SAPIENZA
UNIVERSITÀ DI ROMA

Chinese word segmentation – Homework I

Natural Language Processing - MSc. In AI and Robotics

Professor: Roberto Navigli PhD.
Andrés Fernando Arciniegas Mejía – Matricola No. 1874069

1 Assignment

Processing words and the connection between them is a key requirement for many tasks in language processing, and some languages, like Chinese, do not separate the words in written text. The objective of this work is to create a model that processes a text with no spaces, and generates the corresponding segmentation annotation, in BIES format, with deep learning tools.

2 Datasets and preprocessing

Four datasets were provided for training and testing (AS, CITYU, MSR, PKU), with different sizes each. An analysis of their composition was performed, finding that the distribution of the sentences varying considerably in length from one to another, thus zero-padding to the input of the network was considered. However, that brings some disadvantages, since datasets like AS have an average length per sentence of 11.8, and PKU of 95.85, a zero-padding of 50 might either add a great amount of empty spaces in some samples, leading to unnecessary computation; and truncate too much in others, leading to loss of information. The datasets were redistributed to have at most 50 characters per sentence after removing spaces, improving training efficiency. Considering that the model must generalize to any of these datasets, they were all merged together and shuffled to create a general dataset. **On prediction:** Since the input has no spaces and could be considerably longer than the input of the network (50), all the sentences are split in known single characters like (“。 ， 、 ？ ！ ； ： ”), finding only 40 samples longer than 50 in the 54158 test sentences. These 40 are split by 50, and each part is predicted individually.

3 The model

Most of the configuration suggestions in the architecture in the provided paper were followed, building a bidirectional stacking LSTM, formed with 256 units each; with concatenated embeddings of unigrams and bigrams of length 50 as inputs, which had been processed through a dictionary of all possible unigrams and bigrams available (1’043.604 elements); and a time-distributed dense softmax layer with four values for prediction tags as output. Masking to zero was included to ignore zero-padding. To overcome overfitting, dropout in the LSTM input and in the recurrent units was implemented, and as optimizer Adam was used after showing better results than SGD. The library *Keras* made fairly simple to apply several of these concepts.

4 Results and discussion

It was an interesting, demanding project that provided a good set of challenges that pushed the student forward to come up with a good solution for a real-world problem. Accordingly, as stated by the author, getting better results in the model might be an outcome of an extended process of hyperparameters tuning, since the possible combination can scale bigger quickly. The accuracy of the achieved model reaches ~92 depending on the dataset, which can be considered high, and comparable to other State-of-Art results. Not a requirement, but a helper definitely, is having knowledge of Chinese language, to have a better understanding of the output of the model, and hence be able to provide adequate manual feedback.

5 Tables and Figures

5.1 Dataset

Training preprocess

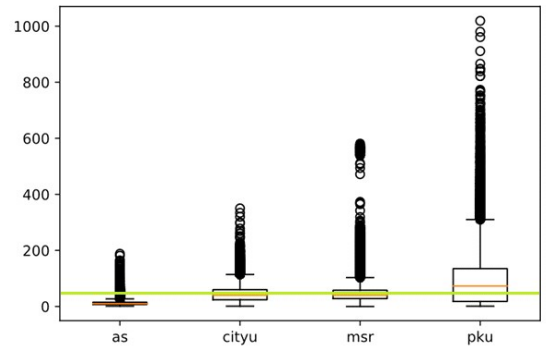

Figure 1. Lengths of sentences in original training dataset. Input maximum length (50)

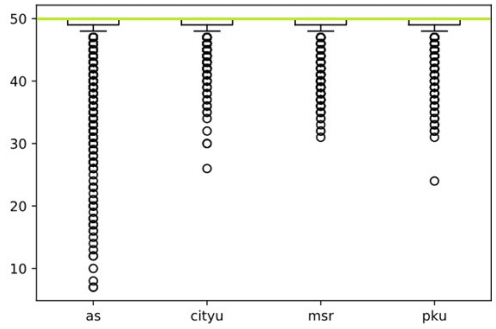

Figure 2. Lengths of sentences in training dataset reordered to have at most a length of 50 characters. Improves training time around 30%.

