NLP HOME WORK 1 “CWE” REPORT

MANOOCHEHR JOODI BIGDELLO, 1860273

KEY FEATURES OF DATA PREPARING AND TENSORFLOW MODEL

I. This report is for Chinese Word Segmentation (BIES format) and I implemented Bidirectional dynamic RNN with stacked LSTM’s and GRU’s as backward and Forward Layers and CRF Layer on Top, while Using Adam optimizer and negative log likelihood with gradient and L2 regularization for optimizing loss function.

II. First I concatenate the 4 sighan 2005 dataset and build our ngram dictionary based on that, I used unigram, bigram (min_freq>2), trigram (min_freq>3) and four gram (min_freq>8) it shows the best result considering lack of google colab resource and GPU allocation (Details shown in Table 1).

III. For every character I’m sending list of 14 ngrams to training algorithm (for every character we assume 2 left and 2 right neighbors with character itself as a 5 length word window and then calculate all possible unigrams (=5), bigrams (5-1=4), trigrams (5-2=3) and fourgrams (5-3=2), in total 14 ngrams.

IV. I am using pre-trained embedding (Embedding size= 100) for unigrams, and for bigram, trigram and fourgrams mean of embedding of their unigrams was used. (boost the performance more than 1% in PKU dataset)

V. Persian Language specific characters as Start, End Sentence, PAD and UNK was used, this characters are impossible to appear in any non-Persian texts. (Because we are using ngrams it’s possible to Start and End chars to appear in Text based on my preprocessing phase.

Training phase

i. For general model I use 18K PKU, 30K MSR, 30K CITYU and 50K AS, 128K in total as train set and I do random shuffling for better distribution of 4 datasets, Also I used 4K (1K from each) as dev set and all 4 gold set (21K) concatenation as test set.

ii. Using (CLIP=5, Hidden Dim=128, Stacked Layers=1) as “Basic Hyper parameters” Because of Lack of resources for computation, and tune Learning Rate (LR), Dropout and Batch size to achieve better results.

iii. First we tune the hyper parameters with PKU dataset to find basic for tuning. Than we use that and tune it slightly in concatenation of all datasets. Somehow same hyper parameters works well for all datasets. (LR=0.001, Dropout=0.4, Batch size= 16).

iv. Using pre-trained embedding’s, dictionary, different ngrams and list of 14 mix of ngrams for every character boost our model performance and On MSR and PKU my model exceed the paper, for AS and CITYU because of lack of time and resource I couldn’t run more epochs for better results. **Almost all models at the end of every epoch was saved on google drive for documentation.** I upload my general model on all 4 datasets 128K subset and models that exceed the paper in MSR and PKU saved in Google Drive, also other datasets are running with model for possible improvements.(lack of time because of calculation).

v. Source of some of errors: words bigger than 4 character in MSR, special characters like ℃ in PKU.

Minimum Frequencies	**Size of Dictionary**	**comment**
2gram = 0, 3gram = 0, 4gram =0	15.5 M	Colab Session Crash (C.S.C)
2gram = 1, 3gram = 1, 4gram =1	3.7 M	C.S.C For Large datasets
2gram = 1, 3gram= 1, 4gram =2	2.8M	C.S.C For Large datasets
2gram = 2, 3gram = 2, 4gram =5	1.5M	1 Epoch 96.1% acc on PKU
2gram = 2, 3gram = 3, 4gram =8	1.1M	10 Epoch 96.27% acc on PKU, 97.95% MSR
2gram = 0, 3gram = 3, 4gram =8	1.7M	20 Epoch 96.42% acc on PKU, 98.14% MSR
2gram = 3, 3gram = 6, 4gram =10	750K	8 Epoch 94.77% acc on All of 4 gold tests

***Table 1**. Ngrams usage tuning on concatenation of 4 datasets*

Epoch	1	2	3	4	5	6	7	8	9	10	Avg
PKU acc	94.05	94.63	94.93	94.95	95.26	95.05	95.12	95.29	95.27	95.38	95.00

***Table 2**. Results on PKU dataset without using pre trained Embedding's*

Hyper Parameters	**Accuracy on PKU**
Basic Hyper parameters + LR = 0.01, Dropout = 0.8, Batch Size = 64 + bigram min_freq=2	95.94 % in 5 Epoch
Basic Hyper parameters + LR = 0.01, Dropout = 0.6, Batch Size = 32 + bigram min_freq=2	95.95 % in 5 Epoch
Basic Hyper parameters + LR = 0.001, Dropout = 0.6, Batch Size = 32 + bigram min_freq=2	96.14 % in 5 Epoch (+0.04% better than paper)
Basic Hyper parameters + LR = 0.001, Dropout = 0.4, Batch Size = 32 + bigram min_freq=2	96.27 % in 5 Epoch (+0.17% better than paper)
Basic Hyper parameters + LR = 0.001, Dropout = 0.4, Batch Size = 16 + bigram min_freq=0	**96.42** % in 10 Epoch (+0.32% better than paper)

