Language Independent POS-tagging Using Automatically Generated Markov Chains.

Joaquim Assunção[1], Paulo Fernandes[2], Lucelene Lopes[2]

[1] UFSM Department of Applied Computing – Santa Maria – Brazil
[2] Roberts Wesleyan College – Rochester, NY – USA
joaquim@inf.ufsm.br, fernandes_paulo@roberts.edu, lopes_lucelene@roberts.edu

Abstract

This paper proposes a method to predict word grammatical classes using automatically generated discrete-time Markov chains to model typical sentences. Such method advantage relies on the availability of input resources needed to build an efficient and effective solution to virtually any language, dialect, or domain lingo. One of the main advantages of the proposed method is its simplicity when compared to other sophisticated approaches based on Hidden Markov Models or even more complex formalisms. The proposed method is instantiated to an example and we show that the achieved efficiency and effectiveness bring advantages to traditional similar solutions.

1 Introduction

Part-Of-Speech (POS) tagging is a very basic task for all Natural Language Processing (NLP) techniques based on linguistic approach. However, only well resourced languages have very effective solution for POS tagging [14]. This fact makes statistical approaches very popular to less resourced languages [6, 13] or even domain lingo [17]. Acquiring information on informal texts, as social networks chats and comments, the use of a tool independent of formal language definition is even more interesting, since it can help to tackle an open NLP problem.

In contrast with the difficulty to find out structured knowledge for less resourced and informal languages, the Internet offers abundant unstructured material (texts) in every language and dialect. Consequently, a procedure capable to produce structured knowledge from textual sources would cope with such limitations.

As for software tools, the basic modules needed for a POS-tagger are a dictionary retriever (to identify possible grammatical classes for known words) and a predictor (to disambiguate words that can be employed with more than one role, or guess the role for unknown words). The dictionary retrieving task for any language can be solved very efficiently by the use of decision diagrams [12], and even multilingual dictionary retrievers can be implemented very efficiently [3].

The gap between a dictionary retriever and a POS-tagger is essentially the syntactic disambiguation task. In the context of this paper we limit ourselves to consider only a shallow POS-tagger assigning a word class to each word, but the methods and techniques discussed in this paper can be applied to even more sophisticated POS-tagger without any loss of generalization.

Our goal is to provide an effective grammatical class predictor to words inside a sentence previously tagged with an efficient dictionary retriever. In order to do so, we propose the construction of Markov chains [16] to describe typical phrases and use the transient analysis of such chains to predict the grammatical class of each word. These typical phrases can be viewed as a training set for the POS-tagger. Consequently, the produced Markov chains can be viewed as a model of the language patterns, *i.e.*, the structured knowledge of the target language. The enlargement of scope of POS-taggers would benefit several NLP tasks like term extraction [9, 10], ontology processing [5], and textual data mining [1], to cite a few.

Our proposed method fits into a POS-tagging landscape as simpler than sophisticated approaches based on Data mining approaches as Support Vector Machines - SVM [4], or inference models as Conditional Random Fields - CRF [8]. Nevertheless, our method is more flexible than traditional linguistic approaches heavily anchored in specific languages [2, 15].

2 Basic Tools

To better understand this paper proposed method, this section presents brief descriptions of Markov chains and Decision Diagrams-based dictionary retrievers.

2.1 Markov Chains Markov chains is a formalism to represent discrete state models as, in our case, a finite

*DOI reference number: 10.18293/SEKE2019-097

automaton where the transitions are fired by the occurrence of a stochastic process [16]. Although there is a very large variety of Markovian models the plain discrete-time Markov chains are sufficient to express a model as a set of states and transitions among them associated to known probabilities. To exemplify such chains, as will be used in the context of this paper, Figure 1 depicts a simple chain with three states: Sunny, Cloudy, and Rainy.

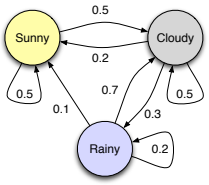

Figure 1: Simple Markov chain example.

Formally, we denote a DTMC by a matrix P where the element $p_{(i,j)}$ corresponds to the probability of leaving state i towards state j, *e.g.*, in the chain of Fig. 1, $p_{(\text{Rainy,Cloudy})} = 0.7$ and $p_{(\text{Sunny,Rainy})} = 0.0$. For a model like this it is possible to make simple predictions, as for example, if a unknown day was preceded by a Sunny day and succeeded by a another Sunny day, it is possible to analyze all three possible intermediate states: Sunny-Sunny-Sunny ($0.5 \times 0.5 = 0.25$); Sunny-Cloudy-Sunny ($0.5 \times 0.3 = 0.15$); and Sunny-Rainy-Sunny ($0.0 \times 0.1 = 0.0$). Since all three possibilities sums up 0.4, it is possible to say that the intermediate day was: Sunny with probability 0.625 ($\frac{0.25}{0.4}$) or Cloudy with probability 0.375 ($\frac{0.15}{0.4}$).

