The goal of the project is to create a word segmenter for simplified Chinese; the input file is a document with sentences without whitespaces, whose characters are labelled using the BIES format.

DATASET

The training and development datasets are based respectively on a concatenation of the four training datasets (simplified AS, simplified MSR, simplified CITYU, simplified PKU) and their four gold datasets. All duplicate sentences, in both training and development sets, are removed, to have unique sentences.

There are in total 797,383 sentences for the training set and 21,372 for the Dev set.
In the training dataset only the sentences that have at least 3 characters are considered, in order to have a similar length distribution for both datasets (as shown in Fig 1), but also to clean the dataset. Indeed, many sentences are short and not useful for the training part (e.g. sentence where there is only a punctuation character "。").

PRE-PROCESSING

In the pre-processing phase, the input and the label files are created: each word is obtained by splitting the sentences with respect to the whitespace and the punctuation [1], because sometimes it happens that the punctuation is merged with the words (e.g. "新堰镇《农业科技简报》").

The two main features used in the model are the unigrams and bigrams, so two different vocabularies are created:

- Unigram Vocabulary: only the most common Chinese ideogram (5,000 ideograms) are included, the punctuation and the Latin words + the Arabic numbers have their own tag [2] (<PUNCT>, <LATIN>, respectively), also the padding tag (<PAD>) and the OOV tag (<UNK>) are added.
- Bigram Vocabulary: only the most common bigrams (40,000 ideograms) are included, the tag for the end of the sentence is added (</s>), to have the unigram and bigram representation of a sentence with the same length, and the padding tag (<PAD>) and the OOV tag (<UNK>) are added.

MODEL

The implemented model [3] (as shown in Fig.2) takes two different inputs per sentence (unigrams and bigrams), whose lengths are equal to 200 (longer sentences are truncated, and shorter sentences are padded) and fed as batches of size 64; their embedding matrices (300 x Vocab_size) are trained during the training phase. A custom embedding matrix from the pre-trained model Fasttext is used as initial weights (for all the unigrams or bigrams not present in the Fasttext vocabulary, a uniform random vector is given). These two Embedding matrices are then concatenated and passed to a Bi-LSTMs model with two stacks of 256 hidden units each, whose outputs are merged and passed to the SoftMax layer predicting the BIES format.

MODELS DESCRIPTION

The first two models used the SGD optimizer with different learning rates and momentum equal to 0.95, both models were trained with different numbers of epochs and steps. Two different pre-processing are adopted (in the first model the matrix is randomly initialized, whilst in the second one the embedding matrix is built using the pretrained Fasttext and training the model using only the sentences whose lengths are between [5-200]).

The SGD optimizer proved to be very slow, as shown in Fig 3 and Fig 4 where it is noticeable that the models were still learning. Therefore, the RMSprop optimiser was used in the third and fourth models, as it was seemingly faster than the SGD. In both models the punctuation tag <PUNCT> is added, however the third model only uses those sentences whose length is between [1-200].

The last three models have a similar precision on the Dev set: the improvement from the fifth model to the seventh one is very small (around 0.03%) even though different optimizers are used: RMSprop in Model 6 and Nadam in Model 5 and 7, where the latter is a version of RMSprop with the momentum parameter.

Although trained with different numbers of epochs, Model 6 and Model 5 show a similar behaviour (as shown in Fig.5). However, Model 5 appears to reach a similar result than its counterpart in half the number of epochs.

The model which works best is Model 7, where the Nadam optimizer and the pre-processing described above are used. In Fig.6 it is possible to see the plot of the loss and the accuracy on the dev set.

Further improvements could be reached through a different evaluation approach since in the adopted evaluation sentences that are longer than the set maximum length (200) are truncated. This implies that each split sentence is considered as a full sentence, meaning that words could be split into single characters, and thus the BIES labels could be assigned without taking into account the linguistic properties of the sentence. For instance, longer words at the end or at the beginning of a sub-sentence could be assigned the wrong labels.

