[RE] Glyce: Glyph-vectors for Chinese Character Representations

Yiheng Lu, Jiyi Wang, Xiaohan Wang

***Abstract*—Based on the Shannon AI team's study of Glyph-vectors for Chinese character and a series of NLP tasks, we implement 2 baselines reported in the original paper, BiLSTM-CRF and BERT and reproduce their results of Chinese NLP Tagging tasks on various datasets. We are unable to reproduce the results for BiLSTM-CRF. However, we obtain a similar result for BERT model. On this basis, we undertake further experiments of hyper-parameter tuning and ablations on CRF+biLSTM and BERT, respectively. By evaluating their performances, we compare and contrast how the components and hyper-parameters can affect the model's accuracy and robustness. We discover that the implementation of BERT embedding as well as adding multiple layers or conditional random field (CRF) can boost the model accuracy to a decent extent.**

I. Introduction

In recent years, deep learning has made great progress in many fields including Natural Language Processing (NLP) and Computer Vision. NLP is one of the most important branches. NLP is a computer technology designed to automatically analyze or generate human languages. In recent ten years, a large number of deep learning models and methods have emerged in the NLP field.

Specifically, the representations based on neural network and dense vector (like word vector) have achieved competitive results in many NLP tasks. In 2019, based on the particularity of Chinese characters, the team of Shannon AI published NeurIPs paper of Glyce, a deep learning model in view of Chinese characters. To compare and contrast this model with other baselines in Chinese NLP tasks, we focused on tasks of tagging at the token level, reproducing and further adjusting the baseline reported in the original paper.

In our project, we choose BiLSTM-CRF and BERT in the original paper as our baselines. BiLSTM-CRF is a mainstream model of *Sequence Labeling* based on deep learning. As for features, this model inherits the advantages of deep learning methodology which does not need feature engineering and this model can make excellent predictions by using word vectors and character vectors.

As a new strong model, BERT transfers sentence input into word vectors from the pre-trained embedding and then passes to the specific downstream NLP tasks. BERT embedding provides a robust and easy way to train models and to do transfer learning. Because of the outstanding characteristics of these two models, we decide to implement them as our main methods. We take the source code and some open source libraries such as huggingface [1]

By using of Kashgari NLP transfer learning framework [2], we establish two baselines and train models to complete the Sequence Labeling Tasks (NER, CWS, POS) on multiple character-level datasets in Chinese language. After comparing the results with those of the original paper, to improve the overall performance, we perform rigorous ablation studies, hyper-parameter tuning, and model adjoining to make them reach the baseline performance and even higher. Since all of tagging datasets have been divided into three parts: train, validate and test, we did not add additional pipelines but we used train and validate sets to fit model, and then we use test sets to evaluate trained model and obtain the final result.

We follow Track 1 of baseline reproduction. The original work is tested on massive models and datasets of different categories. Due to limited computation resources and time, we mainly focus on training model on NER tasks and several datasets of different categories. We select two models among all the baseline models: BERT and BiLSTM-CRF. Our goal is to get the result that is close enough to that of the original work. We use their code and some open source libraries such as HuggingFace Transformer as reference. We will show our results in the following sections, as well as some ablation tests on hyper-parameters and model architectures in Section VII.

As our major finding, we discover that the implementation of BERT embedding is an efficient way to improve the accuracy of the Chinese character labeling. After implementing a few approaches and experiments, we find that the use of multiple layers and the use of conditional random field (CRF) are helpful for increasing accuracy in case of the same batch size and epochs. Also,

we learn a set of hyper-parameters that are optimal for certain dataset.

II. Related Work

Chinese language is a well-known logographic language. The logograms of Chinese characters can encode extremely rich information of their meanings. As a result, NLP tasks for Chinese can benefit from the use of the glyph information. Past literature proves that radical representations are useful in various language understanding tasks ([3]). With the scheme of Wubi, which is a Chinese character encoder that mimics the order of typing of the radical sequence for a character on a computer keyboard, we can boost model performances on Chinese-English machine translation tasks [4]. Besides, Cao et al. [5] propose to solve these tasks from a more in-depth perspective by utilizing stroke n-grams for character modeling.

