(Better than) State-of-the-Art PoS-tagging for Italian Texts

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. The results obtained by the proposed system on standard datasets taken from the EVALITA campaigns show large accuracy improvements when compared with previous systems from the literature.

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. I risultati ottenuti sui dataset delle campagne EVALITA da parte del sistema proposto mostrano incrementi di accuratezza piuttosto rilevanti in confronto ai precedenti sistemi in letteratura.*

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).

All these studies are mainly devoted to show how to find the best combination of new neural network structures and character/word embeddings for reaching the highest classification performances, and typically present solutions that do not make any use of specific language resources (e.g. morphological analysers, gazetteers, guessing procedures for unknown words, etc.). This is, in general, a very desirable feature because it allows for the production of tools not tied to any specific language, but in various evaluation campaigns, at least for highly-inflected languages as Italian, the results showed quite clearly that this task would benefit from the use of specific and rich language resources (Tamburini, 2007; Attardi and Simi, 2009).

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.

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 the CORIS corpus (Rossini Favretti et al., 2002), a 130Mw synchronic reference corpus for Italian, 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

One of the most useful kind of information that increases the performances of PoS-taggers concerns the list of all possible tags for a single word-form. Having a restricted list of possibility enable the tagger to reduce the search space and force it to take reasonable decisions. The results obtained

[1] https://code.google.com/archive/p/word2vec/

in past PoS-taggers evaluations on Italian agree in suggesting that powerful morphological analysers based on large lexica are invaluable resources to increase tagger accuracy. For these reasons, 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 assign them the proper tag(s).

2.3 Unknown words handling and Sentence padding

The source of most tagging errors is certainly the presence of the so called 'unknown words', word-forms for which the tagger did not receive any information during the training phase. A morphological analyser based on a large lexicon could certainly alleviate this problem providing information also for word-forms not belonging to the training set, but there are large classes of tokens that cannot be successfully handled by the analyser, for example proper names, foreign words, etc.

In a previous work (Tamburini, 2007b) we showed that using such a powerful morphological analyser, the word-forms not covered by it in real texts belongs at 95% to the class of proper names, adjectives and common nouns and a simple heuristic correctly assigns most of the cases. In this way AnIta always provides one or more PoS-tag hypothesis for each word-form that can be transformed into a binary vector with 1s in correspondence of possible PoS-tags and 0s otherwise, but if the word-form did not have a computed embedding, the first part of the input features would not be defined. For solving such problem, instead of using the common solution of assigning a random vector to all unknown words, we averaged all the embeddings of the other word presenting exactly the same combination of possible PoS-tags.

It is also a common practice to pad sentences, at the beginning and at the end, using random vectors, but we, instead, used the real embeddings computed for the special tokens '<s>' and '</s>', added for this purpose, with the respective tag 'BoS' and 'EoS'. Due to the internal structuring of the used tensor manipulating application (see later), we were forced to add also an out-of-sentence vector to pad sentences to their maximal length, and the correspondent tag OoS.

2.4 Data structuring

We experimented two different ways of structuring the input features for processing:

- `Win`: this mode of organising input data is based on a sliding window that starts from the beginning of each sentence and concatenates word feature vectors into one single vector. Padding is inserted at sentence borders.
- `Seq`: each sentence is managed as one single sequence padded at the borders.

Each network experimented in this study uses one of these two data structuring type.

3 (Deep) Learning Blocks

All the experiments presented in this paper has been performed using Keras[2] a "a minimalist, highly modular neural networks library, written in Python and capable of running on top of either TensorFlow or Theano", two widely used tensor manipulation libraries. Keras provides some basic neural network blocks as well as different learning procedures for the desired network configuration and simple tools for writing new blocks. In our experiments we used some of them, namely multilayer-perceptrons (MLP) and Long Short-Term Memory (LSTM), and we wrote a new block to handle Conditional Random Fields (CRF).

MLP are simple feedforward neural networks with one or more fully-connected hidden layers. We obtained maximum performances using only one hidden layer.

