NLP HW 1
Chinese Word Segmentation

Karim Ghonim - Matricola: 1774086

25 Apr 2019

SAPIENZA
UNIVERSITÀ DI ROMA

1 Dataset preparation

A dataset consisting of the four available simplified datasets was used for the entirety of the project. They were stripped of the spaces and the corresponding labels were generated. The complete training dataset consists of $\approx$ 870000 samples, while the dev set consists of $\approx$ 22000 samples. The training set was used to generate the vocabulary, here in the form of unigrams and bigrams as all the following presented experiments made use of both in order to dampen the loss of temporal information. Initially 6,592 unique unigrams were found in the dataset, which were all kept, and 1,045,226 unique bigrams, of which only 100,002 were kept with the rest discarded of. The size of the vocabulary of bigrams came as a result of a grid-search procedure in order to have the smallest dictionary without impairing the performance of the network. All the corresponding embeddings for both unigrams and bigrams were learned by the network with no pre-trained embeddings used, thus obtaining task oriented embeddings.

2 Network architecture - Experiments

2.1 Baseline

For the first experiment, an architecture inspired by the State-of-the-art Chinese Word Segmentation with Bi-LSTMs paper, a non-stacking bi-directional LSTM model with a 4-way softmax layer was implemented in order to serve as a baseline for the remaining expirements. Grid-search was performed on some hyperparameters, being the embedding size of the ngrams, number of hidden units and learning rate, of the baseline model with some results provided below showing that the change in performance was very slight between experiments 1 and 2.

2.2 Experiment 3

Just adding a dense layer with 16 hidden units with tanh as its activatiion function before the softmax in order to add some non linearity to the classification, as well as using nesterov momentum exponential decay. This experiment shows how some elementary changes can increase the performance of the model, as even though the final dev accuracy was similar, it reached it much faster.

2.3 Experiment 4

For this experiment, I integrated two new features, being CRF and self attention. CRF is used instead of softmax as the softmax makes local choices, even if the Bi-LSTM captures some temporal information, the tagging decision is still local and the neighbooring tagging decisions are not put into consideration. The self attention layer will be trained to capture the most informative part of the sentence and condensing it in a "context vector", thus in a way capturing higher-level information about the characters.

3 Hyperparameters

Hyperparameters "A" were only used for experiment 1, while all the other experiments had hyperparameters "B". They are both displayed in the table below.

