SAPIENZA
UNIVERSITÀ DI ROMA

Chinese Word Segmenter Report

NLP HOMEWORK 1

MAKINWA, Sayo Michael
[1858198]

23.04.2019

1.0 The Task, The Model

The task is to implement a state-of-the-art word segmenter - sequence tagging - model, encoding the output in the BEIS format with each character marked as belonging to one of the four classes - B (Beginning of a word), E (End of a word), I (Inside of a word), or S (Single character).

I implemented a sequence tagging model which I have described with the image below:

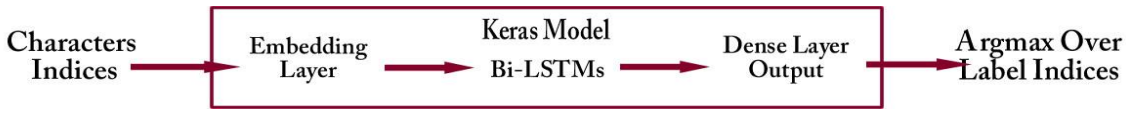

Fig. 1. Description of my model

2.0 The Dataset

At the start of this project, I built an ***English dataset, with 9,500 lines of texts***, on top of which I built all my functions, this is so I can understand what I am doing and all the outputs I get, since I do not understand Chinese characters. This also helped me to ***try out quite a number of hyper-parameters*** since the size of data is considerably small. Afterwards, I ***merged the MSR and PKU*** training dataset into one file, and built everything – vocabulary and model – on top of that. This combined data makes up ***105,980 lines*** of texts, for which each training takes about 1 hour per epoch!

3.0 Model Parameters

Across the literature on word segmentation with RNNs, some network parameters were frequently used and they yielded good results, so ***I fixed those, while I varied the input (X) structure*** since this still varies widely across the literature. It turns out that this indeed, makes a huge difference in the quality of the model. I also varied the number of stacked Bi-LSTM layers. The table below contains the parameters I fixed:

Embedding Size	64	Learning Rate	0.001	Training Data Size	100,673
Hidden Layers Size	256	Batch size	32	Dev. Data Size	5,299
Optimizer	Adam Optimizer	Dropout Rate Per RNN	0.2	Padding Size	50; truncating='pre', padding='post'

Table 1. Model Parameters

4.0 Parameter Tuning

4.1 Tuning The Input (X) Structure

Ji Ma et al., in their paper, noted that they embedded the unigrams and bigram features at each position, concatenated them and fed them to the network. So I tried a few modifications on this structure. Fig. 2 below describes the first set of the structure that I tried. In the structures A and B, I built the vocabulary for both unigrams and bigrams together as one, applied keras pad_sequence to make all input to be of size 50, then fed the corresponding digits to the embedding layer of the model. Fig. 3 compares the accuracy of both structures, both with ***one Bi-LSTM layer, trained over 20 epochs***, and clearly, structure A performs better than B, however, both still fall short. Of course, this is because the values of the numbers of the bigrams are too far apart away from the values of the unigrams in the same vocabulary, this made the data unbalanced. ***This is further proof that the right way to combine unigrams and bigrams is to use different vocabularies and embed them separately before concatenating them. This is proof by refutation***. The better performance of structure A can be explained as be explained as a factor of the pad_sequences truncating the end of each sequence, meaning that the bigram

representations gets chopped off in favour of the unigram representations, which causes the input values to sort of normalize. Using this intuition, I removed the bigrams altogether and this led to really good results (shown in section 4.3).

