Natural Language Processing **Emanuele Giona**
A.Y. 2018/2019

Homework 1: Chinese word segmentation

Task description and base model overview

Chinese word segmentation is an instance of word segmentation, which is the task of splitting a string of written natural language into its component words. The majority of languages relies on a space character to delimit each word in written texts, but there are languages like Chinese, Japanese, Thai, Lao, and Vietnamese which either delimit sentences, phrases, or syllables instead of words, making the task non-trivial [1].

In order to tackle such non-triviality, a variety of neural network approaches has been used, like the one presented in *State-of-the-art Chinese Word Segmentation with Bi-LSTMs* [2], which is the basis for this homework assignment. Instead of a sequence to sequence task, for this homework word segmentation has been reduced to sequence tagging, making use of the BIES format to tag each character of a given sequence.

The authors describe a quite simple model: based on 1-grams and 2-grams, using pretrained word embeddings, their model is composed of two layers: a Bi-LSTM and a dense one.
The concatenation of the embeddings of 1-grams and 2-grams is fed into the Bi-LSTM layer, and its output into the dense layer to obtain a probability distribution over the BIES tags for each character in the sequence.

Data preprocessing

The datasets being used are from the 4th SIGHAN Workshop [3], namely: Academia Sinica (AS), City University of Hong Kong (CITYU), Peking University (PKU), Microsoft Research (MSR); the AS and CITYU datasets have been converted to simplified Chinese via the HanziConv tool [4], as seen in [2]. In order to obtain pretrained embeddings, vocabularies of fixed size were built from a merged dataset of the previous four datasets[1], taking the most frequent n-grams only, and a simple implementation of Word2Vec [5] has been applied to the same merged dataset, as shown in Table 1.
In order to evaluate the model, each original dataset's test set has been split into development set and test set, with 20% and 80% proportions respectively; all the measurements are related to these newly defined test sets.

Implementation and proposed enhancements

There are several directions to improve the model proposed by the authors, and two of them might be using 3-grams features in addition to 1-grams and 2-grams, or adding more Bi-LSTM layers.
Hyper-parameters have been tuned via a grid search approach[2] picking the ones maximizing the test set accuracy, achieving 90.07% on a reduced dataset[3]. The proposed models are trained with the same set of hyper-parameters previously found via grid search (differently from [2]), but, in the same way of the reference paper, they are trained and tested singularly on each dataset.
As shown in Table 3, both the proposed enhancements seem to address overfitting, either virtually increasing the data handled (3-grams addition), or increasing the number of internal parameters to be trained (more layers). As per Table 4, the best performing model is a 2 Bi-LSTM layers model trained on CITYU, when averaging the accuracy achieved across all datasets' test sets.

Shortcomings and future work

The proposed models probably suffer from a subpar hyper-parameters choice due to the grid search being run on a reduced dataset, instead of fine-tuning them, as seen in [2]. They also seem to handle data inconsistencies[4] found in AS and CITYU datasets (as pointed out in [2]) better than the baseline model, but they still perform poorlier than on the other two datasets.
In order to improve performance regardless of the previous considerations, imposing to learn relationships between the BIES tags might help both the baseline and the enhanced models: *i.e.* tagging a certain character in the sequence with 'E' implies the presence of its previous associate 'B'-tagged character, and viceversa.

[1] obtained via UNIX command line concatenation
[2] after 10 epochs, according to Table 2
[3] 20 000 lines dataset, evenly picked from all datasets, then split into training (80%), development (10%), and test (10%) sets
[4] see Figure 1

Tables and graphs

n-grams	Word types	Vocabulary size	Dimension
1-grams	6511	150 000	64
2-grams	1 033 501	350 000	64
3-grams	5 234 998	500 000	64

