Hidden Markov Model Based Part of Speech Tagging for Nepali Language

Abhijit Paul
Department of Computer Science
Assam University Silchar

Bipul Syam Purkayastha
Department of Computer Science
Assam University Silchar

Sunita Sarkar
Department of Computer Science
Assam University Silchar

***Abstract* - Natural Language Processing (NLP) is mainly concerned with the development of computational models and tools of aspects of human (natural) language processing. Part of Speech Tagging (POS) is well studied topic and also one of the most fundamental preprocessing steps for any language in NLP. Natural language processing of Nepali is still lack significant research efforts in the area of NLP in India. POS tagging of Nepali is a necessary component for most NLP applications in Nepali, which analyses the construction of the language, behavior of the language and can be used to develop automated tools for language processing. From the literature survey and related works, it has been found that, not much work has been done previously on POS tagging for Nepali language in India due to lack of comprehensive set of tagged corpus or correct hand written rules. In this paper, Hidden Markov Model (HMM) based Part of Speech (POS) tagging for Nepali language has been discussed. HMM is the most popular used statistical model for POS tagging that uses little amount of knowledge about the language, apart from contextual information of the language. The evaluation of the tagger has been done using the corpora, which are collected from TDIL (Technology Development for Indian Languages) and the BIS tagset of 42 tags. Tagset has been designed to meet the morph-syntactic requirements of the Nepali language. Apart from corpora and the tagset, python programming language and the NLTK's (Natural Language Toolkit) library has been used for implementation. The tagger achieves accuracy over 96% for known words but for unknown words, the research is still continuing.**

Keywords – NLP; POS; HMM

I. INTRODUCTION

In terms of language automation technique, POS is the process of assigning the most accurate part of speech category or word class marker (Noun, Verb, Adjective, Adverb etc.) for each word in a sentence of a natural language. POS tagging is based on both the definition and the context i.e. the relationship of the word with adjacent and related words in the sentence that it exists in **[1].** The input to algorithm is the sequence words of a natural language sentence and specified tag sets (a list of part of speech tags). The output is a single best part of speech tag category for each word in the sentence.

The overall process of POS tagging is depicted in the figure below.

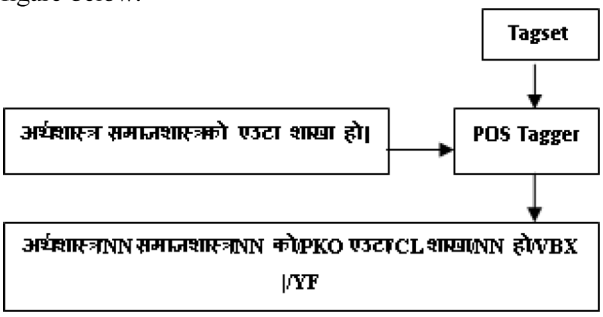

Fig1: the process of part of speech tagging of Nepali Text

Nepali (नेपाली) is an Indo-Aryan language spoken by approximately 45 million speakers in Nepal, Bhutan, Myanmar and some parts of India **[2]**. It is the lingua-franca of Nepal and is one of the 23 official languages of India, incorporated in the Indian constitution. It has official language status in the Indian states of Sikkim and in West Bengal's Darjeeling district. Further it is widely spoken in the Indian states of Uttaranchal and Assam. Nepali language is an inflectionally & morphologically rich in nature i.e. words are inflected with various grammatical features. The grammatical categories like Person, Number and Gender are found distinctively in pronouns and adjectival and verbal inflections. Natural language processing of Nepali is in its infancy means it has just started its journey in the field on NLP in India.

Though the considerable amount of work has already been done in POS tagging for English, European and few Indian languages **[3][4][5][6]**, but not much work has been done previously on part of speech tagging of Nepali language in India. The main reason behind this is the unavailability of linguistic rules and large annotated corpora for Nepali language. So the attempt of this research is to develop POS tagger for Nepali language.

