Neural Networks 66 (2015) 11–21

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# Multilingual part-of-speech tagging with weightless neural networks

Hugo C.C. Carneiro [a,*], Felipe M.G. França [a], Priscila M.V. Lima [b]

[a] Systems Engineering and Computer Science Program/COPPE, Universidade Federal do Rio de Janeiro (UFRJ) - Caixa Postal 68511, Cidade Universitária, Rio de Janeiro, Rio de Janeiro 21941-972, Brazil

[b] Instituto Tércio Pacitti de Aplicações e Pesquisas Computacionais (NCE), Universidade Federal do Rio de Janeiro (UFRJ) - Av. Athos da Silveira Ramos, 274 - Edifício do Centro de Ciências Matemáticas e da Natureza, Bloco E, Cidade Universitária, Rio de Janeiro, Rio de Janeiro 21941-916, Brazil

ARTICLE INFO

*Article history:*
Received 12 April 2014
Received in revised form 17 February 2015
Accepted 22 February 2015
Available online 2 March 2015

*Keywords:*
Weightless neural networks
Part-of-speech tagging

ABSTRACT

Training part-of-speech taggers (POS-taggers) requires iterative time-consuming convergence-dependable steps, which involve either expectation maximization or weight balancing processes, depending on whether the tagger uses stochastic or neural approaches, respectively. Due to the complexity of these steps, multilingual part-of-speech tagging can be an intractable task, where as the number of languages increases so does the time demanded by these steps. WiSARD (**Wi**lkie, **S**tonham and **A**leksander's **R**ecognition **D**evice), a weightless artificial neural network architecture that proved to be both robust and efficient in classification tasks, has been previously used in order to turn the training phase faster. WiSARD is a RAM-based system that requires only one memory writing operation to train each sentence. Additionally, the mechanism is capable of learning new tagged sentences during the classification phase, on an incremental basis. Nevertheless, parameters such as RAM size, context window, and probability bit mapping, make the multilingual part-of-speech tagging task hard. This article proposes mWANN-Tagger (**m**ultilingual **W**eightless **A**rtificial **N**eural **N**etwork **tagger**), a WiSARD POS-tagger. This tagger is proposed due to its one-pass learning capability. It allows language-specific parameter configurations to be thoroughly searched in quite an agile fashion. Experimental evaluation indicates that mWANN-Tagger either outperforms or matches state-of-art methods in accuracy with very low standard deviation, i.e., lower than 0.25%. Experimental results also suggest that the vast majority of the languages can benefit from this architecture.



## 1. Introduction

Part-of-speech tagging (POS-tagging) is a common task in natural language processing. It requires high accuracy since its result is commonly used as input (or as part of the input) to other tasks, e.g., syntactic parsing and machine translation. Multilingual POS-tagging presents a further challenge. Not only its accuracy must be high in every language, but also the tagger used must have an agile language-independent architecture. Nowadays, two different techniques are used: (i) several POS-taggers are trained independently, which can create some overhead, or (ii) cross-lingual POS-taggers are employed, which use previously annotated relations between words of different *corpora* (composed of texts in different languages) in order to remove tagging ambiguities (Naseem, Snyder, Eisenstein, & Barzilay, 2009; Snyder, Naseem, Eisenstein, & Barzilay, 2008, 2009). In the first case, once a new tagger is needed for a particular language, there is no technique to speed up the parameter tuning procedure. In both strategies, the architecture of the tagger is not truly language-independent. This article proposes a tagger with both a language-independent architecture and the ability to train taggers for new languages with little time spent on parameter tuning procedures.

