Baseline Study of Glyce: Glyph-vectors for Chinese Character Representations

Long-Quan Bach
McGill University
Montreal, QC
long-quan.bach@mail.mcgill.ca

Yimeng Hu
McGill University
Montreal, QC
yimeng.hu@mail.mcgill.ca

Sichen Wan
McGill University
Montreal, QC
sichen.wan@mail.mcgill.ca

Abstract

In this report, we reproduce and inspect the performance of the baselines reported in the paper titled "Glyph-Vectors for Chinese Character Representations" by Wu et al., pursuing the baseline track for the reproducibility task. We evaluate a subset of the tasks mentioned in their paper, including sequence labelling, single sentence classification, and sentence pair classification. We are able to implement two of the models mentioned in the original paper and a backtracked model we selected. For sequence labelling task, we beat the baseline mentioned in the paper.

1 Introduction

In the paper *Glyce: Glyph-vectors for Chinese Character Representations* by Wu et al., the authors were able to find an effective method to use Chinese glyph information to handle NLP tasks. They proposed this idea of Glyce, glyph vectors processed from a number of smaller subroutines. These subroutines were explained in great depth in their paper. Essentially, through a number of image classification tasks, they were able to accumulate enough useful information to create their glyph vectors. They embedded these features into deep learning models such as LSTMs, RNNS, and transformers. They were able to show that using Glyce's glyph embeddings achieved impressive results, consistently surpassing many state of the art models in a vast variety of NLP tasks. They proved clearly that using glyph embeddings from Glyce significantly outperformed word and character embeddings which were considered as the industry standard.

This paper pursues the baseline track where we evaluate a subset of the baselines presented by Wu et al.. Because of the ambition of the project, the paper covered an overwhelming number of Chinese NLP tasks in order to firmly prove that the benefits of Glyce embeddings. Many of the baseline metric reports they recorded were taken from other works that were cited. In order to verify the performance of Glyce, we will replicate these baselines to try to beat the Glyce proposed models. In this paper, we will reevaluate the baselines covered in three of their tasks: sequence labelling, single sentence classification, and sentence pair classification.

2 Related Work

Natural Language Processing (NLP) has long been a main research area in machine learning. For Chinese NLP, some main tasks include sequence labeling[17], single sentence classification, sentence

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

pair classification[14], dependency parsing[3], semantic role labeling[9] and so on. Most of the current NLP tasks can be solved by Long Short-Term Memory(LSTM) or BERT. LSTM is special type of recurrent neural network (RNN) that deal with long-range dependencies better than conventional RNNs[10]. BERT is able to pretrain language representations for more effective natural language processing tasks[4].

3 Tasks

Our main goal is to implement and improve the baselines proposed in the paper. Given the time and computational resource constraint, we selected a subset of tasks mentioned in the paper, which includes sequence labelling, single sentence classification and sentence pair classification, and for each task, we implemented one baseline proposed on the selected dataset. The detailed information about the three tasks is described in the following sections.

3.1 Sequence Labelling

Many Chinese NLP tasks can be formalized as character-level sequence labeling tasks, the author proposed three approaches for this task, name entity recognition(NER), Chinese word segmentation (CWS) and part speech tagging (POS). We chose only to evaluate the NER task baseline evaluated on Resume dataset.

NER is a task of finding the start and the end of an entity in a sentence and assigning a class for this entity [13]. The author took the result from [17] as the baseline, which is achieved by the Conditional Random Fields(CRF) based Bidirectional LSTM model (CRF-LSTM) with a 94.53 % F1-score. We evaluated the model on a Chinese NER dataset, Resume, annotated by [17] which consists of randomly selected 1027 resume summaries and 8 manually annotated types of named entities with YEDDA system[15]. For training and evaluation purpose, the dataset was divided into three subsets, train, dev and test set.

3.2 Single Sentence Classification

For the single sentence classification task, our task is to implement the Long short-term memory (LSTM) baseline proposed on ChnSentiCorp dataset provided by the author of the paper[12]. This dataset is divided into train, dev, and test set. The train set contains 9145 entries and the dev and test set have 1199 entries. This task is framed as a binary classification or sentiment analysis task, with the goal of outputting a binary label for each sentence, with 0 represents the negative review and 1 represents the positive review. The paper was able to achieve a 91.7 % accuracy [12]. Our task is to implement this baseline and improve it.