Nepali POS tagging is of interest due to its broad range of applications, also POS tagging gives significant amount of information about the word and

978-1-4673-6708-0/15/$31.00 ©2015 IEEE

its neighbours which can be useful for higher level NLP tasks such as different speech and natural language processing applications, semantics analysis, machine translation and many more [7]. POS tagging is also useful for parsing, as taggers reduce ambiguity from the parser's input sentence, which increases the speed of the parsing by making the computational problem smaller, and the result will less ambiguous. Therefore, there is a pressing necessity to develop an automatic Part-of-Speech tagger for Nepali language.

II. RELATED WORKS

A. English POS Taggers

Most of the work in part of speech tagging is done for English and other European languages. The earlier systems for automatic tagging were rule based. TAGGIT [8] was the first rule based tagger used for the initial tagging of the Brown corpus. It used 3,300 disambiguation rules and was able to tag 70% of the words in the Brown corpus with their correct part of speech. The rest was done manually over several years. Yet another rule based tagger is ENGTWOL [9]. Apart from these CLAWS, Brill Tagger, TnT Tagger, Tree Tagger & Standford Tagger etc are the few POS taggers for English which are already in place in the field of NLP.

B. Indian Languages POS Taggers

In the last few decades of NLP initiatives on Indian languages, different research groups working on various languages have developed POS tagger for respective or a group of Indian languages. Due to broad range of applications, POS tagging makes researcher a lot of interest. Especially in a multilingual country like India where machine translation is highly necessitate. POS tagging in Hindi and other languages in India is constrained heavily due to large annotated corpus. Now the area of automated Part of speech tagging has been enriched due to the development tools, techniques and corpora is going on at several places in India such as CDAC, IIT Bombay, IIIT Hyderabad, CIIIL Mysore. A few part of speech tagging systems reported in recent years use morphological analyzers along with tagged corpus e.g. HMM based part of speech tagging in Bengali [10] and Hindi part of speech tagger [11] over the last few decades by contribution from several researchers. Assamese POS tagger [12] has been developed using HMM and provides an average tagging accuracy of 87%. For Gujarati, machine learning algorithm has been developed [13] following the CRF model in which the features given to CRF are considered with respect to the linguistic aspects of Gujarati. Malayalam POS tagger [14] is designed to capture finer morphological analysis; and consequently, generates the most suitable POS tag using SVM (Support Vector Machine) based approach. It has an accuracy rate of 80% for the sequence generated automatically for the test case.

C. Nepali POS Taggers

A support vector machine based POS tagger has been developed for Nepali Text [15] using NELRALEC (Nepali Language Resources and Localization for Education and Communication) tagset of 112 tags. It shows an accuracy of 93.27%. First order Markov model has been implemented in [16] which uses the same POS tagset as in [15] and reports the good accuracy (91%) for known word. Under NELRALEC project, the “Unitag” - a tool for tagging has been developed for Nepali language. The tagset used in “Unitag” was of 112 tags. It showed more error in tagging since size of the tag set was high i.e. 112 tags. Later, using “Unitag” tool a corpus of 14 million Nepali words was first manually tagged with 112 tags. Then this tagged corpus was served as the training of an automatic tagger.

III. MORPHOLOGY IN NEPALI LANGUAGE

Nepali language is a complex, inflectionally and morphologically rich in nature i.e. words are inflected with various grammatical features and rules. Nepali grammar consists of both inflected and uninflected structure which are open and the closed form of classes, traditionally known as the parts of speech. The open classes are those classes whose membership is in principle indefinite or unlimited and closed form of classes are those whose membership is fixed or limited where the new items are not regularly added. Noun, adjective, verb and adverb belong to open class whereas the pronoun, coordinating conjunction, subordinating, conjunction, postposition, interjection, vocative and nuance particle belongs to closed class [17]. e.g. Nepali noun can be inflected in contrasts for singular vs. plural eg. manis /मानिस 'man' , manisharu / मानिसहरू 'men' and for seven different cases also. Similarly other categorical words are inflected according to gender, case, quantity, tense etc. As Nepali is a highly inflectional & morphologically rich language, it is required to split the word into its smallest constituent parts i.e. morphemes (affixes) and root word for tagging the lexical items with proper lexical categories. Therefore, a morpheme splitter (Stemmer) has been used to improve the performance of the tagger. An affix removal stemmer are been used considering three types

