.30 BLEU in the overall performance. When compare to Results give a strong evidence to show our model's effectiveness.

F. Named Entity Recognition

To verify the ability of our model in sequence labeling, we choose the classic NER task. We follow Lattice LSTM [18] adopting the four frequently-used dataset to show our model effectiveness[4].

From Table IV, we could see that our word char attention based model achieve state-of-the-art performance on all four datasets. This demonstrate the same conclusion with Lattice LSTM that lexical information is really important for named entity recognition task. Comparing to Transfromer and Star-Transfromer, our model beats both of them on no matter

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

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

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

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

TABLE I: An overall of datasets and its hyper-parameters in our model, "HEAD" indicates the number of heads in the Multi-head attention. Parameters in text classification task divided by "/", it denotes that bert based and word2vec based word embedding initialization method. There is no "MAXLEN" for NER because the task particularity.

Dataset	TASK	MAXLEN	BATCH	LR	HEAD	EPOCH	TRAIN	DEV	TEST
Weibo-NER	NER	-	16	1e-05	8	15	1,350	270	270
MSRA	NER	-	12	1e-05	8	15	46,306	4,361	4,361
Resume	NER	-	12	3e-05	8	5	3,821	463	477
Ontonotes	NER	-	16	1e-05	8	15	15,717	4,298	4,345
ChnSenti	SC	256	16/128	3e-05/7e-04	6	3/20	9,601	1,201	1,201
BQ	SPM	128	64/128	1e-05/7e-04	6	3/20	100,001	10,001	10,001
THUCNews	DC	512	16/128	1e-05/7e-04	6	3/20	50,001	5,000	10,001
Weibo	SC	128	16/128	2e-05/7e-04	6	3/20	100,001	9,989	10,001
DRCD	MRC	512	16/10	3e-5	6	-/2	26,932	3,524	3,485

TABLE II: Results on the Text Classification datasets, bold marks highest number among all models. There are two ways to initiate word embedding layer: Word2vec(top) and Bert(Bottom)

	ChnSenti		Weibo		BQ		THUCNews	
Model	dev	test	dev	test	dev	test	dev	test
LSTM	**89.58**	88.17	**94.92**	95.38	**68.82**	**67.19**	77.81	79.21
Transformer	87.33	86.17	94.73	95.19	67.49	65.24	72.41	72.90
StaTransformer	84.25	81.83	94.66	95.04	65.85	65.34	77.23	78.77
Our	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
StaTransformer[Bert]	93.08	93.58	94.71	97.63	**85.54**	83.34	95.88	96.65
Our[Bert]	**94.90**	**94.92**	**97.52**	**98.04**	84.40	**83.69**	**98.01**	**97.55**

TABLE III: Evaluation of translation performance on NIST Zh-En dataset. RNN, Lattice-RNN and Lattice-Transformer results are 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
Our	Matching	**43.35**	43.84	**44.10**	44.11	**42.56**	**33.78**	**41.23**

large(MSRA, Ontonotes) or small dataset, it follows that our model helps learning a better character representation via lexical information. Interestingly, Transformer achieves better performance than LSTM on all four datasets. We conjecture that this is because Bert has compatibility with Transformer instead of LSTM model.

G. Ablation Study

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

We test three variants of our models in three dataset ChnSenti(Text Classification), Ontonotes(NER) and DRCD(MRC). The first one, we remove dynamic connection, keep the global connection. As shown of the Table VI in the first line, without the dynamic local connection, all performance of three dataset declined, demonstrating that our dynamic connection between word and character can fuse different level information into character representation. For the second, we remove the global connection to prove the effectiveness of the 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 adjacent connection. comparing to our model, it performs worse on three of the

TABLE IV: Results on the NER datasets, bold marks highest number among all models. We don't report Transformer, Star-Transformer and BiG-Transformer's results which initiating embedding layer using Context-free embedding, because there is no result being 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
Our(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**

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

TABLE V: Results on the DRCD dataset, bold marks highest number among all models.

Model	ChiSenti Acc	Ontonotes F1	DRCD F1
Our	89.50	80.33	86.23
w/o global	87.21	72.21	82.11
w/o local	80.38	50.10	73.01

TABLE VI: Results on the DRCD dataset, bold marks highest number among all models.

model.

V. Related Work

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

a) Word-Level Compositionality: As word is the fundamental unit of semantic expression, the most intuitive approach to represent a word to vector [27], but to deal with Out-of-vocabulary problem and learn more semantic relation between words, subword level models are proposed. For example, [28] use a character LSTM to solve obtaining competitive results for NER task. [29], [30] pretrain the subword enrich word representation to improve its performance on kinds NLP tasks. [18] design a novel lattice LSTM representation for mixed characters and lexicon words by add 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 integrate global-level semantic information into words. There mainly divid into three parts: recurrent-based, CNN-based and attention-based. Hierarchically stacked CNN layers [33], [34] allows better interaction 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 model get great success because of its bidirection 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 original fully connected self attention constructure in the Transformer. Experiments base on three tasks demonstrated the effectiveness. In other language, there is segment operation divide lexical level word into character or n-grams. So for the future work, we will extend our model to subword or character base approach in other languages.

Acknowledgments

The preparation of these instructions and the LaTeX and BibTeX files that implement them was supported by Schlumberger Palo Alto Research, AT&T Bell Laboratories, and Morgan Kaufmann Publishers. Preparation of the Microsoft Word file was supported by IJCAI. An early version of this document was created by Shirley Jowell and Peter F. Patel-Schneider. It was subsequently modified by Jennifer Ballentine and Thomas Dean, Bernhard Nebel, Daniel Pagenstecher, Kurt Steinkraus, Toby Walsh and Carles Sierra. The current version has been prepared by Marc Pujol-Gonzalez and Francisco Cruz-Mencia.

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] 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.

[11] 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.

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

[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] J. Su, Z. Tan, D. Xiong, R. Ji, X. Shi, and Y. Liu, "Lattice-based recurrent neural network encoders for neural machine translation," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[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] Y. Zhang and J. Yang, "Chinese ner using lattice lstm," *arXiv preprint arXiv:1805.02023*, 2018.

[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.