Assignment 1

Davide Capacchione, Álvaro Esteban Muñoz and **Luca Trambaiollo**
Master's Degree in Artificial Intelligence, University of Bologna
{ davide.capacchione, alvaro.estebanmunoz, luca.trambaiollo }@studio.unibo.it

Abstract

In this assignment, our goal is to tackle the Part-Of-Speech (POS) tagging task, a prevalent sequence labeling challenge in natural language processing. We will begin by establishing a baseline using a neural architecture commonly employed for sequence labeling on the Penn Treebank Corpus. We will compare two model variants, analyze their respective results, and conclude by discussing errors and proposing potential improvements.

1 Introduction

Part-Of-Speech (POS) tagging is a very common task in NLP, namely, it is a sequence labeling task which is normally used as a way of extracting features from the text for other purposes like named entity recognition (NER) or to feed them to a machine learning model. This type of task can be solved using two different paradigms; we can either treat it as an individual text classification task for each token on the sequence, or we can use all of them to learn its proper label. Even though both are good techniques for solving this task, the second approach tends to give better results, this is due to its capacity of using context as a way of recognizing the patterns behind each label.

The experiments were performed on the Penn TreeBank Corpus (Marcus et al., 1999), this is a set of documents in which each sentence is parsed, i.e. each word is tagged with its syntactic function. The baseline model proposed in the following section was confronted against to variants of itself. The results of our experiments made us understand why this task is not as simple as it seems and how context can be crucial to differ homonyms like prepositions and adverbs.

2 System description

We are asked to implement, as a baseline model, an architecture for sequence labeling using a pre-trained embedding (GloVe) a bidirectional LSTM layer and a dense layer with softmax activation function [figure 2]. This model is then compared with two other versions of it, each with a small change on the architecture.

The first model (Model 1) adds an extra LSTM layer while the second one (Model 2) appends another dense layer. The pre-trained embedding is implemented as an embedding layer which is initialized using the embedding matrix. This one is built using the embedding vectors of each word found on the training vocabulary. The vocabulary contains two extra tokens; <PAD> and <UNK>, each with its own embedding vector on the matrix wich are added as random vectors.

Before feeding the data to the model, this is preprocessed following a few stages, these stages can be seen on figure 1 and are:

- Tokenization: Extract the sentences from the documents and split them by sequences of tokens (word-level).
- Lower casing: Words are all lower cased.
- Indexing: Each token is then replaced by its id on the vocabulary.
- Padding: To process tensors we need to have rectangular matrices of sequences, therefor we pad to the maximum possible length in the vocabulary, this way all the sequences will have the same length.

3 Experimental setup and results

Using the architectures described above we perform three experiments using three different random seeds. The optimizer and learning rates used are the ones defined by default by Tensorflow; RMSprop with $\alpha = 0.001$. Model performance is measured using macro F1-Score.

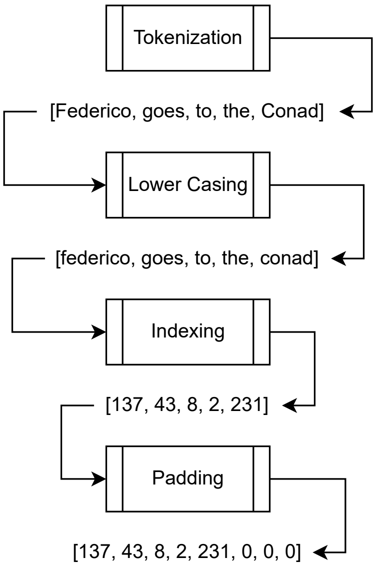


Figure 1: Pre-processing Pipeline

	Seed: 23	Seed: 42	Seed: 87
Baseline	0.7588	0.7539	0.7604
Model 1	0.7503	0.7432	0.7448
Model 2	**0.7650**	0.7599	0.7632

Table 1: Results on the validation set

As seen on table 1, the model which tends to perform better is the Model 2. Results on test set can be checked on table 2

	Seed: 23	Seed: 42	Seed: 87
Model 2	**0.7948**	0.7897	0.7925

Table 2: Results on the test set

4 Discussion

From the results we can see that the variations do not change too much the results and that they tend to improve on test set. We computed other metrics to be able to clarify a bit better what are the weakness of our model. The results are from the validation test because the test set, while similar, is smaller and less stable. The most remarkable things are the following ones:

Adverbs When analyzing the precision and recall of each tag [table 3] we can notice that for adverbs the model have some troubles, specially comparatives and superlatives. Our first intuition is to think that they are being confused with comparative and superlative adjectives (JJR and JJS tags). This pattern was confirmed by looking at the confusion matrix [table 4].

Tag	Precision	Recall	F1-Score	Support
RBR	0.2500	0.1429	0.1818	35
RBS	0.4000	0.1538	0.2222	13

Table 3: Classification report of comparative (RBR) and superlative (RBS) adverbs

RBR	28	0	5	0
RBS	0	10	0	2
True/Predicted	JJR	JJS	RBR	RBS

Table 4: Confusion Matrix of RBR, RBS, JJR and JJS tags

Particles Another tag that gets lower metrics than expect are the particles (RP). Again, first intuition would be to think they are being mistaken with prepositions, since many words used as prepostions (IN) can also be particles like the word "to".

IN	3195	15
RP	16	21
True/Predicted	IN	RP

Table 5: Confusion Matrix of RP and IN

On the confusion matrix shown above [table 5] we can confirm our intuition. Results for prepositions are not that bad, however, the average for the particles in comparison is quite low since only 21 out of 43 (indicated by the support on the classification report) are being correctly classified and 16 of the total of errors are due to prepositions.

