Part Of Speech Tagging Using Statistical Approach for Guajarati Text

ManishaPrajapati, ArchitYajnik
Ph.D Scholar, Gujarat Technological University

***Abstract*—Part of Speech Tagging has always been a challenging task in the era of Natural Language Processing. This article presents POS tagging for Nepali text using Hidden Markov Model and Viterbi algorithm. From the Nepali text annotated corpus training and testing data set are randomly separated. Both the methods are employed on the data sets. Viterbi algorithm is found to be computationally faster and accurate as compared to HMM. The accuracy of 92.87% is achieved using Viterbi algorithm. Error analysis where the mismatches took place is elaborately discussed.**

Keywords—Hidden Markov Model, Viterbi Algorithm, POS Tagging, Natural Language Processing.

I. INTRODUCTION

GUJRATI language is one of the most spoken languages ingujrat,. POS tagging plays a pivotal role in the development of Natural Language Processing applications like Parser and Morphological analyzer. Ample amount of articles are available in the literature on POS tagging task for Indian languages like Hindi, Marathi, Odia, Panjabi etc. but seldom efforts [1, 2] are carried out as far as Gujrati text is concern. Significant literature survey on POS tagging for Indian languages is mentioned in [3]. A remarkable work for Annotating Corpora for Indian languages is made available in "AnnCorra: Annotating Corpora, Guidelines for POS and Chunk Annotation for Indian Languages" by AksharBharati, DiptiMisra Sharma, Rajeev Sangal et al., [4]. A hybrid approach demonstrated in [1] is applying Grammar rules on Statistical approaches for POS tagging task and achieved 93.15% of accuracy whereas [2] took the Support Vector Machine approach to perform this task and achieved around 90% of accuracy. Current approach is lucid as it does not incorporate any grammar rules, it is purely based on statistical approach. Grammar rules may be applied to develop Natural Language Processing applications like Morphological Analysis, Parsing etc. based on the output of the tagger.

The article is divided in 6 sections. After introducing the POS tagging for Indian languages in section 1, the statistical techniques viz. Hidden Markov Model (HMM) and Viterbi algorithm based on HMMare briefed in sections 2 and 3. The methodology and experimental details is covered in section 4 followed by conclusion in section 5. At the end acknowledgement followed by references are presented.

II. HIDDENMARKOV MODEL

POS tagging plays a vital role in the development of the applications of Natural Language Processing in which inputs are the words and outputs are the corresponding tags i.e. Selecting the tag sequence of length *n* that is the most probable given the input word sequence

a) Probabilistic generative model for sequences.

b) Assume an underlying set of hidden (unobserved) states in which the model can be (e.g. parts of speech).

c) Assume probabilistic transitions between states over time (e.g. transition from POS to another POS as sequence is generated).

d) Assume a probabilistic generation of tokens from states (e.g. words generated for each POS).

e) Assume the current event depends only up on the previous event (Bigram Model).

Let x_i and y_i be the ith word and tag respectively forms an order pair (x_i, y_i) where$i = 1, 2, \ldots n$.

Let $p(x_1,x_2,x_3, \ldots\ldots x_n \;\; ,y_1,y_2,y_3 \ldots\ldots y_n)$ be the joint probability for any set of words $x = (x_1,x_2,x_3, \ldots\ldots x_n)$ and tag sequence$y = (y_1,y_2,y_3 \ldots\ldots y_n)$ of the same length.

Then most likely tag sequence for *x* is

$\text{argmax}_{y_1,y_2,y_3 \ldots\ldots y_n}\ p(y_1,y_2,y_3 \ldots\ldots y_n / x_1,x_2, x_3, \ldots\ldots x_n)$
which equals to

$$\frac{p(x_1,x_2, x_3,\ldots\ldots x_n \;/y_1,y_2,y_3 \ldots\ldots y_n).\ P(y_1,y_2,y_3 \ldots\ldots y_n)}{P(x_1,x_2,x_3 \ldots\ldots x_n)} \qquad (1)$$

The same can be written in terms of the joint probability,

$$\frac{p\ (x_1,x_2,x_3, \ldots\ldots x_n \;,y_1,y_2,y_3 \ldots}{P(x_1,x_2,x_3 \ldots\ldots x_n)}$$