The success of AlexNet [6] draws the world's attention to deep learning. Thus, we notice efforts that implement CNN-based algorithms on the visual features of Chinese characters. Nonetheless, consistent performance boosts are not captured [7], and some even yield negative results [Dai and Cai, 2017]. Particularly, Dai and Cai [8] use CNN architectures on character logos to obtain Chinese character representations and also in certain downstream tasks of language modeling. In their work, adding glyph representations impacts negatively the mo del performance and the authors conclude that CNN-based representations is incapable of enhancing language modeling. Quite similarly, the idea is applied onto text classification tasks, and performance boosts are noticeable only in very limited number of scenarios and datasets [7]. However, we do document some positive results from Su and Lee [9], [10] who find glyph embedding helpful for two tasks: word analogy and word similarity, but they only focus on word-level semantic tasks and do not see improvements in the word-level tasks to higher level NLP tasks including sentence or discourse level. Combining radical representations, Shao et al. [3] incorporate CNN architectures on character figures and use the output as additional features in the NLP task of POS tagging.

III. summarized paper

In order to fully utilize the graphical information of Chinese characters, people tried to encode the visual information via CNN, a common technique in computer vision. Glyce is the combination of both visual embedding and textual embedding. It differs from its previous works in that it is trained with the historical scripts, which is more akin to natural images. It also reduces the character graphical features to a 2x2 structure called tianZige. And tianZige is then mapped to output by CNN. This structure with BERT embedding together forms the Glyce-BERT Embedding. With this, Glyce is able to achieve high score among tasks.

Fig. 1: Label distribution of MSRA of NER.

Fig. 2: Label distribution of PKU of CWS.

IV. Dataset and Task Description

A. Dataset

In order to evaluate the performance of our baselines, we use biLSTM-CRF and BERT to undertake the Chinese sequence labeling task on several datasets that are listed as follows.

NER (Name Entity Recognition): Based on the level of Chinese characters, this task is to classify the meaningful name identities into pre-defined categories such as person names, organizations, locations and so on. For this task, we use the widely-used OntoNotes, MSRA, Weibo, and resume datasets. As an example, the label distribution of MSRA is shown in Fig. 1 which is relatively imbalanced.

POS (Part Speech Tagging): This task is to classify characters into their lexical categories which may display different syntactic behaviors. For this task, we utilize the widely-used PKU, MSR, CITYU data sets. We present the distribution of PKU in Fig. 2, an exemplary data of

this type, and we can tell this dataset is less imbalanced compared with the aforementioned MSRA dataset.

CWS (Chinese Word Segmentation): CWS task should classify each character into different label served as grammar functions.In our project we use the CTB5, CTB9 to test our models.

B. Data Pre-processing

Since all datasets house multiple examples of sentences in which each character is noted with its corresponding tag, we load all types of datasets (trainvalidate- and test) into a tuple which make up with two double lists (input x and output y). Each element in the double list is an example of sentence, and each sentence are separate to multi-characters saved as a list In order to study the distribution of various categories in data set, we propose MSRA in NER task and PKU data set in CWS task, then count the number of each class in their train set and draw the relative figure for them. The results show that the distribution of the various categories in the data set is imbalanced.

C. Sentence Pair classification

For sentence pair classification, we have two input texts, and one label to this pair. For this classification, we only work with XNLI [11] (Cross-lingual Natural Language Inference). Its pairs are annotated and translated into 14 languages, thus served as a benchmark of NLP multi language tasks.

