Assignment 1

Gian Marco Baroncini, Matteo Periani, and **Giuseppe Mantineo**
Master's Degree in Artificial Intelligence, University of Bologna
{gianmarco.baroncini, matteo.periani2, giuseppe.mantineo}@studio.unibo.it

Abstract

The present work aims to study the Part-of-speech (POS) tagging task as Sequence Labbelling using Recurrent Neural Architectures. The approach adopted follows some fundamental steps: inspection of data, use of an embedding model, pre-processing of sentences and development of recurrent architectures. Concerning this last step, an increasing development in complexity was performed, starting from a basic architecture.

1 Introduction

Various methods can be used for the classification of natural language parts, e.g. state architectures, moving from recurrent networks (which were the state of the art in most natural language tasks until a few years ago) to the more modern transformers. In this specific work, only recurrent networks were used, concatenating Bi-LSTM/GRU with dense layers.

The dataset used is very unbalanced. Words like names, prepositions are very much present then foreign words or mathematical symbol. This lead to have tag classes with very very few example from which model can learn againts other that are predominant. You can find more information in the figure 2.

More in-depth analyses were carried out and it was seen that most of the sentences consist of less than fifty words. More specifically, more than ten thousand are lowered case words and only three tousand are in upper case. Words can also contains numbers, a mixture of letters and numbers, hypened words and symbol. All these features must be taken into account when developing models, especially for the embedding. In the notebook there are some plots that better explain all these properties.

The metric used to evaluate the classification is F1 score, a choice derived from the nature of the dataset. However the results, although they are not optimal, are pretty good by reaching around a 0,78 of F1 score.

2 System description

We starting with a baseline architecture coposed by a frozen Embedding layer and a Bidirectional LSTM, followed by a Dropout and a FC layer. We use a word tokenization and consequently a word embedding layer, GloVe (Pennington et al., 2014).

Since, any preprocessing has been done on train data, during the creation of the embedding matrix following strategies was applied. Initially, if we don't find words as it, we search it lowercase version. If still not found, we try to search if it's an hypened terms, in this case the resultin embedding is the average of each word embedding. Finally, if no one of the previous methos works, we assign to it a default vector computed as the average of all the GloVe vectors. The vector embedding dimension is 100.

Since we want to predict all the sentence tag at once and by default the FC layer can address one input at time we use the Time-Distributed version. This allows to predict all the sentence tag at the same time and in this way we can compute the loss with the labels tags string. The final TD-FC layer have a number of unit equal to the number of tags, 45 and it will use the softmax activation function

Other model are subsequently created trying to improve the result and these are:

- 2 x Bidirectional LSTM + TD-Dense
- Bidirectional GRU + TD-Dense
- Bidirectional LSTM + 2 x TD-Dense

3 Experimental setup and results

The baseline model was hyperparameter-tuned for 100 epochs in order to obtain best F1 using

EarlyStopping monitoring validation loss. Therefore, all the model was trained with baseline configuration restoring best weight if early stopped. The training hyperparameter are:

- batch size: *256*
- optimizer: *Adam*
- learning rate: *1e3*
- monitor patience: *3*

The metric used to evaluate train is *F1* setting the parameter *zero_division=0* to expoit the problem of some class have support equal to zero. The results of the various models are shown below in Figure 1.

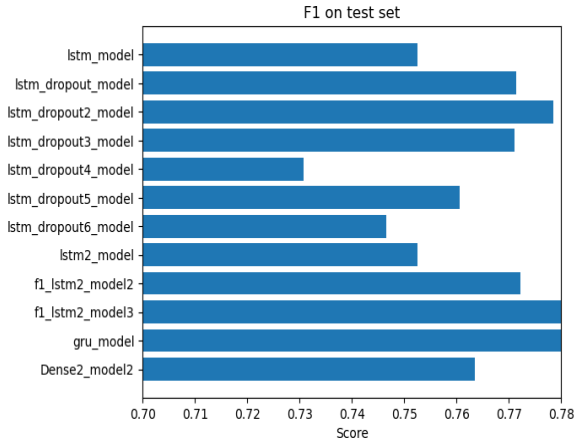

Figure 1: Evaluation Comparison

In order to make a comparison of the four indicated schemes, the best of each was selected and described in the table 1.

