

DL for NLP - Ass. 2 - 1

Itamar Trainin - 315425967

December 2019

1 Best model Parameters

NER

Hidden later: `#_unique_words_in_vocab_file` x 300

Output layer: 300 x `#_of_unique_label`

Learning rate: 0.01

Batch size: 64

Iterations: 100

Window size: 5

`freq_bound`: 2

`balance_coef`: 1.3

POS

Hidden later: `#_unique_words_in_vocab_file` x 50

Output layer = 50 x `#_of_unique_label`

Learning rate: 0.1

Batch size: 512

Iterations: 150

Window size: 5

`freq_bound`: 2

`balance_coef`: 1.3

Why use pre-trained embedding

The reason to use pre-trained embeddings is that the data available to you may be small and/or may not cover much words. When using pre-trained embedding you can get access to much larger embedded vocabulary for example all the words in 'Wikipedia'. Another reason is that the pre-trained embeddings may also have better representations of the words, especially not it could be not over-fitted to your data (for example if learned on 'Wikipedia'). Lastly pre-trained embedding can save computation time and network complexity.

Embedding implementation logic

I have solved the problem with the lower-cased vocabulary by lower-casing the whole text file. Although this solution emits information, a better solution to 'not-ignoring' the upper cased letters is by adding features indicating weather the word has capital letters or not and only embedding the cleaned (normalized) words.

2 Graphs

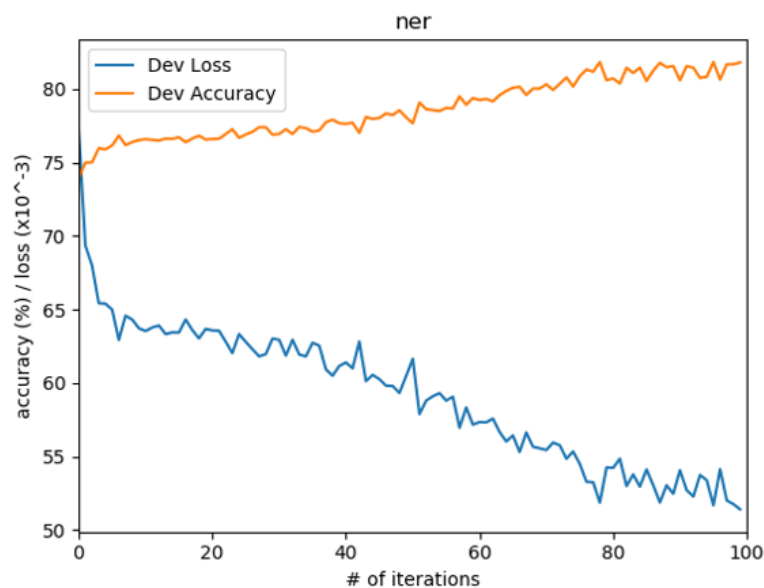


Figure 1: NER

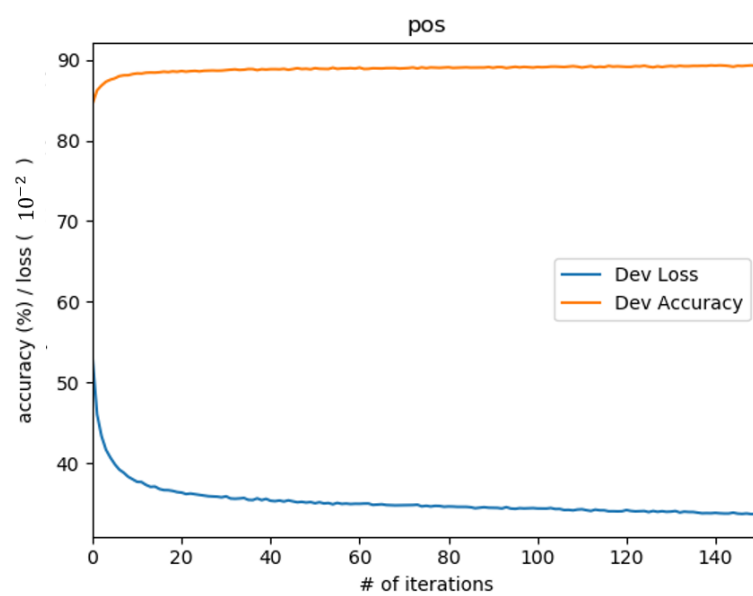


Figure 2: POS