V. Purposed Approach

A. BiLSTM-CRF

Long short-term memory(LSTM) is a type of Recurrent Neural Networks (RNNs). In theory, RNNs are capable of capturing long distance dependencies. However, in practice, RNNs fail due to gradient vanishing and gradient exploding problems. LSTM is first introduced in 1997[14]. LSTM is a variant of RNN and it is designed to solve the gradient vanishing and exploding problems. An LSTM unit contains three multiplicative gates which control the proportions of information to forget and to pass on to the next time step. However, the limitation of LSTM is that the hidden unit h_t only took information from the past. Bidirectional LSTM can capture the past (the last word) and the future (the next word) information effectively. In sequence labelling tasks, consideration of correlations between labels in neighbourhoods and jointly decode the best chain of labels for an input dataset would be beneficial. The label sequence is modelled jointly by using a conditional random field (CRF)[15].

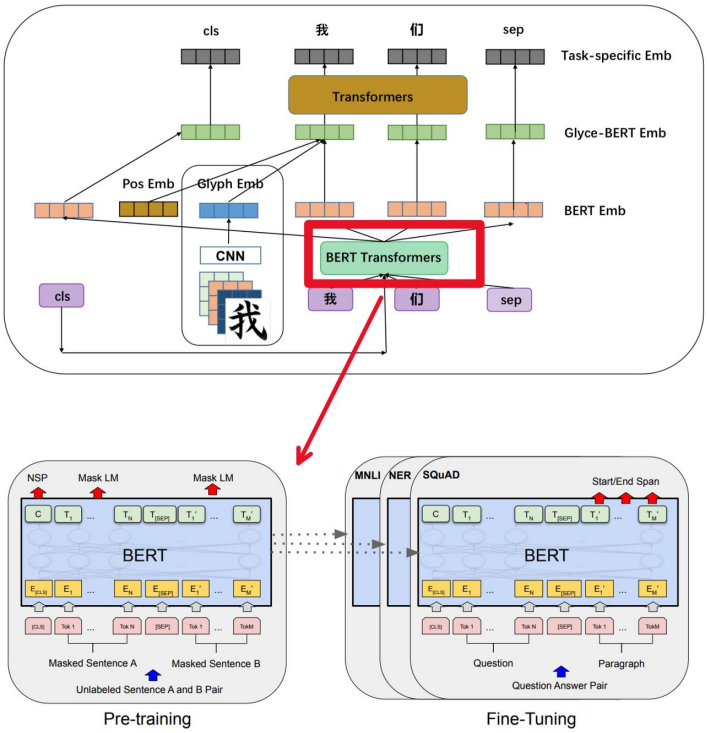

Fig. 3: Model structure of the original work and the architecture we implemented in this project. The 2 figures are from [12] and [13] respectively

The CRF layer is designed to select the best tag sequence from all possible tag sequences with consideration of outputs from BiLSTM and correlations between adjacent tags [16]. Thus the BiLSTM-CRF model is implemented to raise the accuracy.

We use Bidirectional LSTM with CRF (BiLSTM-CRF) as a part of the reproduction. We observe the original paper uses BiLSTM-CRF in the source code [1]. The original paper uses this approach in Named Entity Recognition (NER) task [12]. In this task, BiLSTM-CRF is used for sequence tagging Chinese Characters. We reproduce this task by implementing the BiLSTM-CRF model from Kashgari in GitHub [2]. We use several different datasets as input: Ontonotes, MSRA, Resume and Weibo for the NER task. This API mainly used Tensorflow to built the BiLSTM-CRF model. We modify hyper-parameters, including activation functions, learning rates and optimizers to make the accuracy higher. As indicated in Figure 4, the structure of BiLSTM-CRF is more efficiency with the advantages: without feature abstraction, biLSTM-CRF is able to outperform with only character vectors or word vectors.

