Assignment 1

Michael Magdy Nasr Zaki Ghaly, Simona Scala and **Sara Vorabbi**
Master's Degree in Artificial Intelligence, University of Bologna
{ michael.magdynasr, simona.scala6, sara.vorabbi }@studio.unibo.it

Abstract

The first assignment implements a pipeline for preprocessing and training a part-of-speech (PoS) tagger using Recurrent Neural Networks (RNNs) trained on the Penn Treebank dataset. The focus is on the two best models selected among four different RNNs, based on their F1 scores computed on the validation set. Their performance is evaluated on the test set. The results of the error analysis on the models' predictions show that RNNs can effectively be used for PoS tagging and provided further insight on their limitations and possible improvements.

1 Introduction

Part of speech (PoS) tagging is a task in Natural Language Processing (NLP) that involves labeling words in a sentence with their corresponding grammatical category, such as noun, verb, adjective, etc. PoS tagging is important for a variety of applications, including information retrieval, machine translation, and text generation.

There are several approaches to PoS tagging, including rule-based methods, statistical methods, and machine learning-based methods. Rule-based methods rely on manually-defined rules to assign PoS tags, while statistical methods use frequency counts of PoS tags to predict the most likely tag for a given word. Machine learning-based methods, such as the ones implemented in this work, use algorithms to learn patterns in the data and make predictions.

In particular, this work implements four RNN (Recurrent Neural Network) models for PoS tagging and evaluates their performance to select the best two models for this task. RNNs are a type of neural network that are well-suited to processing sequential data, such as text, and have been used successfully for tasks like language translation and text classification. The reason for using RNNs for PoS tagging is due to their ability to capture contextual information in the sequence of words, which can be important for accurately predicting the PoS tag of a given word.

The experiments are run on the Dependency Treebank Dataset (Marcus et al., 1993), which consists of sentences with PoS tags extracted from the Wall Street Journal, that are split into training, validation, and test sets. The four RNN models are trained on the training set and their performance is evaluated using the F1 score on the validation set. The two models with the highest F1 score are selected for further evaluation on the test set. As a result, RNNs prove to be a good approach for PoS tagging, despite the misclassification of some underrepresented tags.

2 System description

The pipeline implemented in this work consists of the following steps:

1. Download and unzip the Dependency Treebank Dataset from a URL.

2. Preprocess the data: lowercase the tokens, create a list of dictionaries representing each sentence in the dataset with their corresponding document number, sentence number, split (training, validation, or test), tokens and tags then store the data in a Pandas dataframe.

3. Vectorize and pad the sequences of tokens and tags using TextVectorization layers from the TensorFlow library. The maximum number of tokens is set by observing the boxplot of the length of the sequences.

4. Define four RNN models using the Keras functional API:

    - Bidirectional LSTM layer + Dense/Fully-Connected layer

- Bidirectional GRU layer + Dense/Fully-Connected layer
- Two Bi LSTM layers + Dense/Fully-Connected layer
- Bi LSTM layer + Two Dense/Fully-Connected layer

All the models include an embedding layer to map the integer input sequences to dense vectors. The embedding matrix is created by a pre-trained GloVe word embeddings model.

5. Train the models using the training set and evaluate their performance on the validation set using the F1 score.

6. Select the two best models based on their F1 scores on the validation set and evaluate their performance on the test set. Before the test evaluation, the models are retrained with both the training and validation set.

3 Experimental setup and results

All four models are instantiated with the same metric ($accuracy$), loss ($categorical\_crossentropy$) and optimizer ($Adam(0.003)$). The models differ in the number of parameters $\#Param$ but it is kept under $1M$ for all. All models are trained on the training set for ten epochs each. Their performances were evaluated on the validation set using the F1-score. The results are shown in Table 1. Following these results, BiGRU and BiLSTM with

Model	F1-score
BiLSTM	0.66
BiGRU	0.7
Stacked BiLSTM	0.69
BiLSTM w/ Stacked FC	0.7

Table 1: F1 scores of the four RNN models on the validation set.

Stacked FC Layers are selected as models with best performance and are fitted on the training and validation set merged together. The evaluations on the test set give the results shown in Table 2.

4 Discussion

The results of the experiment show that it is possible to implement effectively RNNs for PoS tagging. BiGRU and BiLSTM w/ Stacked FC achieve the best scores on the validation with respect to the other models. Overall, the scores on the test set seem to be a bit better compared to the scores obtained with the validation set. Nevertheless, both models have higher validation accuracy ($\sim 96\%$) but their F1 scores are not as high ($\sim 70\%$).

Model	F1-score
BiGRU	0.72
BiLSTM w/ Stacked FC	0.72

Table 2: F1-scores of the two RNN models on the test set.

Indeed, an error analysis was performed by plotting the confusion matrix of the validation set. The plots show that some tags are misclassified due to the imbalance of the dataset with respect to certain tags as these tags have very few instances in the training data. Still, the model's predictions for these labels were reasonable considering the low amount of training it received for these particular tags.

5 Conclusion

After selecting and evaluating the best two models, we tried to infer the PoS of some sentences of our choice and we observed that the output was not always satisfactory. There are some limitations to the solution implemented in this work. The dataset used is of a very specific domain and is relatively small, and the models may benefit from using a larger dataset on a broader set of domains to be able to make precise predictions for general sentences. In addition, the imbalance of the training set for some tags prevents the models from making correct predictions for those tags. Thus, future works could focus on these aspects to further improve the performance of the RNN models for PoS tagging.

References

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Comput. Linguist.*, 19(2):313–330.