FGN: Fusion Glyph Network for Chinese Named Entity Recognition

Zhenyu Xuan, Rui Bao and Shengyi Jiang

School of Information Science and Technology, Guangdong University of Foreign Studies, Guangdong, China

xuanzhenyu@foxmail.com, jiangshengyi@163.com

Introduction

In the field of Chinese NER, researches generally adopt character-based tagging strategy to label named entities. Currently, distributed representation learning has become the mainstream method to represent Chinese characters. However, these methods overlooked the information inside words or characters like Chinese glyph. For example, character "抓"(grasp) is constitutive of "扌"(hand) and "爪"(claw). Some researchers tried regarding Chinese characters as graphs for glyph encoding and used CNNs to capture glyph information in character graphs. Meng et al. proposed a glyph-based BERT model called Glyce, which achieved SOTA performances in various NLP tasks including NER. They adopted Tianzige-CNN to encode seven historical and contemporary scripts of each Chinese character. However, the previous works ignored the interactive knowledges between glyphs and contexts, which have been studied in the field of multimodal deep learning. we propose the FGN, Fusion Glyph Network for Chinese NER. The major innovations in FGN include:

(1) a novel CNN structure called CGS-CNN, Character Graph Sequence CNN is offered for glyph encoding. CGS-CNN may capture potential information between the glyphs of neighboring characters.

(2) We provide a fusion method with out-of-sync sliding window and Slice-Attention to capture interactive knowledge between glyph representation and character representation.

Model

In this section, we introduce the FGN in detail. As shown in Fig. 1, FGN can be divided into three stages: representation stage, fusion stage and tagging stage. We follow the strategy of character-based sequence tagging for Chinese NER.

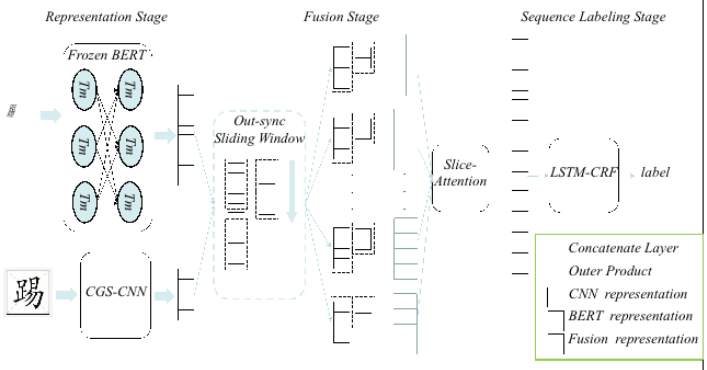

Fig. 1. Architecture of the FGN for named entity recognition

BERT. Different from the normal fine-tuning strategy, we first fine-tune BERT on training set with a CRF layer as tagger. Then freeze the BERT parameters and transfer them to FGN.

CGS-CNN. Fig. 2 depicts the architecture of CGS-CNN.We transform the input sentences to graph sequences for 3D encoding. 3D encoding may capture the interactive knowledge between Adjacent glyphs.

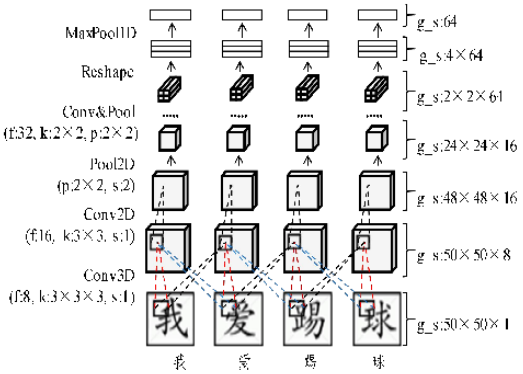

Fig. 2. Architecture of CGS-CNN with a input sample "我爱踢球" (I love playing football). "f", "k", "s", "p" stand for kernel number, kernel size, stride, and pooling window size. "g_s" represents the tensor size of output from each layer.