lexicons viz. prefix, suffix and root and little hand written rules [20].

IV. NEPALI CORPUS

Corpus is a large collection of texts. It is a body of written or spoken material upon which a linguistic analysis is based. Though English and other European languages corpora (both raw and annotated) are easily available in the web but for this research only raw corpora are available, which are developed by TDIL [18]. Tagged corpora are not available for to initiate the tagging task. Tagged corpus which is one of the important issue to work with statistical POS tagging. So using an annotated tool "SANCHAY", developed by IIIT, Hyderabad, a raw corpus which contains around 1,50,839 words, has been tagged.

V. PROPOSED NEPALI TAGSET

Apart from corpus, a well chosen tagset is also important. The proposed tagset has been designed to meet the morph-syntactic requirements of the Nepali language. Also it has been followed the guidelines of ILCI (Indian Languages Corpora Initiative), BSI (Bureau of Indian Standard). The tagset consists of 42 tags including generic attributes and language specific attribute values. Table I presents the Nepali tagset with descriptions.

VI. STOCHASTIC APPROACH FOR TAGGING

Throughout the history of NLP, different approaches have already been tried out for POS tagging and a survey was made on those approaches and found that for high accuracy tagger, researchers are concentrating more on applying mathematical model along with corpus generated lexicon than applying hand written rules. This approach adds a new direction to the development of linguistic tools in unknown linguistic world. So keeping this idea in mind how information and data can be extracted from corpus which can be

Table I: Nepali Tagset

Sl. No	TAG	DESCRIPTION	Sl. No	TAG	DESCRIPTION
1	N-NN	Common Noun	22	PSP-POP	Other Postposition
2	N-NNP	Proper Noun	23	CC-CCD	Coordinating
3	PR-PRP	Personal Pronoun	24	CC-CCS	Subordinating Conjunction
4	PR-PRS	Possessive Pronoun	25	QT-QTC	Cardinal Number
5	PR-PRF	Reflexive Pronoun	26	QT-QTO	Ordinal Number
6	DM-DMR	Marked Demonstrative	27	HRU	Plural Marker
7	DM-DUM	Unmarked Demonstrative	28	QW	Question Word
8	V-VBF	Finite Verb	29	RP-INTF	Intensifier
9	V-VBX	Auxiliary Verb	30	RP-CL	Classifier
10	V-VBI	Verb Infinite	31	RP-INJ	Interjection
11	V-VBNE	Prospective Participle	32	RP-PRD	Default Particle
12	V-VBKO	Aspectual Participle	33	DT	Determiner
13	V-VBO	Other participle Verb	34	RD-UNW	Unknown Word
14	J-JJ	Unmarked Adjective	35	RD-FW	Foreign Word
15	J-JJM	Marked Adjective	36	RD-YF	Sentence Final
16	J-JJD	Degree Adjective	37	RDS-YM	Sentence Medieval
17	RB-RBM	Manner Adverb	38	RD-YQ	Quotation
18	RB-RBO	Other Verb	39	RD-YB	Brackets
19	PSP-PLE	Le-Postposition	40	RD-ALPH	ALPH
20	PSP-PLAI	Lai-Postposition	41	RD-SYM	Symbol
21	PSP-PKO	Ko-Postposition	42	RD-FB	Abbreviation

useful in POS tagging, stochastic (HMM) based approach has been used. HMM is the most popular used mathematical or stochastic model for POS tagging that uses little amount of knowledge about the language, apart from contextual information of the language.

