Assignment 1: Part of Speech (POS) tagging as sequence labelling using Recurrent Neural Architectures

Nicola Carrassi, Gabriele Colasuonno and **Antonio Guerra**
Master's Degree in Artificial Intelligence, University of Bologna
{ nicola.carrassi, gabriele.colasuonno, antonio.guerra7 }@studio.unibo.it

Abstract

In this paper we will present methodologies and results obtained comparing the performances of different recurrent neural network architectures on the POS tagging of sequences. For each network architecture we performed hyperparameter tuning using HyperBand.
We compared the performances of each model on the validation accuracy before testing and comparing the two best architectures, namely Double LSTM and Double FC, obtaining Macro-F1 scores of 0.7003 and 0.6992 respectively.

1 Introduction

POS tagging is the process of marking up a word in text as corresponding to a given part of speech. Manually tagging words to part-of-speech is a laborious task, therefore the need of automating this process emerged. In order to automate the task several approaches have been deployed starting from the mid 70s, when a set of hand-crafted rules was used to determine categories that could co-occur. This simple approach was able to get about 70% of correct tags. Another popular approach, emerged in the 80s, is the use of hidden Markov models in order to disambiguate parts of speech (Kupiec, 1992). This approach is a stochastic technique for POS tagging which exploits the probabilities of particular structures in sentences (for example an adjective followed by a noun is more likely than an adjective followed by a verb).
Some more recent techniques which allow to deal with part-of-speech tagging are: transformers based and recurrent neural architectures.
Transformers based architectures are the current state-of-the-art (Jung et al., 2022) based on the architecture proposed by (Vaswani et al., 2017). Models of this type are realized fine-tuning some big pre-trained architectures to assign a tag to each word.

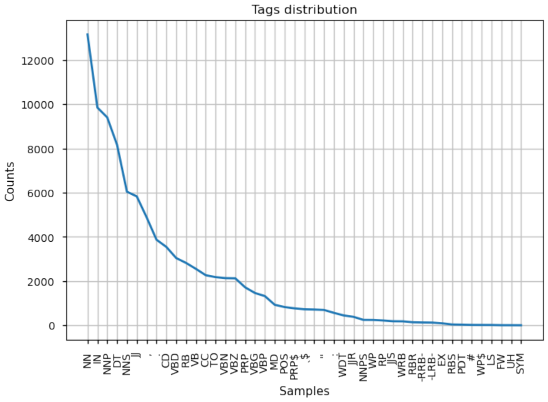
In this report we propose and compare the results obtained with 4 different recurrent neural architectures; we compared the models on the validation accuracy scores obtained and then we tested the two best models on a set of unseen sentences. We computed the Macro-F1 on the test set, considering only the classes which are not punctuation. Finally we tried to improve the best model using some regularization techniques to improve the performances of the network.

2 System description

In the literature there are plenty of recurrent models used to solve NLP tasks and we decided to test four different architectures in this work. We started defining a baseline model, which is a model composed of a Bidirectional Long Short-Term Memory cell (Hochreiter and Schmidhuber, 1997) followed by a Fully Connected layer.
Starting from this we created three more models, modifying or adding layers to it. The other models are:

- **GRU model**: A Bidirectional Gated Recurrent Unit layer (Chung et al., 2014) followed by a Fully Connected layer;
- **Double LSTM**: Two Bidirectional LSTM layers followed by a Fully Connected layer;
- **Double FC**: A Bidirectional LSTM followed by two Fully Connected layers

All models aforementioned contain a non trainable embedding layer placed before the first described layer of the architecture in order to convert each word to its corresponding embedding. Given that we cannot feed neural networks with text we decided to use dense embeddings, in particular the 50-dimensional GloVe embeddings (Pennington et al., 2014). Out-of-vocabulary words (OOV) have been handled creating an embedding based on the mean of the neighbors of such words. An embedding for a handcrafted padding token has also been added in order to make all the sentences of the same length, so the network can accept input of a fixed size.



3 Experimental setup and results

The only pre-processing made was converting the dataset text to lower case since it provided a lower number of OOV words. All the models described in section 2 were tuned using Keras Tuner, obtaining the optimal configuration through HyperBand method. We used the accuracy on the validation set obtained to compare the different models and choose which one to use for the evaluation on the test set. The scores obtained are summarized by the following table:

Model	Best Validation Accuracy
Baseline	0.88532
GRU	0.87693
Double LSTM	0.89546
Double FC	0.89241

Thanks to the tuning procedure we were able to choose the best combination of values for the units of the layers, the optimizer and the learning rate used for training the models. The best results were produced by Double LSTM and Double FC models, so we have evaluated these on the test set. The metrics used for the evaluation are accuracy and Macro-F1 score. The results obtained by the models on the test set are the following:

Model	Accuracy	Macro-F1 score
Double LSTM	0.9038	0.7004
Double FC	0.8990	0.6993

4 Discussion

Despite being a very unbalanced dataset, due to the composition of natural language sentences, we did not notice a direct correlation among the support of classes and the F1-score obtained, even if for some classes with a very low support both networks were not able to learn them. This is the case of predeterminers (PDT), which is never classified correctly and gets labeled as determiner (DT) or adjective (JJ) by the best architectures.

It is also important to highlight that both networks have problems classifying the class of proper plural nouns (NNPS); in fact the words belonging to this class are labeled mainly as plural nouns (NNS) and singular proper names (NNP). We believe that this happens because converting all the text to lower case might help the network with other POS but not with proper nouns.

The last situation we want to mention is related to two classes with low support which are comparative and superlative adverbs (RBR and RBS). These classes are mainly confused by the networks with comparative and superlative adjectives, this may be due to the similarity between adverbs and adjectives in English.

For both networks the results were analogous and both obtained very similar results in each class. After testing both models we tried to improve the results obtained with the best architecture (Double LSTM), clipping the norm of the gradient and adding dropout. With this adjustments we can prevent the issue of exploding gradients, as proposed in (Reimers and Gurevych, 2017). In this way we improved the performance, increasing the Macro-F1 score by 1% on the test set. Thanks to these changes, we managed to partially fix the problem that the network had on the test set with the predeterminers class, since now the network started labeling them correctly.

5 Conclusion

In this work we proposed a network which tackles the problem of automated POS tagging, developing a model which was capable to generalize quite well. One of the main limitation of our solution is that it has problem in recognizing some specific parts of speech, perhaps using a bigger dataset we might obtain more accurate results. We leave that as a future work, as well as introducing some form of weighting in order to balance the importance given to each class.

6 Links to external resources

> Weights and configuration: At this link there are all the best configuration found and the weights for the trained models. The content of the folder must be placed at the same path of the notebook, since the notebook will look for path at a position starting from it.

References

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Jeesu Jung, Sangkeun Jung, Hyein Seo, Hyuk Namgung, and Sungryeol Kim. 2022. Sequence alignment ensemble with a single neural network for sequence labeling. *IEEE Access*, 10:73562–73570.

Julian Kupiec. 1992. Robust part-of-speech tagging using a hidden markov model. *Computer Speech Language*, 6(3):225–242.

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.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *CoRR*, abs/1707.06799.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *CoRR*, abs/1706.03762.