State-of-the-art Chinese Word Segmentation with Bi-LSTMs

Michał Ostyk-Narbutt (1854051)
Prof. Roberto Navigli
Natural Language Processing Homework 1

April 24, 2019

1 Introduction

Chinese word segmentation (CWS) has gained a lot of popularity within the Natural Language Processing (NLP) community lately. This is especially the case since Deep Learning (DL) became prominent and Recurrent Neural Networks (RNN) have been introduced.

In this project, we explore the use of Bi-directional Long Short Term Memory (LSTM) neural networks. This report will present the approach used, the architecture along with several hyperparameters and the end results. However, the biggest problem remains with dealing with Out-of-Vocabulary words (OOV).

2 Dataset and State-of-the-Art

In this project, we draw inspiration from [1], where the researchers have built, and trained two Bi-LSTM's on 4 varying datasets ('as', 'msr', 'pku, 'cityu') each containing 'training' and 'validation' data which can be found here [2]. These include two which were in Traditional Chinese which needed to be translated into simplified prior to training. The authors claim that their model outperforms most other state-of-the art models, reaching accuracy $\geq 95\%$ on most datasets.

Prior to training and building the model, the datasets underwent the following steps of preprocessing. First using HanziConv [3], the Traditional datasets were converted to to Simplified Chinese. Then From the original file, two files were created; An Input File, which contains no spaces and serves as the X-data, as well as a Label File, which in BIES format serves as the target of classification (in reality these would be numerical classes).

3 Model and Approach

The approach for building feature vectors from the dataset which would be fed to the model for training is explained in Algorithm 1 below.

In this paper, the utilised model is the non-stacking model from [1] shown on Figure 1. Initial testing proved that the stacking model gave much worse results. The last layer is of size 4 (number of BIES classes), and is activated using SoftMax. Several hyperparameters were explored, However, not using grid-search as it proved to be too expensive. Some of the most effective ones are presented in Table 1.

Algorithm 1 Model data creation

```
For each dataset, take the Input and label files for training and validation.
For the training Input file (no spaces):
    create unigrams and bigrams for the entire text file
    create separate vocabularies for unigrams and bigrams
    while masking OOV create seperate dictionaries mapped by numbers.
    read the Input file line by line and create feature vectors (scalars) using the above dictionaries.
For the validation Input file (no spaces):
    take the training dictionaries and create feature vectors (scalars) for each line of the text
For both Label files (no spaces):
    Convert from BIES to numerical classes
    perform one-hot-encoding.
for training purposes, pad accordingly for both files.
```

Table 1: Hyperparameters of the BI-LSTM model

Embedding Layer − Unigrams	64	Recurrent dropout	0.35
Embedding Layer − Bigrams	16	Batch size	128
Learning rate	0.0015	Hidden layer size	256
LSTM dropout	0.4	Padding size	30
kernel regularizer l2	0.01	bias regulizer l1	0.01

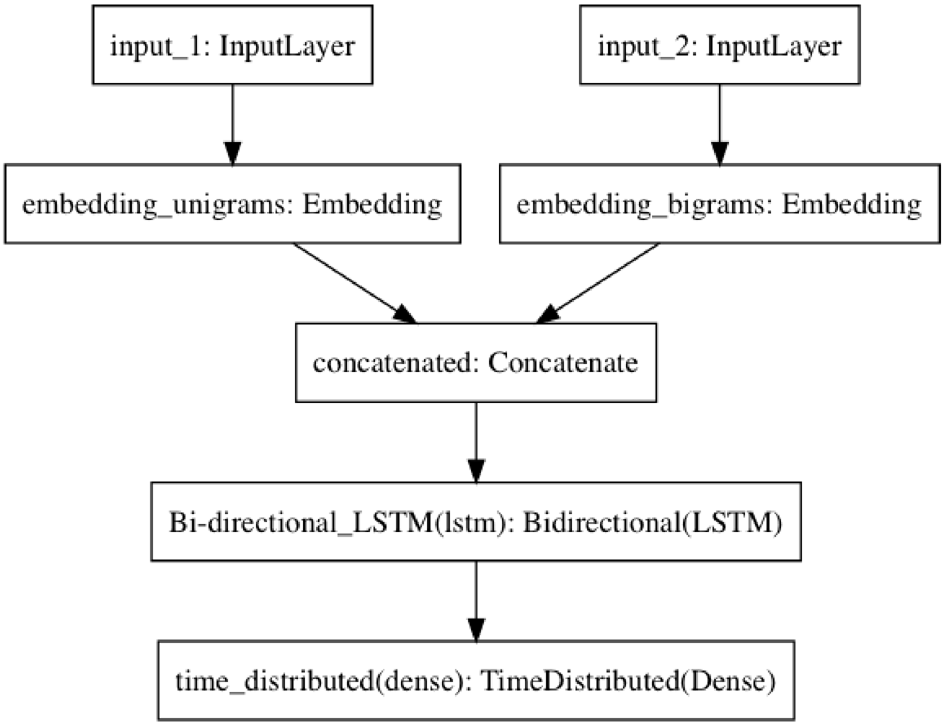
input_1: InputLayer
input_2: InputLayer
embedding_unigrams: Embedding
embedding_bigrams: Embedding
concatenated: Concatenate
Bi-directional_LSTM(lstm): Bidirectional(LSTM)
time_distributed(dense): TimeDistributed(Dense)