Model	Uni/Bigram Embedding size	Uni/Bigram Vocab size	Hidden Size	Optimizer	Learning Rate	Dropout / Recurrent Dropout	Max Length	Epochs/ Steps	Batch Size	Precision
Model 1	64/64	5002/30002	256	SGD	0.001	0.25/0.2	200	20/500	64	70.6%
Model 2	300/300	5002/30002	256	SGD	0.0005	0.25/0.2	200	35/800	64	78.35%
Model 3	300/300	5003/30002	256	RMSprop	0.0005	0.3/0.25	200	20/800	64	92.37%
Model 4	300/300	5003/30002	256	RMSprop	0.001	0.3/0.25	200	20/500	64	92.5%
Model 5	300/300	5004/35002	256	Nadam	0.01	0.3/0.25	200	20/800	64	93.3%
Model 6	300/300	5004/30002	256	RMSprop	0.001	0.3/0.25	200	40/800	64	93.5%
Model 7	**300/300**	**5004/40002**	**256**	**Nadam**	**0.02**	**0.3/0.25**	**200**	**30/800**	**64**	**93.6%**

Table 1: Parameters used to train the models and results (Precision) on the development set.

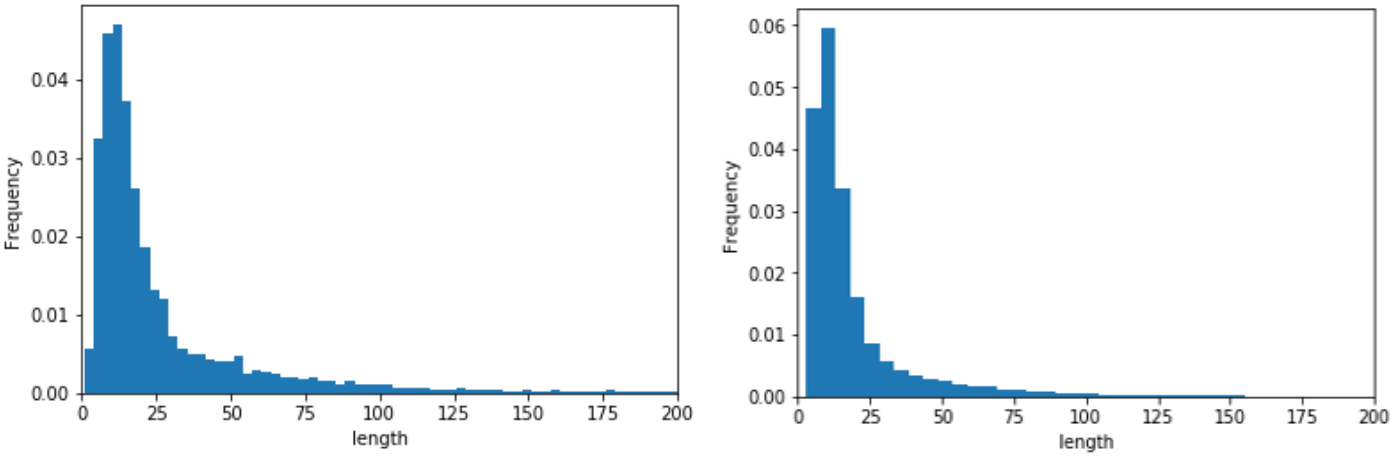

Figure 1: The left histogram shows the distribution of the length for the Dev set; the right histogram shows the distribution of the length for the Train set