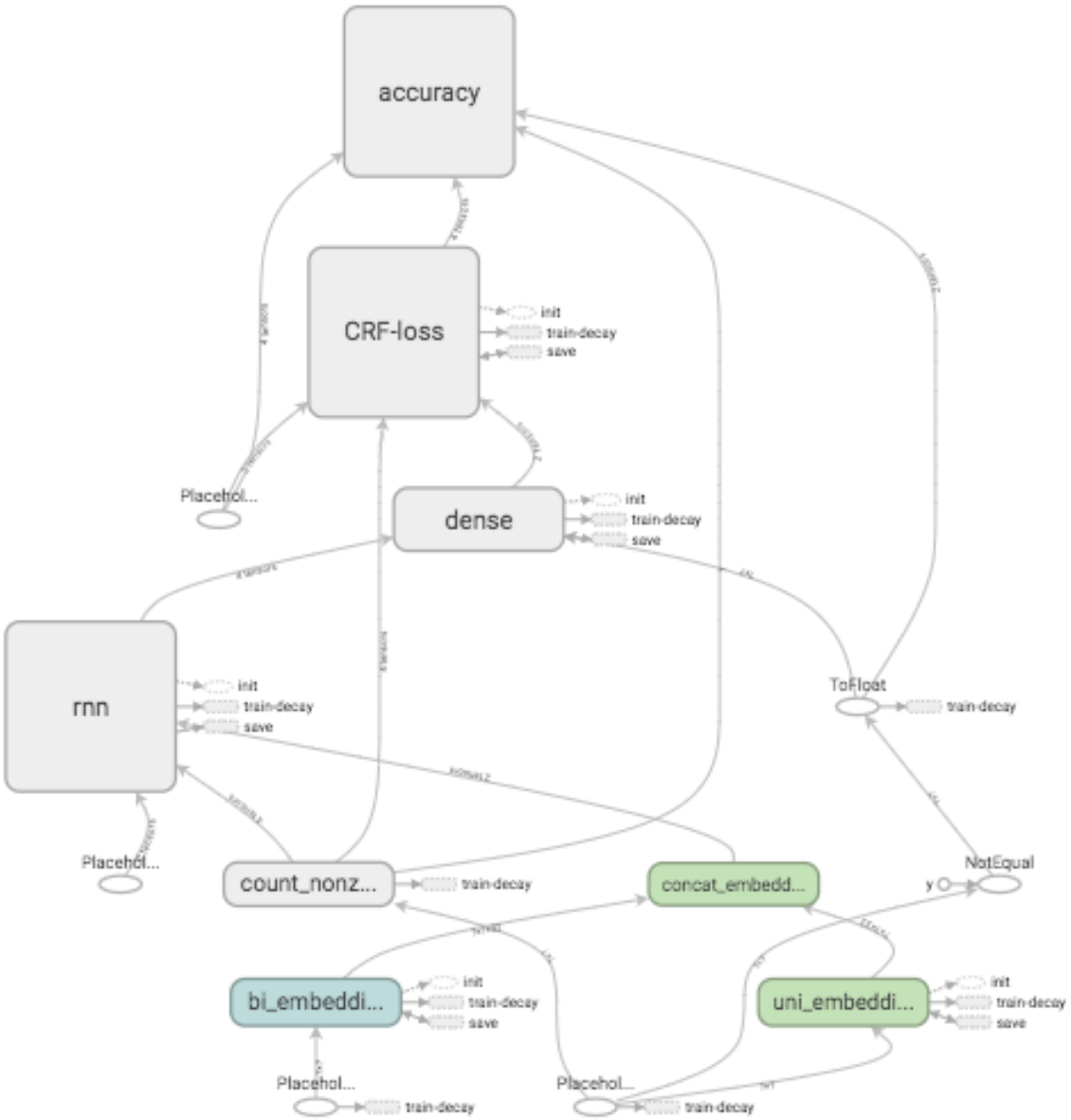
4 Results

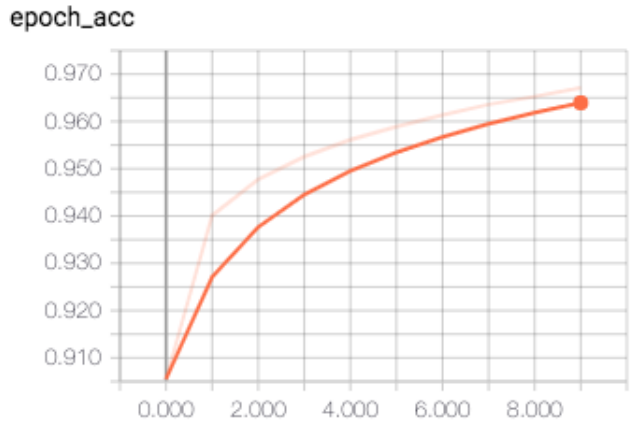
All results are drawn from testing on the complete concatenated dataset and can be found in the table below.

Hyperparameters	A	B
Uni Embedding Size	32	64
Bi Embedding Size	32	64
Hidden Size	128	256

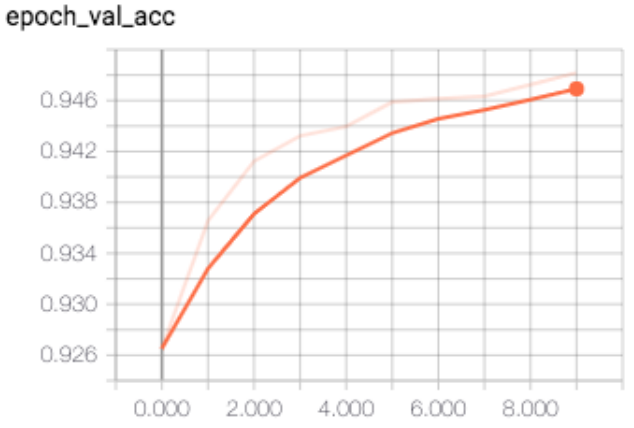
Experiments	Train acc	Dev acc	Precision
1	96.71	94.82	93.9
2	96.32	94.72	93.7
3	97.74	94.88	94.2
4	98.74	95.14	94.3