LSTM networks (Hochreiter and Schmidhuber, 1997; Graves and Schmidhuber, 2005) are a kind of recurrent neural network which received a lot of attention in recent years due to their ability of produce good classification results for sequence problems. Their property of preventing the vanishing (and exploding) gradient problem that affects standard recurrent neural networks made them the default choice for solving sequence classification problems inside the DNN framework. Usually this kind of units are arranged to form a bidirectional chain (BiLSTM) for gathering information both from the past and from the future of the input data sequence, a very desirable issue for such kind of classification problems. In all our experiments using BiLSTM we obtained maximum performances by stacking two layers of them, with

[2] https://github.com/fchollet/keras/tree/master/keras

a dropout layer after each of them (Srivastava et al., 2014), and a final dense softmax layer, or a time-distributed-dense softmax layer, feeded by the BiLSTM output.

Linear CRFs are the simpler Probabilistic Graphical Model (PGM) and it has been successfully used in NLP for sequence classification problems (Lafferty et al., 2001). We did some experiments stacking them after the softmax layer.

Figure 1 shows the most complex DNN structure used in out experiments.

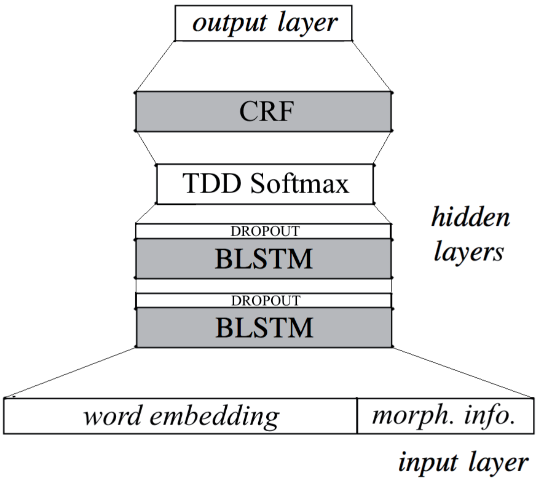

Figure 1: The most complex DNN used in our experiments.

4 Experiments

All the experiments presented in this paper to test the effectiveness of the proposed system refer to two evaluation campaigns organised inside the EVALITA[3] framework. In particular, in 2007 and 2009 were organised specific task to test Italian PoS-taggers performances.

4.1 The EVALITA 2007 evaluation

Two separate data sets were provided: the Development Set (DS), composed of 133,756 tokens, was used for system development and for the training phase, while a Test Set (TS), composed of 17,313 tokens, was used as a reference for systems evaluation. Both contain various documents belonging mainly to journalistic and narrative genres, with small sections containing academic and legal/administrative prose. Each participant was allowed to use any available resource or could freely induce it from the training data.

The original PoS-tagging task involved two different tagsets, but our experiments used only the tags and the annotation named ‘EAGLES-like’.

The evaluation metrics were based on a token-by-token comparison and only one tag was allowed for each token. The EVALITA metric considered in this study is the *Tagging Accuracy*, defined as the number of correct PoS-tag assignments divided by the total number of tokens in the TS. See (Tamburini, 2007) for further details.

4.2 The EVALITA 2009 evaluation

The DS consisted in 113895 word forms (already divided in a training set - 108,874 tokens - and a validation set - 5021 tokens). The TS consisted of 5066 word forms. The training set is formed by newspaper articles from ‘La Repubblica’, while the validation and test set contain documents extracted from the Italian Wikipedia. This test the degree of system adaptation to new domains.

The organisers evaluated the results using a coarse grained (37 tags) and a morphed (336 tags) tagsets inserted in a closed/open task framework, but in this study all the results refer to the open task (one can use external resources) on the coarse grained tagset. The evaluation metric is the same described before in section 4.1. See (Attardi and Simi, 2009) for further details.

4.3 Hyper-Parameters

Considering the large number of hyper-parameters involved in the whole procedure, 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 set-up for the unmodified hyper-parameters.

`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).

[3] http://www.evalita.it/

4.4 The Early Stopping Drama

There are some interesting studies (Bengio, 2012; Prechelt, 2012) dealing with the problem of stopping the learning process at the right point; this issue is known as the 'early stopping' problem. Choosing the correct epoch to stop the learning process helps avoiding overfitting on the training set and usually produces systems exhibiting better generalisations. But, how to choose the correct epoch is not simple. The suggestion given in various studies on this topic is to consider a validation set and stop the learning process when the performances on this set do not increase anymore or even decrease, a clear hint of overfitting.