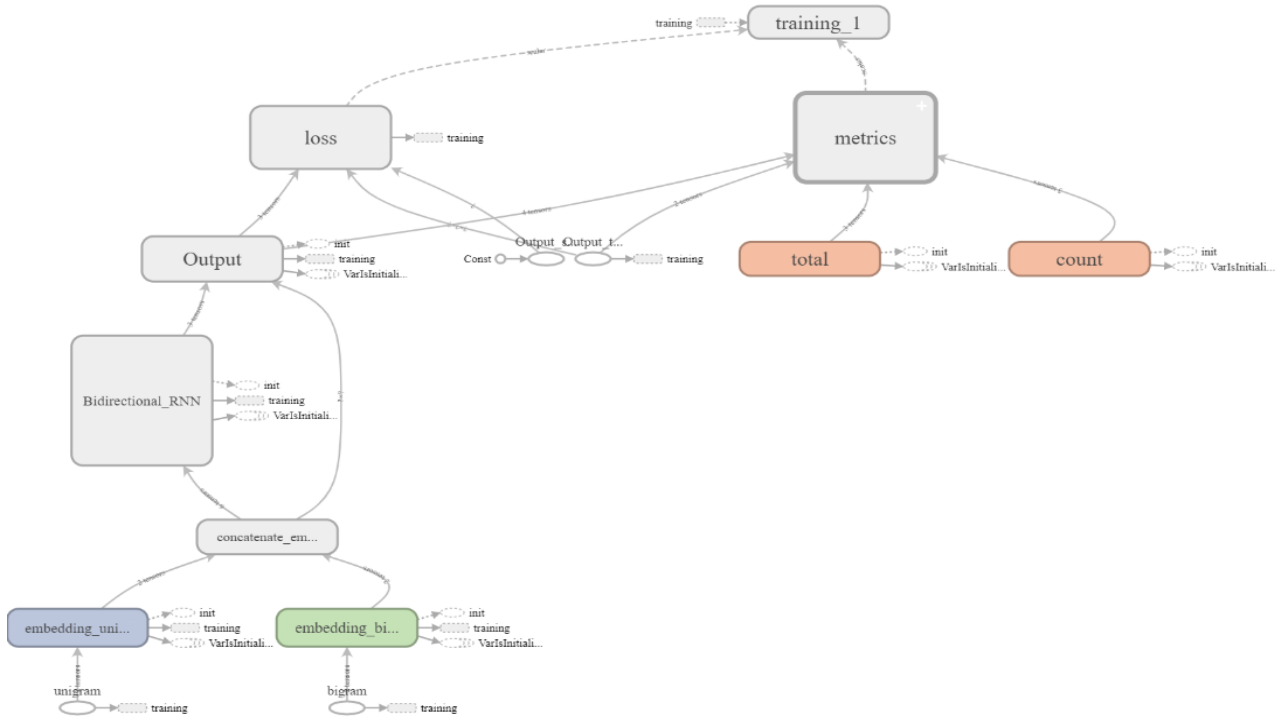

Figure 2: Model Architecture from Tensorboard

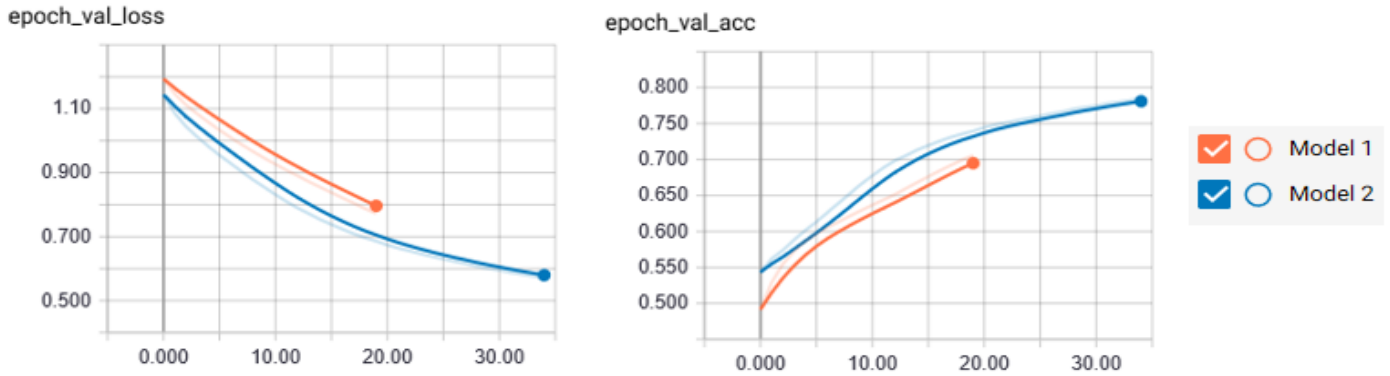

Figure 3: Loss and accuracy plots for the Model 1 and Model 2 on the development set

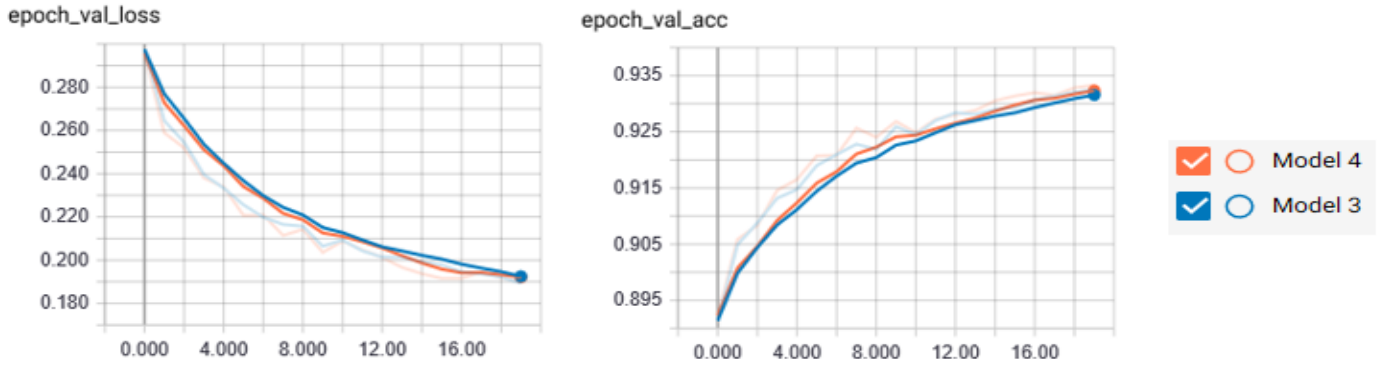

Figure 4: Loss and accuracy plots for the Model 3 and Model 4 on the development set