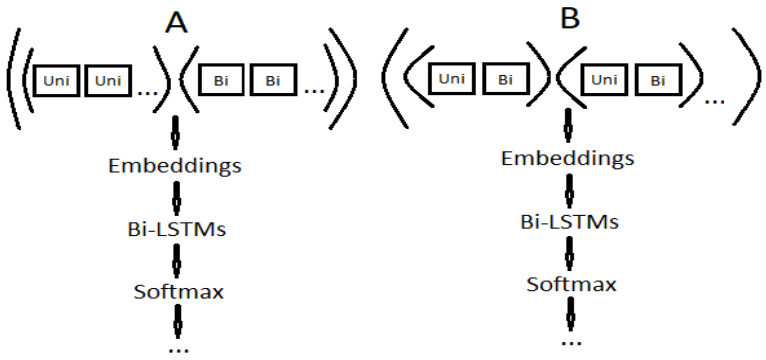

Fig. 2. Two input structures to the model

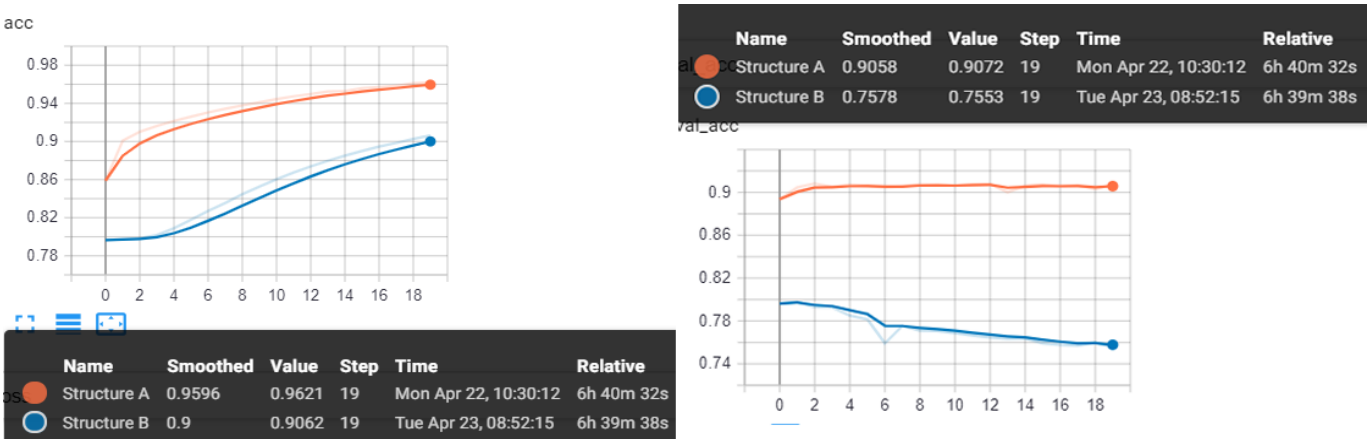

Fig. 3. Charts comparing training accuracy (left) and validation accuracy (right) for structures A and B described in Fig. 2

4.2 Stacking More Bi-LSTM Layers

I proceeded to see how stacking more Bi-LSTM layers may improve structure B in Fig. 2 above. Results on training data is close, but results on validation data is more interesting and shown below:

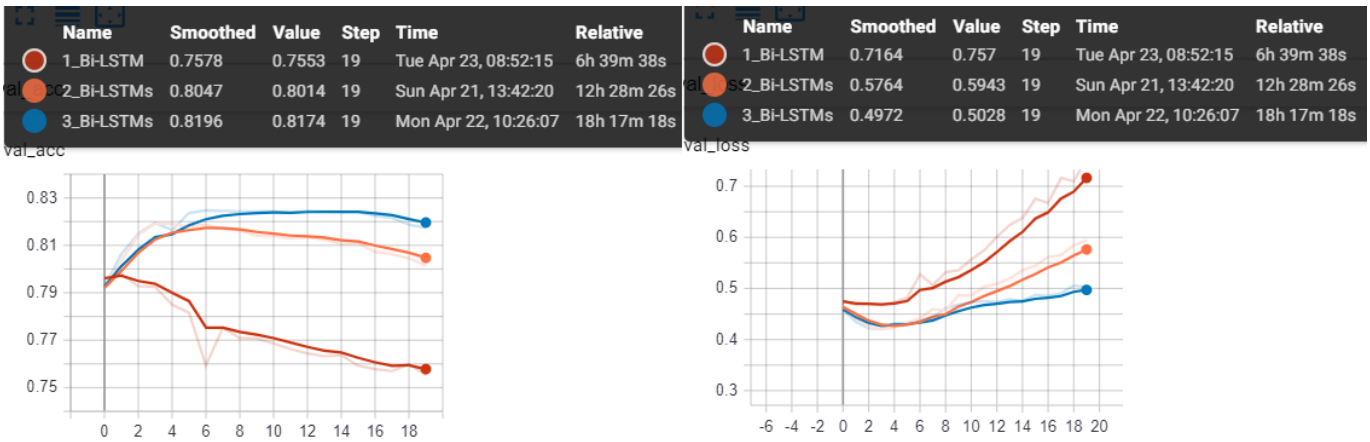

Fig. 4. Charts comparing validation accuracy (left) and validation loss (right) with number of stacked Bi-LSTMs

4.3 My best model (Unigrams-only input)

As explained in the last part of section 4.1 above, I built my next model, using only the unigrams. This time, however, I used 3 stacked Bi-LSTM layers. The performance turned out to be really good, with validation accuracy getting as high as 97.86% after the 20th epoch, and unlike the other structures that overfits, it does not!