[1]https://github.com/ShannonAI/glyce/blob/master/glyce/models/latticeLSTM/model/bilstmcrf.py

[2]https://github.com/BrikerMan/Kashgari

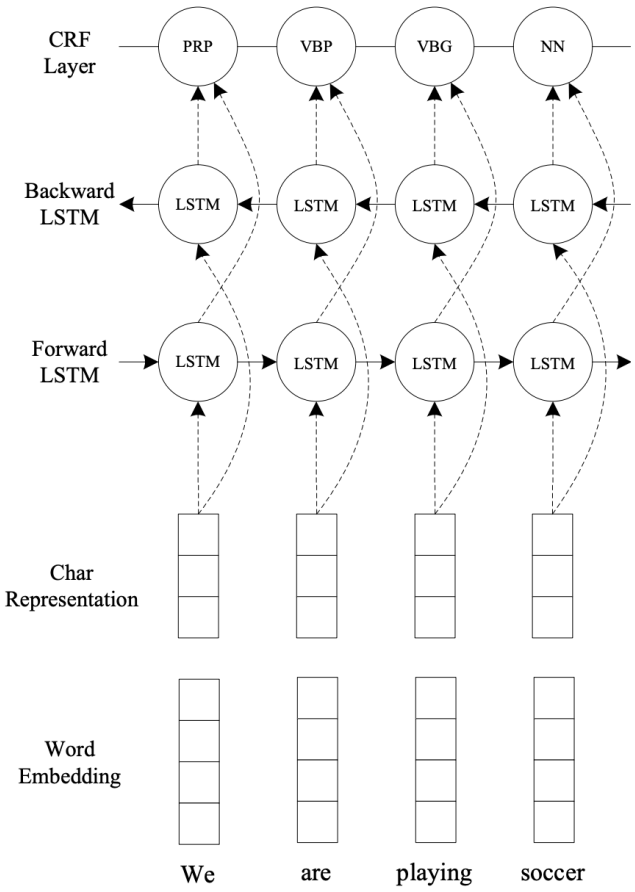

Fig. 4: The structure of BiLSTM-CRF

B. BERT

Bidirectional Encoder Representations from Transformers (refer as BERT below) is recently developed by [13] and reaches a state-of-art performance on many natural language processing tasks. It is built on the architecture of transformers [17]. [17] abandoned the traditional LSTM sequential structure. They use only stacked encoders and decoders with self attention in their layers. Bert uses the pre-trained encoder to represent the word vectors, but instead of using dependencies only on one side, it uses a bidirectional attention. This embedding is then passed to some downstream tasks. In addition, the parameters of the embedding layers would get adjusted according to the specific task. The invariance of the embedding provides a very convenient way to do transfer learning with different NLP models. In the original paper [12], they use both CNN(Glyce Emb) and bert embedding together, which are then followed by a 2 layer transformer model.

Following the set up of the original paper, we use the pretrained bert chinese base model and the default configuration in [13]. The code is adapted from [3][1] and [4]. The max sequence length is set to 150, batch size is set to 128. For sake of efficiency, we use 16 bits float precision. Here we use the default BERT model for the baseline, but in later sections, we also test on BERT embedding combined with task specific models.

C. Additional Experiments with BERT-embedding in POS task

We implement BERT embedding with CNN LSTM Model, BiLSTM Model, BiGRU Model and BiGRU-CRF Model to improve the accuracy to compared with the previous. We use ctb9 dataset from POS task for the models.

D. GRU

Gated recurrent unit is a kind of recurrent neural network. It captures the dependencies of different time scale. [18] GRU is very similar to LSTM except that it does not contain memory cells.

VI. Results

A. Results of BiLSTM-CRF

NER-MSRA:

	precision	recall	f1-score
Original	92.97	90.80	91.87
Reproduced result	85.14	79.45	82.19

NER-Ontonotes:

	precision	recall	f1-score
Original	74.36	69.43	71.81
Reproduced result	54.65	45.56	49.68

NER-weibo:

	precision	recall	f1-score
Original	51.16	51.07	50.95
Reproduced result	27	19	22.3

NER-Resume:

	precision	recall	f1-score
Original	94.53	94.29	94.41
Reproduced result	90.16	88.10	89.11