In this section the approach and the overall architecture of the Nepali POS tagger has been discussed. The intuition behind all stochastic approach is given a sequence of words (sequence), the objective is to find the most probable tag sequence for the sentence. HMM tagger chooses the tag sequence which maximizes probability value.

$$P(word \mid tag) * P(tag \mid previous\ n\ tags)$$

HMM tagger generally chooses a tag sequence for a given sentence rather than for a single word.
Let W be the sequence of words.

$$W = w_1, w_2, w_3, \ldots\ldots\ldots, w_n$$

The task is to find the tag sequence

$$T = t_1, t_2, t_3, \ldots\ldots\ldots, t_n$$

Which maximizes $P(T|W)$, i.e.,

$$= argmax_{t \in T}\ P(T \mid W)$$

Applying Bayes Rule, P(T|W) can be estimated using the expression:

$$P(T|W) = P(W|T) * P(T)/P(W)$$

As the probability of the word sequence, $P(W)$, remains the same for each sequence, so it can be dropped. The expression for the most likely tag sequence becomes:

$$\square = argmax_{t \in T}\ P(T)P(W \mid T)$$

Using the Markov assumption, the probability of a tag sequence can be estimated as the product of the probability of its constituents n-grams, i.e.,

$$P(T) = P(t_1) * P(t_2|\ t_1) * P(t_3 \mid t_1\ t_2), *\ldots\ldots P(t_n|\ t_{1\ldots\ldots} t_{n-1})$$

$P(W \mid T)$ is the probability of seeing a word sequence , given a tag sequence. For example, it is asking the probability of seeing 'The egg is rotten' given 'DT NNP VB JJ'. The following two assumptions can be made:

- The words are independent of each other.
- The probability of a word is dependent only on its tag.

Using these assumptions, the equations becomes
$P(W|T) = P(w_1|t_1) * P(w_2|t_2) \ldots\ldots P(w_i|t_i) * P(w_n|t_n)$

So,

$$P(T)P(W|T) = P(t_1)P(t_2|t_1)\prod_{i=3}^{n} P(t_i|t_{i-2}t_{i-1})\left[\prod_{i=1}^{n} P(w_i|t_i)\right]$$

Approximately the tag history using only the two previous tags, the transition probability, P(T), becomes
$P(T) = P(t_1) * P(t_2|\ t_1) * P(t_3 \mid t_1\ t_2), \ldots\ldots\ldots *P(t_n|\ t_{1\ldots\ldots} t_{n-1})$

Hence, $P(T|W)$ can be estimated as

$$P(t_1)P(t_2|t_1)\prod_{i=3}^{n} P(t_i|t_{i-2}t_{i-1})\left[\prod_{i=1}^{n} P(w_i|t_i)\right]$$

Finally estimate these probabilities from relative frequencies via Maximum Likelihood Estimation.

$$P(t_i|t_{i-2}t_{i-1}) = \frac{c(t_{i-2}t_{i-1}t_i)}{c(t_{i-2}t_{i-1})}$$

and

$$P(w_i|t_i) = \frac{c(w_i t_i)}{c(t_i)}$$

for all w_i where $1 \le i \le n$.
where $c(t_{i-2}, t_{i-1}, t_i)$ is the number of occurrence of t_i followed by $t_{i-2}\ t_{i-1}$.