2.2 Diagram Decision-based Dictionary Retriever The use of decision diagrams to recognize words, and associate its word class (or possible classes), can be very efficient and very effective as proposed in the WAGGER software tool [3]. According to this work, a decision diagram structure holding a multilingual dictionary with English and Portuguese words (a little more than one million words) can be used to tag possible word classes to large corpora (for instance, two million words) in less than three seconds using a personal machine. Such performance leads us to employ WAGGER for all experiments in our paper.

The technology behind such a dictionary retriever are Multi-Terminal Multi-valued Decision Diagrams [7]. Such structures are not only very effective to provide a fast recovery using small amounts of memory, but they also provide flexible structures that can be enhanced, for instance, by including new dictionary words.

The input of WAGGER is a dictionary in textual format, *i.e.*, a list of words with their possible word classes. Then, it generates a MTMDD structure that can acts as dictionary retriever for a list of sentences that are annotated with the possible word classes of each sentences' word.

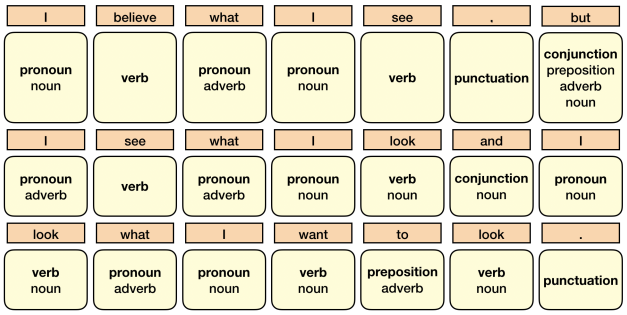

Figure 2: WAGGER output for an example sentence.

The WAGGER output is all words of each sentence followed by the possible word classes. For example, Figure 2 presents the output of WAGGER using our English dictionary [3] for the sentence "I believe what I see, but I see what I look and I look what I want to look". In this figure, the correct word class is indicated in bold. For instance, word "and" plays the role of conjunction in this sentence, but according to the dictionary "and" can also be employed as a noun.

In order to formalize the WAGGER output we will denote by w_i^s the i-th word in sentence s, and $C(w_i^s)$ the set of all possible word classes of word w_i^s. Note that the set $C(w_i^s)$ of a given word is independent of the sentence in which the word is (s). For instance, the word "and" (the thirteenth word of the example sentence) is formally defined by:

$$w_{13}^s = \text{"and"} \quad C(w_{13}^s) = \{\text{conjunction}, \text{noun}\}$$

3 Proposed Method

The proposed method follows the general idea of train and test, since it starts with a set of sentences manually tagged (training set), and only after we attempt to disambiguate word tags for sentences tagged by WAGGER (testing set). Figure 3 depicts the proposed method with the training task to construct the Markov chains, and the testing task to disambiguate WAGGER multiple or absent tags. It is important to notice that the training task needs a supervised action: a human made annotation, required according to the language of the target corpus.. All other tasks, including the annotation made by WAGGER, are fully automated, and, as exemplified in Section 4, memory and time efficient.

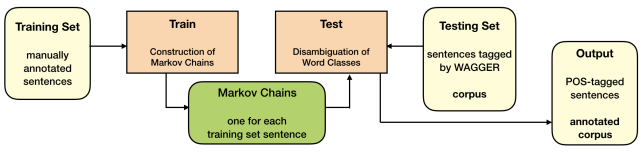

Figure 3: Proposed method.

3.1 Training Task Giving annotated sentences as the example in Figure 4, we construct a Markov chain for each sentence considering the sequences of word classes found in the sentence. This process corresponds to the capturing of an approximative model of the patterns of word class sequences, having each example sentence as an instance of the language usage.

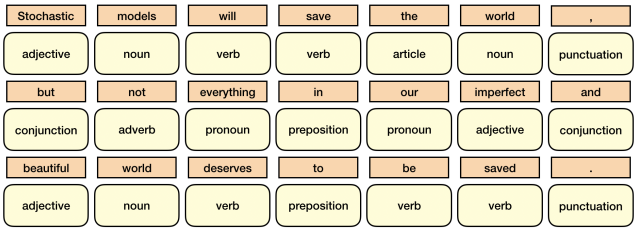

Figure 4: Manually annotated sentence used as training.

