Part of Speech Tagging Using Hidden Markov Model and Viterbi Algorithm

Auhood Hadi Jabbar[*]

[*] University of Thi Qar-College of Education-Dept. of computer science

Abstract

Part-of-Speech tagging is the process of assigning parts of speech (or other classifiers) to the words in a text. In this research , we introduce a tagging algorithm for English sentences based on Viterbi Algorithm and Hidden Markov Model. In traditional part-of-speech taggers, the calculations requires $(2T+1)*N^{T+1}$ multiplications if we used the direct computation. After enhancing the method of calculating, we get the optimal tags sequence by just $2N^2T$ multiplications.

Keywords

Part-Of-Speech (***POS***), Markov Chain Model (***MCM***), Hidden Markov Model (***HMM***), Variable Memory Markov (***VMM***).

1. Introduction

1.1 Hidden Markov Model

We deal with a stochastic or random process which is characterized by the rule that only the current state of the process can influence the choice of the next state. It means the process has no memory of its previous states. Such a process is called a Markov process after the name of a prominent Russian mathematician Andrey Markov (1856-1922). If we assume that the process has only a finite or countable set of states, then it is called a Markov chain. Markov chains can be considered both in discrete and continuous time, but we shall limit our tutorial to the discrete time finite Markov chains.

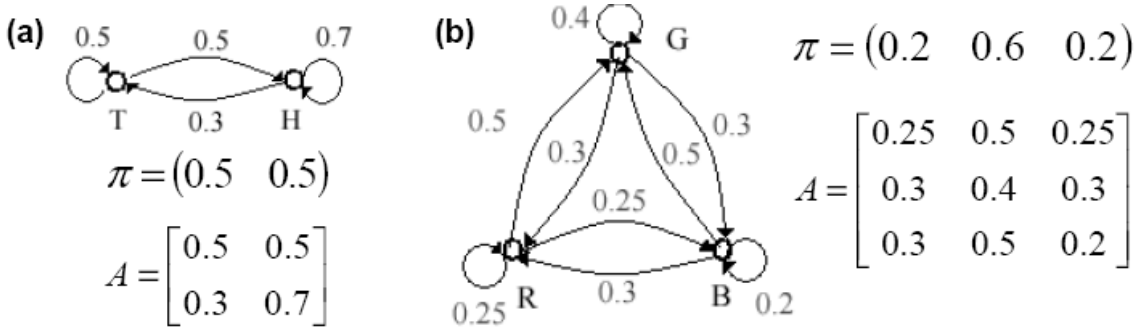

Figure (1) Markov chain models for a biased coin (a), and the paving model.

Such chains can be described by diagrams (Figure 1, extracted from [1]). The nodes of the diagram represent the states (in our case, a state corresponds to a choice of a tile of a particular color) and the edges represent transitions between the states. A transition probability is assigned to each edge. The probabilities of all edges outgoing from a node must sum to one. Beside that, there is an initial state probability distribution to define the first state of the chain [2].

$$M=\langle \pi, A \rangle,\ \pi=(\pi_1, \pi_2, \ldots, \pi_N)\ \ A=\{a_{ij}\}_{i,j=1 \text{ to } N} \qquad \ldots(1)$$

$$P(q_{k+1} \mid q_1, \ldots, q_k) = P(q_{k+1} \mid q_k) \qquad \ldots (2)$$

1.2 Tagging

Part of Speech (POS) tagging is the task of labeling each word in a sentence with its appropriate syntactic category called part of speech. Part of speech tagging is a very important preprocessing task for language processing activities. This helps in doing deep parsing of text and in developing information extraction systems, semantic processing etc. Part of speech tagging for natural language texts are developed using linguistic rules, stochastic models and a combination of both [3, 4]. Many words in English have several parts of speech. For example, "book" is used as a noun in "She read a book" and as a verb in "She didn't book a trip" [5]. Therefore, a correct syntactic classification of words in context is important for most syntactic and other higher level processing of natural language text [5, 6]. The notation that we use in this work is summarized below:

w_i	the word at position ***i*** in the corpus.
t_i	the tag of word.
$w_{i,i+m}$	the words occurring at positions ***i*** through ***i*** +m.
$t_{i,i+m}$	the tags $t_i \ldots t_{i+m}$ for $w_i \ldots w_{i,i+m}$.

w^l	the l^{th} word in the lexicon.
t^j	the tag j in the tag set.
$C(w^l)$	the number of occurrences of w^l in the training set.
$C(t^j)$	the number of occurrences of t^j in the training set.
$C(t^j, t^k)$	the number of occurrences of t^j followed by t^k.
$C(w^l, w^j)$	the number of occurrences of w^l that are tagged as w^j.
T	number of tags in tag set.
W	number of words in the lexicon.
n	sentence length.