Nouns Related to the amount of nouns that exists in the vocabulary errors are not surprising, however we can remark the model has special difficulties to differ proper nouns from the rest of them. This might be a consequence of our pre-processing which removes one of the most noticeable differences for proper nouns, the capital letters.

The image 3 shows the nouns section of the full confusion matrix for all the classes, the order in which the tags are shown is, from top to bottom; Noun (NN), Proper Noun (NNP), Plural Noun (NNS), Proper Plural Noun (NNPS).

Verbs Verbs are a bit special, mostly because some verb tenses can be used as adjectives or nouns.

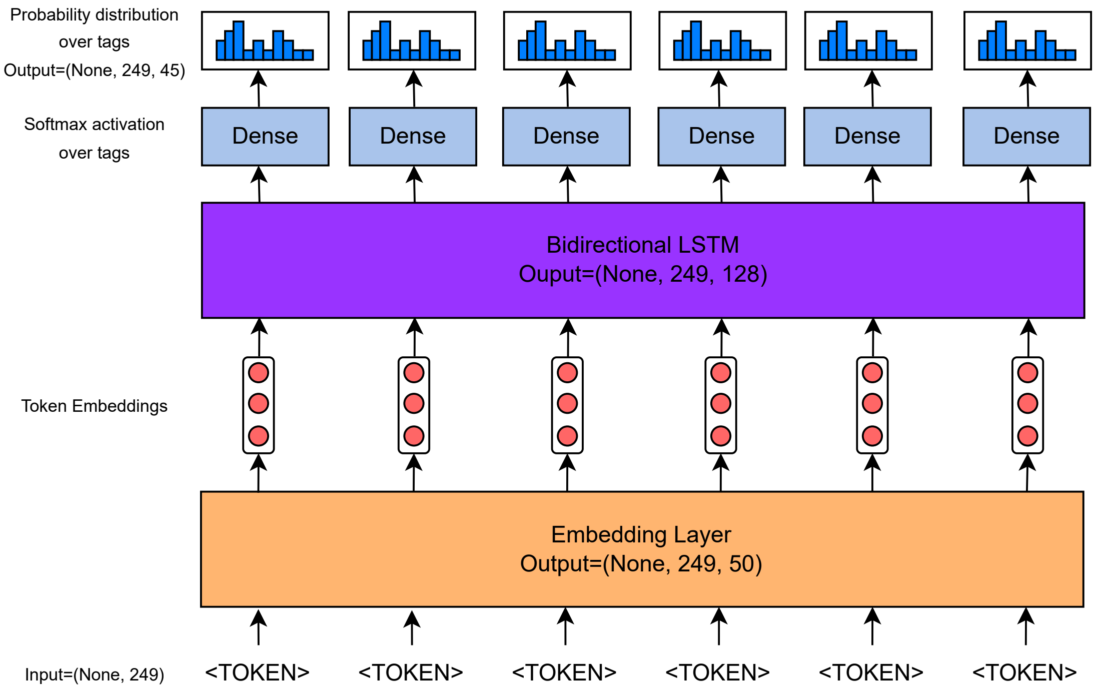
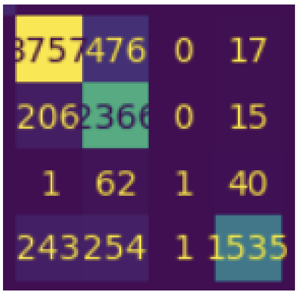
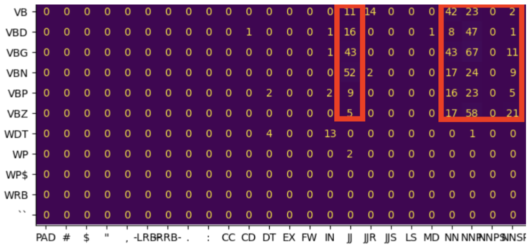


Figure 2: Baseline model architecture



3757	476	0	17
206	2366	0	15
1	62	1	40
243	254	1	1535

Figure 3: Image from nouns section on the generated confusion matrix



	PAD	#	$	''	,	-LRB-	-RRB-	.	:	CC	CD	DT	EX	FW	IN	JJ	JJR	JJS	LS	MD	NN	NNP	NNPS	NNS
VB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	14	0	0	0	42	23	0	2
VBD	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	16	0	0	0	1	8	47	0	1
VBG	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	43	0	0	0	0	43	67	0	11
VBN	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	52	2	0	0	0	17	24	0	9
VBP	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	9	0	0	0	0	16	23	0	5
VBZ	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	0	17	58	0	21
WDT	0	0	0	0	0	0	0	0	0	0	0	4	0	0	13	0	0	0	0	0	0	1	0	0
WP	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0
WP$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
WRB	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
``	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 4: Image from verbs section on the generated confusion matrix

Once more on the confusion matrix we can confirm this [image 4].

5 Conclusion

To sum up, POS-tagging is a task that depends strongly on the context. Preprocessing did not work as expected, we imagined it would simplify the task but instead it might have removed crucial information for the model to distinguish between noun tags. As we expected from verbs, we noticed miss classification increases due to their possible use as adjectives and nouns depending on the context. Moreover, a bigger dataset could be crucial for the model to learn more difficult patterns as it happens between comparative and superlative adverbs and adjectives.

Among possible improvements we can remark removing or trying a different preprocessing and relying on bigger and/or distinct vocabulary pretrained embeddings.

References

Mitchell P. Marcus, Beatrice Santorini, Mary Ann Marcinkiewicz, and Ann Taylor. 1999. Treebank-3 ldc99t42. In *Web Download*. Philadelphia: Linguistic Data Consortium, Philadelphia: Linguistic Data Consortium.