LOCAL CONTEXT INTERACTION-AWARE GLYPH-VECTORS FOR CHINESE SEQUENCE TAGGING

Junyu Lu, Pingjian Zhang[†]

School of Software Engineering, South China University of Technology, Guangzhou, China

ABSTRACT

As hieroglyphics, Chinese characters contain rich semantic and glyphs information, which is beneficial to sequence tagging task. However, it's difficult for shallow CNNs architecture to extract glyphs information from character data and implement the contextual interaction of different glyphs information effectively. In this paper, we address these issues by presenting **LCIN**: a **L**ocal **C**ontext **I**nteraction-aware **N**etwork for glyph-vectors extraction. The network utilizes depthwise separable convolution and attention machine to extract glyphs information from images of Chinese characters. Moreover, we interconnect adjacent attention blocks so that glyphs information can flow within the local context. Experiments on three subtasks for sequence tagging show that our method outperforms other glyph-based models and achieves new SOTA results in a wide range of datasets.

***Index Terms*—** Sequence tagging, hieroglyphic information, convolution neural network, attention mechanism

1. INTRODUCTION

Sequence tagging refers to assigning a categorical label to each unit of a sequence of observed values, whose effect can intuitively reflect the adequacy and validity of underlying semantic information [1].

Chinese is a kind of hieroglyphics, which is the graphic depiction of the objects, and the glyph is a representation of the composition and structure of Chinese characters. The strokes, structures and radicals of Chinese characters encode rich semantic information and representational meaning [2]. For example, "您 (the honorific of you)" is made up of "你 (you)" and "心 (heart)", and "你 (you)" is on the top of "心 (heart)". This structure indicates that "您 (the honorific of you)" is a kind of honorific title to the speaking object and show respect with your heart. Moreover, the characters "燃 (burning)", "烽 (beacon)", "煤 (coal)" all have the radical "火 (fire)", symbolizing that these characters are closely related to fire in semantics. Therefore, it is intuitive that introducing glyph information into Chinese sequence tagging task will bring benefits. Many current studies also support this argument: radical and glyph features play a useful role in natural language understanding task and the training process of Chinese character/word embedding [3–8]. Moreover, introducing glyph information into the models of sequence tagging can improve performance effectively [9–12].

In early researches, to extract visual features from glyphs, many studies regard Chinese characters as images, and focus on applying the CNN-based model to encode glyph information from character images. However, because of the inappropriate design of models structure and insufficient training strategy, the performance of those models is not improved consistently and effectively [13, 14]. Wu et al. [11] point out that not using the correct versions of scripts can also cause negative results and proposes Glyce model to resolve the above problems. They use the Tianzige-CNN structures to encode the ensemble of the historical and the contemporary scripts, enriching pictographic information from the character images through different writing styles, and introduce image character classification task for multi-task learning. However, the Glyce model still has some shortcomings: (1) In the evolution of Chinese characters, the font structure of Chinese characters varies greatly in different dynasties, and the strokes that need to be paid attention to are also different. A single CNNs structure cannot fully extract the complete potential semantics. (2) It only encodes the static distributed representation of current glyph independently and ignores the interactive information between glyphs contexts, which has been extensively studied in the field of multimodal deep learning and proved to be useful [15, 16]. (3) In the choice of the auxiliary task in multi-task learning, neither the image character classification [11] nor the image dimensionality reduction [10] can establish a sound relationship between the semantic information and character images.

In this paper, to resolve the aforementioned issues, we propose LCIN: a Local Context Interaction-aware Network for glyph-vectors extraction. In summary, we make the following contributions:

- We use a combination of the historical and contemporary scripts, and utilize depthwise separable convolution (DSC) to achieve encoding separation and feature aggregation of different scripts.
- In LCIN, we adopt attention mechanism to emphasize meaningful glyph features along the channel and spatial axes, and propose a context feature refinement and interaction module to integrate contextual glyphs information.
- We propose an auxiliary training task of image classification based on objective entity categories, and add its loss function to the sequence tagging task for multi-task learning.

Experiments illustrate that the proposed LCIN achieves significant improvement in a wide range of sequence tagging datasets and achieves SOTA performance.

