

Assignment 2

Substitute the ↑ title ↑ with your project's title, or with Assignment 1 / 2

↓ Keep only one of the following three labels / leave empty for assignments: ↓

NLP Course Project

Daniel Bernardi, Daniele Santini, Hiari Pizzini Cavagna and Muhammad Saleem Ghulam

Master's Degree in Artificial Intelligence, University of Bologna

{ daniel.bernardi, daniele.santini2, hiari.pizzinicavagna, muhammad.ghulam }@studio.unibo.it

DO NOT MODIFY THIS TEMPLATE - EXCEPT, OF COURSE FOR TITLE, SUBTITLE AND AUTHORS. IN THE FINAL VERSION, IN THE L^AT_EX SOURCE REMOVE THE `guidelines` OPTION FROM `\usepackage[guidelines]{nlpreport}`.

Abstract

The abstract is 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?)

Comparative analysis of Neural-Network based implementations of text generation: combining Transformer based architectures for history-aware question answering.

NOTICE: THIS REPORT'S LENGTH MUST RESPECT THE FOLLOWING PAGE LIMITS:

- ASSIGNMENT: **2 PAGES**
- NLP PROJECT OR PROJECT WORK: **8 PAGES**
- COMBINED NLP PROJECT + PW: **12 PAGES**

PLUS LINKS, REFERENCES AND APPENDICES. THIS MEANS THAT YOU CANNOT FILL ALL SECTIONS TO MAXIMUM LENGTH. IT ALSO MEANS THAT, QUITE POSSIBLY, YOU WILL HAVE TO LEAVE OUT OF THE REPORT PART OF THE WORK YOU HAVE DONE OR OBSERVATIONS YOU HAVE. THIS IS NORMAL: THE REPORT SHOULD EMPHASIZE WHAT IS MOST SIGNIFICANT, NOTEWORTHY, AND REFER TO THE NOTEBOOK FOR ANYTHING ELSE. FOR ANY OTHER ASPECT OF YOUR WORK THAT YOU WOULD LIKE TO EMPHASIZE BUT CANNOT EXPLAIN HERE FOR LACK OF SPACE, FEEL FREE TO ADD COMMENTS IN THE NOTEBOOK. INTERESTING TEXT EXAMPLES THAT EXCEED THE MAXIMUM LENGTH OF THE REPORT CAN BE PLACED IN A DEDICATED APPENDIX AFTER THE REFERENCES.

1 Introduction

MAX 1 COLUMN FOR ASSIGNMENT REPORTS / 2 COLUMNS FOR PROJECT OR PW / 3 FOR COMBINED REPORTS.

The Introduction is an executive summary, which you can think of as an extended abstract.

Start by writing a brief description of the problem you are tackling and why it is important. (Skip it if this is an assignment report).

Then give a short overview of known/standard-/possible approaches to that problems, if any, and what are their advantages/limitations.

After that, discuss your approach, and motivate why you follow that approach. If you are drawing inspiration from an existing model, study, paper, textbook example, challenge, ..., be sure to add all the necessary references (?????).¹

Next, give a brief summary of your experimental setup: how many experiments did you run on which dataset. Last, make a list of the main results or take-home lessons from your work.

HERE AND EVERYWHERE ELSE: ALWAYS KEEP IN MIND THAT, CRUCIALLY, WHATEVER TEXT/CODE/-FIGURES/IDEAS/... YOU TAKE FROM ELSEWHERE MUST BE CLEARLY IDENTIFIED AND PROPERLY REFERENCED IN THE REPORT.

The goal of this assignment was to perform Part-Of-Speech tagging using neural architectures on the Dependency Treebanks corpus from NLTK.