A. SYSTEM ARCHITECTURE

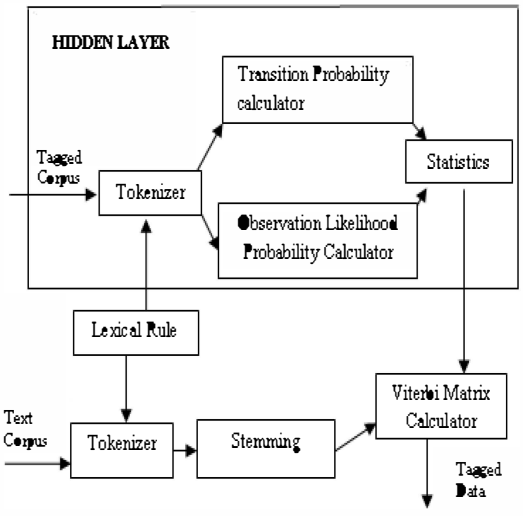

fig 2: Architecture of the tagger

The architecture of the tagger (fig. 2) uses two layers of states: a visible layer corresponding to the input words, and a hidden layer learnt by the system corresponds to the tags. While tagging the input data, one can only observe the words- the tags (states) are hidden. States of the model are visible in training, not during tagging task. The upper part of the architecture is hidden from the user. Following are the modules used in the system.

Lexical Rule: Checks boundary conditions of each sentences and words according to the lexical rules. It is used both in training tagged corpus and input text corpus.

Tokenization: It runs through the tagged corpus first in the hidden layer, separate out the words and tags and makes it ready for probability calculation. It also runs in text corpus, separate out the tokens only.

Observation Likelihood Probability Calculation: Calculate the observation probability for each word in the tagged corpus and prepare observation likelihood matrix and keep this matrix in the statistics module.

Transition Probability Calculation: Calculate the Transition probability for each tag sequence in the tagged corpus and prepare transition probability matrix and also keep this matrix in the statistics module

.

Viterbi Matrix Calculator: It retrieves all the statistics from statistics module and prepares a state graph that has all possible state transitions for the given text input, calculate and assign state transition probability for each transition in the matrix. Finally it finds the best sequence of tags based on maximum probability.

Stemming: Stemming is one of the important phase to developing an inflectional language POS tagger. Along with HMM, a stemmer has been used to improve the performance of the tagger.

B. VITERBI ALGORITHM

The algorithm that has been used for tagging is the viterbi algorithm and it is widely used algorithm for most of the NLP applications viz. parsing, chunking etc. The algorithm considers all the words in the given sentence simultaneously and computes the most probable tag sequence against given sentence. For best tag sequence computation Viterbi algorithm makes a matrix with i+2 (two extra rows: one for starting symbol and another for end state) number of rows and j+1 (One extra for start symbol) number of columns, where i is the total no tags in tagset and j is the total no of words in an input sentence. Each cell V [i, j] of the matrix contains the probability of the path considering the previous probability.

The algorithm fills up all the cells, column by column with one column for each observation and one row for each state in the state graph. The first column in the matrix is the initial observation, which is the start of the sequence, then next corresponds to the first observation, then next corresponds to the second observation, and so on. Next sets the probability of the cell V[0,0] to 1, and other probabilities V[0,k]=0 for all k= 1 to j+2. Then the cell V [k,l], for each column of the matrix against each row of the matrix , will contain the probability of the most likely path to end in that cell. Then calculate this probability recursively, by maximizing over the probability of the coming out from all possible preceding states. Then gradually move to the next state and compute the probability for each cell and finally, get the probability for the best path, which will appear in the final state. At last back track the highest probability path from the final state to the initial state and get the path that gives the best possible tag sequence.

C. WORKING OF TAGGER

During training the tagged corpus, what statistics and how these statistics can be generated are the important steps. The method that that has been considered in this paper is the supervised training methods. It runs on the tagged corpus and estimates the probabilities of transition, P (tag | previous tag) and observation likelihood P (word | tag) for the HMM.

The transition probability $P(t_i|t_{i-1})$ is calculated by using the following formula.

$$P(t_i|t_{i-1}) = \frac{c(t_{i-1}t_i)}{Total\ number\ of\ bigrams\ starts\ with\ t_{i-1}}$$

Where c $(t_{i-1}t_i)$ is the frequency count of tag sequence t_{i-1},t_i in the corpus.

