A BiLSTM-CRF PoS-tagger for Italian tweets using morphological information

Fabio Tamburini
FICLIT - University of Bologna, Italy
fabio.tamburini@unibo.it

Abstract

English. This paper presents some experiments for the construction of an high-performance PoS-tagger for Italian using deep neural networks techniques (DNN) integrated with an Italian powerful morphological analyser that has been applied to tag Italian tweets. The proposed system ranked third at the EVALITA2016-PoSTWITA campaign.

Italiano. *Questo contributo presenta alcuni esperimenti per la costruzione di un PoS-tagger ad alte prestazioni per l'italiano utilizzando reti neurali 'deep' integrate con un potente analizzatore morfologico che è stato applicato all'annotazione di tweet. Il sistema si è classificato terzo nella campagna di valutazione EVALITA2016-PoSTWITA.*

1 Introduction

In recent years there were a large number of works trying to push the accuracy of the PoS-tagging task forward using new techniques, mainly from the deep learning domain (Collobert et al., 2011; Søgaard, 2011; dos Santos and Zadrozny, 2014; Huang et al., 2015; Wang et al., 2015; Chiu and Nichols, 2016).

In this study, still work-in-progress, we set-up a PoS-tagger for Italian able to gather the highest classification performances by using any available language resource and the most up-to-date DNN. We used AnIta (Tamburini and Melandri, 2012), one of the most powerful morphological analysers for Italian, based on a wide lexicon (about 110.000 lemmas), for providing the PoS-tagger with a large set of useful information.

The general PoS-tagger has been described in (Tamburini, 2016). This paper briefly describes the adaptation process we made for annotating Italian tweets.

2 Input features

The set of input features for each token is basically formed by two different components: the word embedding and some morphological information.

2.1 Word Embeddings

All the embeddings used in our experiments were extracted from a twitter corpus composed by 200 millions of tokens, belonging to 11 millions of tweets downloaded at the beginning of 2012 (February and March), by using the tool `word2vec`[1] (Mikolov et al., 2013). We added two special tokens to mark the sentence beginning '<s>' and ending '</s>'.

2.2 Morphological features, Unknown words handling and Sentence padding

As described in (Tamburini, 2016), we extended the word embeddings computed in a completely unsupervised way by concatenating to them a vector containing the possible PoS-tags provided by the AnIta analyser. This tool is also able to identify, through the use of simple regular expressions, numbers, dates, URLs, emails, etc., and to assign them the proper tag(s).

With regard to unknown words handling and sentence padding we followed the same procedure for the general tagger described in the cited paper, managing each sentence as one single sequence padded at the borders.

3 (Deep) Learning Blocks

All the experiments presented in this paper has been performed using Keras[2]. Keras provides some basic neural network blocks as well as different learning procedures for the desired network

[1] https://code.google.com/archive/p/word2vec/
[2] https://github.com/fchollet/keras/tree/master/keras

configuration and simple tools for writing new blocks. In our experiments we used Bidirectional Long Short-Term Memory - LSTM (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005), and a new layer we wrote to handle Conditional Random Fields (CRF). We did some experiments stacking them after the softmax layer.

Figure 1 shows the DNN structure used in our experiments.

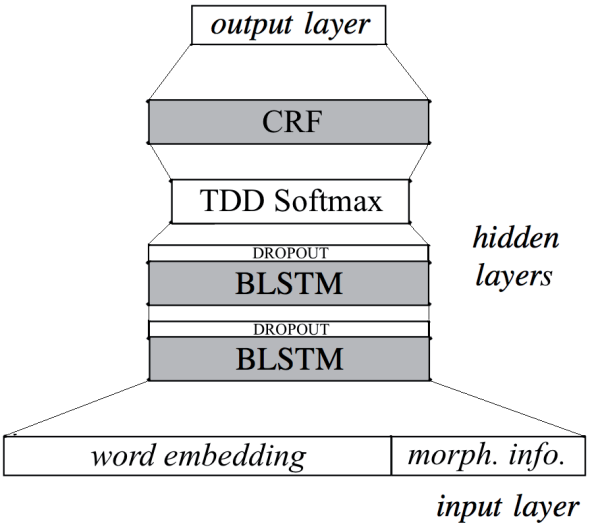

Figure 1: The DNN used in our experiments.

4 Experiments

During the set up phase, we did a lot of experiments for tuning the PoS-tagger using the Development Set. The following section describes the setup and the obtained results.

4.1 Hyper-Parameters

As for the general tagger (Tamburini, 2016), we did not test all the possible combinations; we used, instead, the most common set-up of parameters gathered from the literature. Table 1 outlines the whole setup for the unmodified hyper-parameters.

The DNN hidden layers were composed by 256 neurons.

4.2 The Early Stopping procedure

The usual way to set up an experiment following this suggestions involves splitting the gold standard into three different instance sets: the training set, for training, the validation set, to determine the stopping point, and the test set to evaluate the system. However, we are testing our systems on real evaluation data that has been already

`word2vec` **Embed.**		**Feature extraction**	
Hyperpar.	**Value**	**Hyperpar.**	**Value**
type	SkipGr.	window	5
size	100	**Learning Params.**	
(1/2) win.	5	batch (win)	1/4*NU
neg. sampl.	25	batch (seq)	1
sample	1e-4	Opt. Alg.	`Adam`
iter	15	Loss Func.	Categ.CE

Table 1: Unmodified hyper-parameters and algorithms used in our experiments. NU means the number of hidden or LSTM units per layer (the same for all layers). For `Adam` refer to (Kingma and Ba, 2015).

split by the organisers into development and test set. Thus, we can divide the development set into training/validation set for optimising the hyper-parameters and define the stopping epoch, but, for the final evaluation, we would like to train the final system on the complete development set to adhere to the evaluation constraints and to benefit from using more training data.