2. METHODOLOGY

As shown in Figure 1, the proposed network LCIN takes the stacked character image as input, and extracts lower-level graphic feature through DSC and max-pooling. The scripts are shown in Table 1. The lower-level graphic feature passes through the context feature refinement and interaction module and loop four times to obtain glyph-vector. The dimension of a character image is

[†]Corresponding author (pjzhang@scut.edu.cn). This work was supported by National Natural Science Foundation of China (62076100).

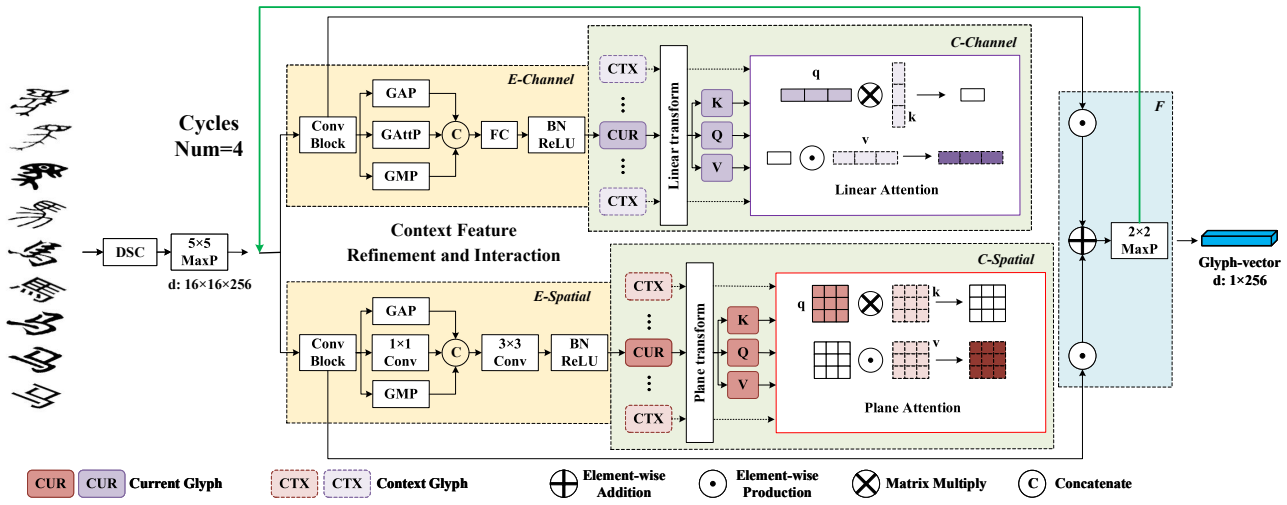

Fig. 1. Overall architecture of LCIN

$80\times80\times3N_{script}$, N_{script} is the number of scripts we use.

Table 1. scripts used in LCIN

Script	Time Period / Dynastic
Oracle Bone	Shang
Bronze	Shang, Zhou
Ancient/Small Seal	Spring and Autumn, Qin
Clerical	Han
Regular Simplified/Traditional	Eastern Han – Now
Running Simplified/Traditional	Wei and Jin – Now

2.1. Lower-level Graphic Feature Extraction

The input image x_{image} is first passed through a DSC, which consists of a group convolution of kernel size 5, group number N_{script} and output channels 512, and a 1×1 convolution of output channels 256. Group convolution can divide the input stacked images into groups according to different scripts, which is beneficial to learn the feature information of different scripts respectively. And the 1×1 convolution can aggregate valid features on the channel dimension. Finally, we apply a max-pooling of kernel size 5 on the feature map to capture the lower-level graphic feature, which reduces the resolution from 80×80 to 16×16. Since the character image is characterized by sparse features (i.e., a large number of white pixels), we choose a larger kernel size for the first max-pooling.

2.2. Context Feature Refinement and Interaction

Unlike the common images, character images are characterized by small scale, sparse pixel points and abstract content, which makes it difficult to extract effective glyph information using deep convolution networks. Therefore, inspired by current works on image classification [17] that encapsulate and formulate attention modules and their components, we propose a generalized attention framework for context glyphs feature refinement and interaction, which consists of three components: extraction, connection and fusion. Extraction serves as a feature refiner to obtain intermediate attention weight. Connection connects the current and contextual refined features, and then transforms them to a new attention space. Fusion merges attention and input feature maps. The E, T, F modules in Figure 1 show the context feature refinement and interaction modules in the LCIN.