So the equation for determining the optimal tags for a sentence is:

$$t_{1,n} := \arg\max_{1,n} P(t_{1,n} \mid w_{1,n}) = \prod_{i=1}^{n} P(w_i \mid t_i) P(t_i \mid t_{i-1}) \quad \ldots (3)$$

1 for all tags t^j do
2 for all tags t^k do
3 $P(t^k \mid t^j) := \frac{C(t^j \mid t^k)}{C(t^j)}$
4 end
5 end
6 for all tags t^j do
7 for all words w^l do
8 $P(w^l \mid t^j) := \frac{C(w^l \mid t^j)}{C(t^j)}$
9 end
10 end

In traditional part-of-speech taggers, the calculations requires $(2T+1)*N^{T+1}$ multiplications. In this reserach, we introduce a tagging algorithm for English sentences based on Viterbi Algorithm and Hidden Markov Model, in which we tried to enhance the manner of calculations to decrease the number of multiplications.

2. Related Works

M. Andrews and G. Vigliocco [7] have described a model that learns semantic representations from the distributional statistics of language Nizar Habash and Owen Rambow present an approach to using a morphological analyzer for tokenizing and morphologically tagging (including part-of-speech tagging) Arabic words in one process [8]. Dipanjan Das and Slav describe a novel approach for inducing unsupervised part-of-speech taggers for languages that have no labeled training data, but have translated text in a resource-rich language [9]. Hinrich Schiitze and Yoram Singer present a new approach to disambiguating syntactically ambiguous words in context, based on Variable Memory Markov (VMM) models [10]. Levent Altunyurt, Zihni Orhan, Tunga Güngör present a composite part of speech tagger for Turkish which combines the rule-based and statistical approaches [11]. Asif Ekbal, Rejwanul Haque, and Sivaji Bandyopadhyay presents a POS tagger for Bengali using the statistical Maximum Entropy (ME) model. The system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the various POS classes [12].

3. The Proposed Method

We could evaluate equation (3) for all possible taggings of a sentence of length ***n***, but that would make tagging exponential in the length of the input that is to be tagged. An efficient tagging algorithm is the Viterbi algorithm. The Viterbi algorithm has three steps: (i) initialization, (ii) recursion, and (iii) termination. We compute two functions $\gamma_i(j)$, which gives us the probability of being in state ***j*** tag at word ***i,*** and (j), which gives us the most likely state (or tag) at word ***i*** given that we are in state ***j*** (=tag ***j***) at word ***i***, and the function $\psi_{i+1}(j)$, which gives us the most likely state (or tag) at word ***i*** given that we are in state ***j*** at a word ***i+1***. Throughout, we will refer to states as tags in this section because the states of the model correspond to tags. The initialization step is to assign probability 1.0 to the tag HASH #. We start each sentence with a HASH and end it with a HASH also. That is we assume that sentences are delimited by HASHs. Then, we implemented the following algorithm to tag the sentences we choose randomly, as mentioned previously. After enhancing the method of calculating, we get the optimal tags sequence by just $2N^2T$ multiplications (see algorithm below).

3.1 The Algorithm

Below, we present the Viterbi algorithm that we used to calculate the optimal tag sequence path:

1 Input: Sentence of length n
2 $\gamma_1(HASH) = 1.0$
3 $\gamma_1(t) = 1.0$ for $(t \neq HASH)$
4 for i:= 1 to n do
5 for all tags t^j do
6 $\gamma_{i+1}(t^j) := \max_{1 \le k \le T}[\gamma_i(t^k) * P(w_{i+1} | (t^j)) * P((t^j) | (t^k))]$
7 $\psi_{i+1}(t^j) := \arg\max_{1 \le k \le T}[\gamma_i(t^k) * P(w_{i+1} | (t^j)) * P((t^j) | (t^k))]$
8 end
9 end
10 $X_{i+1} := \arg\max_{1 \le k \le T} \gamma_{n+1}(\gamma(j))$
11 for j:= n to 1 do
12 $X_j := \psi_{j+1}(X_{j+1})$
13 end
14 $P(X_1, X_2, ..., X_n) = \max_{1 \le k \le T}[\gamma_{n+1}(t^j)$

