World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:11, No:1, 2017

Part of Speech Tagging Using Statistical Approach for Nepali Text

Archit Yajnik

***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 methods are employed on the data sets. Viterbi algorithm is found to be computationally faster and accurate as compared to HMM. The accuracy of 95.43% 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

NEPALI language is one of the most spoken languages in Sikkim, Nepal, Bhutan, etc. 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 Nepali text is concerned. Significant literature survey on POS tagging for Indian languages is mentioned in [3]. In order to develop automatic POS tagger, either a comprehensive set of linguistically characterized rules or a large annotated corpus is required. But such rules are not generated for most of the Indian languages which can meet the requirement of an automatic POS tagger. There are two types of taggers observed in the literature viz. Rule based tagger and statistical tagger. 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 Bharati 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, Local Word Grouping (LWG), 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 I, the statistical techniques viz. Hidden Markov Model (HMM) and Viterbi algorithm based on HMM are briefed in Sections II and III. The methodology and experimental details is covered in Section IV followed by conclusion in Section V. At the end acknowledgment followed by references are presented.

Archit Yajnik is with the Sikkim Manipal Institute of Technology, Sikkim Manipal University, India (phone: 0091-96353 19656; e-mail: archit.yaajnikr@ gmail.com).

II. Hidden Markov 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. HMM is a probabilistic (nondeterministic) finite state automaton, with probabilistic outputs. Let x_i and y_i be the ith word and tag respectively forms an order pair (x_i, y_i) where $i = 1, 2, \ldots n$. For a given input sequence of words, the task of HMM tagger is to choose the tag sequence that maximizes

$$p\left(\frac{word}{tag}\right) * p\left(\frac{tag}{previous\ n\ tags}\right)$$

Consider a joint probability distribution function,

$$p(x, y) = p(y)p(x/y) \tag{1}$$

where $p(y)$ is a prior probability distribution over the tag y. $\mathrm{p(x/y)}$ is a conditional probability of yielding the input x given the tag y. By using Bayes rule to derive the conditional probability $\mathrm{p(y/x)}$ for any order pair $\mathrm{(x, y)}$,

$$\mathrm{p(y/x)} = \frac{\mathrm{p(y)p(x/y)}}{\mathrm{p(x)}} \tag{2}$$

where $p(x) = \sum_{n \epsilon y} p(x, n) = \sum_{n \epsilon y} p(n)\, p(x/n)$ (from (1)).

Our goal is to derive the relation $y = f(x)$ for the given word x as following model

$$f(x) = \arg \max_{y} \mathrm{p(y/x)}$$

From (2),

$$f(x) = arg \max_{y} \frac{p(y)p(x/y)}{p(x)}$$

$p(x)$ is independent of y, hence, does not affect arg max. Therefore the equation looks like

$$f(x) = arg \max_{y} p(y)p\left(\frac{x}{y}\right) \tag{3}$$

Using first Markov assumption, in a sequence $\{x_1, x_2, \dots, x_n\}$

$$(x_n/x_{n-1}, \quad x_{n-2}, \dots x_1) \approx p(x_n/x_{n-1})$$

We can express the joint probability using Markov assumption,

$$p(x_1, x_2, \dots, x_n) = \prod_{i=1}^{n} p(x_i/x_{i-1}) \tag{4}$$

For the n^{th} element of the sequence of word

$$f(x_n) = arg \max_{yn} \; p(y_n) p\left(\frac{x_n}{y_n}\right) \tag{5}$$

Assume that the probability of a word is only dependent on its own POS tag, then

$$p(x_n/y_n) \approx \prod_{i=1}^{n} p(x_i/y_i) \tag{6}$$

which constitutes the emission probability matrix.

If the probability of POS tag is only dependent on the previous POS tag, then

$$p(y_n) \approx \prod_{i=2}^{n} p(y_i/y_{i-1}) \tag{7}$$

which constitutes the transition probability matrix. Using (6) and (7) in (5)

$$,f(x_n) = arg \max_{yn} \prod_{i=1}^{n} p(x_i/y_i) \prod_{i=2}^{n} p(y_i/y_{i-1}) \tag{8}$$

This model is known as bigram model.

III. Viterbi Algorithm

Viterbi algorithm makes some assumptions.

- There are two states of events with respect to time viz. Emitted events and hidden events.
- There has to be one – one correspondence of these to sequence of events.

