Assignment 1

Francesca Boccardi, Luigi Podda, Matteo Nestola
Master's Degree in Artificial Intelligence, University of Bologna
{francesca.boccardi, luigi.podda, matteo.nestola}@studio.unibo.it

Abstract

The purpose of this study is to compare different models based on recurrent neural architectures on a Part-of-Speech (POS) tagging task. The four neural networks employed are based on BiLSTM and GRU models, with final Dense layers. The experiments runned on the Dependency Treebank corpus show that, according to macro-f1 measure, the performances of the four models are positive and comparable. From results obtained by further analyzing the two best models out of the four, it is possible to state that the presence of LSTM layers has a strong impact in this type of task. Finally, the error analysis shows that the unbalance nature of the tags classes slightly affects the results.

1 Introduction

In this study, the main objective is to address a Part-of-Speech (POS) tagging task. Possible approaches to this problem are based on probabilistic methods, like Hidden Markov Model, on CRF models or on Recurrent Neural Networks, as BiLSTMs and GRUs, using contextual or non-contextual word embeddings.

In this case, the task was addressed as Sequence Labelling by using four different neural networks, based on BiLSTM and GRU (3) models with non-contextual GloVe-100 embedding (1).

The corpus on which experiments were runned is Dependency Treebank (2), which consists of 199 documents containing tagged sentences. The first 100 documents were considered for the training phase, while the rest was divided in two parts of 50 and 49 documents, acting as validation and test sets respectively. Then, for each subgroup so made, the documents were furtherly split in sentences.

From dataset analysis, tag classes turned out to be unbalanced, which should be taken into account when discussing the results, as it affects networks performances. The two best models turned out to be LSTM-based, reaching on the test set satisfactory results.

2 System description

To make them meaningful for the models, words are represented through the GloVe-100 embedding. Starting from the GloVe vocabulary, a costumed one is created by adding to it a random embedding for the OOV terms of the corpus used. Then, each vocabulary word is converted into an index according to its position in the vocabulary, such that each sentence becomes a sequence of indexes.

In this analysis, four models were proposed:

- **LSTM-1Dense** consists of a bidirectional LSTM layer of size 100 with an additional Time Distributed Dense layer of size 46 (6). This simple architecture constitutes the baseline for all other models.
- **GRU-1Dense** is composed by replacing the previous model's LSTM layer with a GRU layer. Layers dimensions are the same as the baseline model.
- **LSTM-2Dense** is composed by adding to the baseline an additional Time Distributed Dense layer. Therefore, this model consists of a first LSTM layer of size 100, followed by two dense layers with respective sizes of 256 and 46.
- **LSTM2-1Dense** consists of two bidirectional LSTM layers, the first one of size 100 and the second one with a double size. They are followed by a Time Distributed Dense layer of size 46.

Since not all sentences have the same length, some sentences are padded and some are truncated according to the maximum length computed as to handle the 99% of the sentences, which is 56 words. So, padding corresponds to an important percentage of each sentence and it is necessary not to consider it during models training. To do this, an embedding layer was introduced in each model (7). Embedding layers take as input sentences as sequences of indexes and, using a so-called embedding matrix (not trainable, in this case) which contains the embedding of each vocabulary word and which is indexed by the corresponding word index, retrieves the corresponding words embeddings. This layer then lets to compute the embedding for each input sentence, while allowing the network to ignore padding during the training phase.

3 Experimental setup and results

During the experimental phase, the four models were trained on the same training set for a maximum of 100 epochs and with a batch size of 128 using the following setup: the early stopping technique for the actual number of epochs; Adam as optimizer; softmax as activation function; the accuracy as metric during the training process and the categorical cross entropy loss. In addition, the models were evaluated on the validation set accordingly to the average f1-macro metric without considering the punctuation (5) and padding classes (Table 1). According to the results obtained, only the best two models were also evaluated on the test set. Out of conducted experiments, **LSTM-1Dense** and **LSTM2-1Dense** showed the best performances.