3.2 The Model

Depending on the data chosen for training the model, we build Hidden Markov model (HMM) shown in Figure (2) below to tag each word within each sentence. We have selected 100 sentences randomly and calculate the ratios of each word tagging by hand (see theFigure (3), sentences chosen to represent the sample). After setting the appropriate initialization parameters of the model we train it on these sentences. All suffixes to verbs are remove (ing, ed, … etc) to make the tagging process easier. Then we test the model against sentences similar to training data. We have reached to results explained in tables below (see the Figure(4)).

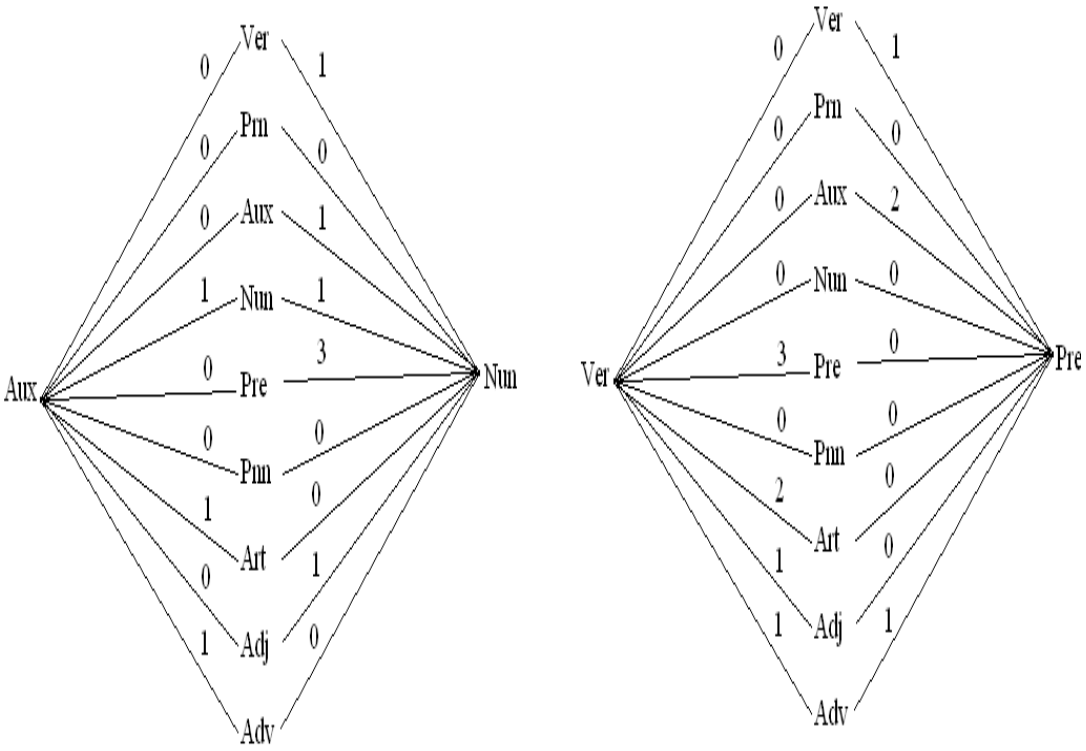

Figure (2) Hidden Markov Model for Sentence Tagging.

1 #I am going to youth center#
2 #Bertrand wrote a book#
3 #Terry writes a program#
4 #The main objective of an accounting#
…….
55 #I went to the bank#
56 #He always read about wars#
57 #I will frequently refer to variable names
……
100 # To turn, bank the airplane#

Figure (3) Sentences chosen to represent the sample.

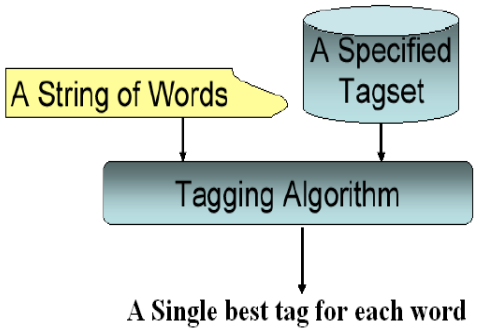

Figure (4) the model of training.

