Assignment 1

Fabio Zanotti, Antonio Morelli, Federico Rullo, and **Edoardo Conca**
Master's Degree in Artificial Intelligence, University of Bologna
{ fabio.zanotti, antonio.morelli, federico.rullo, edoardo.conca }@studio.unibo.it

Abstract

The abstract is a very brief summary of your report. Try to keep it no longer than 15-20 lines at most. Write your objective, your approach, and your main observations (what are the findings that make this report worthwhile reading?)

1 Introduction

The Assignment consisted in implementing a Part Of Speech Tagging system using neural architectures, given a corpus of documents, the objective was to predict the Part Of Speech tag for each word. The system has been implemented using 3 different models, and comparing them in order to see which one would perform better on the given data. The Three different models are:

- Baseline - consisting of a single Bidirectional LSTM layer;
- Gated Recurrent Units (GRU);
- Baseline with an additional Bidirectional LSTM layer.
- Baseline with an additional Dense layer.

Bidirectional LSTM layers are able to process sequential data in both forward and backward directions, this allows the model to capture contextual information from both past and future. This is particularly useful for natural language processing tasks as the meaning of words can sometimes depend on the context in which they are used, one advantage of using it for POS tagging is that it allows one to predict the tag for each word in a sentence simultaneously, this is particularly useful when dealing with long sentences. On the other hand Gated Recurrent Units are a type of recurrent neural networks that are often used in natural language processing tasks such as **Part of Speech** tagging, they are similar to LSTM networks but have a simpler structure and fewer parameters, Gru layers can be used to process a sequence of words and predict its POS tag for each word in the sequence. Our approach was to implement these different models and compare them based on the average Macro-F1 score which is a common metric used to evaluate the performance of models and was also the requested metric.

2 System description

We initially imported the dataset from the Natural Language Toolkit(NLTK) library, which we preprocessed. We removed the "-NONE-" tag, this way the model will have fewer examples of unstructured data to learn, and can focus on the examples that are most relevant to the POS tagging increasing the accuracy of the results. After pre-processing we build the vocabulary, for this, we used GloVe (Global Vectors for Word Representation) which is a method for learning vector representations of words called "word embeddings" from a large corpus of text. Because some words may not be present in the GloVe vocabulary we expanded it by integrating Out Of Vocabulary words which are present in the dataset. Since there are different versions of GloVe, each of which includes an increasing number of dimensions we settled for the smaller one, 50-dimensional, which is easier to work with and computationally more efficient. By using GloVe embedding to set the initial weights of the model we could take advantage of the pre-trained word representations and fine-tune them to suit our task. Finally, we implemented the three models for the POS Tagging system. All these models have the same input and output layers, we used Glove Embedding as an input layer and TimeDistributed as an output layer. The models differ in the hidden layers, the first model, which is the Baseline model, is composed of only one Bidirectional LSTM hidden layer1. The Second model has a GRU hidden layer2. The third model is the Baseline model extended with an additional Bidirectional LSTM

layer3. And the fourth and final model is the Baseline with an additional Dense layer4.

Describe the system or systems you have implemented (architectures, pipelines, etc), and used to run your experiments. If you reuse parts of code written by others, be sure to make very clear your original contribution in terms of

- architecture: is the architecture your design or did you take it from somewhere else
- coding: which parts of code are original or heavily adapted? adapted from existing sources? taken from external sources with minimal adaptations?

3 Experimental setup and results

3.1 Parameter Tuning

Before running the experiments on the data we performed some hyperparameter tuning on each model. During the tuning phase, we observed that the most evident changes presented themselves in the scores due to the units being used in the LSTM layer and the batch size. Generally, many units in the LSTM layer mean that the model has more capacity for complex representations. however, too many units can lead to overfitting, while a number of units between 128 and 256 can help the model memorize only the important feature of the data, which results in a more general model. For the batch size, we used a value of 32 because a small batch improves performance and alleviates the problem of high variance in the estimated mean. Finally, with a small batch, the model is more likely to observe a more diverse range of samples in each batch, mitigating class imbalance. The same numbers of units and batch size are then used for all the models.

3.2 Metrics

For the Metrics we implemented a custom function which first computes the per-sample accuracy, a binary tensor indicating if the prediction for each sample is corrected or not and then multiplies it with the weights for the corresponding true class to obtain a weighted per-sample accuracy. After it creates a binary mask which indicates what samples to ignore in the computation of the accuracy, this mask is initialized as all ones and then updated to exclude samples with class labels specified in the arguments. Finally, the overall weighted accuracy is computed by summing the weighted per-sample accuracy and dividing it by the number of non-ignored samples.