We are unable to reproduce the full results of BiLSTM-CRF model. Especially Ontonotes dataset and weibo dataset. The first reason is that the dataset itself

[3]https://github.com/huggingface/transformers

[4]https://github.com/lemonhu/NER-BERT-pytorch

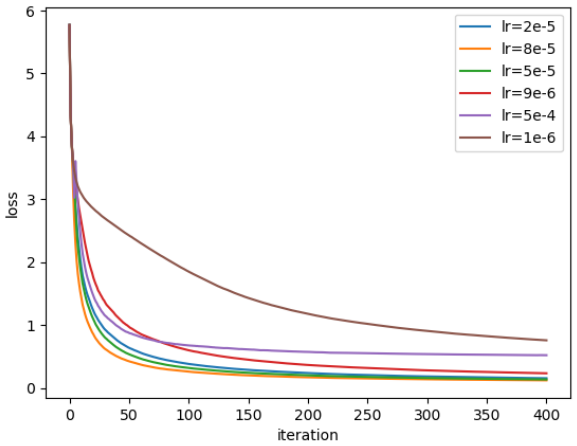

Fig. 5: Loss vs. Iteration

contains some poorly structured data, which results in the relative low precision, recall and f1-score among all models. The second reason is that we might miss some pre-processing steps before the training and did not explore all the input space during fine-tuning process.

B. Results of BERT

We tried to reproduce the result of BERT on ontonotes dataset. The result is slightly lower than the original result.

NER-Ontonotes:

	precision	recall	f1-score
Original	78.01	80.35	79.16
Reproduced result	77.82	80.11	78.94

NER-MSRA:

	precision	recall	f1-score
Original	94.97	95.04	94.80
Reproduced result	94.60	95.01	94.80

Sentence Pair Classification-XNLI:

	accuracy
Original	78.4
Reproduced result	71.3

C. Fine-tuning on learning rate

We change the learning rate and the maximum sequence length of the model in order to obtain an optimal configuration. We obtain our best result at learning rate=2e-5 and max sequence length=150.

CWS-PKU			
BERT+BiGRU	96.04	95.48	96.76
Glyce+BERT+Transformer	97.1	96.4	96.7
CWS-Cityu			
BERT+BiGRU	96.64	97.07	96.85
Glyce+BERT+Transformer	97.9	98.0	97.9
CWS-MSR			
BERT+BiGRU	96	96.24	96.12
Glyce+BERT+Transformer	98.2	98.3	98.3
NER-MSRA			
BERT+BiGRU	91.07	91.97	91.5
Glyce+BERT+Transformer	97.1	96.4	96.7
NER-Weibo			
BERT+BiGRU	57.57	60.92	58.68
Glyce+BERT+Transformer	67.68.1	67.71	67.60
POS-CTB5			
BERT+BiGRU	94.62	94.95	95.26
Glyce+BERT+Transformer	96.5	96.74	96.61
POS-CTB9			
BERT+BiGRU	92.45	93.67	93.05
Glyce+BERT+Transformer	93.49	92.84	92.38

TABLE I: The precision, recall and f1-score obtained on different datasets using bert+BiGRU respectively

Fig. 5 plots the loss function under different learning rate during training.

D. Ablation Studies

As shown in Table I, we try different model concatenations such as BERT+CNN+BiLSTM, BERT+CNN+BiMPM, BERT+CNN+BiCNN, etc.([12]) We work with an additional model BiGRU. However, we do not use Glyce embedding and BERT embedding together. We train only with BERT embedding to predict the data result. This model is trained with drop-out rate=0.4, max-sequence-size=128, batch-size=128 and trained for 10 epochs. We show the some evaluation results of several datasets. From I, we find that BERT+BiGRU overall performed well. It mainly take advantage of Bert embedding layer. This architecture even outperforms Glyce on CTB9 dataset. To be more precise, BiGRU can achieve a competitive result on POS datasets, which are sequence labeling tasks in favor of RNN.