Out-of-sync Sliding Window. Sliding window has been applied in multimodal affective computing [3] as mentioned above. The reason for using sliding windows is that directly fusing vectors with outer product would exponentially expand vector size, which increases space and time complexity for subsequent network. However, this method requires the multimodal representations to have the same size, which is not suitable to slide through both BERT vector and glyph vector. Because character representations of BERT have richer semantic information than glyph representations, requiring a bigger vector size. Here we provide an out-of-sync sliding window that can satisfy different vector sizes while keeping the same number of slices.

Slice-Attention. Outer product offers interactive information for character-level rep-resentation at the same time generates more noises, as many features are irrelevant. With reference to attention mechanism, we propose the Slice-Attention, which may adaptively quantify the importance of each slice pair and combined them to represent a character.

Tagging Stage. We concatenate each vector in character-level before tagging. The final representation of a sentence can be defined as , where stands for the length of sentence. Then BiLSTM is adopted as sequence encoder and CRF is adopted as decoder for named entity tagging.

Model

Four widely-used NER datasets are chosen for experiments, including OntoNotes 4, MSRA, Weibo and Resume. Table 1 and Table 2 show some detailed statistics of FGN, which is compared with other SOTA models on four NER datasets. Here FGN represents the proposed glyph model with LSTM-CRF as tagger; Lattice LSTM and WC-LSTM are the SOTA model without BERT, combining both word embedding and character embedding. BERT-LMCRF represent the BERT model with BiLSTM-CRF as NER tagger. Glyce [1] is the SOTA BERT-based glyph network as mentioned earlier. GlyNN [2] is another SOTA BERT-based glyph network. For example, some interesting and extraordinary entity words in Weibo and OntoNote4 like "铼德" (company name) and "啊滋猫" (milk tea shop), which were successfully identified only by FGN. We guess the reason is because the character "铼" contain the radical "钅" which means "metal" and the character "滋" contains the radical "氵" which means "water".

Table 1. Detailed statistics of FGN on Weibo and OntoNote 4.

Model	Weibo			OntoNote 4		
	P	R	F1	P	R	F1
Lattice-LSTM	53.04	62.25	58.79	76.35	71.56	73.88
WC-LSTM	52.55	67.41	59.84	76.09	72.85	74.43
BERT-LMCRF	66.88	67.33	67.12	78.01	80.35	79.16
Glyce	67.68	67.71	67.70	80.87	80.40	80.62
GlyNN	N/A	N/A	69.20	N/A	N/A	N/A
FGN	69.02	73.65	71.25	82.61	81.48	82.04

Table 2. Detailed statistics of FGN on Resume and MSRA.

Model	Resume			MSRA		
	P	R	F1	P	P	F1
Lattice-LSTM	93.57	92.79	93.18	93.57	92.79	93.18
WC-LSTM	95.27	95.15	95.21	94.58	92.91	93.74
BERT-LMCRF	96.12	95.45	95.78	94.97	94.62	94.80
Glyce	96.62	96.48	96.54	95.57	95.51	95.07
GlyNN	N/A	N/A	95.66	N/A	N/A	95.21
FGN	96.49	97.08	96.79	95.45	95.81	95.64

References

[1]Yuxian, M., Wei, W., Fei, W., et al.: Glyce:Glyph-vectors for Chinese Character Represen-tations. In: Advances in Neural Information Processing Systems, pp. 2742-2753 (2019).

[2] Arijit, S., Chan, S.. Using Chinese Glyphs for Named Entity Recognition. arXiv preprint arXiv:1909.09922, Computer Science (2019).

[3]Sijie, M., Haifeng, H., Songlong, X.: Divide, Conquer and Combine: Hierarchical Feature Fusion Network with Local and Global Perspectives for Multimodal Affective Computing. In: Annual Meeting of the Association for Computational Linguistics, pp. 481-492 (2019).