Testing preprocess

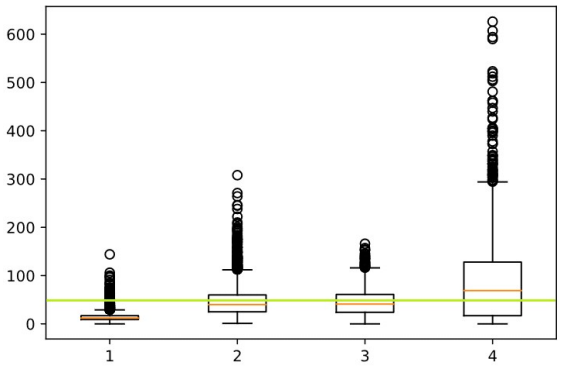

Figure 3. Lengths of sentences in original testing dataset. Many samples above the maximum input length.

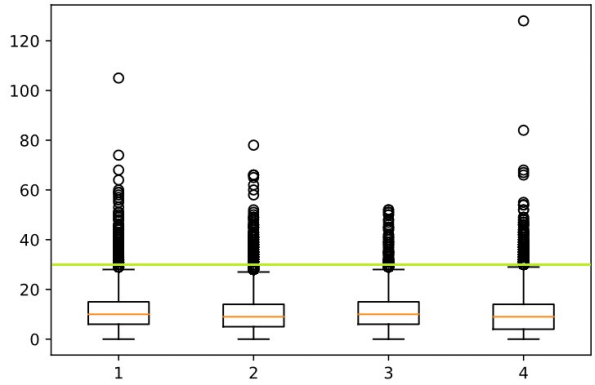

Figure 4. Lengths of sentences in testing dataset Split in known single characters. Only few samples above 50

5.2 Model

Model architecture

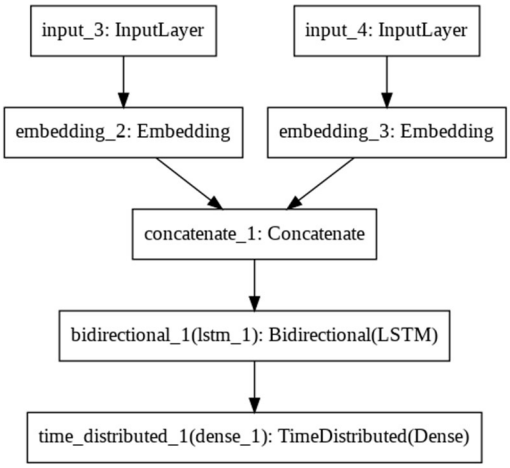

Figure 5. Architecture of the model

Training

Training was performed with 30% of the merged dataset, and final model with 100% after tuning parameters. The following figures depict one of the performed grid search variations. Static learning rate of 0.0005, and variable dropout in recurrent units with values [0, 0.1,0.4,0.6]. Shows sings of overfitting.

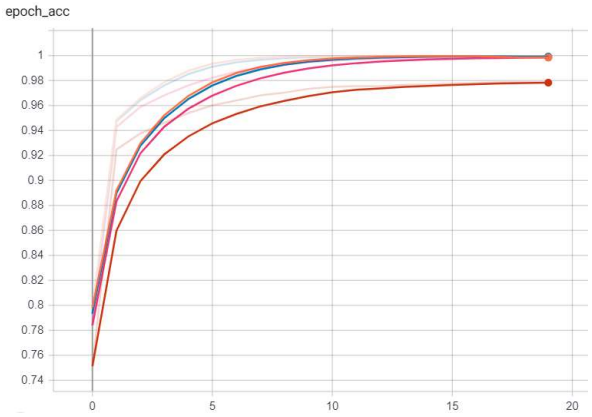

Figure 6. Training accuracy in a sample of grid search.