Figure 1: Final model graph

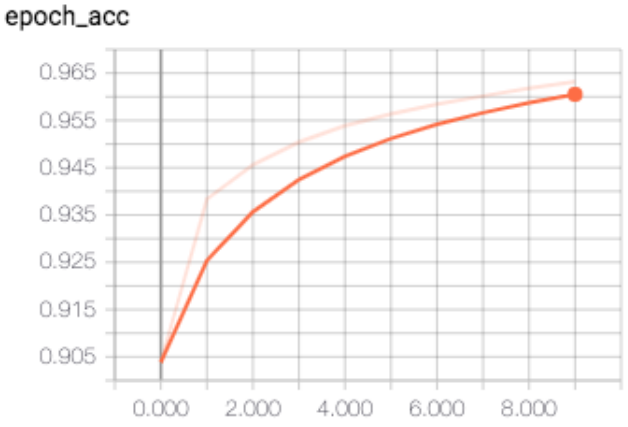




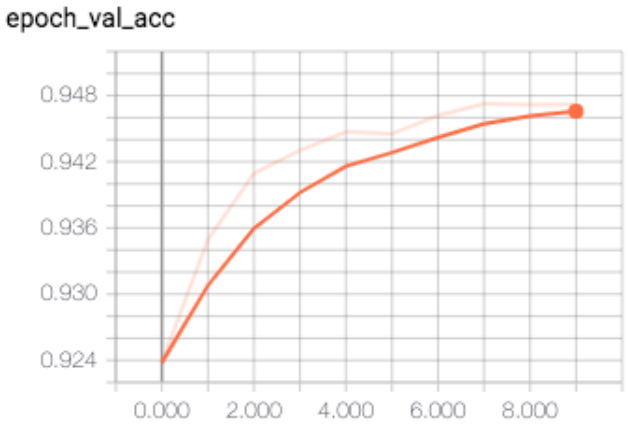
(a) Experiment 1 Train accuracy



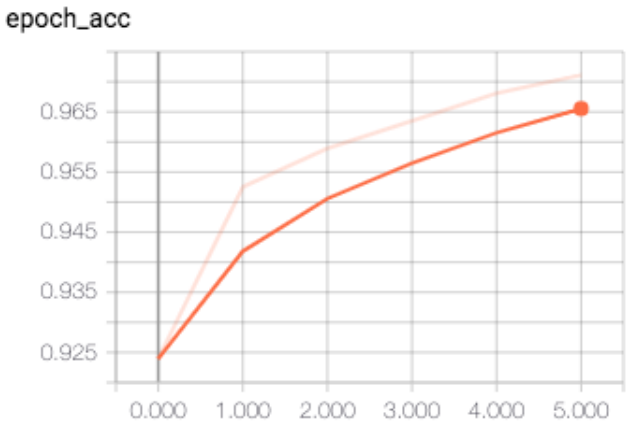
(b) Experiment 1 Dev accuracy



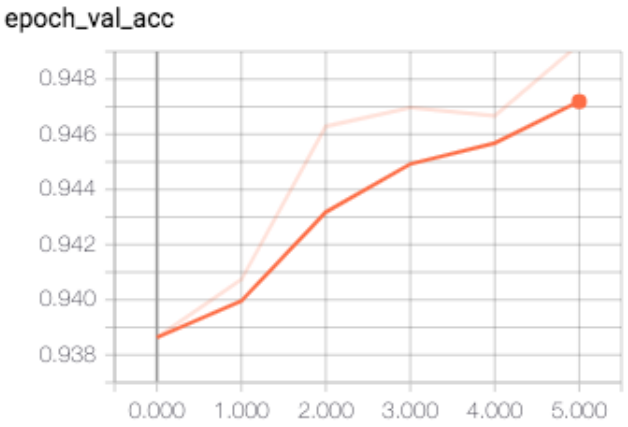
(a) Experiment 2 Train accuracy



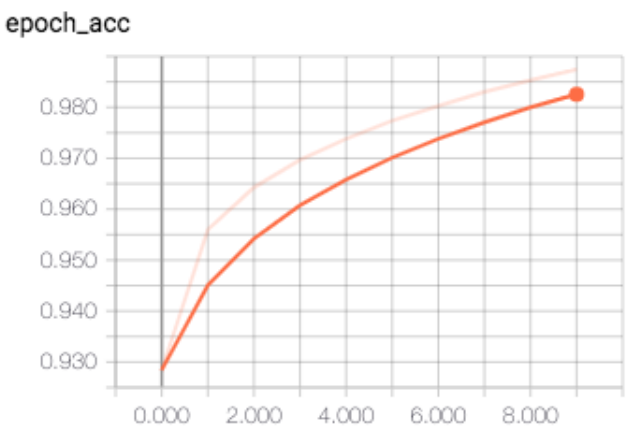
(b) Experiment 2 Dev accuracy



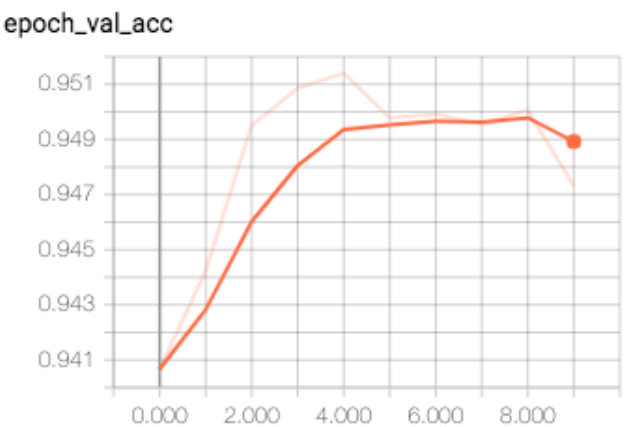
(a) Experiment 3 Train accuracy



(b) Experiment 3 Dev accuracy



(a) Experiment 4 Train accuracy



(b) Experiment 4 Dev accuracy