Proceedings of the Second Workshop on Language Technologies for Historical and Ancient Languages (LT4HALA 2022), pages 178–182
Language Resources and Evaluation Conference (LREC 2022), Marseille, 25 June 2022

Glyph Features Matter: a Multimodal Solution for EvaHan in LT4HALA2022

Xinyuan Wei*, Weihao Liu*, Qing Zong*, Shaoqing Zhang*, Baotian Hu†
Harbin Institute of Technology (Shenzhen)
{21S151175, 200110921, 200110513, 1190200721}@stu.hit.edu.cn
hubaotian@hit.edu.cn

Abstract

We participate in the LT4HALA2022 shared task EvaHan. There are two subtasks in this task. Subtask 1 is word segmentation, and subtask 2 is part-of-speech tagging. Each subtask consists of two tracks, a close track that can only use the data and models provided by the organizer, and an open track without restrictions. We employ three pre-trained models, two of which are open-source pre-trained models for ancient Chinese (Siku-Roberta and roberta-classical-chinese), and one is our pre-trained GlyphBERT combined with glyph features. Our methods include data augmentation, data pre-processing, model pretraining, downstream fine-tuning, k-fold cross validation and model ensemble. We achieve competitive P, R, and F1 scores on both our own validation set and the final public test set. For the word segmentation task and the part-of-speech tagging task, respectively, on F1 on the close track, we achieved 91.89 and 85.74 on test A, and 80.75 and 69.62 on test B; similarly, on the open track, we achieved 92.33 and 86.47 for test A, and 81.24 and 70.05 for test B.

Keywords: ancient Chinese, glyph features, pre-trained language model

1. Introduction

Our team HITszTMG participates in the LT4HALA shared task EvaHan 2022. This task contains two subtasks: Chinese word segmentation and part-of-speech tagging. Chinese word segmentation and part-of-speech tagging tasks are two basic tasks in natural language processing. Chinese word segmentation aims to divide the continuous word sequence into word units. The input is a continuous word sequence (a sentence), and the output is a segmented word unit sequence. The part-of-speech tagging task is to tag each word with a separate label that represents usage and its syntactic effect, such as noun, verb, adjective, etc. The input is a sequence of consecutive words (a sentence), and the output is the sequence of parts of speech corresponding to each word.

Each subtask consists of two tracks, a close track that can only use the data and models provided by the organizer, and an open track without restrictions. For close tracks, we employ Siku-Roberta model [王东波 et al.] , utilize some data post-processing methods, and try some downstream fine-tuning tricks to improve performance. For the open track, we obtain some ancient text data and use the jiayan[1] toolkit for data augmentation; we also use multiple pretraining models: GlyphBERT (pre-trained by us) [Li et al.2021], Siku-Roberta and roberta-classical-chinese, [2] for downstream fine-tuning, and use some fine-tuning tricks; finally, we employ model ensemble. We achieve competitive scores on P, R, and F1 in our test set.

2. Related Work

2.1. Chinese Word Segmentation (CWS)

Chinese Word Segentation is a fundamental task in Chinese language processing. There is extensive research ([Sproat and Shih1990], [Xue and Shen2003], [Huang et al.2007], [Liu et al.2014]). In recent years, deep neural networks have also been widely used to solve the CWS problem with great success. ([Zhou et al.2017], [Yang et al.2017], [Ma et al.2018], [Yang et al.2019]). They can better perform word segmentation through contextual information and knowledge learned in the pre-training process.

2.2. Part-of-speech Tagging

Part-of-speech (POS) tagging is a fundamental task in NLP as well. It's one of the first stages in natural language processing, as an initial stage of information extraction, summarization, retrieval, machine tranlation and speech conversion. [Patil et al.2014]One of classical approaches is generally done with a maximum entroph Markov model(MEMM) [Ratnaparkhi1996]. Recently, deep models are employed to achieve a better performance for this task ([Józefowicz et al.2016], [Choi2016]).