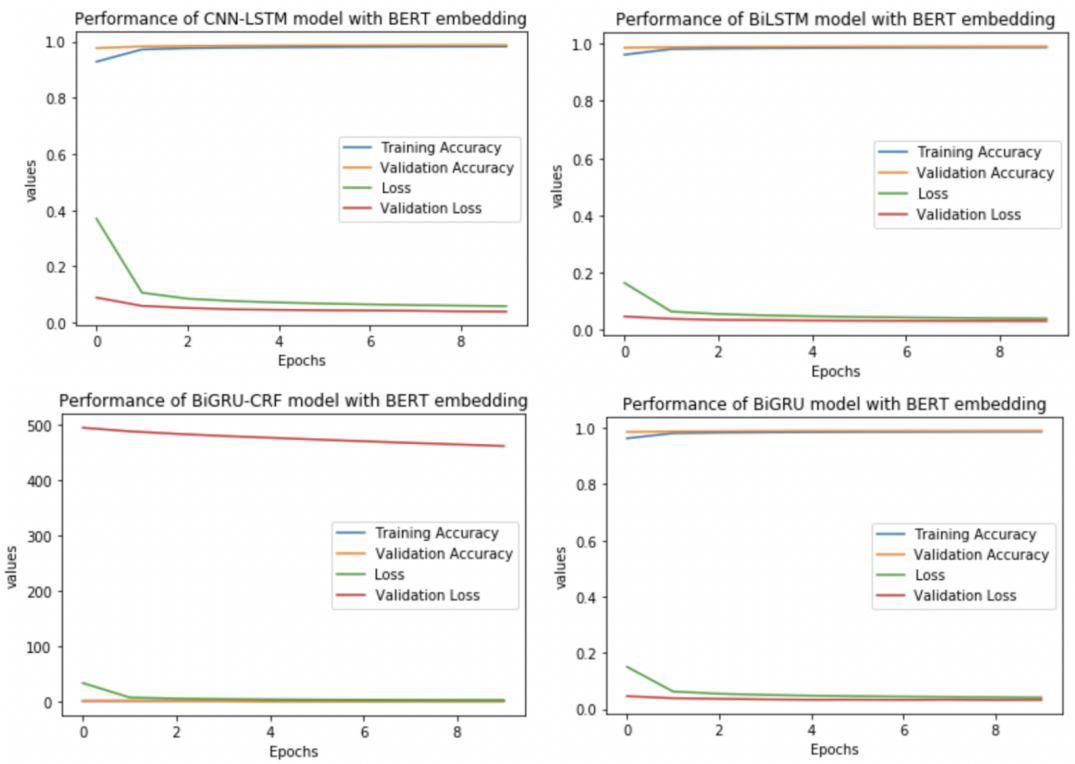

Fig. 6: The comparison of training accuracy, validation accuracy, loss and validation loss in the four experiments

E. Results of Additional Experiments with BERT embedding in POS task

	precision	recall	f1-score
CNN-LSTM	91.07	92.68	91.87
BiLSTM	92.74	93.91	93.92
Bi-GRU	92.61	93.77	93.18
BiGRU-CRF	94.18	94.04	94.11
Original BERT	92.43	92.15	92.29

We use BERT embedding with the above models to observe if an improvement of accuracy is achieved. With batch-size=256, epochs=10, we perform four different kinds of approaches for ctb9 dataset in POS task. Compared with the previous results in baseline achievement, the BiGRU-CRF model with BERT embedding outperforms all the experiments and baseline indicated in the paper with BERT model. The CNN-LSTM model with BERT embedding performs the least accurate among the four experiments and the original baseline implemented in the paper. As Fig. 6 indicates, the loss and overall performance of BiGRU-CRF model with BERT embedding is the best, although the validation in the last 6 epochs is significantly lower than the previous epochs and the loss is much higher than other 3 experiments.