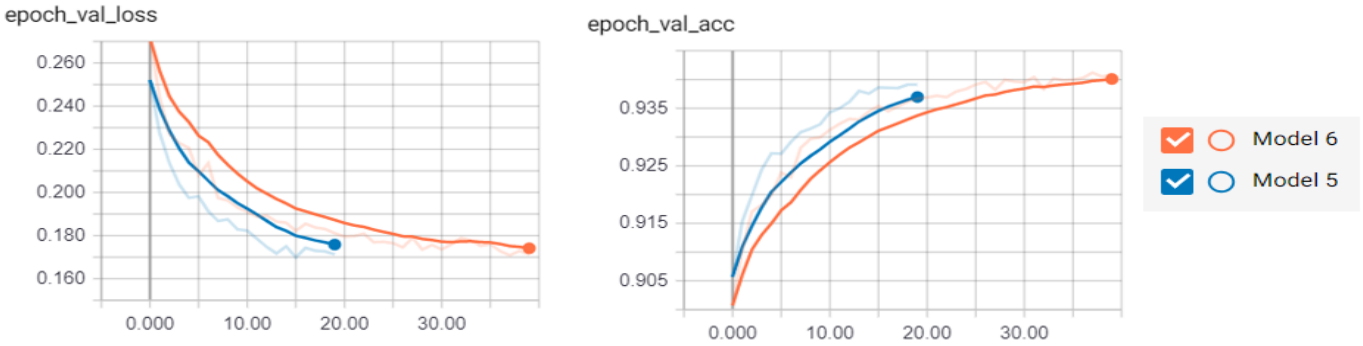

Figure 5: Loss and accuracy plots for the Model 5 and Model 6 on the development set

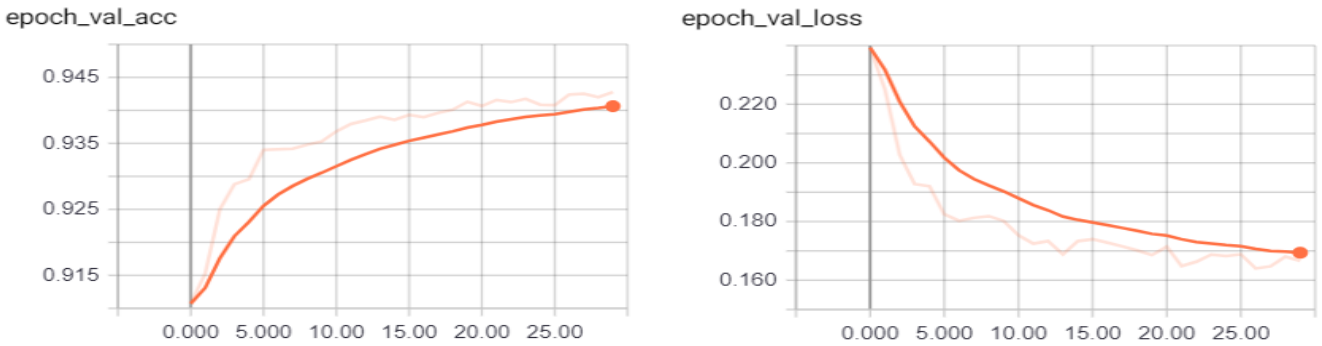

Figure 6: Loss and accuracy plots for the Model 7 on the development set

REFERENCES

[1] Wang, J., Zhu, Y., Jin Y., (2014): A Rule-Based Method for Chinese Punctuations Processing in Sentences Segmentation. International Conference on Asian Language Processing (IALP), pages 195

[2] Huang, W., Cheng, X., Chen, K., Wang, T., Chu, W., (2019): Toward Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning.

[3] Ma, J., Ganchev, K., Weiss, D., (2018): State-of-the-art Chinese Word Segmentation with Bi-LSTMs. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4902–4908. Association for Computational Linguistics