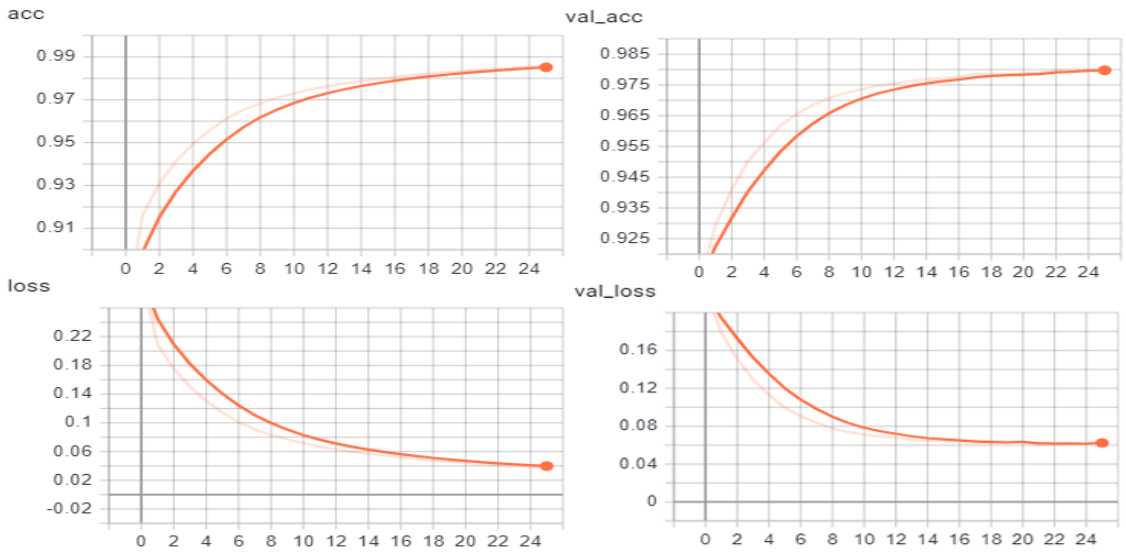

Fig. 5. Charts comparing training and validation accuracy on the unigrams-only model

	Training accuracy	Training Loss	Validation Accuracy	Validation Loss
Unigrams-only Input	0.9845	0.0415	0.9786	0.0688
Structure B Input	0.8789	0.3109	0.8174	0.5028

Table 2. Results from the models from the different input structures after 20 epochs

	MSR Gold Data	PKU Gold Data
Precision Score	0.9483	0.9153

Table 3. Precision score of the unigrams only model on the MSR and PKU gold data using score.py

5.0 References

Bo Zheng, Wanxiang Che, Jiang Guo, Ting Liu. 2017. Enhancing LSTM-based Word Segmentation Using Unlabeled Data. *In CCL* 2017 DOI:10.1007/978-3-319-69005-6_6

Ji Ma, Kuzman Ganchev, and David Weiss. 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 31 - November 4, 2018. c 2018 Association for Computational Linguistics

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, Xuanjing Huang. 2015. Long Short-Term Memory Neural Networks for Chinese Word Segmentation. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal, 17-21 September 2015. c 2015 Association for Computational Linguistics.

Yan Shao, Christian Hardmeier, Joakim Nivre. 2018. Universal Word Segmentation: Implementation and Interpretation. *In Transactions of the Association for Computational Linguistics*, vol. 6, pp. 421–435, 2018. Action Editor: Sebastian Pado. Submission batch: 3/2018; Revision batch: 6/2018; Published 7/2018.
c 2018 Association for Computational Linguistics. Distributed under a CC-BY 4.0 license

Yossi Adi, Joseph Keshet, Emily Cibelli, and Matthew Goldrick. 2017. Sequence Segmentation Using Joint RNN and Structured Prediction Models. *Proc IEEE Int Conf Acoust Speech Signal Process*. 2017 Mar; 2017: 2422–2426. doi: 10.1109/ICASSP.2017.7952591