2.3. Pre-trained Language Model (PLM)

The classic word embedding technology, such as Word2Vec [Mikolov et al.2013] and GloVe [Pennington et al.2014] is static. These methods learn the word embeddings with fixed dimensions and meaning rather than contextual information through training on large-scale corpora. To address this problem, researchers

---

∗ equal contribution

† corresponding author

[1]Jiayan: ancient Chinese toolkit https://github.com/jiaeyan/Jiayan

[2]roberta-classical-chinese https://huggingface.co/KoichiYasuoka/roberta-classical-chinese-large-char

study how to learn word embeddings that can contain more comprehensive contextual information. ELMo [Peters et al.2018] is proposed to capture contextual features. BERT [Devlin et al.2018] employs masking language model (MLM) and Next Sentence Prediction (NSP) as pre-train tasks, and then the neural network can learn the context information very well. Based on BERT's architecture and idea, some studies have proposed different pre-training methods to enhance the effect of BERT. Roberta [Liu et al.2019] improves the performance of BERT by employing the MLM by dynamically masking computation while abandoning the NSP task. Roberta optimizes its pre-training process to make the language representation learned by the model more robust, showing better performance than BERT in many tasks.

In addition, researchers are concerned that pre-trained models do not generalize to all problems in all domains, so they start training models that fit for unique domains. In the field of ancient Chinese, roberta-classical-chinese and Siku-Roberta both show excellent performance in the field of ancient Chinese by adopting different training corpora. We also pre-train GlyphBERT, a pre-train BERT model that can capture glyph information to train a better ability of representation.

2.4. Glyph Vector

Compared with English words, Chinese characters consist of more complex symbolic results. Chinese characters often have unique structures and radicals, and these radicals are often related to the meaning of the word, so obtaining glyph information can help models better understand contextual semantics.There have also been many researches ([Su and Lee2017], [Meng et al.2019], [Chen et al.2020]) that demonstrate the effectiveness of incorporating glyph information into pre-trained models. The typical method is to use a deep convolutional neural network to extract glyph features of Chinese characters from images. Then combining glyph information and word embeddings can enhance the representation of Chinese characters. We use HanGlyph module as a feature extraction module, and pre-train our own glyph pre-training model GlyphBERT, which also gets competitive results in this competition.

3. Our Methods

Our methods include data augmentation, data preprocessing, model pre-training, downstream fine-tuning, K-fold cross validation and model ensemble. We achieve competitive P, R, and F1 scores on both our own validation set and the final public test set.

3.1. Data Augmentation

This part focuses on the open track. Some research has shown that the larger corpus and the more distribution, the better the generalization performance and robustness of the trained model. Since this, we decide to expand a part of the pseudo-corpus as data augmentation first.

We have expanded Modern Chinese and Ancient Chinese respectively. For modern Chinese, we use the named entity datasets MSRA and People, which are two NER datasets commonly used in the field of Chinese natural language processing. And then we preprocess their test set according to our BIOE labeling way, to be consistent with our training set. The size of this corpus is about 20k. For ancient Chinese, we find a collected open source project that includes the twenty-four histories. After randomly shuffling these ancient Chinese texts, we randomly select a part of them using another open-source project Jiayan for part-of-speech tagging. The size of this corpus is about 20k.

In addition, after the test set is open, we observe the results of the model and find that the models have insufficient labeling ability for some special symbols (such as ", ", [,], etc.). We analyze that it is due to the lack of corpus of special symbols in the training set. So we collect the part of the training set that contains special symbols and perform a fine-tuning as the augmented data.

3.2. Preprocessing

In this task, we combine Chinese word segmentation and part-of-speech tagging into a sequence tagging task. After tagging the part-of-speech of each word with the BIOE tagging method, we then segmented the words according to the tags.

First, we mark all parts of speech involved in this task through the BIOE tagging method, with a total of 88 kinds.

At the same time, since there was no public test set in the early stage of the competition, we divide 1-7000 into the training set, 7001-7700 as the validation set, and the rest into the test set.