3.3 Sentence Pair Classification

Sentence pair classification can be broken down into many different subcategories: paraphrase identification, semantic textual similarity, or question and answer(QA) tasks. The goal was to construct model classifying the relationship between two sentences. In the case of semantic textual similarity, classification would be based on the similarity between the sentences. For QA tasks, classification would be based on whether the answers appropriately addresses the question or not. Sentence pair classification is normally a binary task; an output of 0 represents no similarity and an output of 1 shows sentence correlation.

The paper used four different datasets for this task. The dataset chosen for this task was the Bank Quesion (BQ) corpus, The BQ corpus contains 120,000 question pairs and is used commonly for sentence semantic equivalence identification [2]. The LCQMC corpus contains 260,000 question pairs and is used for QA tasks [7]. The author was able to provide us the necessary data files that were split into separate training, validation, and testing files. Three baselines were mentioned in the paper: Bilateral Multi-Perspective Matching model (BiMPM), BiMPM with Glyce, and BERT. These baselines are all extremely intricate models. For the sake of simplicity, we decided to backtrack to an Attention Based CNN (ABCNN), one of the baselines that was mentioned by Wang et al. when their BiMPM model with other competitors. The ABCNN was a much simpler and more familiar model that was easier for us to implement and make ablations to.

4 Implementation and Approach

4.1 Sequence Labelling

We implemented a character-based model, CRF-LSTM, for this task. It is a bidirectional LSTM structed model with an additional layer of CRF. At each time step, a standard LSTM's hidden state h_t only collect the information from the past knowing nothing about the future. However, for many sequence labelling tasks, it is proven to be beneficial to have access to both past and future contexts. Bidirectional LSTM is an elegant solution. The idea is simple, the algorithm is applied to the embedded input characters to obtain compute the hidden state in both the left-to-right and right-to-left directions with two distinct sets of parameters. Then the two hidden states are concatenated to form the final output [5].

The work was broken down into two steps. First, a BiLSTM and a CRF model was implemented separately. Then combined the two models by feeding the output vectors of BiLSTM into a CRF layer to jointly decode the best label sequence. Moreover, the dropout layers were applied on both input and output vectors of BiLSTM. It has been shown that using the dropout layers significantly improves the performance of the model[8].

4.1.1 HyperParameter Tuning

Hyper-parameter tuning technique were applied to improve the performance of the model including random search. Table 1 demonstrates the values of hyper-parameters the author used for the model compared to the final tuned hyper-parameters we found, only the altered hyperparameters are listed.

Model	learning rate	embedding size	hidden size
(from paper)	0.015	50	200
CRF-LSTM	0.001	128	128

Table 1: Hyperparameter values before and after tuning.

Moreover, during the training session, after each epoch, the validation accuracy is compared with the best performance has obtained so far and the model that achieves the best performance is saved for later test evaluation.

4.2 Single Sentence Classification

4.2.1 Data Preprocessing

The first step is to vectorize the sentences into number sequences to make computation easier. This is achieved by using an external dictionary obtained from Jclian's Github repository [6]. Second, the dataset contains sentences with non-equal length (Figure 1). We performed sequence padding to make all the sentences to equal length. The input length is chosen by using the 90 percentile of all the sentence length in the training dataset, which is 188.

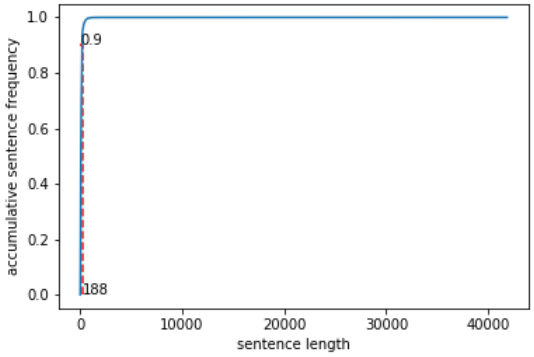

Figure 1: Accumulative Sentence Frequency Distribution with 90 percentile

4.2.2 LSTM Implementation and Parameter Tuning

Our LSTM model contains 5 layers, which include a embedding layer as the input layer, a LSTM layer, a dropout layer and a dense layer for regularizing the model. Next, We fine-tuned several parameters, including input sentence length, number of hidden unit, and number of epochs. We also added some L1 and L2 regularizers to prevent overfitting.

