Assignment 1

Giorgio Adragna, Chiara Bellatreccia, Francesco Cavaleri and **Marcello Mancino**
Master's Degree in Artificial Intelligence, University of Bologna
{ giorgio.adragna, chiara.bellatreccia, francesco.cavaleri2, marcello.mancino }@studio.unibo.it

Abstract

Part-of-Speech tagging is an essential preprocessing step in Natural Language Processing, where words in a text are labeled with tags that indicate their grammatical role. We investigate the results that can be achieved within a constrained setting, utilizing a small dataset and three very simple models. Despite the inherent challenges posed, under such circumstances, by the unbalanceness of data, our approach consistently achieves a macro F1 score greater than 0.8.

1 Introduction

Modern day reaserch has managed to obtain astonishing results for **POS tagging**, basically solving it: state-of-the-art models manage to achieve an impressive result of 97% accuracy (Akbik et al., 2018). However, these solutions are highly dependent on substantial computational power and really large datasets, and this is often impractical. We investigate how far we can get within a constrained setting: a small dataset and a group of very simple models. Under these circumstances, we tried to squeeze the best out of our models in terms of runtime performance and robustness. Other than the obvious generalization issues coming from such a small domain, the main challenge of POS tagging is the fact that it is intrinsically **unbalanced**, and it is only natural that the vast majority of the (much limited) capacity of the model is employed for the more frequent tags, resulting in a poor performance for the minority classes. Given these premises, results were satisfactory, managing to consistently obtain a macro F1 score greater than 0.8. We also discovered that, probably because our baseline reaches perfect accuracy on training data, augmenting its capacity with one additional layer does not necessarily yield better results.

2 System description

Our three models have the following structure:

- The **Baseline** is composed of a Bidirectional LSTM with a Dense layer on top;
- The "**Model 1**" adds an additional LSTM layer to the Baseline model, before the final Dense layer;
- The "**Model 2**" adds an additional Dense layer to the Baseline model, before the final Dense layer.

All dense layers are enclosed with a TimeDistributed wrapper.

Each model has, of course, an Embedding layer at the beginning.

3 Experimental setup and results

For the vocabulary, we chose to use the union of **GloVe** (Pennington et al., 2014) and all the **training set words**. The only preprocessing technique we opted to use is **lowercasing**.

All three models were trained with 3 different seeds for robust estimation, and **macro F1 score** was chosen as metric for evaluation.

The following details were shared by all 3 models:

- The **biases** of the last layer were initialized with the log-frequencies of their corresponding tags. This should ensure a good starting point for the gradients (Karpathy, 2019);
- **Dropout**, **Recurrent Dropout**, **Label Smoothing**, **Early Stopping** and **Weight Decay** were all used to perform regularization;
- **Adam** was used as optimizer: the best learning rate was found to be 0.0003.

#units	Embedding	Baseline	Model 1	Model 2
64/64	50	0.877	0.878	0.878
256/512	50	0.864	0.852	0.872
256/512	100	**0.885**	**0.887**	**0.885**

Table 1: Average validation accuracy of the three models with different hyperparameter values.

Split	Baseline	Model 1	Model 2
Validation	0.780	0.769	**0.782**
Test	**0.836**	0.825	0.821

Table 2: Average macro F1 scores of the three models for the Validation and Test Set.

In the training process, two main hyperparameters were optimized: the **number of units** per layer and the **embedding dimension**. These hyperparameters were selected based on the validation accuracy they produced, shown in Table 1.

4 Discussion

The F1 scores (Table 2) are good: they are all around 0.8 for both Validation and Test Sets, which is a strong result considering the circumstances. Nonetheless, we can observe that the two models do not succeed in beating the baseline, with marginally better scores on the Validation Set and even slightly worse scores on Test Set. It is not easy to find a reason for this, but we can infer something by looking at the training results: our baseline model managed to reach a training accuracy of 0.996, basically solving the training set, while its validation accuracy was much smaller (0.877); this model suffers from high variance, but certainly it does not have a bias problem, based on the results we get on the training set. If we wanted to get better scores, we should have not been focusing on the capacity of the model, which was plenty enough, but even more on the distance between the training and validation scores. This can be reduced by some regularization, but, as shown here, regularization only takes us as far as the data distribution allows us to.

Regardless of the architecture, every model seems to struggle to get over 0.83. If we divide our tags in bins depending on their frequency, we notice that tags with a frequency less than 100 have a macro F1 of 0.631, while all other bins go from 0.810 to 0.914. This seems to prove that the least frequent tags really have a negative impact on the overall performance.

This is further confirmed by the analysys of the *Relative Error Frequency*, which is computed by dividing, for each tag, the number of errors by their support: highest percentages often regarded rare tags.

There were some exceptions, though, with tags are mistaken in spite of their relatively high frequency. These were the tags whose morphology or syntactic role was often ambiguous. Therefore, we may think of tackling these two problems with a **custom loss** that initializes the class weights with their inverse frequency, but adds a constant penalty term a to the weights of the frequent yet highly mistaken tags.

5 Conclusion

Engaging in the task of POS tagging in such a limited environment was an interesting challenge. We had to overcome the problems of overfitting and unbalancedness with limited tools.

The main issue was that the least frequent tags were, as expected, the ones that had the worst performance, and this was not solved by a naïve class weighting. In addition, also some tags with bigger support were highly misclassified. These were the tags whose morphology or syntactic role was often ambiguous.

The biggest surprise that we encountered has to be the fact that the models did not manage to *clearly* beat the baseline, proving that augmenting the capacity of the model when it already gains perfect accuracy over the training set does not pay in our setting.

On a positive note, heavy regularization allowed to mitigate the inevitable overfitting that comes with having to deal with such a small dataset, leading to really satisfactory F1 scores.

In conclusion, on the one hand, it is true that we are still far from the nearly perfect results of the state-of-the-art models, and that the infamous *Bitter Lesson* (Sutton, 2019) still stands: in modern, data driven, (deep) machine learning, the amount of data and resources at the disposal of the researcher are, by far, the key factors that drive the performance. On the other hand though, the results that we show here are a testament of the remarkable and surprising robustness that we can achieve with simpler models if we are willing to commit to some heavy engineering.

6 Links to external resources

The details of our work can be found in our GitHub repository.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics.

Andrej Karpathy. 2019. A recipe for training neural networks. http://karpathy.github.io/2019/04/25/recipe/.

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.

Richard S. Sutton. 2019. The bitter lesson. http://www.incompleteideas.net/IncIdeas/BitterLesson.html.