NLP Assignment 1

Andrea Zecca, Samuele Marro and **Stefano Colamonaco**
Master's Degree in Artificial Intelligence, University of Bologna
{ andrea.zecca3, samuele.marro, stefano.colamonaco}@studio.unibo.it

Abstract

In this assignment we implemented and trained a Part-of-Speech tagging model based for the Penn Treebank dataset. The models, which featured a combination of traditional preprocessing techniques and LSTMs, achieved a remarkable F1 score of 0.867, showing that such a fusion of classical and neural techniques is indeed effective. We also studied the effect of different preprocessing techniques and hyperparameters, finding that lowercasing leads to the best performance, although at the cost of a higher misclassification rate of proper nouns.

1 Introduction

In this assignment we trained a model capable of performing Part-of-Speech (PoS) tagging. There are several approaches that can be used to accomplish this task, such as rule-based taggers, which rely on a set of handcrafted rules to assign tags to words based on their surface form and context, or statistical taggers, which use models to predict the most likely tag for a given word based on its context and surrounding words. Additionally, neural network-based taggers have recently been found to be a promising approach, combining the strengths of the previous models.

We employed a standard NLP pipeline, with an initial part where the data is loaded and analyzed, followed by a preprocessing step, then training and finally an evaluation of three different models. We trained and tested each model with three seeds to identify the best performing architecture, whose test performance was then analyzed.

The models were trained on the Penn Treebank dataset, which consists in a set of sentences where each word has a corresponding PoS tag.

2 System description

For the purpose of this task we implemented a pipeline involving data preprocessing, model definition, training, evaluation and error analysis. In the preprocessing phase, we tested the following transformations:

- Named Entity Recognition (NER) to map personal/organisation names into special tokens;
- Lowercasing of each token;
- Mapping of numerical values into a unique token.

We empirically found that the best approach consists in only using lowercasing.

After preprocessing, we created an embedding matrix initialized on GloVe, with Out-Of-Vocabulary (OOV) tokens mapped to embeddings generated by applying various techniques (random, mean and fixed embeddings). We then defined three models:

- Baseline: one bidirectional LSTM layer combined with a dense one;
- Model1: same as the baseline model, with one additional bidirectional LSTM layer;
- Model2: same as the baseline model, with one additional dense layer.

We also tested several activation functions for each model, but we found the effect to be negligible. After defining the models, we trained them and evaluated their performances on the validation set using the macro F1-score. We repeated these steps with three different seeds to choose the best performing model based on the average score.

3 Experimental setup and results

In the numerous tests that we carried out, we always trained the three models using the same configurations, preprocessing pipelines and embedding matrices, with the goal of having an objective comparison of the results. We found that the preprocessing pipeline had the largest influence on the results, specifically:

- The usage of NER reduces the percentage of OOV tokens from 10.75% to 6.29%, although the overall model performance is worse;

Model	Seed	F1-score	Mean
Baseline	42	0.78811	0.788957
	69	**0.77749**	
	420	0.79566	
Model 1	42	0.78709	0.771363
	69	0.76096	
	420	0.79995	
Model 2	42	**0.79167**	0.798927
	69	0.77564	
	420	**0.80117**	

Table 1: Results on the validation set. The best F1-score is highlighted for each seed.

Model	Accuracy	F1-score
Model 1	0.9281	0.8677

Table 2: Results on test set for the best model.

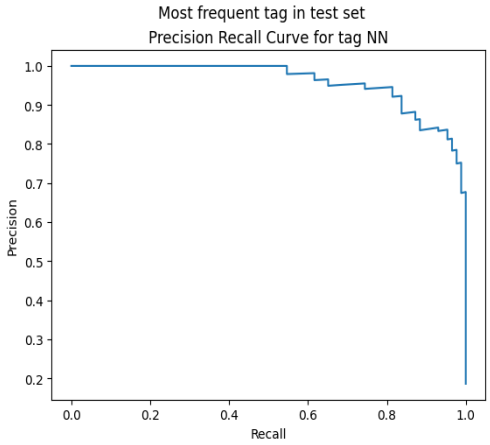

Figure 1: Precision-recall curve for NN.

- Using lowercasing reduces the percentage of OOV to 1.61% and leads to the best performance;
- The combined use of NER and lowercasing reduces the amount of OOV to 1.30% and results in intermediate performances compared to the two previous configurations;
- We found that among the various OOV embedding strategies (e.g. random, mean or fixed embedding), random embedding led to the best results;
- In general, using larger GloVe models with higher embedding dimensions had a positive effect on the results.

The choice of hyperparameters also had a minor effect, with the best set being `lstm_size` = 32, `batch_size` = 32, `initial_lr` = 5e-3, `lr_decay_factor` = 0.1 and `lr_decay_patience` = 2. Moreover, we found that a small number of epochs was crucial in preventing overfitting, with 10 epochs being the ideal value for all architectures. We report in Table 1 the metrics on the validation set for all models trained with this parameter set. Additionally, Table 2 contains the metrics of the best-performing model on the test set.

4 Discussion

The results of the experiment show that RNNs represent an effective architecture for PoS tagging, as shown by the fact that all three models achieved a satisfying F1-score on the validation set. The best-performing model appears to be *Model2*, i.e. the one with the additional Dense layer. Surprisingly, the score obtained on the test set is slightly higher compared to the one obtained on the validation set, with an accuracy of $\sim 93\%$ and an F1-score of $\sim 87\%$. We carried out an error analysis by plotting a confusion matrix, which shows for each tag a breakdown of its misclassifications. Thanks to this analysis, we found that our model performed poorly on some tags, in particular the ones representing *noun* (NN) and *proper noun* (NNP). These errors could be due to the lowercasing phase, since casing is an important factor when distinguishing NNs from NNPs. We also studied the behavior of *Model2* on the most and least frequent tags for both the validation and test set; we found that not only NN and NNP are the most commonly misclassified tags, but they are also the most frequent. Therefore, improving the accuracy on these two tags would have a significantly positive impact on the performance. Note that our tests found that using NER in the preprocessing phase, while beneficial for distinguishing NNs and NNPs, had a negative effect on the model performance on other tags, leading to an overall lower accuracy and F1-score. Finally, we plot in Figure 1 the precision-recall curve for the most frequent tag in the test set (i.e. NN).

5 Conclusion

In this assignment, we trained a model capable of performing Part-of-Speech (PoS) tagging using neural network-based approaches. Our observations reveal that while the model performed poorly on some tags (particularly NN and NNP, which was probably the result of the lowercasing step), in general LSTM-based architectures are highly accurate PoS taggers, with F1 scores of around 0.8. Future directions include exploring different architectures (e.g. transformers) and using larger datasets in order to achieve even better performances.