Extraction is designed for gathering attention maps from the channel and spatial axes of the input graphic feature. For a given input graphic feature map $F_I \in \mathbb{R}^{C\times L\times L}$, we first transfer it to a convolution block for feature encoding to obtain the feature map $F \in \mathbb{R}^{C\times L\times L}$. The kernel size of the convolution block is 5 in the first two cycles and 3 in the last two cycles. Then, we utilize global attention pooling and 1×1 convolution in the channel and spatial axes respectively, and combine them with global max-pooling and global avg-pooling as feature detectors. The global attention pooling utilizes an attention map to aggregate the spatial information. Finally, we concatenate the deduced feature map along the channel axes and apply a non-linear transformation to calculate channel attention $F^C \in \mathbb{R}^{C\times1\times1}$ and spatial attention $F^S \in \mathbb{R}^{1\times L\times L}$:

$$F^C = FC(Concat[GMP(F), GAP(F), GAttP(F)]) \quad (1)$$

$$F^S = f^{3\times3}(Concat[GMP(F), GAP(F), f^{1\times1}(F)]) \quad (2)$$

Connection applies the multi-head attention mechanism as Transformer [18] to extracted attention features to implement the interaction of contextual glyphs information. We concatenate the F^C, F^S of current and context glyphs to obtain keys, values and queries $Q_C, K_C, V_C \in \mathbb{R}^{l\times C}$ and $Q_S, K_S, V_S \in \mathbb{R}^{l\times L\times L}$. Projection matrices $W_C^Q, W_C^K, W_C^V \in \mathbb{R}^{C\times d_C}$ and $W_S^Q, W_S^K, W_S^V \in \mathbb{R}^{L\times d_L}$ are used to transform linear and plane KV-Q pairs into new attention space, where d_C, d_L=C/h, L/h. We employ $l = 7$ to represent the local context window size and $h = 8$ for parallel attention heads. In the calculation of the attention value of the context glyph, we apply a position weight function $\sigma(\cdot)$ so that the context glyph near the center gets more attention:

$$\begin{aligned} \text{Att}\,(q_{cur}, k_{ctx}, v_{ctx}) &= \sigma(cur, ctx)\frac{q_{cur}k_{ctx}^T}{\sqrt{l}}v_{ctx} \\ \text{where } \sigma(cur, ctx) &= \cos\left(\frac{\pi}{2}\cdot\frac{cur-ctx}{4}\right) \end{aligned} \quad (3)$$

Fusion integrates the attention map $\widetilde{F^C}$, $\widetilde{F^S}$ output in the connection component with the input graphic feature. We perform an

element-wise production between $\widetilde{F^C}$, $\widetilde{F^S}$ and F_I, add the results and transfer them to a max-pooling of kernel size 2. The output feature map $F^{'}$ can be expressed as:

$$F' = \text{maxpool}\left(\widetilde{F^C} \otimes F_I \oplus \widetilde{F^S} \otimes F_I\right) \quad (4)$$

After the lower-level graphic feature map pass through the attention framework and loop four times, we get the glyph-vector h_{image} of current character image.

2.3. Image Classification as an Auxiliary Task

To further prevent overfitting and enhance character semantics, we propose a task of image classification based on objective entity categories. The glyph-vector h_{image} from LCIN will be forwarded to a classification layer to predict its corresponding categories, which are formulated according to entity categories (such as PER, LOC, and ORG) in the NER task. In this case, the target labels of the character image classification are determined based on the entity statistical distribution of the character in the corpus. The training object for the image classification task $\mathcal{L}(cls)$ is given as below:

$$\mathcal{L}(cls) = -\log softmax\left(h_{image} \times W_{cls} + b_{cls}\right) \quad (5)$$

Let $\mathcal{L}(seq)$ denote the training object of the sequence tagging task. We use a trade-off function $\lambda(t)$ to linearly combine $\mathcal{L}(seq)$ and $\mathcal{L}(cls)$, making the final training object as follow:

$$\mathcal{L} = (1 - \lambda(t))\mathcal{L}(seq) + \lambda(t)\mathcal{L}(cls) \quad (6)$$

$\lambda(t)$ is a two-stage function of epoch t: given a threshold τ,when $\lambda(t)>\tau$, $\lambda(t)=\lambda_0\lambda_1^t$, where $\lambda_0\in[0,1]$ denotes the starting value and $\lambda_1\in[0,1]$ denotes the decaying value. When $\lambda(t)\leq\tau$, we will keep $\lambda(t)=\tau$.