Neural network models have proven useful in solving natural language processing tasks (Caridakis, Karpouzis, Drosopoulos, & Kollias, 2012; Hinoshita, Arie, Tani, Okuno, & Ogata, 2011; Klein, Kamp, Palm, & Doya, 2010). Neural-based taggers have been proposed since Schmid (1994), some of which employed the neuro-symbolic paradigm, such as Ma, Murata, Uchimoto, and Isahara (2000) and Marques, Bader, Rocio, and Hölldobler (2007). More recently, a weightless neural-based tagger was proposed in Carneiro, França, and Lima (2010). Despite the variety of techniques and parameters adjustment employed, it is observed that every neural tagger created ever since has only been used for monolingual part-of-speech tagging. This work explores the weightless neural

* Corresponding author.
*E-mail addresses:* hcesar@cos.ufrj.br (H.C.C. Carneiro), felipe@cos.ufrj.br (F.M.G. França), priscilamvl@gmail.com (P.M.V. Lima).

http://dx.doi.org/10.1016/j.neunet.2015.02.012
0893-6080/

paradigm for multilingual part-of-speech tagging through the proposal of mWANN-Tagger(**m**ultilingual **W**eightless **A**rtificial **N**eural **N**etwork **tagger**), in order to speed up the search of language-specific parameter configurations.

Several other POS-tagger models were proposed, such as rule-based ones (Brill, 1992, 1994) and those that work as a finite-state machine that employs sliding windows (Sánchez-Villamil, Forcada, & Carrasco, 2004, 2005). Of the very widespread probabilistic graphical models for POS-tagging, *hidden Markov models* (HMM) (Jurafsky & Martin, 2008; Manning & Schütze, 1999), *maximum entropy Markov model* (MEMM) (McCallum, Freitag, & Pereira, 2000) and *Conditional random fields* (CRF) (Lafferty, McCallum, & Pereira, 2001) constitute some of the most used techniques. However, HMMs may present some drawbacks to correct classification: (i) if the part of speech of a word cannot be inferred from the words in its vicinity, or (ii) if a word was not presented in the training *corpus*. The former problem was solved with the use of second-order Markov models through the use of trigrams (Brants, 2000; Petrov, Das, & McDonald, 2012). The latter incorporated the use of features to the probabilistic nature of HMM, initially proposed in Ratnaparkhi (1996).

Ratnaparkhi (1996) proposed the use of binary feature functions $f_j(w_i, t_i)$ to represent that word $w_i$ appears with tag $t_i$ in the *corpus*. Examples of feature functions can include information about if the word ends in a particular suffix, if it is capitalized, if it is a number and so on. This way, words could be substituted by a feature vector $\mathbf{f}(w_i, t_i)$, enabling the tagging of unpresented words. These words do not appear in the training *corpus*, however some of its characteristics could appear in the feature vector. A feature $f_j(w_i, t_i)$ could assume the value 1 if $w_i$ possesses the characteristic associated to the feature, and 0 otherwise.

The model makes use of the *maximum entropy* formalism (ME), which states that the probability which best represents a given state of knowledge is the one with highest entropy. This probability is known as *maximum entropy probability distribution* or *Gibbs distribution* (Levine & Tribus, 1978). The model was optimized by maximizing the log-likelihood of this probability distribution.

The feature functions were incorporated to the Markovian architecture of HMM to create a more robust probabilistic graphical model, MEMM (McCallum et al., 2000). This model proved to be very effective by incorporating the global knowledge from probabilistic graphical models and the ability to tag unseen words more precisely through the use of arbitrary features. However, this model presented a drawback called the "label bias problem". This means that if there are states with low-entropy transitions to its following states, they will take little notice of an observation (Bottou, 1991; Lafferty et al., 2001). In order to overcome this limitation of MEMMs, Lafferty et al. (2001) proposed the CRF.

CRFs are defined according to two random variables, **X** over data sequences and **Y** over label sequences. In POS-tagging tasks, **X** range over natural language sentences and **Y** range over the possible strings of tags associated with **X**. Also, CRFs can be used in any possible graph $G = (V, E)$, but for the POS-tagging task it is recommended to use a chain-like graph. This particular case of CRF was called HMM-like CRF by Lafferty et al. (2001). This is the only kind of CRF that is detailed in this paper, since it is the one used in POS-tagging tasks. CRFs substitute the notion of dependency between states of HMMs and MEMMs by the use of features. This way, CRFs work with undirected graphs, differently from HMMs and MEMMs which use directed graphs. HMM-like CRFs employ two distinct types of features: one that is defined for each pair of states $(y', y)$ and another for each pair of state-observations $(y, x)$.