4.3 Sentence Pair Classification

4.3.1 Data Preprocessing

The text was first cleaned by removing any punctuation and parentheses and then preprocessed through the Jieba package designed for Chinese word segmentation. Then, the text documents were mapped into corresponding word vectors so that they would be easily transformed into word embeddings.

4.3.2 ABCNN Implementation

The ABCNN we built follows the exact structure presented by Yin et al. [16]. It consists of a simple BiCNN, essentially 2 CNNs that share the same weight but each processes a different sentence. These two CNNS are connected by a final layer that determines the correlation between a paired sentences. There are four layers in the BCNN: the input layer which deals with the word embeddings, the convolutional layer, the average pooling layer, and finally the output layer. The two CNNs are connected by an attention matrix at the convolution layer.

Because of the size of our datasets, using grid search for our deep learning model would be very slow. Therefore, due to the time constraint, we used random search to selectively choose the few hyperparameters that are the most important and relevant to our dataset. Random search has been shown to produce good performance in many examples and datasets even though it lacks the exhaustive approach used by grid search [1]. The hyperparameters that we tested was the L2 regularization constant, the number of convolution layers, and the sentence length. For the values we chose, the number of convolution layers were limited to the discrete set from 1 to 3 and the sentence length was chosen to be from the average sentence length in the dataset to the maximum sentence length in the document. It is worth to note that maximizing the sentence length would guarantee the highest possible accuracy scores but greatly hinders the performance and running time off our models. We were curious to see this relationship and if there was an optimal way to maximize accuracy without the cost of overusing memory.

5 Results

In general, we successfully implemented the three baseline models discussed above and some fine-tuning were done to improve the models' performance. All experiments were done through Google Colaboratory using a Tesla K80 12GB GPU and a Xeon 2.3Ghz CPU.

5.1 Sequence Labelling

Table 2 demonstrates the result achieved by the baseline we implemented compared with the ones proposed in the paper. After the tuning, all scores have significantly improved. Our tuned model was able to reach a F1-score of 95.81% compared to 94.41%. It is worth noticing that after tuning, the performance of our model outperforms the 3 other baseline models, Lattice-LSTM, Lattice-LSTM+Glyce and BERT mentioned in the paper with F1-score of 94.46%, 95.67% and 95.78% respectively. It is only slightly below the performance achieved by the Glyce+BERT model the author proposed with a F1-score of 96.54%.

Model	**Precision %**	**Recall %**	**F1-Score %**
CRF-LSTM	94.53	94.29	94.41
CRF-LSTM(our implementation)	95.85	95.82	95.81

Table 2: Results for Sequence Labeling Task

5.2 Single Sentence Classification

Table 3 shows the results for single sentence classification task. Our model achieves an accuracy score of 87.5%, which is lower than the baseline proposed in the paper.

Model	Accuracy %
LSTM	91.7
LSTM(Our implementation)	87.5

Table 3: Results for Single Sentence Classifcation Task

5.3 Sentence Pair Classification

Table 4 shows the performance of ABCNN compared to the baselines covered by Chen et al. for the BQ dataset. As explained above, we decided to choose our model to be the baseline of the baseline for the sake of familiarity and flexibility. We were not able to beat the baseline scores recorded, and were about 4% off compared with the accuracy. We mentioned before that we were curious to see if maximizing sentence length would have

Model	Precision %	Recall %	F1-Score %	Accuracy %
BiMPM	82.3	81.2	81.7	81.9
ABCNN (our implementation)	-	-	77.77	77.65

Table 4: Results for Sentence Pairing Task on the Bank Question corpus

6 Discussion and Conclusion

Due to the time constraint and computational resource limitation, we only chose to work on a subset of baselines on a selected dataset proposed in the paper. We successfully implemented the baseline model of Sequence Labelling and Single Sentence Classification task mentioned by Wu et al., and managed to made ablations that outperforms the author proposed baseline performance for Sequence Labelling task. For the Sentence Pair Classification task, we went one step back and implemented the baseline of the baseline proposed in the paper.