3.3 Results

Table 1: F1 Scores on Training Set

	Macro F1 Scores
Baseline Model	0.7031
GRU Model	0.6389
Additional LSTM	0.7415
Additional Dense	0.7002

Table 2: F1 Scores on Test Set

	Macro F1 Scores
Baseline Model	0.8345
GRU Model	0.7446
Additional LSTM	0.7808
Additional Dense	0.7866

4 Discussion

MAX 1.5 COLUMNS FOR ASSIGNMENT REPORTS. ADDITIONAL EXAMPLES COULD BE PLACED IN AN APPENDIX AFTER THE REFERENCES IF THEY DO NOT FIT HERE.

Here you should make your analysis of the results you obtained in your experiments. Your discussion should be structured in two parts:

- discussion of quantitative results (based on the metrics you have identified earlier; compare with baselines);
- error analysis: show some examples of odd-/wrong/unwanted outputs; reason about why you are getting those results, elaborate on what could/should be changed in future developments of this work.

5 Conclusion

In one or two paragraphs, recap your work and main results. What did you observe? Did all go according to expectations? Was there anything surprising or worthwhile mentioning? After that, discuss the main limitations of the solution you have implemented, and indicate promising directions for future improvement.

6 Links to external resources

THIS SECTION IS OPTIONAL

Insert here:

- a link to your GitHub or any other public repo where one can find your code (only if you did not submit your code on Virtuale);
- a link to your dataset (only for non-standard projects or project works).

DO NOT INSERT CODE IN THIS REPORT

References

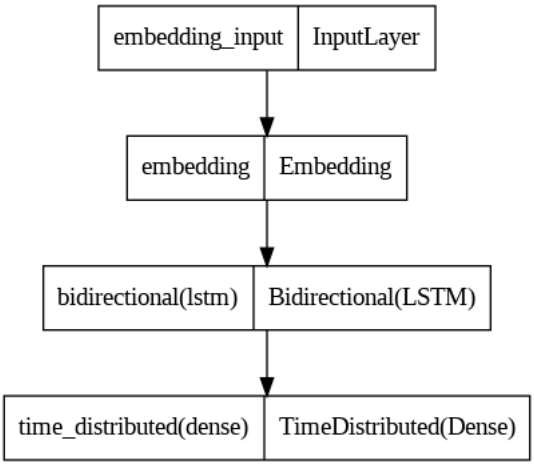

Figure 1: Baseline architecture

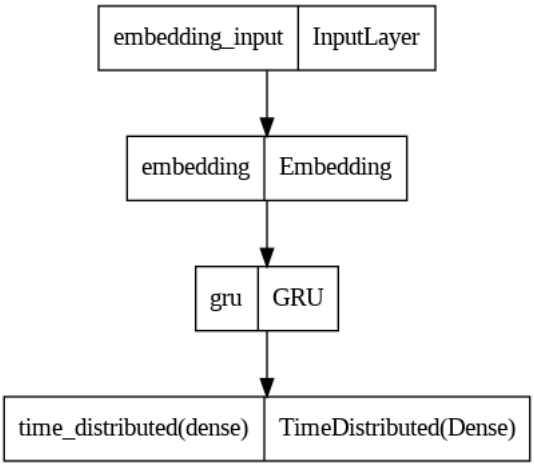

Figure 2: GRU architecture

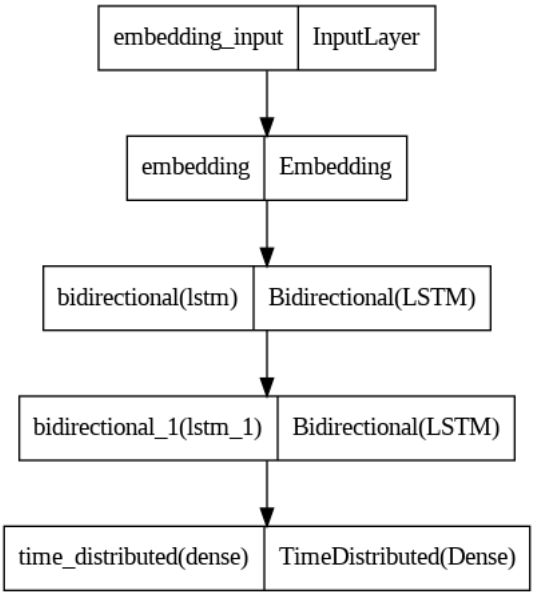

Figure 3: Additional LSTM architecture

Figure 4: Additional Dense architecture