3.3. Pre-training Models and GlyphBERT

Since most of the pre-trained models are trained on modern texts, it is also important to select suitable pre-trained models. On the close track, we use the Siku-Roberta provided by the organizer. On the open track, in addition to Siku-Roberta, we also select roberta-classical-chinese and our own pre-trained GlyphBERT. Although GlyphBERT is trained through modern Chinese corpus, experiments show that GlyphBERT also has an excellent performance in this task. This may benefit from the good learning and application of glyph features by GlyphBERT, which make this model has a great ability of transfer.

3.4. Downstream Fine-tuning

Downstream fine-tuning has always been an important step that affects model performance. In this task, we add a CRF layer to the output results before the fully connected layer in the downstream, and set a different learning rate for the CRF layer. The experimental results show that the CRF layer has an excellent effect on

Models	Segmentation			Pos tagging		
	P	R	F1	P	R	F1
Siku-Roberta	88.2762	88.2762	89.3116	80.0600	81.9605	80.9991
+CRF	88.4167	92.0458	90.1948	80.3061	83.6022	81.9210
+Data augmentation	90.4368	91.1646	90.7993	82.6507	83.3158	82.9819
+Change Lr	91.7447	92.3494	92.0460	84.4133	84.9697	84.6906
+K-fold	**92.7101**	**94.8314**	**93.7588**	**87.4430**	**89.4438**	**88.4321**

Table 1: The experimental results of Siku-Roberta on our dividing test set. The methods we take have effectively improved the model performance.

Models	Segmentation			Pos tagging		
	P	R	F1	P	R	F1
roberta-classical-chinese	95.6615	95.6692	95.6654	90.4941	90.5014	90.4978
+CRF	**95.7000**	**95.7541**	**95.7270**	**90.4664**	**90.5176**	**90.4920**
+Data augmentation	95.5804	95.4953	95.5378	90.2623	90.1820	90.2221
+Change Lr	95.6294	95.2002	95.4143	90.5764	90.1698	90.3727

Table 2: The experimental results of roberta-classical-chinese on our delineated test set. Despite our use of these methods, the results are not much different from the original. So, we choose the roberta-classical-chinese model with CRF when doing model ensemble.

Models	Segmentation			Pos tagging		
	P	R	F1	P	R	F1
GlyphBERT	**93.9186**	**93.5467**	**93.7323**	**87.3382**	**86.9924**	**87.1650**
+CRF	92.6289	92.3450	93.3370	86.1587	86.5049	85.7965
+Data augmentation	92.6731	92.2838	92.4780	85.3341	84.9756	85.1544
+Change Lr	92.4743	92.6058	92.5400	85.5207	85.6423	85.5815

Table 3: The experimental results of GlyphBERT on our dividing test set. The methods we take are not very effective on GlyphBERT, so we choose to use GlyphBERT baseline when doing model ensemble.

the sequence labeling task, and setting learning rates for the CRF layer different from the base model is also very effective.

3.5. K-fold Cross Validation

We divide the original data into K groups (K-Fold), use each subset data as a validation set, and use the remaining K-1 sets of subset data as a training set, so that we obtain K models accordingly. The K models evaluate the results in the validation set respectively, then make predictions in the test set, and finally combine the prediction results of the K models to obtain the prediction labels of the test set. Cross-validation effectively utilizes limited data, and the evaluation results can be as close as possible to the performance of the model on the test set, which can be used as an indicator for model optimization.

3.6. Model Ensemble

Ensemble of multiple models is a common method used in competitions. The ensemble of models often requires certain differences between several models, such as using different corpora for training, or using different architectures. In this task we use 4 different models for ensemble: Siku-Roberta, roberta-classical-chinese-base-char, roberta-classical-chinese-large-char, GlyphBERT. Among them, Siku-Roberta and roberta-classical-chinese have similar architectures, but their training corpora are quite different. GlyphBERT is unique in its architecture, training corpus, and feature extraction method. So we think they will have a great effect in ensemble.