Viterbi is a common decoding algorithm to find the y^*, the most likely path of tags given the emissions[5].

$$y^* = arg \max_{y} p(y/x) = arg \max_{y} p(y, x)$$

It is based on Bottom-up dynamic programming approach. For an input sentence$\{x_1, x_2, \dots, x_n\}$. This is the problem of finding

$$arg \max_{y_1, \dots, y_{n+1}} p(x_1, x_2, \dots, x_n, y_1, \dots, y_n, y_{n+1})$$

where $y_1, \dots, y_n$ belongs to the set of Tags and y_{n+1} = STOP.

$$p(x_1, x_2, \dots, x_n, y_1, \dots, y_n, y_{n+1}) = \prod_{i=1}^{n} p(x_i/y_i) \prod_{i=1}^{n+1} p(y_i/y_{i-2}, y_{i-1}) \tag{9}$$

Application of Brute-force search technique in (9) yields Viterbi algorithm.

IV. Experiment and Discussion

The database is generated from NELRALEC Tagset [6] with 41 tags. A report on Nepali Computational Grammar is made available by Prajwal Rupakheti et al. [7] which contains frequently used tag set. A POS tagged sentence of Nepali text with English conversion is illustrated in Figs. 1 (a) and (b) respectively. In Fig. 1 (a), DUM represents unmarked demonstrative pronoun, PKO indicates KO postposition whereas RBO and PP$ demonstrate other adverb and possessive pronoun respectively. Moreover, NN and POP represent common noun and other post position respectively. At the end of the sentence VBX and YF indicate auxiliary verb and sentence final respectively. The tagset of the database used in the experiment is having the format shown in Fig. 1 (a).

यस DUM को PKO आज RBO हाम्रो PP$ श्रमसमुह NN सँग POP कुनै DUM सरोकार NN छैन VBX । YF

Fig. 1(a) POS Tagged Nepali text

AT PRESENT THIS HAS GOT NO CONCERN TO OUR WORK.

Fig. 1(b) English Translation

The information of tagset is Database contains 45,000 Nepali words with corresponding Tag, out of which 15005 samples are randomly collected for testing purpose. Remaining 29995 words are utilized to construct Transition matrix, made up of 41 states i.e. Tags, which is of the size 41 x 41 using (7). And the emission matrix, made up from the emission of words, is constructed from the testing samples. It is of the dimension 41 x 15005 using (6). Transition and Emission matrices are the outcome of emitted events and hidden events respectively as discussed in Section III. In order to generate these matrices, the computer program is implemented in Java. These matrices are supplied to MATLAB program along with the original tag sequence to get the accuracy. Table I depicts the performance of both the methods. Using hmmgenerate() command of Matlab, states and arbitrary sequence are generated. These two arguments are supplied to hmmtrain() and hmmviterbi() to get the output.

TABLE I
Classification Accuracy

Sr. No	Technique	No of Mismatches	Accuracy
1	*HMM*	*3455*	*76.97*
2	*Viterbi*	*574*	*95.43*

Tag wise error occurrence analysis of 574 mismatches (Table I) with Viterbi algorithm is depicted in Figs. 2 and 3. Out of 574 mismatches 309 are wrongly identified as NN (Common Noun) as shown in Fig. 1. Upon observing 309 incorrectly identified NN's, 35 are misidentified as NNP (Proper Noun). Hence, out of the total 48 NNP Tags from Fig. 3, 35 are confused with NN. The transition probability of occurrence of NN after NN a 0.1519 which is the highest among all, whereas the occurrence of NNP after NN is 0.0056 and NN after NNP is 0.0065. Maximum mismatch (309) takes

place for NN only as appeared the tallest pick in Fig. 2 which is due to the highest probability from NN to NN.

V. Conclusion and Future Work

Viterbi algorithm is found to be very accurate as compared to HMM for POS Tagging of Nepali text as depicted in Table I. The frequency of misclassification of the Tag NN (common noun) is 309 as shown in Fig. 2 which is the highest compared all other tags. In 35 out of 48 cases (Fig. 3) where the original tag is supposed to be NNP is wrongly tagged as NN. The second highest frequency of misclassified Tag CD (Cardinal Number) is 128. In 73 out of 128 cases (Fig. 2), the Cardinal Numbers are confused due to the availability of mixed and improper fractions in the database. Such errors can be minimized by applying proper grammar rule and that is the future task to work on. Another approach to improve accuracy is to apply Conditional Random Fields (CRF) with morphological analyzer presented by Patel and Gali [8].