	F1-macro validation	**F1-macro test**
LSTM-1Dense	0.7413	0.8144
GRU-1Dense	0.6987	/
LSTM-2Dense	0.7208	/
LSTM2-1Dense	0.7500	0.8000

Table 1: F1-macro analysis

4 Discussion

As shown in Table 1, according to the macro f1-score metric, **LSTM-1Dense** and **LSTM2-1Dense** perform similarly, both of them obtaining better results on the test set than on the validation set, gaining almost 0.1 f1-score points. In order to understand the nature of this difference, a further analysis was required.

The per-class f1-scores on test and validation sets, for only tags present in both, were found to be comparable for both models and the small differences found do not justify the significant difference in performance. By analyzing instead the per-class f1-scores for those tags present in the validation but not in the test set, it turned out that they are misclassified by both models, negatively contributing to the macro f1-score on the validation set.

Deeper analyzing the results of the two models on the test set, it can be noticed that they well classify the most frequent tags with similar and stable performances, while they struggle on less frequent ones, obtaining different and less regular results.

An example of this kind of behavior can be found in how the two models classify the *plural proper noun* NNPS tag. They both show on it an f1-score equal to 0, classifying it most of the time as a *singular proper noun* NNP tag. The reason behind this kind of error could be that NNP and NNPS both refer to proper nouns and often occur one after the other, showing a close correlation. However, the NNP shows a test set support of 1471, while NNPS only of 44. Hence, the networks tend to confuse the two tags, probably because they have been trained with many more examples of NNP than NNPS, thus not being able to effectively encode and learn the difference between the two.

Another interesting misclassified sample is *comparative adverb* RBR tag, half the time predicted as *comparative adjective* JJR tag by at least one of the models. In this case, there is no significative difference between the supports size of the two tags, but they both refer to comparative words. In particular, the word *more* can be used as a RBR as well as a JJR, depending on its function in the sentence. Hence, when trying to classify *more* the networks tend to confuse the two tags.

A possible effective solution to both these sources of error could be the deployment of data augmentation techniques or the employment of a contextual word embedding, which assigns to each word a representation based on its context (4).

5 Conclusion

To conclude, the purpose of this work was to compare four different RNN architectures on a POS tagging task. The two best models turned out to be **LSTM-1Dense** and **LSTM2-1Dense**, reaching on the test set respectively 0.81 and 0.80 f1-macro scores. However, some tags with lower support were strongly misclassified, probably due to the unbalanced nature of the dataset classes. In order to improve performances, future works might focus on the employment of a contextual word embedding, which represents each word based on its context, better encoding its sense.

References

[1] Jeffrey Pennington and Richard Socher and Christopher D. Manning, *Empirical Methods in Natural Language Processing (EMNLP)*, 2014. (http://www.aclweb.org/anthology/D14-1162)

[2] Dependecy TreeBank dataset: (https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/packages/corpora/dependency_treebank.zip)

[3] Kamath, Shilpa and Shivanagoudar, Chaitra and Karibasappa, K.G. *Title: Part of Speech Tagging Using Bi-LSTM-CRF and Performance Evaluation Based on Tagging Accuracy*, 2021. (https://link.springer.com/chapter/10.1007/978-981-33-6987-0_25#Bib1)

[4] Kawin Ethayarajh *How Contextual are Contextualized Word Representations?*, 2020. (http://ai.stanford.edu/blog/contextual/)

[5] Wikipedia contributors. (2022, November 24). English punctuation. In Wikipedia, The Free Encyclopedia. Retrieved 11:30, December 7, 2022, (https://en.wikipedia.org/wiki/English_punctuation)

[6] Bogdani, *Build a POS tagger with an LSTM using Keras*, 2018. (https://nlpforhackers.io/lstm-pos-tagger-keras/)

[7] Bogdani, *Getting started with Keras for NLP*, 2018. (https://nlpforhackers.io/keras-intro/)