***Table 3**. Hyper parameter tunings*

Hyper parameters	**PKU**	**MSR**	**CITYU**	**AS**
Basic Hyper parameters + LR= 0.001, Dropout=0.4, Batch Size= 16 + bigram min_freq=2, Epoch= 10	96.27 **(+0.17)**	97.93	96.04	C.R.C
Basic Hyper parameters + LR= 0.001, Dropout=0.4, Batch Size= 16 + bigram min_freq=0, Epoch= 20	96.42 **(+0.32)**	98.14 **(+0.04)**	C.R.C	C.R.C

***Table 4**. Accuracy of every dataset in gold test set with just training on their own training set and concatenated dictionary and pre trained embedding’s*

Hyper parameters	**PKU**	**MSR**	**CITYU**	**AS**	**All 4 test sets Concat**
Basic Hyper parameters + LR= 0.001, Dropout=0.4, Batch Size= 16, Epoch=8	95.44	96.6	93.4	94.2	94.77

Table 5. *. Accuracy of every dataset in gold test set with training on 128K shuffled subset with concatenated dictionary 750K size and pre trained embedding's.*

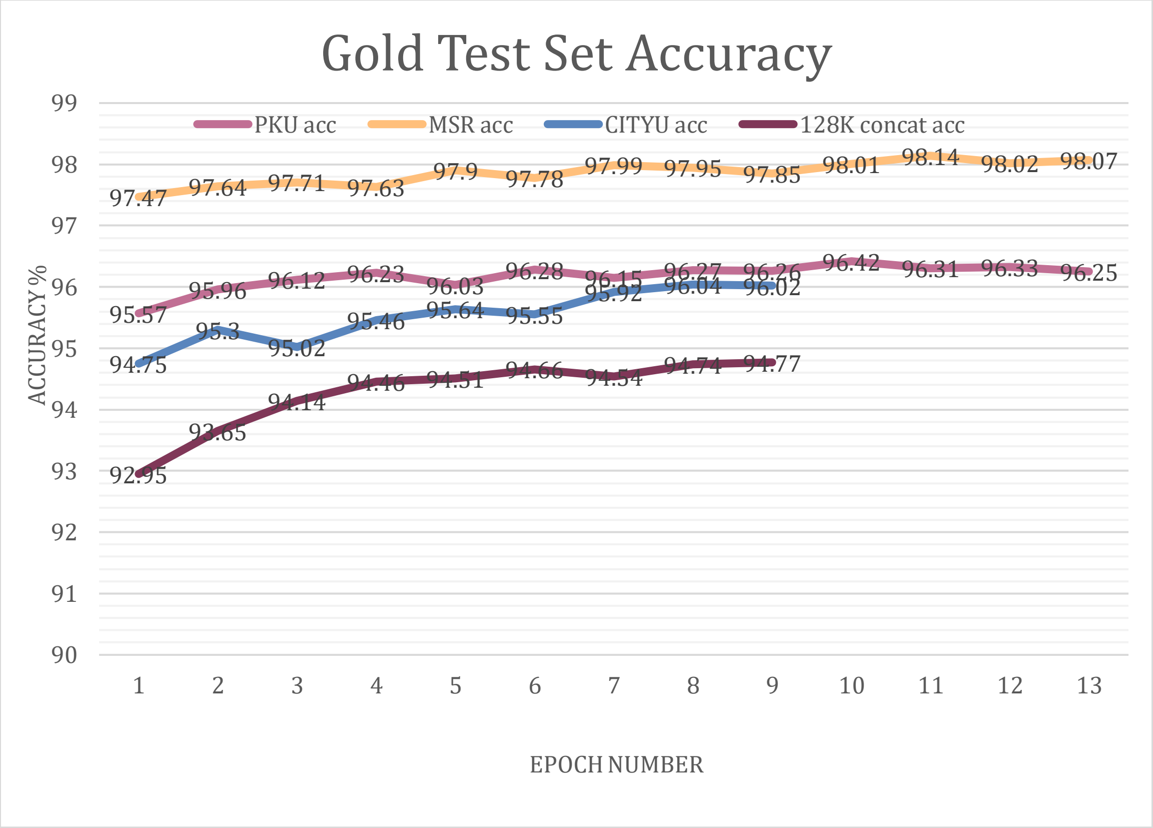

Figure 1. Accuracy on gold test in different epochs, for every dataset we consider the best hyper parameters and report the corresponding result of its own gold set(like paper) and for 128K result is for all 4 gold test set 21K.

- Saving tensor board in google colab cause session crash.

References

1- Wwww.tensorflow.com

2- State-of-the-art Chinese Word Segmentation with Bi-LSTMs, Ji Ma, Kuzman Ganchev and David Weiss, EMNLP 2018