Considering the first term of the numerator of (1),

$p(x_1,x_2,x_3, \ldots\ldots x_n \;,y_1,y_2,y_3 \ldots\ldots y_n)$
$= \; p(x_1,x_2, x_3, \ldots x_n \;/y_1,y_2,y_3 \ldots y_n).\ p(y_1,y_2,y_3 \ldots y_n)$

$= p(x_1/y_1,y_2, \ldots y_n)\ p(x_2/x_1,y_1,y_2, \ldots y_n)$
$\qquad p(x_3/x_2,x_1,y_1,y_2, \ldots y_n) \ldots\ldots$
$p(x_n/x_{n-1}x_{n-2} \ldots y_1,y_2, \ldots y_n)$

Using the assumptions of HMM mentioned above
$= p(x_1/y_1)p(x_2/y_2)p(x_3/y_3)$ ……. $p(x_n/y_n)$

$$= \prod_{i=1}^{n} p(x_i/y_i) \quad (2)$$

(2) is called Emission probability because it is based on the emission of each of the word and corresponding tag assigned to it. It may defer depending up on the context of the sentence.

Considering the second term of the numerator of (1),
$P(y_1, y_2, y_3 \ldots . y_n) =$
$p(y_1).p(y_2/y_1).\ p(y_3/y_1\ y_2).$ ……
$p(y_n/y_{n-1}y_{n-2}y_{n-3} \ldots . y_1).$

Using the assumptions of HMM mentioned above

$P(y_1, y_2, y_3 \ldots . y_n) =$
$= p(y_1).p(y_2/y_1).p(y_3/y_2) \ldots p(y_n/y_{n-1})$

$$= \prod_{i=0}^{n} p(y_{i+1}/y_i) \ (3)$$

(3) is called Transition probability because it is based on the transition of the states (i.e. tags).

The denominator of (1) is fixed for any word sequences so that can be neglected.

which constitutes the transition probability matrix.
Using (2) and (3) in (1),

$$f(x_n) = arg\ max_{y_n} \prod_{i=1}^{n} p(x_i/y_i) \prod_{i=0}^{n} p(y_{i+1}/y_i) \quad (4)$$

This model is known a bigram model of HMM.The block diagram is depicted in Fig.1.

III. VITERBI ALGORITHM

It is based on Bottom-up dynamic programming approach. For an input sentence$\{x_1, x_2, \ldots, x_n\}$.

$$p(x_1, x_2, \ldots, x_n, y_1, \ldots, y_n, y_{n+1}) = \prod_{i=1}^{n} e(x_i/y_i) \prod_{i=1}^{n+1} q(y_i/y_{i-2}, y_{i-1})$$

Define n to be the length of the sequences.
Define s_k for k= -1…….n to be the set of possible tag at position k

a) $s_{-1} = s_0 = \{*\}$
b) s_k = s for k∈ $\{1 \ldots . n\}$
c) Define
$r(y_{-1}, y_0 y_1, \ldots . y_k) =$
$\prod_{i=1}^{k} q(y_i/y_{i-2}, y_{i-1}) \prod_{i=1}^{k} e(x_i/y_i)$

Define a dynamic programing table
$\pi(k, u, v) =$ maximum probabilityof tag sequences endings in tags u, v at position k
That is, {1,2….n} $s_{k-1}s_k$
$\pi(k, u, v) = max(\ y_{-1}, y_0 y_1, \ldots . y_k);$
$y_{k-1}=u;\ y_k=vr(y_{-1}, y_0 y_1, \ldots . y_k)$
For an input $x_1, x_2, x_3 \ldots . x_n$
$argmax_{y_1, y_2, y_3 \ldots . y_{n+1}}\ p(\ x_1, x_2, x_3 \ldots x_n y_1, y_2, y_3 \ldots y_{n+1})$

Where the arg max is taken all over sequences
$y_1, y_2, y_3 \ldots . y_{n+1}$

Such that $y_i \in S\ for\ i = 1,2, \ldots n\ and y_{n+1}$=STOP.
We assume that p again takes the form

$$p(x_1, x_2, \ldots, x_n, y_1, \ldots, y_n, y_{n+1}) = \prod_{i=1}^{n} e(x_i/y_i) \prod_{i=1}^{n+1} q(y_i/y_{i-2}, y_{i-1})$$

In this definition we assume that $y_0 = y_{-1}$=*, $and y_{n+1}$=STOP.

