DL for NLP - Ass. 2 - 4

Itamar Trainin - 315425967

December 2019

1 Best model Parameters

NER

Train words embedding layer: #_unique_words_in_train x 50
Prefix/Suffix embedding layer: #_unique_prefixes_suffixes_in_train x 50
Hidden later: 50 x 300
Output layer: 300 x #_of_unique_label

Learing rate: 0.005
Batch size: 64
Iterations: 50

Window size: 5
Prefix/Suffix size: 3

freq_bound: 2
balance_coef: 1.3

POS

Embedding layer: #_unique_words_in_train x 50
Prefix/Suffix embedding layer: #_unique_prefixes_suffixes_in_train x 50
Hidden later: 50 x 50
Output layer = 50 x #_of_unique_label

Learing rate: 0.1
Batch size: 512
Iterations: 150

Window size: 5
Prefix/Suffix size: 3

freq_bound: 2
balance_coef: 1.3

Considerations

When constructing the prefixes/suffixes, I had to deal with the case where the length of the word is less than the prefix/suffix_size flag. In that case I have assign a unique constant prefix/suffix: '-under-' to the word.

Conclusions

It can be observed from the data that the best case scenario is when using the pre-traiend embedding vectors AND adding the embeddings of the prefixes/suffixes to the embedding of the word. In addition we see that we have very good results for the POS task however the results of the NER are not as good. This is because first the POS is a easier problem than the NER, however in addition we have more data.

Next I will turn to examining the influence of pre-trained embeddings and prefix/suffix approaches. From the results, we can see that the influence of the two is different for the POS and the NER problems. As both are meant to help with dealing with unknown words, the pre-trained embeddings can give better representations to words than can be extracted from the train, it is more likely to have the words that are in the dev set and it should have better word representations since our data is not a special case of any problem (for example of text specifically about 'soccer'). Therefore this has great influence on the result. In the pre/suf-fix solution we will see that the embeddings of the words tend to be closer to similarly 'starting'/'ending' words helping representing the words. Since the essence of the word itself is missing in this method (the addition), it shouldn't have much influence on the results.
In the results we see that the influence on the POS data was similar for the two cases. However this task the results were all ready very good in the first place. As for the NER problem we clearly see that the pre-trained embeddings have better effect than the pre/suf-fix approach.

2 Graphs

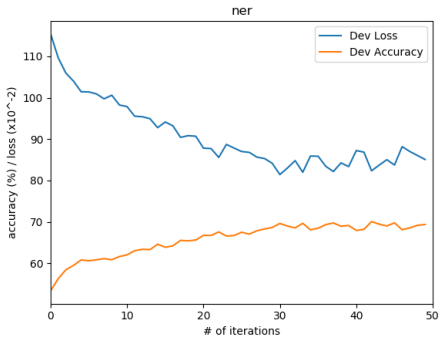

Figure 1: NER (part a)

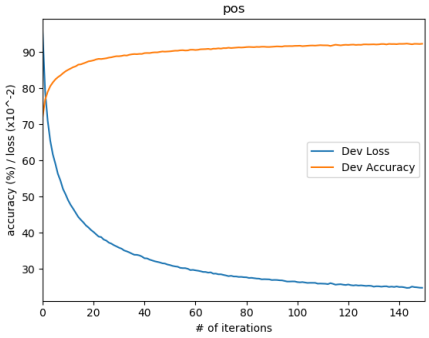

Figure 2: POS (part a)

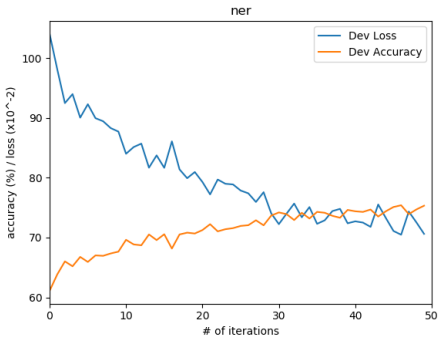

Figure 3: NER (part b)