2.4. Combining Glyph-vectors with BERT

As a large-scale pre-trained language model, BERT [19] has a remarkable effect on sequence tagging task. We explore the method of combining LCIN and BERT as Figure 2. For a given input sentence S, we forward each token of S to BERT and use output of the last layer of the BERT transformer to represent the final hidden state of the token as h_{hidden}. Then, we transfer the character image corresponding to each token of S to the LCIN to obtain the glyph-vector h_{image}. Finally, we concatenate the h_{image} and h_{hidden} to obtain the complete representation of a token, and transfer it to the BiLSTM-CRF decoder to obtain the predicted label.

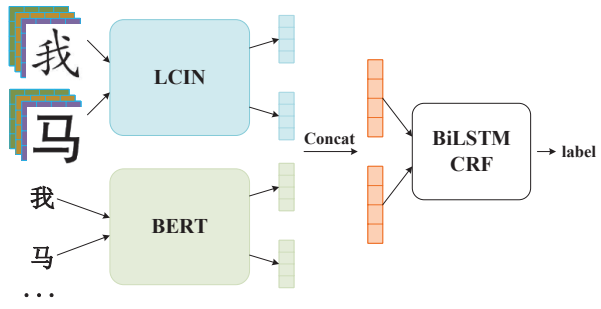

Fig. 2. Combining glyph information with BERT

Table 2. Results on Chinese NER

OntoNotes			
model	P	R	F
FLAT+BERT	-	-	81.82
Lattice-LSTM	76.35	71.56	73.88
Glyce+Lattice-LSTM	82.06	68.74	74.81
LCIN+Lattice-LSTM	**80.82**	**72.66**	**76.52**
			(+2.64)
BERT	78.01	80.35	79.16
Glyce+BERT	81.87	81.40	81.63
LCIN+BERT	**83.31**	**82.26**	**82.78**
			(+3.62)
MSRA			
model	P	R	F
FLAT+BERT	-	-	96.09
Lattice-LSTM	93.57	92.79	93.18
Glyce+Lattice-LSTM	93.86	93.92	93.89
LCIN+Lattice-LSTM	**94.65**	**94.39**	**94.52**
			(+1.34)
BERT	94.97	94.62	94.80
Glyce+BERT	95.57	95.51	95.54
LCIN+BERT	**97.01**	**96.47**	**96.74**
			(+1.94)

Table 3. Results on CWS

PKU			
model	P	R	F
Huang et al. [20]	-	-	96.7
BERT	96.8	96.3	96.5
Glyce+BERT	97.1	96.4	96.7
LCIN+BERT	**97.94**	**96.82**	**97.38**
			(+0.88)

Table 4. Results on Chinese POS

CTB6			
model	P	R	F
Lattice-LSTM	92.00	90.86	91.43
Glyce+Lattice-LSTM	92.72	91.14	91.92
LCIN+Lattice-LSTM	**93.84**	**91.63**	**92.72**
			(+1.29)
BERT	94.91	94.63	94.77
Glyce+BERT	95.56	95.26	95.41
LCIN+BERT	**96.47**	**95.61**	**96.04**
			(+1.27)

3. EXPERIMENT

3.1. Experiment Setting

Datasets We evaluate our method on four datasets of three different Chinese sequence tagging tasks, including named entity recognition (NER), Chinese word segmentation (CWS), and part-of-speech tagging (POS). We conduct experiments on OntoNotes and MSRA

benchmark [21, 22] datasets in NER, which are in the news domain. For CWS and POS, we employ the benchmark datasets PKU [23] and CTB6 [24] respectively. We use the official splits of train/dev/test in our experiments.