Basically, the Markov chain created has all word classes as the chain states, and every exiting arc represents a possible succession of words classes. For instance, the adjectives (*adj*) are succeeded twice by a noun (*n*) and once by a conjunction (*conj*). Hence, the *adj* state has transition towards *n* with probability $\frac{2}{3}$ and a transition towards *conj* with probability $\frac{1}{3}$. Repeating the same procedure for all word classes creates the Markov chain depicted in Figure 5.

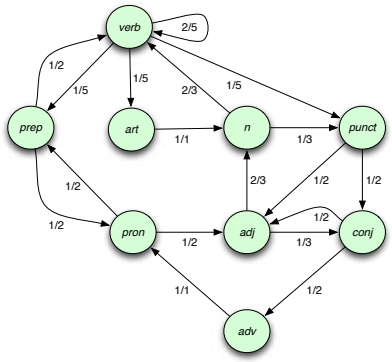

Figure 5: Markov chain for sentence in Figure 4.

This same process of creation of Markov chain is repeated for a certain number of sentences. For example, Figure 2 sentence correctly annotated would produce the Markov chain depicted in Figure 6.

3.2 Disambiguation using Markov Chains The disambiguation process is made through the estimation of probabilities of neighbors words according to the constructed chains. Every word that needs disambiguation is analyzed to all possible word classes, *i.e.*, we compute the probability of its immediate predecessor and successors according to each chain.

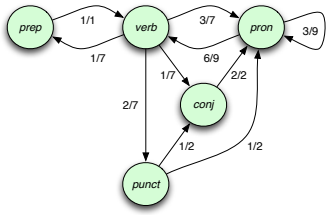

Figure 6: Markov chain for sentence in Figure 2.

Formally, a word w_i in sentence s, has its probability of being of word class c computed according its predecessor (Eq. 3.1) and successor (Eq. 3.2) to chain m respectively by:

$$\overleftarrow{c}^{(m)}_{w_i^s} = \frac{\sum_{k \in C(w_{i-1})} p^{(m)}_{[k(w^s_{i-1}),c(w^s_i)]}}{\mid C(w^s_{i-1}) \mid} \tag{3.1}$$

$$\overrightarrow{c}^{(m)}_{w_i^s} = \frac{\sum_{k \in C(w_{i+1})} p^{(m)}_{[c(w^s_i),k(w^s_{i+1})]}}{\mid C(w^s_{i+1}) \mid} \tag{3.2}$$

where $p^{(m)}_{[x,y]}$ is the probability of leaving state x and going to state y in the Markov chain m.

The overall probability of a word w_i in sentence s be of class c according to all training chains of a set M is given by:

$$\widehat{c}_{w_i^s} = \frac{\sum_{m \in M} \overleftarrow{c}^{(m)}_{w_i^s} + \overrightarrow{c}^{(m)}_{w_i^s}}{2 \mid M \mid} \tag{3.3}$$

Hence, the disambiguation of a word w_i^s will be made by choosing the word class $c \in C(w_i)$ with the maximum value $\widehat{c}_{w_i^s}$.

For example, considering the sentence "*The old book is dusty and black.*", the WAGGER annotation is depicted in Figure 7. In this sentence the words "*book*" - w_3^s (either a noun or a verb) and "*black*" - w_7^s (either an adjective, a noun or a verb) need disambiguation.

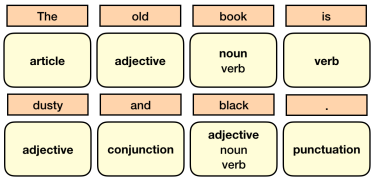

Figure 7: Example sentence to disambiguate.

Using the first chain (Figure 5), let us call it m_1, we observe that the probability of a noun be preceded by an adjective ("*old*" precedes "*book*") is equal to $\frac{2}{3}$, and since "*old*" can only be an adjective, *i.e.*, $C(\text{"old"}) = \{\text{adjective}\}$,

we express formally:

$$\overleftarrow{n}^{(m_1)}\text{“book”} = \frac{p^{(m_1)}_{[adj,n]}}{\mid C(\text{“old”}) \mid} = \frac{0.6667}{1} = 0.6667$$

We also observe that the probability of a noun to precede a verb (“*book*” precedes “*is*”) is also equal to $\frac{2}{3}$, and since “*is*” can only be a verb, *i.e.*, $C(\text{“is”}) = \{\text{verb}\}$, we express formally:

$$\overrightarrow{n}^{(m_1)}\text{“book”} = \frac{p^{(m_1)}_{[n,verb]}}{\mid C(\text{“is”}) \mid} = \frac{0.6667}{1} = 0.6667$$