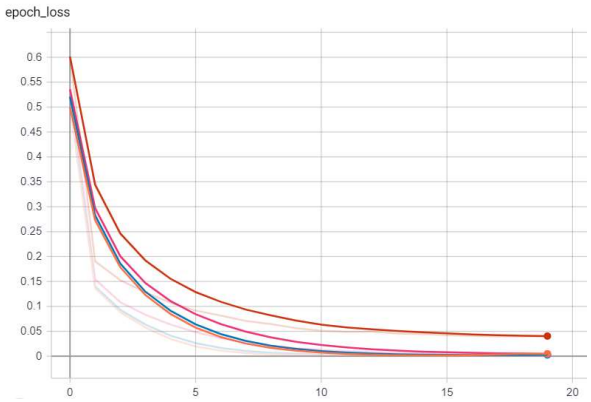

Figure 7. Training accuracy in grid search variation.

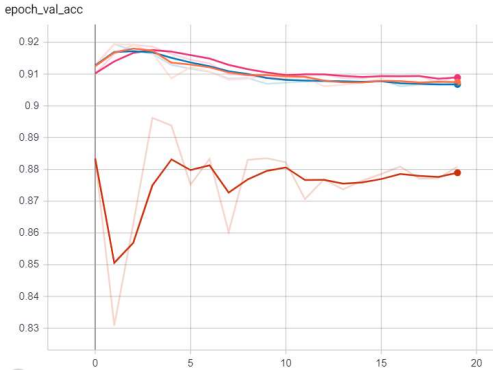

Figure 8. Validation accuracy in grid search variation.

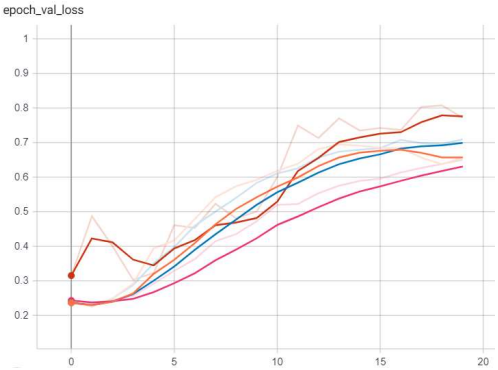

Figure 9. Validation loss in grid search variation.

Final Model

Parameter	Value
Optimizer	Adam
Learning rate	0.0005
Batch size	512
Epochs	20

Parameter	Value
LSTM units	256 each
LSTM Input dropout	0.4
LSTM Recurrent dropout	0.4
Input size	50

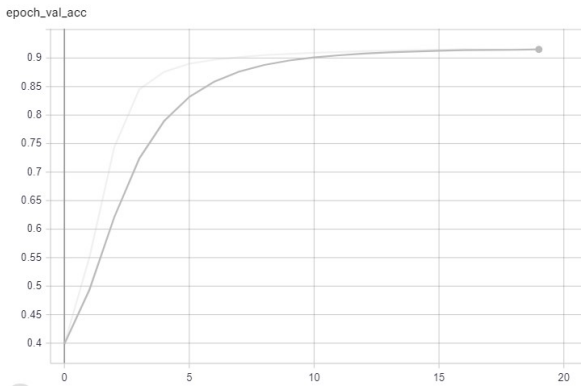

Figure 10. Validation accuracy in final model.
Maximum Accuracy: 0.925

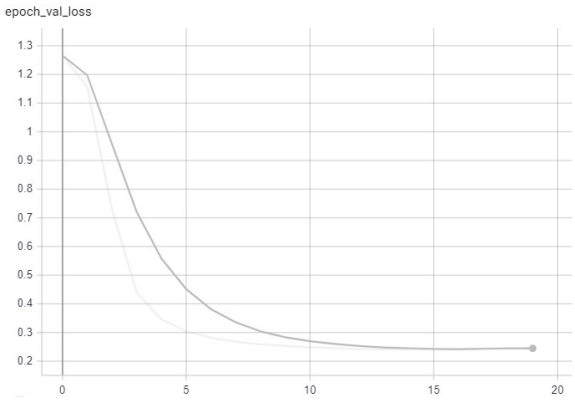

Figure 11. Validation accuracy in final model
Minimum Accuracy: 0.2466