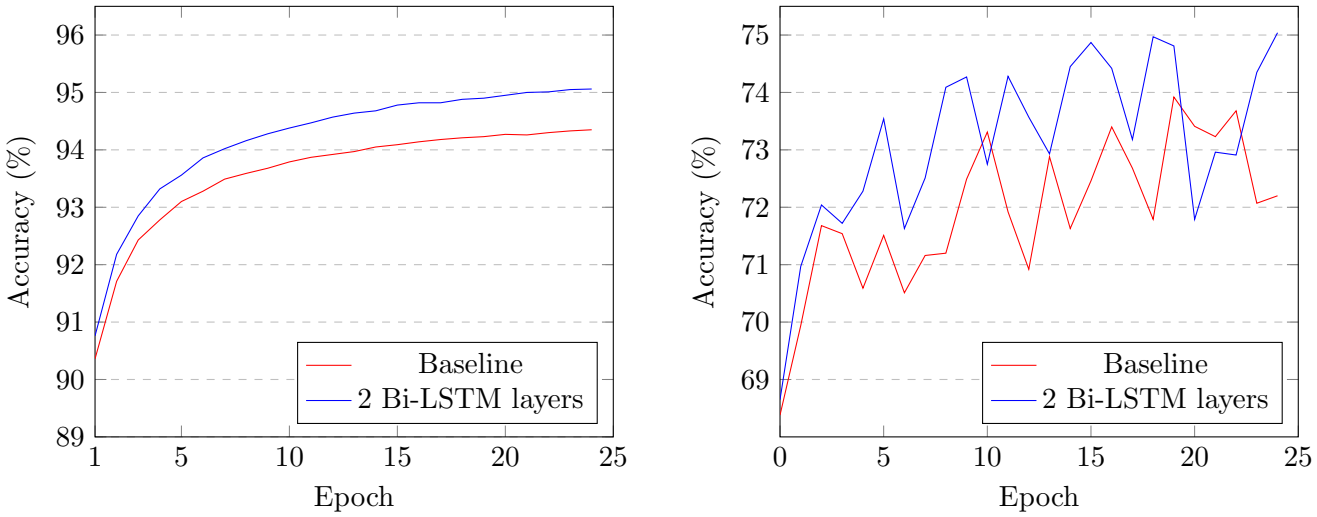
Table 1: Pretrained embeddings sizes.

Hyper-parameter	Set of values	Best tuning
Hidden units	64, 96	96
Learning rate	0.001, 0.005	0.005
Dropout	0.20, 0.35	0.20
Recurrent dropout	0.10, 0.20	0.10

Table 2: Grid search used for hyper-parameter tuning and best scoring settings.

Model	AS	CITYU	MSR	PKU	Gain
Base model	69.19	70.77	94.11	94.25	-
with 3-grams	71.60	71.85	94.33	94.49	+0.99
with 2 Bi-LSTM layers	73.08	**73.90**	**95.03**	94.70	**+2.10**
with 3-grams and 2 Bi-LSTM layers	**75.59**	72.51	90.90	**94.94**	+1.40
with 3 Bi-LSTM layers	71.72	72.80	94.72	94.90	+1.46

Table 3: Test set accuracy on each dataset after 25 epochs; 'Gain' is the average performance increase w.r.t. the baseline model.



(a) Training set **(b)** Development set

Figure 1: Performance comparison on the CITYU dataset.

Training dataset	AS	CITYU	MSR	PKU	Average
AS	73.08	70.70	84.70	88.85	79.33
CITYU	72.48	73.90	85.38	90.53	**80.57**
MSR	58.86	54.39	95.03	87.33	73.90
PKU	63.14	60.30	87.27	94.70	76.35

Table 4: Accuracies against all datasets' test sets for a 2 Bi-LSTM layers model trained on a specific dataset.

References

[1] Wikipedia. *Text segmentation*. URL: https://en.wikipedia.org/wiki/Text_segmentation.

[2] Ma, Ji & Ganchev, Kuzman & Weiss, David. (2018). *State-of-the-art Chinese Word Segmentation with Bi-LSTMs*. URL: https://arxiv.org/abs/1808.06511

[3] SIGHAN. (2005). *Second International Chinese Word Segmentation Bakeoff*. URL: http://sighan.cs.uchicago.edu/bakeoff2005/

[4] Yue, Bernard. *Hanzi Converter*. URL: https://github.com/berniey/hanziconv

[5] Mikolov, Tomas & Chen, Kai & Corrado, Greg & Dean, Jeffrey. (2013) *Efficient Estimation of Word Representations in Vector Space*. URL: https://arxiv.org/abs/1301.3781