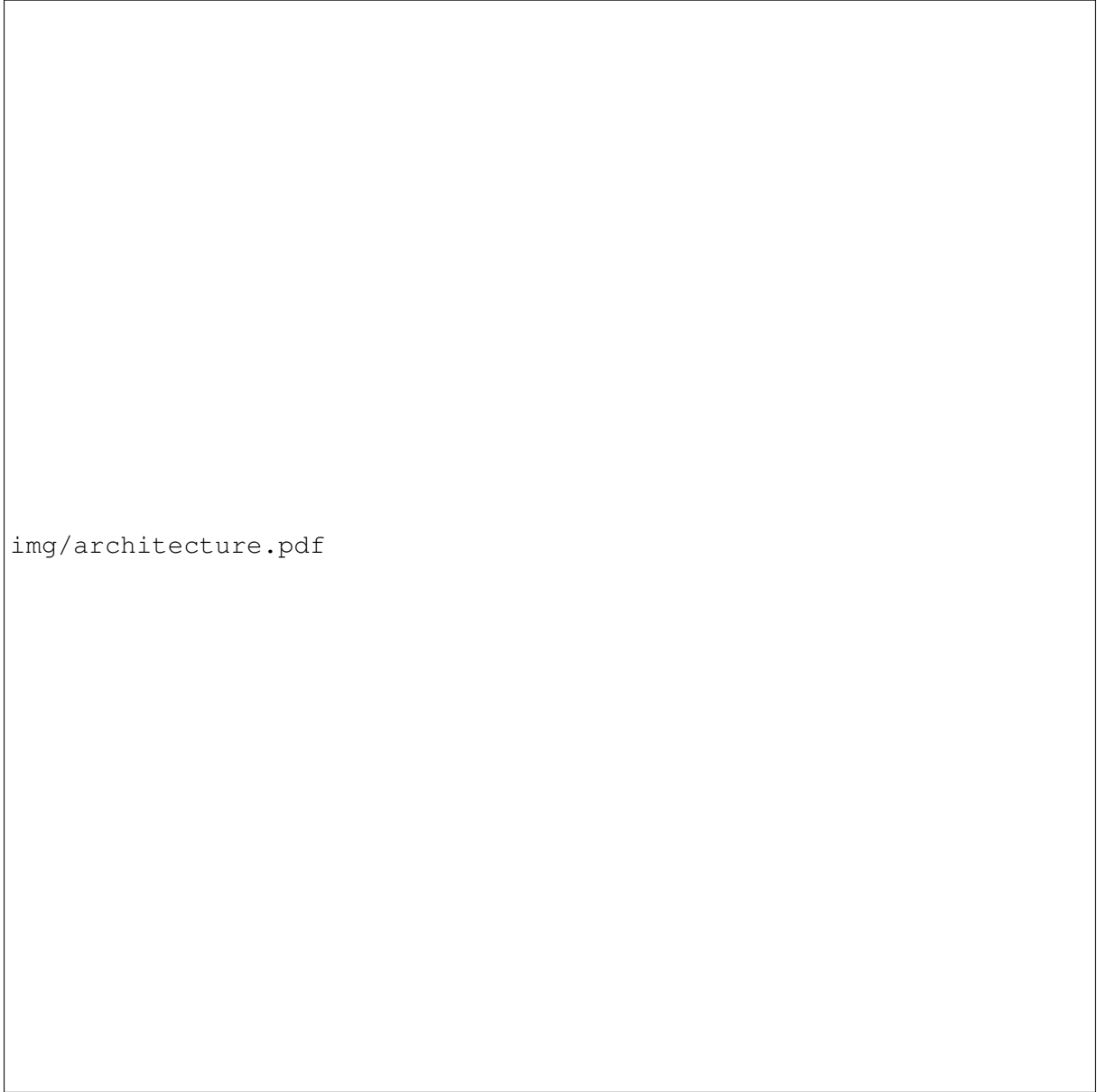
The necessary tasks included the download, pre-processing, and analysis of the corpus, the dense embedding of the words in the corpora and the comparison of the results of four Neural Network architectures chosen ahead of time:

- Bidirectional LSTM layer + Dense layer (baseline)
- Bidirectional GRU layer + Dense layer
- 2x Bidirectional LSTM layer + 1x Dense layer
- 1x Bidirectional LSTM layer + 2x Dense layer

The intended evaluation metric on the test set is F1-Macro, without considering punctuation classes.

¹Add only what is relevant.

Then we calibrated the configurations for the four different architectures using Keras Tuner^{??}. The hyper-parameters considered and their range (min, max) were: the number of units (16, 256), the activation functions (relu, tanh), dropout (0, 0.2) and learning rate (1e-4, 1e-2), with Adam as optimizer. The chosen tuner algorithm was Hyperband^(?), that allows to quickly converge on a



img/architecture.pdf

Figure 1: Model architecture

Architecture	Units	Dropout	Activation	LR	Epochs	Acc.	F1 Val.
Bi-LSTM + Dense	32	0	tanh	28e-4	9	0.885	0.707
Bi-GRU + Dense	112	0.05	tanh	64e-5	17	0.891	0.701
Two Bi-LSTM + Dense	128+112	0+0.2	tanh+relu	14e-4	7	0.890	0.708
Bi-LSTM + Two Dense	112+64	0	relu+tanh	12e-4	6	0.885	0.716

Table 1: Results obtained using Keras Tuner for every different architecture

high-performing model, comparing possible combinations through a "championship style" bracket. The algorithm used accuracy to determine the best model, training at most for 30 epochs, using the Early Stopping callback with a min. value of 15e-4 to increase the performances. The results obtained and the F1-score calculated are showed in the Table ???. The F1-score is macro-averaged, calculated excluding all the punctuation classes. In the end, the F1-score on the test set has been calculated, both best models had a better score compared with the validation score.

Model	F1 Val.	F1 Test
2xBi-LSTM + Dense	0.708	0.782
Bi-LSTM + 2xDense	0.716	0.793

6 Discussion

MAX 1.5 COLUMNS FOR ASSIGNMENT REPORTS / 3 COLUMNS FOR PROJECT / 4 FOR COMBINED 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.

All the models ended up with a F1-score result on the test set around 0.8, good but not perfect.

Since Gated Recurrent Units are less complex than Long Short Term Memory units we weren't surprised to see that the GRU model didn't outperform the baseline LSTM model.

During the initial manual phase we started with a high number of units in the first Dense layer

of the last model and noticed that it was particularly susceptible to over-fitting. This problem was addressed and solved by using dropout and early stopping. KerasTuner then demonstrated that a better result could be obtained simply reducing the number of units in the dense layer while keeping early stopping. The optimized hyper-parameters also changed the learning rate and the activation functions, which could have an impact.

As we expected the best performing architectures were the last two, however they didn't substantially outperform the baseline, even after tuning the hyper-parameters with Keras Tuner.

7 Conclusion

MAX 1 COLUMN.

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.

Since this problem relies heavily on identifying the relationship between the elements in the string, a possible way to improve the output would be to use attention-based architectures, such as transformer-derived techniques like BERT. Recurrent architectures like LSTM and GRU are partially able to learn these patterns but attention based architectures are precisely designed to exploit these relationships.

8 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).

- [Corpus](#)
- [English punctuation on Wikipedia](#)
- [PUNCT POS tags on Universal Dependencies](#)
- [gensim library docs](#)
- [gensim available models](#)
- [KerasTuner API - Keras](#)
- [Keras Tuner - Tensorflow](#)
- [Our GitHub repository](#)

DO NOT INSERT CODE IN THIS REPORT
