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

SAPIENZA
UNIVERSITÀ DI ROMA

Contents

1 Introduction

Chinese word segmentation (CWS), which is no easy task 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 [2], where the researcher's 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 [3]. These include two which were in Traditional Chinese which needed to be translated into simplified prior to training. The authours' claim to outperform 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 [4], 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.

Algorithm 1 Model data creation

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

In this paper, the utilised model is the non-stacking model from [2] 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.

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

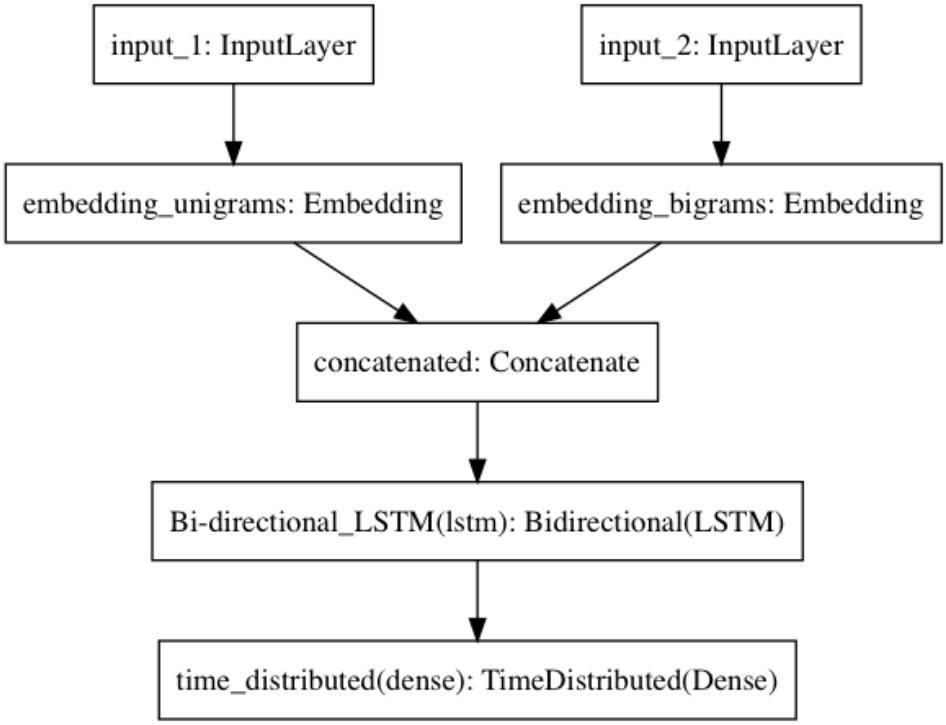

Figure 1: Non-Stacking Model used for training, including two separate inputs for unigrams and bigrams.

4 Results

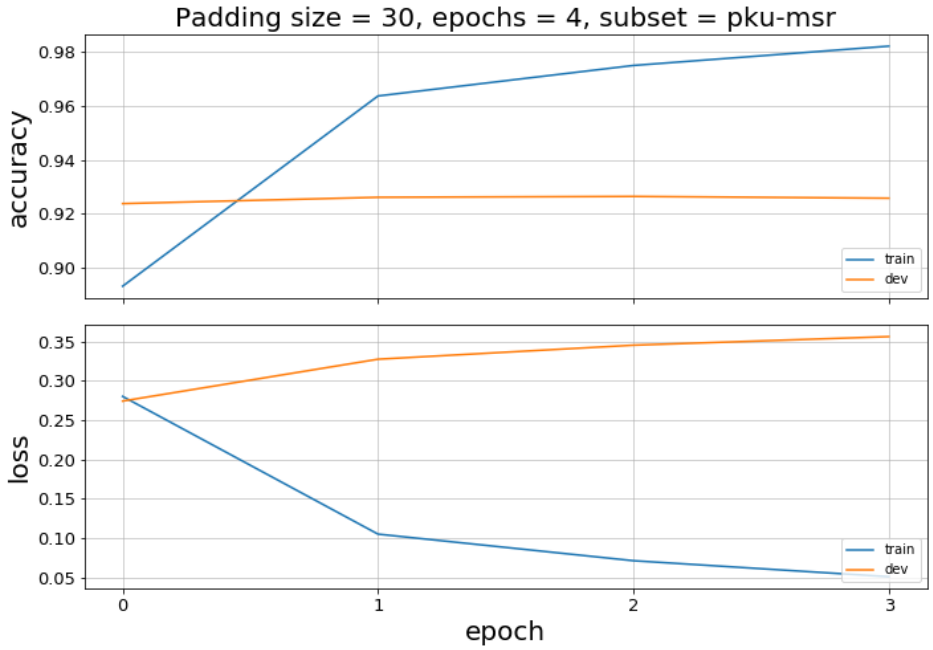

Figure 2: Training and validation results compared using loss and accuracy

5 Conclusion

The student has to deliver via the Google form: a link to the Gitlab shared project with the source code and any additional data needed to run the software a paper of up to 4 pages (+infinite pages for references, images, tables, graphs, etc.) including: a brief introduction to the project problem, a brief state of the art, an illustration of the methods/approach/techniques (min. 1 page), a quantitative (and ideally a small qualitative) evaluation of the system, some analysis of the results.

References

[1] `https://github.com/Ostyk/Chinese-LSTM`

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

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

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