4. Experiments and Analysis

4.1. Experimental Settings

Our implementations of Siku-Roberta, roberta-classical-chinese-char, GlyphBERT are based on the public pytorch implementation from Transformers. Siku-Roberta is in large size, while roberta-classical-chinese-char models of both large and base versions are used. GlyphBERT is implemented base on Pytorch and Transformers library. During pre-training, we follow the hyper-parameters setting of the original implementation. During fine-tuning, We set the maximum length of the sentence to 512. We use a single Tesla v100s GPU with 32gb memory, and fine-tuning time varies from 6 to 12 hours for each model.

	Segmentation			Pos tagging		
	P	**R**	**F1**	**P**	**R**	**F1**
test A close1	90.8050	92.9935	91.8862	84.7235	86.7655	85.7323
test A close2	**90.7833**	**93.0326**	**91.8942**	**84.7024**	**86.8010**	**85.7389**
test A open1	91.0912	93.4130	92.2375	85.2745	87.4480	86.3476
test A open2	**91.1994**	**93.4947**	**92.3328**	**85.4086**	**87.5582**	**86.4701**
test B close1	82.1870	77.8193	79.9435	70.2067	66.4456	68.2744
test B close2	**82.7873**	**78.8168**	**80.7533**	**71.3723**	**67.9465**	**69.6173**
test B open1	**83.2716**	**79.2979**	**81.2361**	**71.8098**	**68.3830**	**70.0545**
test B open2	82.2262	78.3115	80.2211	70.7657	67.3967	69.0401

Table 4: The results of our eight submitted texts using the official final release evaluation script. On test B, the performance degradation of our model is more obvious. We think this is mainly due to the large differences in language habits in test B due to dynasties or other factors.

4.2. Experimental Results and Analysis

In the early stage of the competition, We intercept the last 1100 records of the dataset as the test set. Table 1 shows the experimental results of the baseline on this test set after using different tricks. The baseline is a Siku-Roberta model used on the close track. We set the learning rate to 1e-4, the batch size to 2, and the epoch to 5, and then obtained 89.3116 and 80.9991 points on the F1 score of word segmentation and part-of-speech tagging, respectively. After adding a CRF layer to get the prediction results, the F1 value of both tasks improved by 1 point. Then we add additional corpus besides CRF, and the scores of the two tasks also increased steadily. Finally, we set the learning rate of the CRF layer to 10 times that of the base model, and get 93.7588 and 88.4321 points in the two tasks, respectively. Table 2 and Table 3 show the cases of roberta-classical-chinese-large-char model and GlyphBERT model, respectively. If only using the roberta-classical-chinese-large-char model, we will get scores of 95.6654 and 90.4978 on the F1 score of the two tasks, which already exceeds the performance of the Siku-Roberta model. Although the GlyphBERT model basically exceeds the Siku-Robera in all indicators, it is not as good as the roberta-classical-chinese-large-char model. Before the release of the official test data, we finally use several models to predict the original 1100 pieces of test set with various tricks. These models include roberta-classical-chinese-char (both base and large), Siku-Roberta and GlyphBERT. After the ensemble at the logits, we achieve F1 scores of 95.9438 and 90.9540 on the two tasks respectively.

In the latter stage of the competition, each team has two submission opportunities for each of the two test sets in each track. Table 4 shows the final results of our model on the competition test set. For the close track of test A, We seperately submit the Siku-Roberta model with 10-fold cross-validation, and the combined results of 5-fold and 10-fold cross-validation at a logits ratio of 1:2. For the open track of test A, based on the close track, we add the results of roberta-classical-chinese-char and Siku-Roberta training on the expanded data set, as well as results of roberta-classical-chinese-char (including both base and large versions) using 5-fold cross validation.

The model usage on test B is the same as that on test A. However, results of test B are much worse than results of test A. We argue that the results of test B may come from other dynasties, and the usage of some words is slightly different from that of Siku Quanshu, resulting in a decline in the model prediction effect.