Analogously, repeating the computations for the second chain (Figure 6), let us call it m_2, we obtain zero values, since there is no word class noun in it, formally:

$$\overleftarrow{n}^{(m_2)}\text{“book”} = \frac{p^{(m_2)}_{[adj,n]}}{\mid C(\text{“old”}) \mid} = \frac{0.0}{1} = 0.0$$

$$\overrightarrow{n}^{(m_2)}\text{“book”} = \frac{p^{(m_2)}_{[n,verb]}}{\mid C(\text{“is”}) \mid} = \frac{0.0}{1} = 0.0$$

Computing the overall probability for “*book*” be a noun considering the two Markov chains ($M = \{m_1, m_2\}$) will be formally:

$$\widehat{n}\text{“book”} = \frac{\overleftarrow{n}^{(m_1)}\text{“book”} + \overrightarrow{n}^{(m_1)}\text{“book”} + \overleftarrow{n}^{(m_2)}\text{“book”} + \overrightarrow{n}^{(m_2)}\text{“book”}}{2 \mid M \mid}$$

$$\widehat{n}\text{“book”} = 0.3333$$

Analogously, computing the probability of “*book*” being a verb in the sentence of Figure 7, we formally obtain:

$$\widehat{verb}\text{“book”} = \frac{0.4}{4} = 0.1$$

As result, we conclude (correctly) that “*book*” must be tagged as a noun (probability 0.33), and not a verb (probability 0.1). Analogously, we conclude (also correctly) that “*black*” must be considered an adjective, since:

$$\widehat{adj}_{w_7^s} = 0.13 \quad \widehat{n}_{w_7^s} = 0.08 \quad \widehat{verb}_{w_7^s} = 0.12$$

4 Application Example

We employ the proposed method in a larger scale to illustrate the efficiency and effectiveness benefits of our approach. To do so, we choose the following test bed:

- Two POS-tagger to Portuguese: PALAVRAS [2] and LX-Center Suite [15];
- A Portuguese dictionary and the WAGGER dictionary retriever [3];
- Fifty Brazilian Portuguese manually annotated sentences to be used as training set;
- One hundred and twenty eight Brazilian Portuguese manually annotated sentences to be used as test set.

Applying our proposed method to the sentences of the training set has produced 50 discrete-time Markov chains. As mentioned, these chains can be considered an approximative model of Brazilian language usage in terms of word class sequences. The choice of these sentences is not particularly planed, since these sentences were randomly taken from domain corpora composed of articles and other academic texts of several scientific domains [11].

After that, 128 sentences were also randomly chosen from the same corpora in order to test our proposed method. The only concern choosing these 128 test sentences were not to choose a sentence that was already used as training set. For both training and testing sets, the manual annotation was performed by two linguist specialists.

The application of our method to the training set delivered 2,455 correctly classified words from a total of 2,958 words from the testing set. This result corresponds to a 83.0% precision, which is comparable to the precision of word class tagging made by PALAVRAS (86.1%) and LX-Center (88.8%).

In terms of efficiency, however, our proposed method is far more impressive, while the time spent for PALAVRAS and LX-Center were measured in terms of minutes, the performance of our method took less than one second. Table 1 summarizes the effectiveness (precision) and efficiency (time to tag) of the prototype implementing our proposed method in comparison with PALAVRAS and LX-Center running on a portable machine with i7 2.2 GHz processor, 8 Gbytes memory. This table results indicate a reasonable effectiveness and an extraordinary efficiency improvement. Such performance let us believe that the proposed method is a worthy option for real-time POS-tagger.

Table 1: Overall comparative performance of the proposed method prototype.

	correct words	precision	time to tag
our method	2,455	83.0%	<1 sec.
PALAVRAS	2,548	86.1%	2.3 min.
LX-Center	2,627	88.8%	1.5 min.

Finally, it is important to remember that unlike PALAVRAS and LX-Center rigid approaches, our method is based on train and test phases. Therefore, a careful choice of a training set may improve the effectiveness of our method.

5 Final Considerations

This work lays down the basic ideas to a novel approach to predict grammatical classes from corpora. This novel idea is much simpler than sophisticated HMM, CRF or SVM. Nevertheless, our method is more prone to evolution than traditional rigid approaches available at existing POS-taggers like PALAVRAS and LX-Center Suite.

The examples presented here embody the core concept of our technique, yet the results delivered a reasonable precision and very impressive efficiency. We believe that a trusted Markov chain database can make a difference in this kind of scenario. Thus, we are currently working on the construction of a solid training databases for several formal and informal languages in order to promote a broader test. For instance, we are building an additional lexicon of informal words and abbreviations usually employed in informal chats to try to grasp a reasonable POS-tagging for such a complex dialect.