	U	E	D	F1 Val	F1 Test
LSTM+FC	150	68	0.3	0.775	0.779
2xLSTM+FC	200	52	0.3	0.79	0.785
GRU+FC	128	59	0.2	0.784	0.784
LSTM+2FC	128/96	45	0.2	0.759	0.763

Table 1: Where **U** refers to the number of units of each layer, **E** is the number of training epochs at wich we obtain best F1 score and **D** stands for the dropout rate

4 Discussion

We alanized the date through the Confusion Matrix and we highlighted that, there are classes with very low support, even zero. Nevertheless some of these, are still correctly classified, while other classes with substantial support are still difficult to predict well. The combination of this information can lead to a more in-depth study of the nature of the same: for example, tags that are scarce but well predicted could represent a class formed by particular words or with the presence of clearly distinguishable features such as capital letters or numbers.

Regarding the problem of classes with zero support, it was decided to use zero division equal to 0 in the calculation of F1. This led to the realisation that the final performance would be lower, so after testing various models, a test was made of the evaluation with zero division equal to 1 on the best of them. The final F1-score was much better. The results of the various architectures are approximately similar. There is no excessive gain in performance with the slightly more complex models. As some classes are difficult to interpret, the task is limited to sub-optimal results that do not deviate widely from the baseline. For future work to be more thorough, it is necessary to cover the poorer classes with more examples so that the networks can benefit.

5 Conclusion

The work went as expected. Despite limitations due to unbalanced data and overall very simple architectures, the results are satisfactory. It is certainly not a complicated task, but handling the language still requires a great deal of effort, especially in terms of representation in this case. Handle fuller models and pretrained embeddings requires substantial computation. Hence, having tackled the problem with models that are now obsolete give us confidence in future developments in the technology and research field of natural language processing. The most amazing thing about this task is the fact that a pretrained model made available to everyone can be so powerful. It already contains so much information and links between the components of the language that we were able to score well even on classes that were objectively difficult to predict for the few units in the dataset. This ability to capture the features of words by studying only their proximity and position/repetition will later be exploited in the group project where we will deal with the recognition of offensive sentences and pre-processing on semantics will be fundamental.

6 Additional material

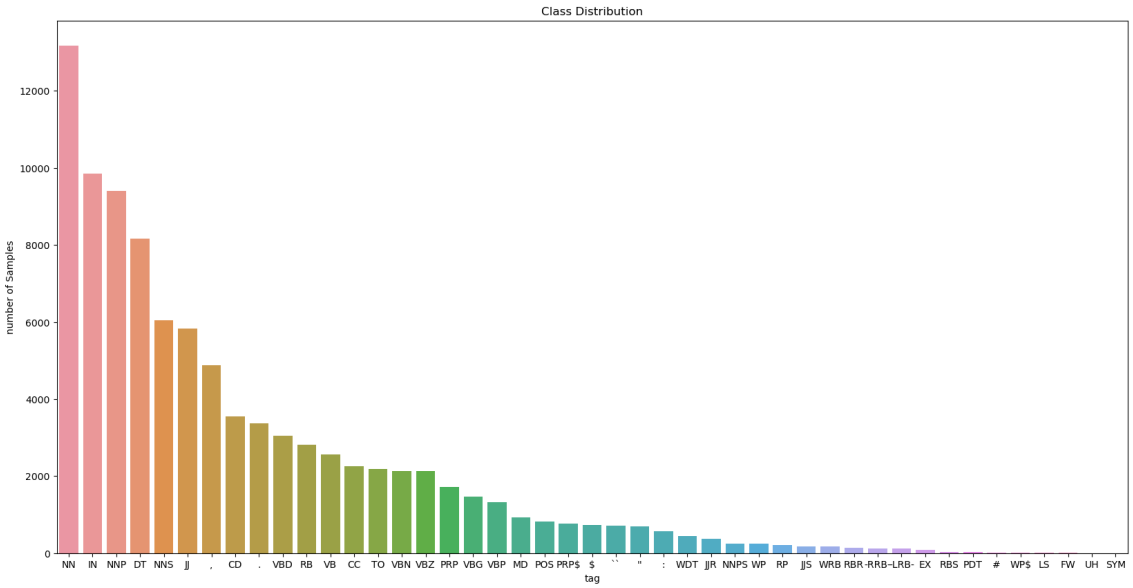

Figure 2: Tags distribution on train set

- GitHub project: `https://github.com/matteoperiani2/pos_tag_rnns.git`

References

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1532–1543. ACL.