5. Conclusion

We introduce our submission for LT4HALA shared task EvaHan2022. For the close track, we propose some simple but efficient data augmentation methods and fine-tune methods. For the open track, we propose methods including data augmentation, data pre-processing, model pretraining, downstream fine-tuning, K-fold cross validation and model ensemble. We find that our model GlyphBERT performs well on transfer learning in this task. For the word segmentation task and the part-of-speech tagging task, respectively, on F1 on the close track, we achieved 91.89 and 85.74 on test A, and 80.75 and 69.62 on test B; similarly, on the open track, we achieved 92.33 and 86.47 for test A, and 81.24 and 70.05 for test B.

6. Bibliographical References

Chen, H.-Y., Yu, S.-H., and Lin, S.-d. (2020). Glyph2Vec: Learning Chinese out-of-vocabulary word embedding from glyphs. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2865–2871, Online, July. Association for Computational Linguistics.

Choi, J. D. (2016). Dynamic feature induction: The last gist to the state-of-the-art. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 271–281, San Diego, California, June. Association for Computational Linguistics.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Huang, C.-R., Šimon, P., Hsieh, S.-K., and Prévot, L. (2007). Rethinking Chinese word segmentation: Tokenization, character classification, or word-break identification. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 69–72, Prague, Czech Republic, June. Association for Computational Linguistics.

Józefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., and Wu, Y. (2016). Exploring the limits of language modeling. *CoRR*, abs/1602.02410.

Li, Y., Zhao, Y., Hu, B., Chen, Q., Xiang, Y., Wang, X., Ding, Y., and Ma, L. (2021). Glyphcrm: Bidirectional encoder representation for chinese character with its glyph. *CoRR*, abs/2107.00395.

Liu, Y., Zhang, Y., Che, W., Liu, T., and Wu, F. (2014). Domain adaptation for CRF-based Chinese word segmentation using free annotations. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 864–874, Doha, Qatar, October. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ma, J., Ganchev, K., and Weiss, D. (2018). State-of-the-art Chinese word segmentation with Bi-LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium, October-November. Association for Computational Linguistics.

Meng, Y., Wu, W., Wang, F., Li, X., Nie, P., Yin, F., Li, M., Han, Q., Sun, X., and Li, J. (2019). Glyce: Glyph-vectors for chinese character representations. *CoRR*, abs/1901.10125.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26, 10.

Patil, H. B., Patil, A. S., and Pawar, B. V. (2014). Article: Part-of-speech tagger for marathi language using limited training corpora. *IJCA Proceedings on National Conference on Recent Advances in Information Technology*, NCRAIT(4):33–37, February. Full text available.

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *CoRR*, abs/1802.05365.

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Conference on Empirical Methods in Natural Language Processing*.

Sproat, R. and Shih, C. (1990). A statistical method for finding word boundaries in chinese text. *Computer Processing of Chinese Oriental Languages*, 4(4):336–351, March.

Su, T.-R. and Lee, H.-Y. (2017). Learning Chinese word representations from glyphs of characters. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 264–273, Copenhagen, Denmark, September. Association for Computational Linguistics.

Xue, N. and Shen, L. (2003). Chinese word segmentation as lmr tagging. 07.

Yang, J., Zhang, Y., and Dong, F. (2017). Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada, July. Association for Computational Linguistics.

Yang, J., Zhang, Y., and Liang, S. (2019). Subword encoding in lattice LSTM for Chinese word segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2720–2725, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Zhou, H., Yu, Z., Zhang, Y., Huang, S., Dai, X., and Chen, J. (2017). Word-context character embeddings for Chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 760–766, Copenhagen, Denmark, September. Association for Computational Linguistics.

王东波, 刘畅, 朱子赫, 刘江峰, 胡昊天, 沈思, and 李斌.). Sikubert与sikuroberta:面向数字人文的《四库全书》预训练模型构建及应用研究.