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 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.

4.5 Results

Table 2 outlines the systems' accuracies for different configurations for both datasets. We can observe that by using AnIta morphological information, as well as all the techniques described in section 2.3, improves the systems' results by more than 1%. Considering the data structuring described in section 2.4, the management of an entire sentence as a complete sequence allows recurrent configurations to work with larger contexts producing better results. Adding a CRF layer after the BiLSTM seems to slightly improve the performances, but not in a significant way.

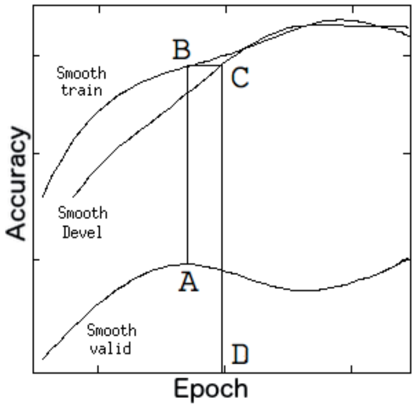

Figure 2: The early stopping procedure.

SYSTEM	TA		Notes
	E07	E09	
MLP-256	96.45	95.57	Win=5
MLP-256	97.75	96.84	M,Win=5
2-BiLSTM-256	98.12	97.30	M,Win=5
2-BiLSTM-256	98.14	97.45	M,Seq
2-BiLSTM-256-CRF	**98.18**	**97.48**	M,Seq

Table 2: Tagging accuracies (TA) for different configurations for both datasets. ('M' marks the use of AnIta morphological information).

In Table 3 we can see our best system performances, namely AnIta-BiLSTM-CRF, compared with the three best systems of the considered EVALITA campaigns. As you can see, in both cases the proposed system ranked first improving the scoring by large quantities.

5 Conclusions

The proposed system for PoS-tagging, integrating DNNs and a powerful morphological analyser, exhibited very good accuracy results when applied to standard Italian evaluation datasets from the EVALITA campaigns. The information from AnIta proved to be crucial to reach such accuracy values as well as stacked BiLSTM networks processing entire sentence sequences.

EVALITA 2007	
SYSTEM	TA
AnIta-BiLSTM-CRF	**98.18**
FBKirst_Zanoli	98.04
UniTn_Baroni	97.89
ILCcnrUniPi_Lenci	97.65
EVALITA 2009	
AnIta-BiLSTM-CRF	**97.48**
UniPi_SemaWiki 2	97.03
UniPi_SemaWiki 1	96.73
UniPi_SemaWiki 4	96.67

Table 3: Participants' results with respect to Tagging Accuracy (TA) at EVALITA 2007 and 2009.

We have to further test different DNN configurations and their integration with other kind of PGMs as well as make more experiments with different hyperparameters.

References

Giuseppe Attardi and Maria Simi. 2009. Overview of the EVALITA 2009 Part-of-Speech Tagging Task. In *Proc. of Workshop Evalita 2009*.

Yoshua Bengio. 2012. Practical Recommendations for Gradient-Based Training of Deep Architectures. In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg.

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.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. 18th International Conf. on Machine Learning*, pages 282–289.

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

Lutz Prechelt. 2012. Early Stopping — But When? In Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors, *Neural Networks: Tricks of the Trade: Second Edition*, pages 53–67. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rema Rossini Favretti, Fabio Tamburini, and Cristiana De Santis. 2002. CORIS/CODIS: A corpus of written Italian based on a defined and a dynamic model. In Andrew Wilson, Paul Rayson, and Tony McEnery, editors, *A Rainbow of Corpora: Corpus Linguistics and the Languages of the World*, pages 27–38. Lincom-Europa, Munich.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

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. 2007. EVALITA 2007: the Part-of-Speech Tagging Task. *Intelligenza Artificiale*, IV(2):4–7.

Fabio Tamburini. 2007b. CORISTagger: a high-performance PoS tagger for Italian. Intelligenza Artificiale. *Intelligenza Artificiale*, IV(2):14–15.

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*.