VII. Discussion and Conclusion

Since our reproduction does not reach most of the baselines. Our future work will focus on the downstream task models (transformers). We do not run our models on every dataset, so sentence pairs and other types of datasets should be handled. Reasons behind the fact that certain models fail on some datasets should be further analysed. In conclusion, we implement several baselines from the original paper and reproduce their results of Chinese NLP Tagging tasks on various datasets. We undertake further experiments of hyper-parameter tuning and ablations on two core models By evaluating their performances, we compare and contrast how the components and hyper-parameters can affect the model's accuracy and robustness. In ablation studies, we further explore different aspects of the models and manipulate parameter values. By reconstructing the architectures, our models outperform the original one on one dataset. We discover that the implementation of BERT embedding as well as adding multiple layers or conditional random field (CRF) can boost the model accuracy to a decent extent.

VIII. Statement of Contribution

1) Yiheng Lu (260789862)
yiheng.lu@mail.mcgill.ca,
Implemented the baseline LSTM and ablation model architecture, additional four experiments, tuned hyperparameters, and contributed the the corresponding part of the report
2) Jiyi Wang (26077138),
jiyi.wang@mail.mcgill.ca
Worked on dataset processing, research on available APIs, contributed to fine tuning and corresponding part of the report
3) Xiaohan Wang (260739056)
xiaohan.wang2@mail.mcgill.ca
Implemented the baseline bert and ablation model architecture, adpated code from the source code of the original paper, and contributed to the corresponding part of the report.

References

[1] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and J. Brew, "Huggingface's transformers: State-of-the-art natural language processing," *ArXiv*, vol. abs/1910.03771, 2019.

[2] M. Maimaiti, A. Wumaier, K. Abiderexiti, and T. Yibulayin, "Bidirectional long short-term memory network with a conditional random field layer for uyghur part-of-speech tagging," *Information*, vol. 8, no. 4, p. 157, 2017.

[3] Y. Shao, C. Hardmeier, J. Tiedemann, and J. Nivre, "Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf," *arXiv preprint arXiv:1704.01314*, 2017.

[4] N. I. Nikolov, Y. Hu, M. X. Tan, and R. H. Hahnloser, "Character-level chinese-english translation through ascii encoding," *arXiv preprint arXiv:1805.03330*, 2018.

[5] S. Cao, W. Lu, J. Zhou, and X. Li, "Cw2vec: Learning chinese word embeddings with stroke n-gram information," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[7] F. Liu, H. Lu, C. Lo, and G. Neubig, "Learning character-level compositionality with visual features," *arXiv preprint arXiv:1704.04859*, 2017.

[8] F. Z. Dai and Z. Cai, "Glyph-aware embedding of chinese characters," *arXiv preprint arXiv:1709.00028*, 2017.

[9] T.-R. Su and H.-Y. Lee, "Learning chinese word representations from glyphs of characters," *arXiv preprint arXiv:1708.04755*, 2017.

[10] Y. Sun, L. Lin, N. Yang, Z. Ji, and X. Wang, "Radical-enhanced chinese character embedding," in *International Conference on Neural Information Processing*, Springer, 2014, pp. 279–286.

[11] A. Conneau, R. Rinott, G. Lample, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov, "Xnli: Evaluating cross-lingual sentence representations," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium: Association for Computational Linguistics, 2018.

[12] Y. Meng, W. Wu, F. Wang, X. Li, P. Nie, F. Yin, M. Li, Q. Han, X. Sun, and J. Li, "Glyce: Glyph-vectors for chinese character representations," *arXiv preprint arXiv:1901.10125*, 2019.

[13] 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.

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

[15] X. Ma and E. Hovy, "End-to-end sequence labeling via bi-directional lstm-cnns-crf," *arXiv preprint arXiv:1603.01354*, 2016.

[16] S. Misawa, M. Taniguchi, Y. Miura, and T. Ohkuma, "Character-based bidirectional lstm-crf with words and characters for japanese named entity recognition," in *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, 2017, pp. 97–102.

[17] 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.

[18] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.