For calculating observation likelihood probability $P(w_i|t_i)$, calculate the unigram of a word along with its tag assigned in the tagged data. The likelihood probability is calculated by the following formula.

$$P(w_i|t_i) - \frac{c(t_i w_i)}{Total\ number\ of\ bigrams\ starts\ with\ t_i}$$

Where c (t_i,w_i) is the frequency count of word i (w_i) is assigned tag i (t_i) in the corpus.

VII. IMPLEMENTATION & RESULT ANALYSIS

Apart from corpora and the tagset, Natural Language Toolkit (NLTK) [19] has been used for the implementation. NLTK, which is a set of computational linguistics and NLP program modules, annotated corpora and tutorials supporting research and teaching aid for NLP. NLTK is completely written with Python programming language. NLTK allows various NLP tasks by providing implementation of various algorithms such as the Brill tagger, HMM based POS tagger, n-gram based taggers etc. Though it provides the implementation of such taggers but these works only for English language. For English, NLTK's authority has fully implemented the important modules of NLP. Also they provide tagged corpus like Brown, Gutenberg etc to work with HMM tagger. To work with Indian languages, some changes have been made on the source code. For the experiment, the Unigram, Bigram and the HMM tagging modules of NLTK has been used. HMM tagger gives the result based on transition probability matrix, observation likelihood probability matrix and viterbi matrix. Accuracy has been calculated by the following formula,

$$\text{Accuracy} = \frac{\text{No. of single correct tags}}{\text{Total no. of words}} * 100\%$$

which is already defined in the NLTK's different tagger module. Entropy is also calculated against each sentence. Each test sentence (actual result), untagged Sentence (input sentence) and HMM tagged sentence (output sentence) along with its entropy will display at run time. Testing of text corpus has been done using two steps: first, tested with only known words (i.e. the words which are already in the training tagged corpus) and it gives accuracy over 96% i.e. approximately 4% is the error rate. On second step, unknown words (i.e. the words which are not in the tagged corpus) along with known words have been taken for testing and it gives less accuracy (over 85%) and the tagger doesn't perform well. For training we have trained the same amount of tagged corpus (i.e. of 1,50,839 words) on both the steps. Figure 3 shows the accuracy of the system for known words on different test cases. Here we consider the number of sentences instead of number of words. But the systems i.e. the NLTK's modules automatically calculate the number of words present in the sentence to find the accuracy.

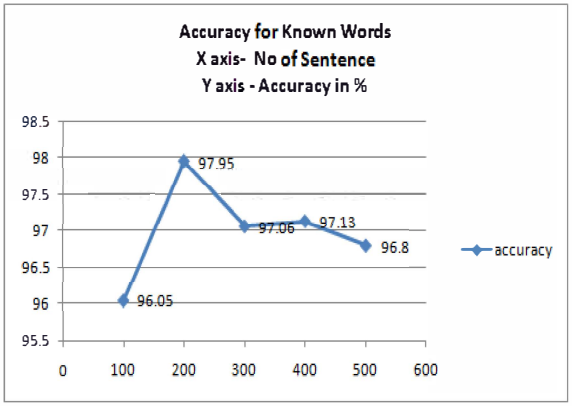

Fig 3. Accuracy for known words

For second test, 70% known sentence and 30% unknown sentence have been taken for testing. Figure 4 shows the accuracy of the system for different test cases for unknown words along with known words.