4. Results

We implemented this technique for the Viterbi algorithm HMM taggers. From our training data, we were able to extract data for on the order of 100 unique unambiguous tag sequences which were then be used for better initializing the state transition probabilities. As shown in Table 2, this method improved tagging accuracy of the Viterbi algorithm HMM tagger over traditional simultaneous HMM training:

Table (1)

	#	Ver	Prn	Aux	Nun	Pre	Pnn	Art	Adj	Adv
I	0	0.0	0.6726	0.0	0.01392	0.0	0.0	0.0	0.0	0.0
Am	0	0.0	0.0	0.881	0.0	0.0	0.0	0.0	0.0	0.0
Go	0	0.8234 2	0.0	0.0	0.0	0.0	0.0	.00	0.0	0.0
To	0	0.0	0.0		0.0	0.0	0.999	0.0	0.0	0.0
youth	0	0.0	0.0	0.967	0.0	0.0	0.0	0.0	0.0	0.0
center	0	0.0	0.0	0.0	0.911	0.0	0.0	0.0	0.0	0.0
#	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Bertran	0	0.0	0.0	0.0	0.9993	0.	0.0	0.0	0.0	0.0

d						0				
wrote	0	0.8731 1	0.0	0.0	0.0	0. 0	0.0	0.0	0.0	0.0
A	0	0.0	0.0	0.0	0.0	0. 0	0.0	00.95 4	0.0	0.0
book	0	0.0	0.0	0.0	0.9330	0. 0	0.0	0.0	0.0	0.0
… …		… …	… …	… …	… …	… …	… …	… …	… …	… …

5. Conclusion

Viterbi algorithm HMM tagger method improved tagging accuracy, in which we get the optimal tags sequence by just $2N^2T$ multiplications, over the traditional simultaneous HMM training, in which, the calculations requires $(2T+1)*N^{T+1}$ multiplications. We have selected 100 sentences randomly and calculate the ratios of each word tagging by hand, so in future we recommend using large corpora of sentences instead of using just hundred of them.

References

[1] V. A. Petrushin, **"**Hidden Markov Models: Fundamentals and Applications Part 1: Markov Chains and Mixture Models", the Online Symposium for Electronics Engineer, (2000).

[2] W.J. Stewart, "Introduction to the Numerical Solutions of Markov Chains", Princeton University Press, (1994).

[3] Cutting, D., Kupiec, J., Pedersen, J., Sibun, P.: "A practical part-of-speech tagger", In: Proceedings of the Third Conference on Applied Natural Language Processing, pp 133-140, (1992).

[4] Merialdo, B.: "Tagging english text with a probabilistic model", Comput. Linguist. 20(2) pp 155-171, (1994).

[5] Hinrich Schiitze and Yoram Singer, PART-OF-SPEECH TAGGING USING A VARIABLE MEMORY MARKOV MODEL, unpublished (1995).

[6] S. Klein and R. Simmons, "A computational approach to grammatical coding of English words", JACM, 10: pp 334–337, (1963).

[7] Mark Andrews, Gabriella Vigliocco, "The Hidden Markov Topic Model: A Probabilistic Model of Semantic Representation", Topics in Cognitive Science (2) pp 101–113 (2010).

[8] Nizar Habash and Owen Rambow, "Arabic Tokenization, Part-of-Speech Tagging and morphological Disambiguation in One Fell Swoop", Proceedings of the 43rd Annual Meeting of the ACL, pages 573–580, Ann Arbor, (2005).
[9] Dipanjan Das and Slav Petrov,"Unsupervised Part-of-Speech Tagging with Bilingual Graph-Based Projections" Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, Portland, Oregon, pp 600–609,, (2011).
[10] Hinrich Schiitze and Yoram Singer, PART-OF-SPEECH TAGGING USING A VARIABLE MEMORY MARKOV MODEL, unpublished (1995).
[11] Levent Altunyurt, Zihni Orhan, Tunga Güngör, "A Composite Approach for Part of Speech Tagging in Turkish", International Scientific Conference Computer Science, (2006).
[12] Asif Ekbal, Rejwanul Haque, and Sivaji Bandyopadhyay, "Maximum Entropy Based Bengali Part of Speech Tagging", Advances in Natural Language Processing and Applications Research in Computing Science 33, pp. 67-78, (2008).