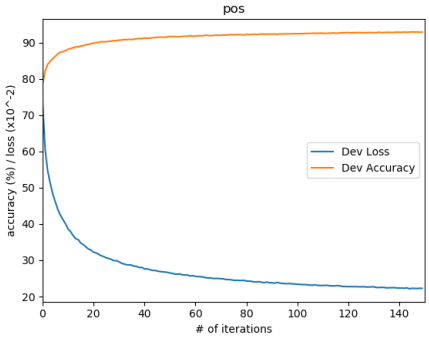

Figure 4: POS (part b)

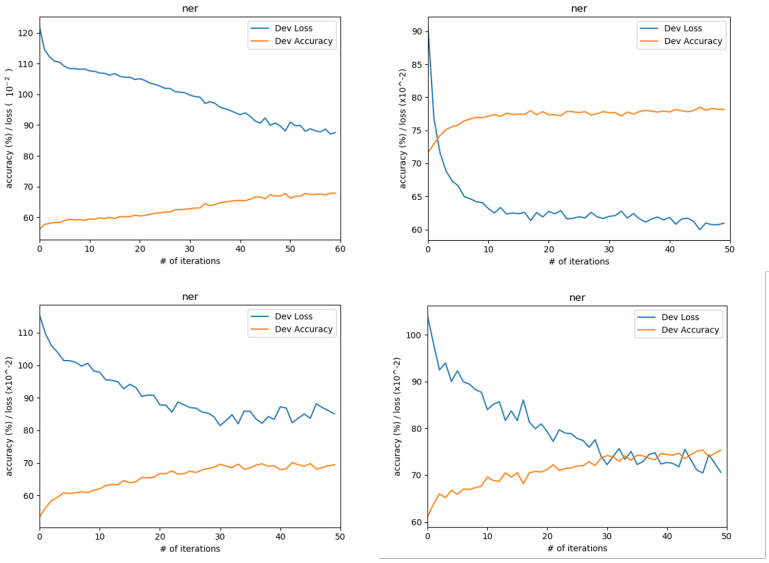

Figure 5: NER (1 - upper left, 3 - upper right, 4a - lower left, 4b - lower right)

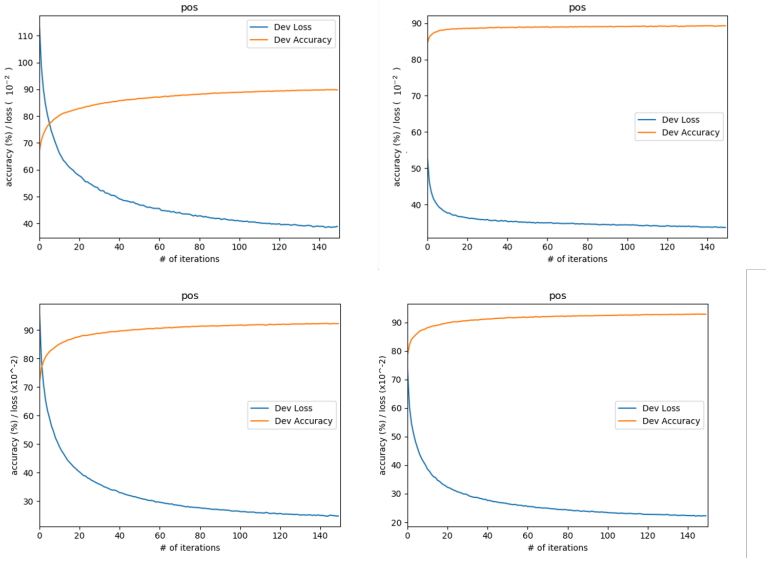

Figure 6: POS (1 - upper left, 3 - upper right, 4a - lower left, 4b - lower right)