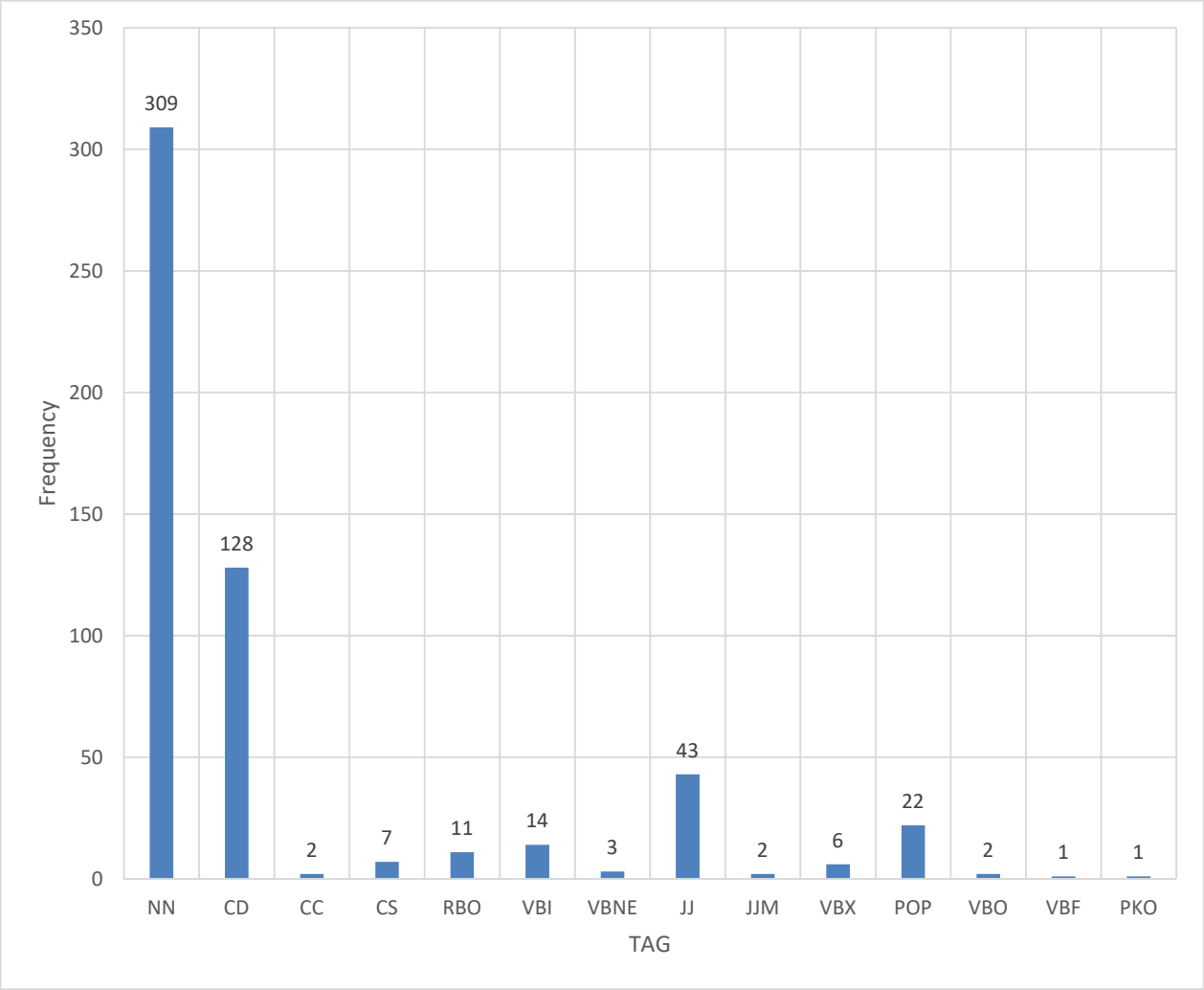

Fig. 2 Tag Wise Error Analysis for Viterbi

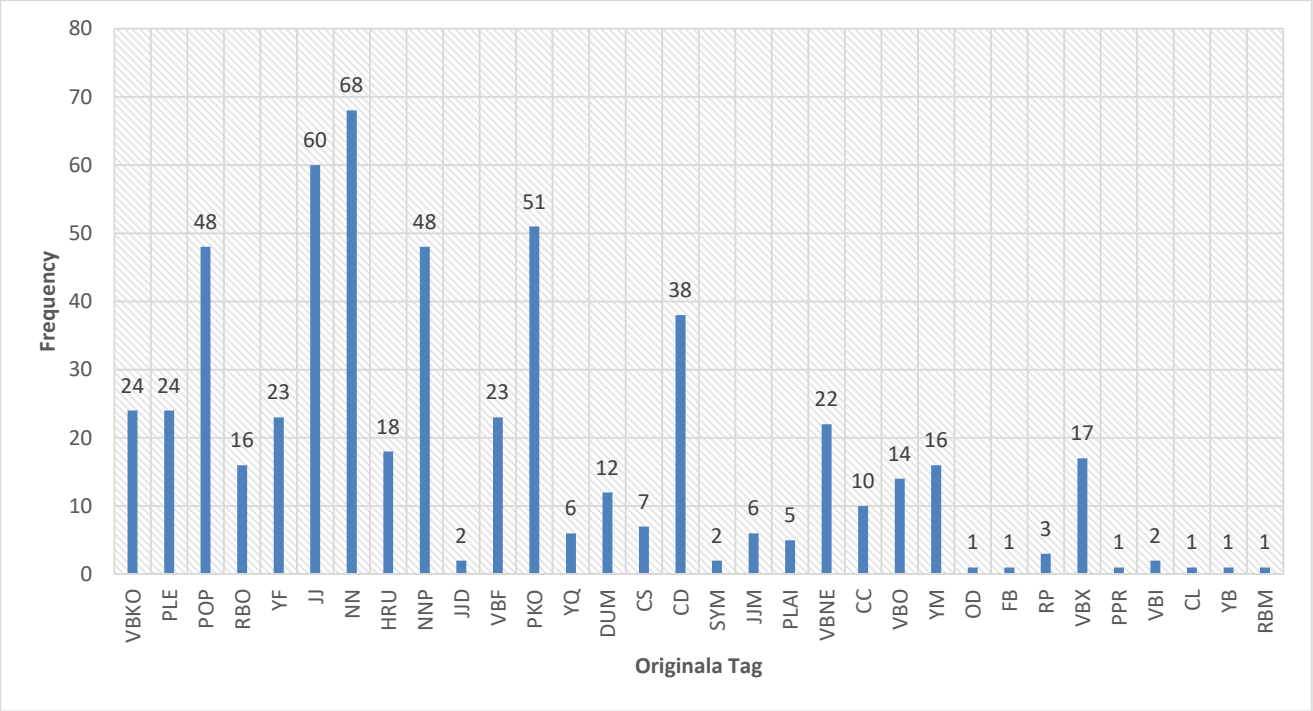

Fig. 3 Original Tag Wise Error Analysis Using Viterbi

ACKNOWLEDGMENT

Dr. Archit Yajnik is 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. He extends his 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 he expresses his sincere thanks to DST for providing an opportunity to work on the platform of NLP.

REFERENCES

[1] Prajadip Sinha et al. Enhancing the Performance of Part of Speech tagging of Nepali language through Hybrid approach, *International Journal of Emerging Technology and Advanced Engineering*.2015 Vol 5(5).

[2] Tej Bahadur Shai et al. 2013. Support Vector Machines based Part of Speech Tagging for Nepali Text, *International Journal of Computer Applications,* May 2013, Vol: 70-No. 24, pp. 0975-8887.

[3] Antony P J et al. 2011.*Parts of Speech Tagging for Indian Languages: A* Literature Survey, *International Journal of Computer Applications*, 2011, Vol. 34(8), pp. 0975-8887.

[4] Akshar Bharati, Dipti Misra Sharma, Rajeev Sangal et al., (15th December, 2006), AnnCorra: Annotating Corpora, Guidelines for POS and Chunk Annotation for Indian Languages. Retrieved from http://researchweb.iiit.ac.in/~rashid.ahmedpg08/ilmtdocs/chunk-pos-ann-guidelines-15-Dec-06.pdf

[5] Ben Langmead. (n.d.) Hidden Markov Models. Retrieved from http://www.cs.jhu.edu/~langmea/resources/lecture_notes/hidden_markov_models.pdf

[6] A part-of-speech tagger for Nepali. (17th May 2006). Retrieved from http://www.lancaster.ac.uk/staff/hardiea/nepali/postag.php

[7] PAN Localization.(n.d.). Retrieved from http://www.panl10n.net/english /Outputs%20 Phase%202/CCs/ Nepal /MPP/Papers/2008/Report% 20on% 20Nepali%20Computational%20Grammar.pdf

[8] Chirag Patel et. al., Part-Of- Speech Tagging for Gujarati Using Conditional Random Fields, *Proc. Of the IJCNLP-08 Workshop on NLP for Less Privileged Languages,* 2008, pp.117-122.