IV. EXPERIMENT AND DISCUSSION

The database is generated from NELRALEC Tagset [6] with 28 tags. A report on Gujarati Computational Grammar is made available by PrajwalRupakheti et al. [7] which contains frequently used tag set. A POS tagged sentence of Gujarati language is illustrated below:

- સપ્તપુરીઓના NNP દર્શનથી NN મળે VM છે VAUX મોક્ષ NN . PUNC
- હિંદુધર્મમાં NN તીર્થનું NN ઘણું QTF મહત્વ JJ છે VM . PUNC

4.1 Experimental Procedure for Gujarati text

Experiment is carried out using Matlab. The particulars of the inputs to the Matlab code is given below:

Total tags : 28
Database : 351 Gujrati words
Size of transition matrix : 28x 28
Size of Emission matrix: 28 x 351.

The transition and emission matrices are computed by implementing in Java, a programming language. The tables 1 and 2 depict the classification accuracy and error analysis whether the mismatch takes place respectively. The Tagwise error graph is shown in fig. 2.

Table 1.Classification accuracy

Technique	No of mismatch	Accuracy
Viterbi	25	92.87

Table 2. Error Analysis Table

ORIGINAL TAG SEQUENCE	ORIGINAL TAGS	VITERBI TAG SEQUENCE	VITERBI TAGS	POSITION WHERE ERROR TOOK PLACE	WORD
7	DMD	1	NN	5	આ
5	VM	11	VAUX	8	છે
8	CCS	17	CCD	30	કે
11	VAUX	5	VM	73	લાગી
11	VAUX	5	VM	111	રહ્યા
5	VM	11	VAUX	145	છે
5	VM	11	VAUX	154	છે
17	CCD	13	RPD	156	પણ
5	VM	11	VAUX	162	છે
7	DMD	9	DMR	164	આ
8	CCS	17	CCD	178	કે
7	DMD	9	DMR	180	આ
7	DMD	1	NN	194	એ
8	CCS	13	RPD	198	પણ
9	DMR	1	NN	200	એ
13	RPD	20	PSP	209	એટલે
1	NN	2	NNP	210	મહાજન
16	VNG	1	NN	211	કામે
11	VAUX	5	VM	213	ગયા
7	DMD	1	NN	234	એ
1	NN	5	VM	235	જગ્યા
13	RPD	20	PSP	246	એટલે
7	DMD	9	DMR	251	આ
11	VAUX	5	VM	274	ગયા
3	VNF	16	VNG	336	આવીને

V. Conclusion

Table 2 demonstrates the error analysis of the words whether mismatch takes place. In 14 cases out of 25 mismatches which is highlighted are confusing with in the same category. For ex. In the second row the original Tag is VM while the predicted tag using Viterbi algorithm is VAUX for the word "છે". Ignoring such cases only 9 places are there which the actual mismatch takes place the accuracy is achieved around 97.4 %. The accuracy can be ameliorated by taking significantly large database.

Acknowledgment

I am sincerely thankful to Dr. Samarjeet Borah, Associate Professor, Department of Computer Applications, Sikkim Manipal Institute of Technology, India for providing useful information about programing in Java. I extend my gratitude to Dr. Samar Sinha, Department of Nepali,Sikkim University, Gangtok, India for his valuable linguistic suggestions in Nepali Grammer and Annotated Corpora for POS Tagging in Nepali text. Without the project funded by Department of Science and Technology, New Delhi, this work would not have been initiated, so I express my sincere thanks to DST for providing an opportunity to work on the platform of NLP.

References

[1] PrajadipSinha et al. 2015.Enhancing the Performance of Part of Speech tagging of Nepali language through Hybrid approach, 5(5) International Journal of Emerging Technology and Advanced Engineering.
[2] TejBahadurShai et al. 2013. Support Vector Machines based Part of Speech Tagging for Nepali Text, Vol: 70-No. 24 International Journal of Computer Applications.
[3] Antony P J et al. 2011.*Parts of Speech Tagging for Indian Languages: A Literature Survey*, International Journal of Computer Applications (0975-8887), 34(8).
[4] http://ltrc.iiit.ac.in/tr031/posguidelines.pdf
[5]http://www.csjhu.edu/~langmea/resources/lecture_notes/hidden_markov_models.pdf
[6] http://www.lancaster.ac.uk/staff/hardiea/nepali/postag.php
[7]http://www.panl10n.net/english/Outputs%20Phase%202/CCs/Nepal/MPP/Papers/2008/Report%20on%20Nepali%20Computational%20Grammar.pdf