We also expect to refine the disambiguation phase by considering more information than just the predecessor and successor of the target word. It is important to call the reader attention that the number of Markov chains is not an obstacle to the efficiency, since the weight computed according to each Markov chain may be computed independently, *i.e.*, a parallel implementation of the proposed method would scale very well to a very large number of Markov chains.

Despite those promising future works, the initial results encourage this line of research. In fact, our approach seems to gather the flexibility of sophisticated learning methods, the easiness of development, and computational efficiency. Therefore, our proposed method can bring effective and efficient POS-tagging to virtually any language, dialect or domain lingo.

References

[1] J. Assunção, P. Fernandes, L. Lopes, and S. Normey, *Distributed Stochastic Aware Random Forests - Efficient Data Mining for Big Data*, in 2013 IEEE Conference on Big Data, San Clara, CA, USA, June 2013, IEEE Computer Society, pp. 484–485.

[2] E. Bick, *The parsing system PALAVRAS: automatic grammatical analysis of portuguese in constraint grammar framework*, PhD thesis, Arhus University, Arhus, Danemark, 2000.

[3] P. Fernandes, L. Lopes, C. A. Prolo, A. Sales, and R. Vieira, *A fast, memory efficient, scalable and multilingual dictionary retriever*, in Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), N. Calzolari, ed., Istanbul, Turkey, may 2012, European Language Resources Association (ELRA), pp. 2520–2524.

[4] J. Giménez and L. Màrquez, *Svmtool: A general pos tagger generator based on support vector machines.*, in LREC, European Language Resources Association, 2004.

[5] R. Granada, L. Lopes, C. Ramisch, C. Trojahn, R. Vieira, and A. Villavicencio, *A comparable corpus based on aligned multilingual ontologies*, in Proceedings of the First Workshop on Multilingual Modeling, 2012, pp. 25–31.

[6] F. M. Hasan, N. Uz Zaman, and M. Khan, *Comparison of different pos tagging techniques (n-gram, hmm and brill's tagger) for bangla*, in Advances and Innovations in Systems, Computing Sciences and Software Engineering, K. Elleithy, ed., Springer Netherlands, 2007, pp. 121–126.

[7] T. Kam, T. Villa, R. K. Bryaton, and A. Sangiovanni-Vincentelli, *Multi-valued decision diagrams: theory and applications*, Multiplie-Valued Logic, 4 (1998), pp. 9–62.

[8] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, *Conditional random fields: Probabilistic models for segmenting and labeling sequence data*, in Proceedings of the Eighteenth International Conference on Machine Learning, ICML '01, San Francisco, CA, USA, 2001, Morgan Kaufmann Publishers Inc., pp. 282–289.

[9] L. Lopes, P. Fernandes, and R. Vieira, *Estimating term domain relevance through term frequency, disjoint corpora frequency - tf-dcf*, Knowledge-Based Systems, 97 (2016), pp. 237 – 249.

[10] ———, *Exato – high quality term extraction for portuguese and english*, in Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence, WI 2016, San Francisco, CA, USA, Oct. 2016, IEEE Computer Society, pp. 531–536.

[11] L. Lopes and R. Vieira, *Building domain specific parsed corpora in portuguese language*, in Proceedings of the X National Meeting on Artificial and Computational Intelligence (ENIAC), 2013, pp. 1–12.

[12] C. L. Lucchesi and T. Kowaltowski, *Applications of finite automata representing large vocabularies*, Software: Practice and Experience, 23 (1993), pp. 15–30.

[13] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*, Cambridge University Press, Cambridge, 2008.

[14] F. Segond, A. Schiller, G. Grefenstette, and J. Chanod, *An experiment in semantic tagging using hidden markov model tagging*, in ACL/EACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications, 1997, pp. 78–81.

[15] J. Silva, A. Branco, S. Castro, and R. Reis, *Out-of-the-box robust parsing of portuguese*, in PROPOR 2010, 2010, pp. 75–85.

[16] W. J. Stewart, *Probability, Markov Chains, Queues, and Simulation*, Princeton University Press, USA, 2009.

[17] Y. Tsuruoka, Y. Tateishi, J.-D. Kim, T. Ohta, J. McNaught, S. Ananiadou, and J. Tsujii, *Developing a robust part-of-speech tagger for biomedical text*, in Advances in Informatics, P. Bozanis and E. Houstis, eds., vol. 3746 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, 2005, pp. 382–392.