Figure 1: Non-Stacking Model used for training, including two separate inputs for unigrams and bigrams.

4 Results

The best model was trained on the 'pku' and 'msr' datasets which amount 105980 samples for training and 5930 sample for validation. Figure 2 present the training results. As shown, the training stops quite early within 4 epochs. This is due to the imposed early stopping monitoring of the validation loss and precision. As previously stated the padding size was set to 30. Other sizes were explored however, smaller proved to be to unstable, and larger was not representative of the data as the median line length far below 40. For testing purposes the model has an extra None input layer which can intake variable line lengths, hence ideal for predictions of large files.

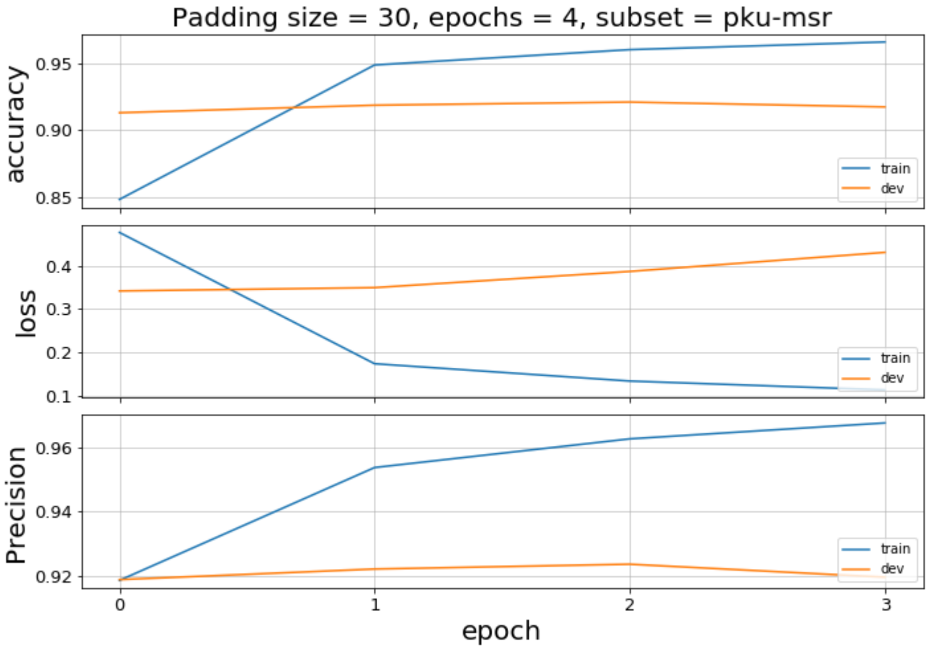

Figure 2: Training and validation results compared using loss and accuracy

5 Conclusion

To sum up, the model achieves quite good accuracy, however, in this case we are more interested in precision which grows on slightly and then falls. Some of these flaws can be associated to OOV words, which could be slightly resolved better regularization and more data. The final validation (dev) accuracy seems promising but for the growing loss.

References

[1] Ma, Ji & Ganchev, Kuzman & Weiss, David. (2018). State-of-the-art Chinese Word Segmentation with Bi-LSTMs.

[2] `http://sighan.cs.uchicago.edu/bakeoff2005/`

[3] `https://pypi.org/project/hanziconv/`