Implementation Details In our experiments, we use the pre-trained Chinese-BERT, which constructed based on BERT-base [19] with 12 layers of transformer. We choose Adam for optimization, and fix the initial learning rate at 4e-5 for BERT and BiLSTM, 2e-4 for LCIN model and CRF decoder. The max length of the sequence is set to 512, and the training batch size is set to 16. A maximum epoch number of 30 is used for training all datasets. For the hyperparameters, we set $\lambda_0 = 0.95$, $\lambda_1 = 0.8$ and $\tau = 0.1$.

3.2. Experiment Results

In NER and POS tasks, except for BERT, we also select the widely used non-BERT model Lattice-LSTM [25] as baseline. Meanwhile, we compare our model with another glyph-based model Glyce, which uses shallow CNNs structure to extract glyph information, and other SOTA models on four datasets. The main results for NER, CWS and POS are respectively shown in Tables 2,3 and 4.

Experiment results show that both BERT and non-BERT models achieve stable and significant improvement in all datasets after being combined with LCIN. The LCIN+BERT model outperforms Glyce+BERT model and sets new SOTA results across all datasets. It manifests that compared with shallow CNNs structure, LCIN can incorporate glyph information more effectively.

In NER task, the F1 scores of LCIN+BERT outperform BERT model by 3.62% and 1.94% on OntoNotes and MSRA datasets respectively. We deduce that this significant improvement is mainly due to the introduction of the auxiliary task of character image classification, which makes the glyph-vectors establish a strong association with the objective entities.

Meanwhile, the introduction of LCIN brings more promotion in precision than recall. We conclude this is because the semantic complement brought by the glyph-vectors leads the decoding process to predict in a higher confidence direction.

Moreover, as shown in Table 5, we compare the prediction results of BERT and LCIN+BERT for Out-of-Vocabulary (OOV) entities in the NER datasets. The significant performance improvement shows that the heuristic bootstrap of glyphs information enables some never-seen entities to be correctly identified, probably because the perception of radical features and inference of similar structures are beneficial for ambiguous contexts.

Table 5. Results on OOV entities in Chinese NER

Model	OntoNotes			MSRA		
	P	R	F	P	R	F
BERT	64.75	65.42	65.08	85.97	84.83	85.41
LCIN+BERT	**72.24**	**71.69**	**71.96**	**91.07**	**89.52**	**90.29**

3.3. Ablation Studies

We discuss the influence of different factors of LCIN. We use OntoNotes dataset for illustration. The factors we investigate contain training strategy and model architecture.

Training Strategy We denote our training tactic as Joint-Weight, which fine-tune BERT and LCIN jointly according to the predefined weighting function until convergence. We compare our strategy with other tactics, including (1) Joint-Average strategy, in which we directly jointly training BERT and LCIN with the same weight. (2) Glyph-Joint strategy: in which we freeze BERT to tune the LCIN in the first five epochs, and then jointly tune both models until convergence. (3) We also compare the ordinary image classification as an auxiliary task, which predicts the charID of the image. We denote it as Ord-image-cls.

Results are shown in Table 6. As we can see, the Joint-Weight outperforms the rest three strategies. This indicates that at the early stage of training, we need to learn more semantic information about glyphs from the image classification task based on objective entity categories, and as training proceeds, the proportion of the auxiliary task should be reduced to a relatively low threshold so as not to affect the training process of the main task. We speculated the inferior performance of Joint-Average is as follows: In the early training, the BERT is pre-trained but the parameters of LCIN are randomly initialized. The fitting process of BERT could be misled by the invalid information in glyph-vectors. Moreover, instead of applying the Ord-image-cls task, LCIN can fully learn semantic information related to objective entities by attaching the entity categories to the image classification.

Table 6. Impact of different training strategies

Strategy	P	R	F
Joint-Weight	**83.31**	**82.26**	**82.78**
Joint-Average	79.44	82.78	81.08
Glyph-Joint	81.18	83.02	82.10
Ord-image-cls	82.15	81.39	81.77

Model Architecture We explore the role of the connection component in LCIN, and compare the results of replacing LCIN with shallow CNNs architectures: Tianzige-CNN and CGS-CNN. Results are shown in Table 7. As can be seen, the performance of LCIN deteriorates significantly without connection component. Shallow CNNs architectures only achieve limited improvements after combining with BERT. This indicates that after introducing the attention mechanism and context glyphs information interaction structure, the glyph-vectors extracted from character images contain richer semantic information.