Fig. 1 Block diagram of HMM

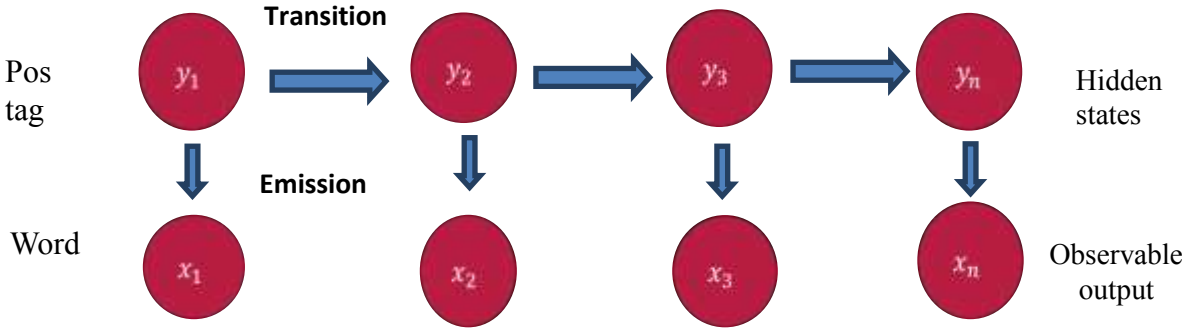

Fig. 2 Tag Wise Error Analysis For Viterbi

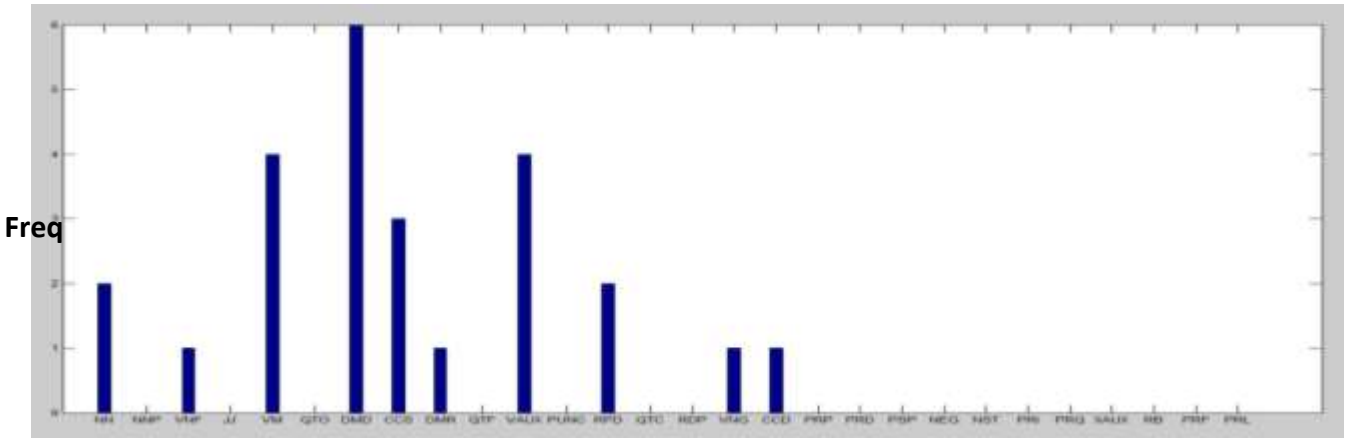

Table 3.TAGSET

tagset[0]=NN	tagset[14]=RDP
tagset[1]=NNP	tagset[15]=VNG
tagset[2]=VNF	tagset[16]=CCD
tagset[3]=JJ	tagset[17]=PRP
tagset[4]=VM	tagset[18]=PRD
tagset[5]=QTO	tagset[19]=PSP
tagset[6]=DMD	tagset[20]=NEG
tagset[7]=CCS	tagset[21]=NST
tagset[8]=DMR	tagset[22]=PRI
tagset[9]=QTF	tagset[23]=PRQ
tagset[10]=VAUX	tagset[24]=XAUX
tagset[11]=PUNC	tagset[25]=RB
tagset[12]=RPD	tagset[26]=PRF
tagset[13]=QTC	tagset[27]=PRL