Those maximum entropy models proved to be quite versatile and able to tag texts quite accurately, especially CRF (Lafferty et al., 2001). However, as the number of features grows, their accuracy may diminish and the time spent during the training step increases considerably. This article proposes a POS-tagger that employs a one-pass learning model, whose optimal parameter configuration can be thoroughly searched in feasible time. The choice of a non-overtraining-prone model helps in producing taggers that keep a high accuracy despite the complexity of the feature space. This way, it is possible to create new accurate taggers more quickly. Comparison of language-specific characteristics could benefit from these studies.

A review of weightless neural models, especially WiSARD (**Wi**lkie, **S**tonham and **A**leksander's **R**ecognition **D**evice), is presented in Section 2, so that the reader is capable to understand how the WiSARD model tags sentences (Section 3). The experimental methodology used to test mWANN-Tagger capabilities on tagging texts in languages of distinct natures is discussed in Section 4. Experimental results are described and analyzed in depth in Section 5. Conclusion and future work directions are presented in Section 6.

## 2. Weightless artificial neural networks and WiSARD model

Weightless Artificial Neural Networks (WANNs) are a set of ANN models in which there is no synaptic weight balancing during the training phase. This lack of synaptic weight is compensated by the use of Random Access Memories (RAMs) inside its neural nodes, whereas traditional neural network neurons do not store any information, but only applies a multivariate nonlinear continuous function whose arguments are either the outputs given by the nodes in the previous layer or the network inputs.

There are several weightless artificial neural models, e.g., WiSARD (Aleksander, Thomas, & Bowden, 1984), and its variants, WiSART (a portmanteau of WiSARD and ART—Adaptive Resonance Theory Grossberg, 1987) (Fulcher, 1992), AUTOWISARD (an unsupervised learning extension of WiSARD that allows automatic generation of new discriminators) (Wickert, França, & Prieto, 2001) and others; Probabilistic Logic Nodes (Kan & Aleksander, 1987); Goal Seeking Neuron (Filho, Fairhurst, & Bisset, 1991); General Neural Unit (Aleksander & Morton, 1991); G-RAM (Generalizing Random Access Memory) (Aleksander, 1990a), as well as its most common implementation Virtual G-RAM (VG-RAM) (Mrsic-Flogel, 1991); Sparse Distributed Memory (Kanerva, 1988) and its integer counterpart (Snaider, Franklin, Strain, & George, 2013), and others. A detailed comparison between several weightless neural models can be found in Aleksander, Gregorio, França, Lima, and Morton (2009). This work adopts the WiSARD model with bleaching (Carvalho, Carneiro, França, & Lima, 2013; Grieco, Lima, Gregorio, & França, 2010), because it has the best trade-off between training agility, memory consumption and capability of avoiding saturation. Besides, the neural model proved promising in monolingual POS-tagging (Carneiro et al., 2010).

### 2.1. WiSARD model

The pioneering WiSARD $n$-tuple classifier constitutes the RAM-based neural network chosen as the basis of mWANN-Tagger architecture. Its main difference from other RAM-based models is the use of a structure called RAM-discriminator, depicted in Fig. 1(a). The discriminator receives inputs from a "retina" (a matrix of **0**s and **1**s, see Fig. 1(a)) mapped to a set of $N$ RAMs via $n$ RAM address bits and a summation device. The summation device $\Sigma$ outputs the number of RAMs that responded positively to an input pattern. A set of address bits and RAMs constitutes a **RAM node**.

Mapping of retina pixels to the RAM nodes is effected via the address bits, usually in a *pseudorandom* (invariant for a discriminator) and biunivocal fashion (one retina pixel is associated to one and only one address bit of only one RAM).