Assignment 1: POS Tagging

Andrea Alfonsi, Gianluca Di Tuccio, Lorenzo Orsini
Master's Degree in Artificial Intelligence, University of Bologna
{andrea.alfonsi2, gianluca.dituccio, lorenzo.orsini}@studio.unibo.it

Abstract

In this assignment, we tested four different neural architectures (BiLSTM/ GRU + Dense Layers) for PoS tagging. One important aspect of this task was the embedding of OOVs words, which we handled by choosing the mean of the word context embeddings when it was possible. Finally, the two best models according to the macro F1 score are the BiLSTM followed by a Dense layer (BiLSTM model) and the two-layer BiLSTM followed by a Dense Layer (ML-BiLSTM model), which both perform with an F1 score equal to 0.79.

1 Introduction

Part-of-Speech (PoS) tagging is the task of labelling each word in a sentence with its appropriate part of speech; in the last years, the majority of the researchers mainly used Deep Learning tools for developing POS tagging models [1], which includes neural network architectures.
For this assignment, we used four different models: one is a two layers architecture composed by a BiLSTM followed by a Dense Layer (which acts as "classification head"), while the other three are simply variations of this, experimented by considering a GRU instead of the LSTM layer, an additional LSTM layer, and an additional Dense Layer.
Before training and testing them, each word of the dataset was embedded by using the GloVe model with dimension 100 (notebook chapter 6). Then, we handled the OOV words by starting first with the random embedding and then, as an alternative, we replace each OVV with the mean of the context embedding vectors, taking as context the three words before and after the OOV words; when it was not possible, the OOV word was replaced by a random vector. With the latter method, we got a higher F1 score than random embedding of 2.7% and so we used this as embedding.
Regarding the performance of each architecture, the macro F1 score was computed on the validation set to choose the two best models which we then evaluate on the test data. The two best models were the initial architecture with a BiLSTM plus a Dense layer (BiLSTM) and its variation with two BiLSTM (Multi-Layer BiLSTM, ML-BiLSTM), which both performed on the test set with a macro F1 score of 0.79.

2 System description

First of all, we have downloaded the dependecy_treebank dataset and then split the document into train, validation and test set according to the instruction; each sentence of a document has been preprocessed and stored in its respective dataframe (train, validation and test).
Then, we embedded each word of each sentence by using the GloVe model with dimension 100 and handled the OOV words by averaging the embedding vectors of the context when it was possible, otherwise we substitute them with a random vector. Starting from the GloVe vocabulary, the embedding matrix was built by adding the OOV words of the training set plus the ones of the validation and test set: for simplicity, the embedding layer of each model contains the embedding of the words that appear only in the training, validation, and test set, without loading the rest of the GloVe vocabulary.
Since each sentence will be the input of each architecture, we need to define a fixed length of the input. Here we chose the length of the 75 percentile of the distribution of the length over the training set, which is a compromise between the mean and the max length (notebook chapter 4). Then we applied padding and truncation to have sentences of the same length, which are now ready to be the inputs of our models.

3 Experimental setup and results

For this POS-tagging task, we considered the four architectures suggested in the instructions: a two layers architecture composed of a BiLSTM followed by a Dense Layer (BiLSTM model) and three variations of this one, experimented by considering a GRU instead of the LSTM layer (BiGRU model), an additional LSTM layer (ML-BiLSTM model), and an additional Dense Layer (BiDense model). The tuning of the hyperparameters was performed on the validation set, choosing the best ones according to the macro F1 score. The following table sums up the configuration of each model (chapter 10 of the notebook), reporting also the F1 score on the validation set:

Model	LSTM/GRU Units	Total Par.	F1
BiLSTM	64	90k	0.75
BiGRU	64	70k	0.73
ML-BiLSTM	64-32	129k	0.74
BiDense	30	36k	0.70

Table 1: F1 macro score for all the models, with their best configuration, evaluated with the validation split.

4 Discussion

In this assignment, we have compared four different architectures on the POS-tagging task. First, as expected, we noticed that handling the OOV words with the mean of the embeddings of the context words showed a better performance than taking a random vector, since with the first method the encoding of a word is based on its semantic meaning. As we can see from Table 1, the BiLSTM model, the BiGRU model and the ML-BiLSTM model perform with a similar macro F1 score on the validation set, while the BiDense has a lower macro F1 score. Regarding the two-single-layer architectures, we can see that with an equal number of units the BiLSTM model performs better than the GRU model, as we expected since BiLSTM are more powerful than BiGRU. Concerning the two layers architectures, we can see that the ML-BiLSTM model has a macro F1 score higher than BiDense of 4%, but it also has 4 times the number of parameters than the other. In addition, the results suggest that an architecture with two BiLSTM layers doesn't improve the performance of an architecture with just one BiLSTM layer, since the performance of the BiLSTM model is slightly better than the ML-BiLSTM. However, adding a Dense layer after a BiLSTM/BiGRU rather than a BiLSTM worsens the model's performance, as we can see from the F1 score of BiLSTM and BiGRU compared with BiDense model.
In the end, the two best models on the validation set were BiLSTM and ML-BiLSTM, whose F1 scores on the test set are for both equal to 0.79. These two performances are higher than the ones on the validation set of 4% and 5% respectively. This behaviour is caused by the fact that the validation set contains two classes, UH and FW, which are predicted with low results and not present in the test set; by computing the macro F1 score without considering these two classes also on the validation set the performance is equal to the one on the test set (notebook chapter 12-13).
It is noteworthy that both models struggle to predict the classes NNPS, RBR, RBS and PDT. In particular, the BiLSTM model is more accurate on RBR (by 9%), while the ML-BILSTM model on RBS (by 12%). The common factor between the four previous classes is that they have little support both on the training and test set, therefore for both models is more difficult to predict these labels than the more frequent ones, like DT, IN, NN and NNP, whose performances are great. It is also remarkable the different performances of the two models on the class WP$, which is predicted perfectly by the BiLSTM (F1 = 1.0) and not by the ML-BiLSTM (F1 = 0.67).

5 Conclusion

In this assignment we implemented four models that use variations of a BiLSTM followed by a Dense Layer (as 'classification head') to solve POS tagging. The results showed that the best models are a BiLSTM followed by a Dense Layer (BiLSTM model) and its variation that uses another BiLSTM before the Dense Layer (ML-BiLSTM). Our work also showed that using a BiGRU instead of a BiLSTM doesn't improve the performance, nor does adding a Dense Layer after the BiLSTM. It is noteworthy that both models perform equally on the test set, with almost the same performances on all the tags.
In the end, we have only tested models with traditional BiLSTM/GRU layers: nowadays are very popular architectures that make use of the attention mechanism to reach better results such as BiLSTM-LAN [2], which outperforms the previous models.

References

[1] Alebachew Chiche and Betselot Yitagesu. 2022. Part of speech tagging: a systematic review of deep learning and machine learning approaches. *Journal of Big Data*, 9(1):10.

[2] Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4115–4128, Hong Kong, China. Association for Computational Linguistics.