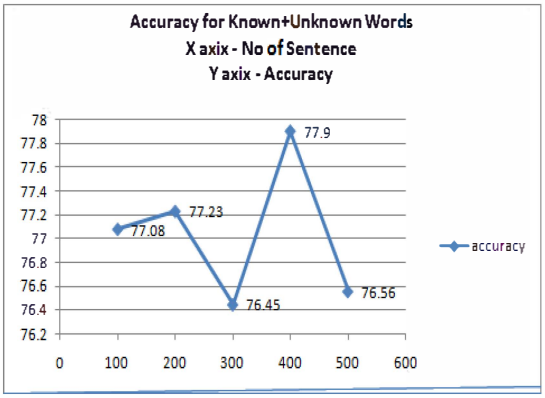

Fig 4: Accuracy for known & unknown words

Another testing has been done for unknown words only and this time it achieves less accuracy compare to others two. Here the size of the test corpus (data) has been kept fixed but training tagged corpus will be varying on different test cases.

Table II: Accuracy for unknown words

Corpus (No. of words)			Unknown Words
Total Corpus	Training Data	Test Data	Acc. (%)
1,50,839	40,000	1,000	41.19
	80,000	1,000	43.02
	1,20,000	1,000	46.5

From the above observations it can be conclude that the overall accuracy of the tagger depends upon the size of the corpus and the accuracy of the tags assigned. Therefore, training corpus and the accuracy of tags should be as high as possible so that the overall performance of the tagger could be improved.

VIII. COMPARISON

NLTK's HMM tagging module for English language using Brown tagset and tagged corpora which are provided by the NLTK's authority has been tested and it reports accuracy over 96% for both known and unknown words. The reason behind where performance was reported over 96% is very large corpora was employed for training the model. Similarly using NLTK's HMM module, a test has been made on four Indian languages corpora Hindi, Bangla, Telegu and Marathi, which are also provided by the NLTK authority. But for these four cases (Hindi, Bangla, Telugu & Marathi), it does not show more accuracy like English.

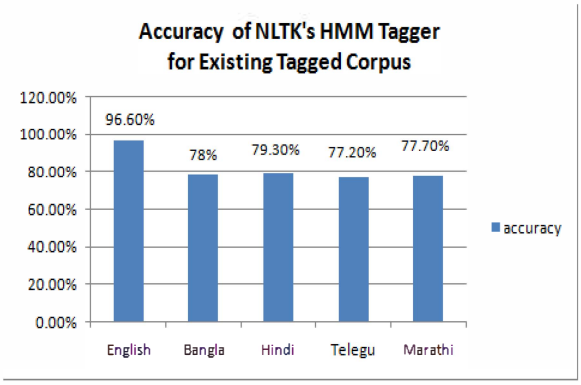

Fig 5 Accuracy (known and unknown Words) of NLTK's HMM Tagger for Existing Tagged Corpus

In the paper [15], accuracy of different existing taggers for Nepali text is explained, which are available in Thribhuvan University, Kathmandu, Nepal. Table III shows the comparison.

Table III. Accuracy of existing Nepali Taggers

	Accuracy		
Tagger	Known Words	Unknown Words	Overall
TnT	91%	56%	73.50%
SVM	96.48%	90.06%	93.27%

IX. CONCLUSION & FUTURE DIRECTION

Part of speech tagging is playing an important role in various applications of NLP. Since many of the reputed companies like Google and Microsoft are concentrating on NLP applications, it has got more importance. In this paper, HMM based statistical model has been used for POS tagging of Nepali language. Viterbi algorithm has been incorporated with the said model. The system disambiguates correct word-tag combinations using the probabilistic information available in the tagged corpus. The model has been tested against various sizes sentences of Nepali text corpus. Accuracy over 96% for known words has been achieved, but the system is not performing well for the text with unknown words yet. It had shown that the performance of the tagger depend upon the size of the training data, as well as number of tokens that are present and absent in the training data.

Though accuracy over 96% for known words has been achieved but for unknown words it stills an open area for further research. Also future enhancement is required to improve the performance of the tagger. This can be achieved by refining the tagset and adding more tags and words in the training tagged corpus so that the tagger can face less ambiguous classification of the text. Smoothing technique can be applied to get a better outcome. A comparison can also be made between the results obtained by the system and other Indian languages tagging system.

