Assignment 1

Yuri Noviello, Enrico Pallotta, Flavio Pinzarrone and **Giuseppe Tanzi**
Master's Degree in Artificial Intelligence, University of Bologna
{ yuri.noviello, enrico.pallotta, flavio.pinzarrone, giuseppe.tanzi }@studio.unibo.it

Abstract

In Natural Language Processing, it is important the use of part-of-speech tagging. By assigning each word a specific tag, algorithms can improve the representation of similar words in different situations. The objective of this study is to tackle part-of-speech tagging with bidirectional recurrent neural models with a small number of parameters. Different networks with different combinations of layers were compared, using recurrent networks such as BiLSTM and BiGRU. The embedding was pretrained using GloVe-50 and the Out-Of-Vocabulary words were initialized by taking the average of a 3-words context. The two best models, which contained BiLSTM layer, both achieved a Macro-F1 scores of 0.77.

1 Introduction

Many studies have been carried out on different techniques to tackle POS tagging, starting from simpler RNNs like in [3] up to more complex transformer-based approaches, like in [5]. In this study, four different small recurrent models were compared according to their performance on the Dependency Parsed Treebank dataset by University of Pennsylvania. The documents of the dataset were firstly split into sentences in this way: 1958 sentences for the training set, 1242 for the validation set and 628 for the test set. With an inspection of the classes of the samples in each split, it was clear that the dataset was unbalanced, due to the intrinsic infrequency of some parts of speech in the English language. After handling the out-of-vocabulary words with a context-based embedding, we performed hyper-parameter tuning on the validation set for each model and then we compared the performance of the two best models on the test set. The study shows how deeper models with BiLSTM [2] layers reached higher Macro-F1 scores (0.77) with respect to shallower models, both with BiLSTM or BiGRU [1] layers.

2 System description

In this study the usual pipeline of an NLP task was followed. During the preprocessing step, it was decided only to transform each word to lower case, without removing any stopwords, since they make the classification of the tags more effective. Regarding the embedding, it was used GloVe [6] embedding model with 50 as embedding length. In terms of handling out-of-vocabulary terms, all words that were found in train, test and validation, but weren't included in GloVe, were considered to be OOV. At this point, the OOV words embeddings were created as the average of the word embeddings in a context window of size 3 of a random sentence in which the OOV word was found. It was used a baseline model structured as following. The first layer of the model is a pre-trained embedding layer, followed by one Bidirectional LSTM layer that focuses on the sequential relationship between the terms of each sentence. On top of that there is a Time-Distributed dense layer that aims to classify each token in the sentence. Three different variants of this architecture have been tested. For simplicity from now on we'll call the first variant GRU, the second variant 2-LSTM and the last variant 2-Dense. In the first variant the BiLSTM layer was replaced by a BiGRU layer. In the second variant, an additional BiLSTM layer was added, resulting in two BiLSTM layers. And in the last variant, an additional Time-Distributed dense layer before the last dense layer was added, resulting in two final Time-Distributed dense layers. During the training of the models a step of hyperparameters tuning was performed for each model.

3 Experimental setup and results

The experiments of this work were performed on the four aforementioned models: baseline, GRU, 2-LSTM e 2-Dense. For each of these models the embedding layer was set to be non-trainable and we selected a set of hyper-parameters to be tuned. For the first two models the tuning involved the number of units of the LSTM/GRU layer and the learning rate, whereas for the deeper models it involved also the number of units of the additional LSTM/Dense layer. Hyperband [4] was used to tune the hyper-parameters over a small set possible values: $[64, 128, 256]$ for the number of units and $[0.01, 0.001, 0.0001]$ for the learning rates. All the models were compiled with a Categorical Crossentropy loss with label smoothing $[\alpha = 0.1]$ and Adam optimizer. The choice of the hyper-parameters was performed according to the accuracy score on the validation set. The results of the hyper-parameter tuning are presented in Table 1.

	N. units LSTM	**N. units add. layer**	**Learning rate**
Baseline	256	/	0.01
GRU	256	/	0.01
2-LSTM	128	128	0.01
2-Dense	128	256	0.01

Table 1: Hyper-parameters

Proceeding in the pipeline, the models were retrained with the selected hyper-parameters for 70 epochs with an early stopping callback on the validation accuracy, in order to prevent overfitting. After training, precision, recall and Macro F1-Score of each model were computed on the validation set as reported in Table 2. The punctuation and symbols classes were not included in the evaluation.

	Macro Precision	Macro Recall	Macro F1
Baseline	0.77	0.77	0.76
GRU	0.79	0.72	0.73
2-LSTM	0.81	0.77	0.77
2-Dense	0.79	0.78	0.78

Table 2: Performances on the validation set

As you can see from Table 2, the two best models were selected to be 2-LSTM and 2-Dense, as they showed higher F1-Scores, respectively 0.77 and 0.78. For this reason their performances were then evaluated on the test set and proved to be in line with the ones on the validation set. The final Macro F1-Scores for the 2-LSTM and 2-Dense models are both 0.77.

4 Discussion

Looking again at the results in Table 2, you can see how replacing the BiLSTM layer with a BiGRU layer has led to a small increase in macro precision but also to a significant drop in macro recall, resulting in an overall undesirable decrease in macro F1-Score. On the other hand, increasing the complexity of the models by adding either a BiLSTM or a Time Distributed Dense layer actually increased the capability of the model of classifying tags correctly, while still generalizing well on unseen data. Indeed, in both cases the additional layer led to a small increase in precision. Concerning the recall, only the addition of the Dense layer led to an improvement, but still both models managed to reach higher macro F1-Scores. Looking at the misclassification errors of the evaluated model it was noticeable that there were classes which were harder to classify. For example both the models had a very low F1-Score for the NNPS class, i.e. plural proper nouns. Almost every example belonging to this class was misclassified either as singular proper noun (NNP) or plural common noun (NNS). Moreover, the presence in the dataset of some very infrequent classes led to other misclassification issues such as pre-determiners (PDT) classified as determiners (DT) and comparative adverbs (RBR) classified as comparative adjectives (JJR). However, the difference between this parts of speech is sometimes difficult to notice even for human annotators. For example in the case of comparative adverbs and adjectives some words can be used as both depending only on the context, e.g. words like *longer, faster, etc.*

5 Conclusion

The task of tagging parts of speech plays an important role to make algorithms learn the semantics of sentences and improve their performances. To meet this need, we have proposed a POS tagging solution with bidirectional recurrent neural models with a small number of parameters. This study shows that the best models are the deepest models, which are 2-LSTM and 2-Dense, both of which achieving a Macro F1-Scores of 0.77. However, there is still a presence of ambiguity when tagging terms with different contextual meanings within a sentence. Indeed, the results show that, for example, comparative adverbs (RBR) are often misclassified as comparative adjectives (JJR). But, the main problem to face to improve the performances of both models is the unbalancing of the dataset, which could be tackled with some techniques such as class weighting or data augmentation.

References

[1] Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555.

[2] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

[3] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging.

[4] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2016. Hyperband: A novel bandit-based approach to hyperparameter optimization.

[5] Artem A. Maksutov, Vladimir I. Zamyatovskiy, Viacheslav O. Morozov, and Sviatoslav O. Dmitriev. 2021. The transformer neural network architecture for part-of-speech tagging. In *2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus)*, pages 536–540.

[6] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.