For the Sequence Labelling task, we successfully implemented the baseline, the CRF-LSTM model on Resume Dataset, proposed in the paper. We then further explored different hyper-parameter tuning method to improve the model's performance including random search. The performance of the model during the training session was noted after each epoch base on the validation accuracy. We then saved the model with the best performance to evaluate on the test set. As a result, the our implemented model not only beat the baseline accuracy, but outperforms other 3 models mentioned in the paper done on the Resume dataset with a F1-score of 95.81%. Future training can be done on the other three chinese datasets, OntoNote, Weibo and MSRA, with our implemented model to confirm the improvement in accuracy.

In terms of the single sentence classification task, we were able to implement the model, whereas our accuracy cannot beat the baseline proposed in the paper despite the tunning of several parameters. We noticed that accuracy score on this dataset yields much lower accuracy score then other binary classification dataset proposed in the paper [12] and the model trained overfit significantly. We assume this could be related to the nuture of this dataset. This dataset covers three different areas including hotel review, book review and PC review, and it was collected by web-scraping. Besides, this dataset contains both traditional Chinese and simple Chinese sentences, so the dictionary we used might not be able to cover all the words in the dataset. Further research can be done to better understand the dataset and try to implement the model on some other datasets used in the paper.

For the sentence pair classification task, we were not able to completely replicate the BiMPM baseline model but instead chose the ABCNN model. Our results from using ABCNN were not able to beat the BiMPM baseline scores recorded in the paper, but we managed to bring the difference margin close to 4%. At the end of implementing the ABCNN model, we returned to the BiMPM model. There were a few problems that made implementing this model hard. The first and biggest issue was that we were not given a vector file containing the word or character embeddings related to the dataset. Without these vector files, we were at risk of reproducing the baselines incorrectly compared to the authors. There were some possible solutions such as pretraining our own embeddings through

packages like Word2Vec or downloading popular open source word vectors like GloVe or FastText, but this does not guarantee that we have the same vector files as the authors. The second problem was the complexity of the model. Due to the amount of work we had to do to replicate the baselines, not enough time was allocated for BiMPM implementation.

7 Statement of Contribution

The project was split up into 3 tasks and each group member was in charge of their own task. Yimeng focused on sequence labelling, Sichen worked on single sentence classification, and Long-Quan worked on sentence pair classification. The group worked together to collectively write the paper.

References

[1] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[2] Jing Chen, Qingcai Chen, Xin Liu, Haijun Yang, Daohe Lu, and Buzhou 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*, pages 4946–4951, 2018.

[3] Hao Cheng, Hao Fang, Xiaodong He, Jianfeng Gao, and Li Deng. Bi-directional attention with agreement for dependency parsing. *arXiv preprint arXiv:1608.02076*, 2016.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 4, pages 2047–2052. IEEE, 2005.

[6] Jclian. Sentiment Analysis, 2019. URL `https://github.com/percent4/Sentiment_Analysis/tree/master/sentiment_analysis`.

[7] Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962, 2018.

[8] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*, 2016.

[9] Michael Roth and Mirella Lapata. Neural semantic role labeling with dependency path embeddings. *arXiv preprint arXiv:1605.07515*, 2016.

[10] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *Fifteenth annual conference of the international speech communication association*, 2014.

[11] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*, 2017.

[12] Wei Wu, Yuxian Meng, Qinghong Han, Muyu Li, Xiaoya Li, Jie Mei, Ping Nie, Xiaofei Sun, and Jiwei Li. Glyce: Glyph-vectors for chinese character representations. *arXiv preprint arXiv:1901.10125*, 2019.

[13] Hang Yan, Bocao Deng, Xiaonan Li, and Xipeng Qiu. Tener: Adapting transformer encoder for name entity recognition. *arXiv preprint arXiv:1911.04474*, 2019.

[14] Jie Yang, Yue Zhang, and Fei Dong. Neural word segmentation with rich pretraining. *arXiv preprint arXiv:1704.08960*, 2017.

[15] Jie Yang, Yue Zhang, Linwei Li, and Xingxuan Li. Yedda: A lightweight collaborative text span annotation tool. *arXiv preprint arXiv:1711.03759*, 2017.

[16] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016.

[17] Yue Zhang and Jie Yang. Chinese ner using lattice lstm. *arXiv preprint arXiv:1805.02023*, 2018.