REFERENCES

[1] T. Siddiqui and U.S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press Publication, 2010.

[2] Isaac Y. Arredondo, B.S. & Heather Ballard, "NEPALI MANUAL: LANGUAGE AND CULTURE" Prepared by: B.S. Texas State University – San arcos, Class of 2012.

[3] M. Shrivastav, R. Melz, S. Singh, K. Gupta and P. Bhattacharyya, 2006, "Conditional Random Field Based POS Tagger for Hindi", in Proceedings of the MSPIL, Bombay, 63-68.

[4] P. Arulmozhi., R.K. Rao and L. Sobha, 2006, "A Hybrid POS Tagger for a Relatively Free Word Order Language", in Proceedings of the Modeling and Shallow Parsing of Indian Language (MSPIL), Bombay. 79-85.

[5] S. Singh, K. Gupta, M. Shrivastav and V. Bhattacharyya, 2006. "Morphological Richness Offset Resource Demand – Experience in constructing a POS Tagger for Hindi", in Proceedings of COLLING/ACL 06. 779-786.

[6] K. Nagaraj, U. Swant, S. Shelke and P. Bhattacharyya, 2007, "Building Feature Rich POS Tagger for Morphologically Rich Languages: Experience in Hindi", in Proceedings of ICON, India.

[7] A. MacKinlay, "The Effects of Part-of-Speech Tagset on Tagger Performance", Undergraduate Thesis, University of Melbourne, 2005.

[8] B. Green and G. Rubin, 1971, “Automated grammatical tagging of English”, Department of Linguistics, Brown University.
[9] R. Garside, 1987, “The CLAWS Word tagging System’ The Computational Analysis of English: A Corpus based Approach”, Longman, London, pp. 167-180.
[10] S. Dandapat, S. Sarkar and A. Basu, 2004, “A Hybrid Model for Part-of- Speech Tagging and its Application to Bengali”, International Journal of Information Technology Volume 1, Number 4.
[11] S. Singh, K. Gupta, M. Shrivastav. And V. Bhattacharyya, 2006. “Morphological Richness Offset Resource Demand – Experience in constructing a POS Tagger for Hindi”, in Proceedings of COLLING/ACL 06. 779-786.
[12] N. Saharia, D. Das, U. Sharma and J. Kalita, “Part of Speech Tagger for Assamese Text”, Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 33–36, Suntec, Singapore, 4 August 2009.
[13] C. Patel and K. Gali, “Part-Of-Speech Tagging for Gujarati Using Conditional Random Fields”, Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages, 2008.
[14] P.J. Antony, P. Santhanu, P. Mohan and K.P. Soman, "SVM Based Part of Speech Tagger for Malayalam", itc, pp.339-341, International Conference on Recent Trends in Information, Telecommunication and Computing, 2010.
[15] T.B. Shahi, T.N. Dhamala and B. Balami, “Support Vector Machines based Part of Speech Tagging for Nepali Text”, Central Department of Computer Science and IT, Tribhuvan University 2013, Nepal.
[16] M.R. Jaishi, “Hidden Markov Model Based Probabilistic Part Of Speech Tagging For Nepali Text”, (Masters Dissertation, Central Department of Computer Science and IT ,Tribhuvan University 2009, Nepal).
[17] D. Simkhada, “Implementing the GF Resource Grammar for Nepali Language”, “Master of Science Thesis in Software Engineering and Technology”, Chalmers University of Technology, University of Gothenburg, Sweden 2012.
[18] http://tdil.mit.gov.in
[19] NLTK, The Natural Language Toolkit, available online at: http://nltk.sourceforge.net/ index.html.
[20] A. Paul, A. Dey, B.S. Purkayastha, “An affix reomval stemmer for Natural Text in Nepali” International Journal of Computer Applications (IJCA), ISBN: 973-93-80881-06-7, April, 2014.