Table 7. Impact of different model architecture

Model architecture	P	R	F
LCIN	**83.31**	**82.26**	**82.78**
LCIN w/o Connection	82.68	81.55	82.11
Tianzige-CNN	82.23	81.38	81.80
CGS-CNN	82.46	81.32	81.89

4. CONCLUSION

In this paper, we propose a local context interaction-aware network for glyph-vectors extraction, which utilizes attention machine to implement the refinement and interaction of context glyphs, overcoming the difficulty of shallow CNNs for extracting glyph information. We introduce an auxiliary task of image classification based on objective entity categories to tune LCIN and combine it with BERT for Chinese Sequence tagging. Experiments demonstrate the capability of our methods on glyph-vectors extraction.

References

[1] Jurafsky Daniel and H Martin James, "Speech and language processing," 2000.

[2] Yun Zhang, Yongguo Liu, Jiajing Zhu, Ziqiang Zheng, Xiaofeng Liu, and Weiguang Wang, "Learning chinese word embeddings from stroke, structure and pinyin of characters," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1011–1020.

[3] Xinlei Shi, Junjie Zhai, Xudong Yang, and Zehua Xie, "Radical embedding: Delving deeper to chinese radicals," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 2015, pp. 594–598.

[4] Yanran Li, Wenjie Li, Fei Sun, and Sujian Li, "Component-enhanced chinese character embeddings," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 829–834.

[5] Shaosheng Cao, Wei Lu, Jun Zhou, and Xiaolong Li, "cw2vec: Learning chinese word embeddings with stroke n-gram information," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[6] Hong-You Chen, Sz-Han Yu, and Shou-De Lin, "Glyph2vec: Learning chinese out-of-vocabulary word embedding from glyphs," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2865–2871.

[7] Zijun Sun, Xiaoya Li, Xiaofei Sun, Yuxian Meng, and Xiang Ao, "Chinesebert: Chinese pretraining enhanced by glyph and pinyin information," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 2021, pp. 2065–2075.

[8] Yun Zhang, Yongguo Liu, Jiajing Zhu, and Xindong Wu, "Fsprm: A feature subsequence based probability representation model for chinese word embedding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1702–1716, 2021.

[9] Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre, "Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing*, 2017, pp. 173–183.

[10] Chan Hee Song and Arijit Sehanobish, "Using chinese glyphs for named entity recognition (student abstract)," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, vol. 34, pp. 13921–13922.

[11] Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, and Muyu Li, "Glyce: Glyph-vectors for chinese character representations," *Advances in Neural Information Processing Systems*, vol. 32, pp. 2746–2757, 2019.

[12] Zhenyu Xuan, Rui Bao, and Shengyi Jiang, "Fgn: Fusion glyph network for chinese named entity recognition," in *China Conference on Knowledge Graph and Semantic Computing*. Springer, 2020, pp. 28–40.

[13] Falcon Dai and Zheng Cai, "Glyph-aware embedding of chinese characters," in *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, 2017, pp. 64–69.

[14] Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig, "Learning character-level compositionality with visual features," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 2059–2068.

[15] Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang, "Adaptive co-attention network for named entity recognition in tweets," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[16] Jianfei Yu and Jing Jiang, "Adapting bert for target-oriented multimodal sentiment classification," in *IJCAI*, 2019.

[17] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[20] Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu, "Towards fast and accurate neural chinese word segmentation with multi-criteria learning," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 2062–2072.

[21] Ralph Weischedel, Sameer Pradhan, Lance Ramshaw, Martha Palmer, et al., "Ontonotes release 4.0," *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2011.

[22] Gina-Anne Levow, "The third international chinese language processing bakeoff: Word segmentation and named entity recognition," in *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, 2006, pp. 108–117.

[23] Thomas Emerson, "The second international chinese word segmentation bakeoff," in *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*, 2005.

[24] Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer, "The penn chinese treebank: Phrase structure annotation of a large corpus," *Natural language engineering*, vol. 11, no. 2, pp. 207–238, 2005.

[25] Yue Zhang and Jie Yang, "Chinese ner using lattice lstm," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 1554–1564.