Having two different training procedures for the optimisation and evaluation phases leads to a more complex procedure for determining the stopping epoch. Moreover, the typical accuracy profile for DNN systems is not smooth and oscillate heavily during training. To avoid any problem in determining the stopping point we smoothed all the profiles using a bezier spline. The procedure we adopted to determine the stopping epoch is (please look at Fig. 2): (1) find the first maximum in the validation smoothed profile - A; (2) find the corresponding value of accuracy on the smoothed training profile - B; (3) find the point in the smoothed development set profile having the same accuracy as in B - C; (4) select the epoch corresponding at point C as the stopping epoch - D.

4.3 Results

First of all we split the Development Set into a proper training set (109,273 tokens) and a validation set (12,132 tokens) for setting up the entire system, to verify the correctness of the whole tagging process and to derive a first estimate of the tagger performances. We ran some experiments with three different seeds and, after having applied the early stop procedure described above, we derived the optimal stopping epoch to be used for the final testing and the tagging performances on the

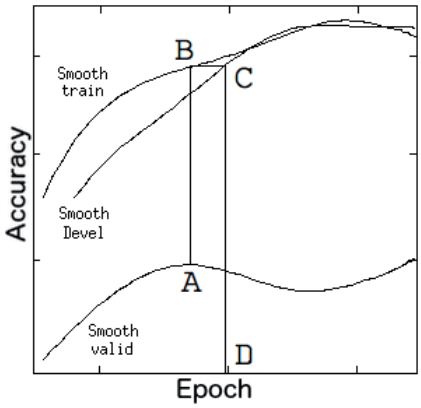

Figure 2: The early stopping procedure.

training/validation pair. Table 2 outlines these results.

	Tagging Accuracy	Stopping epoch
(A)	95.56	12
(B)	95.49	13
(C)	95.53	10
Avg.	95.53	

Table 2: The tagging results obtained in the setup phase.

We presented two kinds of results for the final evaluation: (1) the first official run was derived by applying the same random seed as the configuration (A), and (2) we submitted also, as an unofficial run, a tagged version obtained by combining all the three configurations using a voting scheme.

In Table 3 we can see our system performances, namely AnIta-BiLSTM-CRF (ABC), compared with all the systems that participated at the PoSTWITA 2016 task.

5 Conclusions and discussion

The proposed system for PoS-tagging, integrating DNNs and a powerful morphological analyser, exhibited very good accuracy results when applied to the PoSTWITA task of the EVALITA 2016 campaign.

Looking at the official results, and comparing them with the experiments we devised to set up our system, it is easy to note the large difference in performances. During the setup phase we obtained coherent results well above 95% of accuracy, while the best performing system in the official evaluation exhibit performances slightly above 93%. It is a huge difference for this kind of task, rarely observed in real experiments.

#	TEAM	Tagging Accuracy
1	Team1	0.931918 (4435/4759)
2	Team2	0.928556 (4419/4759)
3	**ABC_UnOFF**	0.927926 (4416/4759)
4	Team4	0.927086 (4412/4759)
5	**ABC**	0.924564 (4400/4759)
6	Team5	0.922463 (4390/4759)
7	Team5_UnOFF	0.918470 (4371/4759)
8	Team6	0.915739 (4358/4759)
9	Team6_UnOFF	0.915318 (4356/4759)
10	Team7	0.878966 (4183/4759)
11	Team8	0.859634 (4091/4759)
12	Team2_UnOFF	0.817819 (3892/4759)
13	Team9	0.760034 (3617/4759)

Table 3: EVALITA2016 - PoSTWITA participants' results with respect to Tagging Accuracy. "UnOFF" marks unofficial results.

In my opinion there is only one reason that explains this difference in performances: the documents in the test set are not drawn from the same kind of corpus as the development set and this is not a desirable condition unless you explicitly organise a domain adaptation task. The TS, as well as the DS, have been inherited from the SENTIPOLC task of the same evaluation campaign, thus the problem could be the same also for other tasks of the same evaluation campaign.

References

Jason Chiu and Eric Nichols. 2016. Sequential Labeling with Bidirectional LSTM-CNNs. In *Proc. International Conf. of Japanese Association for NLP*, pages 937–940.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537.

Cicero dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proc. of the 31st International Conference on Machine Learning, JMLR*, volume 32. JMLR W&CP.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

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

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. *ArXiv e-prints, 1508.01991*.

D.P. Kingma and J.L. Ba. 2015. Adam: a method for stochastic optimization. In *Proc. International Conference on Learning Representations - ICLR.*, pages 1–13.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *Proc. of Workshop at ICLR*.

Anders Søgaard. 2011. Semi-supervised condensed nearest neighbor for part-of-speech tagging. In *Proc. of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 48–52, Portland, Oregon, USA.

Fabio Tamburini and Matias Melandri. 2012. AnIta: a powerful morphological analyser for Italian. In *Proc. 8th International Conference on Language Resources and Evaluation - LREC 2012*, pages 941–947, Istanbul.

Fabio Tamburini. 2016. (Better than) State-of-the-Art PoS-tagging for Italian Texts. In *Proc. Third Italian Conference on Computational Linguistics - CLiC-it*, Napoli.

Peilu Wang, Yao Qian, Frank. K Soong, Lei He, and Hai Zhao. 2015. A Unified Tagging Solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. *ArXiv e-prints, 1511.00215*.