Part of Speech Tagging of Hindi using Markov Model

Aarti Singh, Nisheeth Joshi

Abstract: ***Hindi is free word-order language and morphologically rich language due to this applying Part of Speech tagging is very challenging task. Here, we have used HMM and Trigram model for POS tagging of Hindi. We have trained our tagger on 50000 manually POS tagged sentences. These two algorithms gave us an accuracy of 97.05% and 98.28% respectively for trigram and HMM models.***

Index Terms: Trigram model, Hidden Markov model, Hindi Language, POS Tagging, Sequence Labelling.

I. INTRODUCTION

NLP (Natural language processing) is a computation of natural language by machine in the same way as human do. It has ability to extract meaning from the text, machine translation and language analysis. NLP provide interaction between machine and the human. Initial step to develop NLP application is POST (part-of-speech tagging). POS Tagging is sequence labelling task in which we assign Part-of-speech to every word (Wi) which is sequence in sentence and tag (Ti) to corresponding word as label such as (Wi/Ti…. Wn/Tn). To implement tagging we have different-different approach such as: Rule Base approach in which tagging rules are given manually, Statistical approach in this mathematical formula are used to finding the probability of tag or Hybrid approach is the combination of both Rule base and statistical approach. In this paper we worked on Trigram and HMM (Hidden markov model) tagging. This POS tagging is done for various purpose such as parsing which is done using chunking with the help of POS tagging. For NER (Name Entity Recognition) POS tagging is required which help in identify name entity such as Location, Numeric, Entity, Human, Description etc. POS tagging is also use for Question Answering task by finding noun phrase which highlight the entity ask about and also in word sense disambiguation. POS tagging is also known as grammatical tagging. Every language has their own grammar rules, thus developing tagger is a big challenge. Structuring tagger for Hindi language is difficult due to its morphological rich nature and free word-orderings. These are the reasons for which finding ambiguities tag in Hindi is challenging. Also, we do not have working tagger for Hindi and small corpus prone to more error.

II. LITERATURE SURVEY

In this section we will discuss work done on POS tagging in Indian language using different-different approach. Joshi [1] defined POS tagging for Hindi language using HMM base tagger with 24tagset and attend accuracy of 92.13%. Modi [2] This tagger accepts data in Devanagari Hindi use Rule based approach which includes grammatical rules and regular expression,100% correctness for split and tokenizer,91.84% for POS tagging. This system achieved 91.84% of average precision and 85.45% of average accuracy. Ekbal [3] statistical maximum entropy model.13 is used for Bengali Part of Speech Tagging, ME system has been compared with HMM based POS tagger in which trigram model is considered. Performance of ME POS tagger is better than HMM based POS tagger and also worked with [7] Conditional Random Field features that used *Context word feature, *Word suffix, *POS information *Name entity information, *Length of word, *Lexicon feature and perform work with two different data set with the following accuracy of 92.1% and 90.3% etc. Dandapat [4] developed a hybrid tagging model, the training model is based on partially supervised learning, for supervised and unsupervised learning HMM is used. This POS tagger dealt with Chalitabhasha of Bengali with 40 different tags are used. Singh [5] POS tagger for morphologically rich language Hindi using Decision tree-based learning algorithm CN2(Clark and Niblett, 1989). The accuracy obtained by simple lexicon lookup, based approach is 61.19%. If multiple tags for a word is considered as an error then the accuracy is 73.62%. Final accuracy was gained by the learning-based tagger after 4-fold cross validation, and the result is 93.45% which is quite satisfactory. Dandapat [6] worked on POST for Bengali with HMM use combination of supervise (HMM-S) and unsupervised (HMM-SS) model. These models are experimented with (HMM-S+IMA and HMM-SS+IMA) show complete morphological restriction and (HMM-S-CMA and HMM-SS-CMA) show complete morphological restriction. Singh [8] POS tagger for Marathi language using Trigram model Corpus consists of 2000 sentences i.e. 48,635 words Accuracy gained is 91.63%. Patil [9] use rule base POS tagger for Marathi language, corpus consists of 576 unique words and are manually tagged by 9 tag sets and for evaluation of the system three different tag sets are formed. The highest accuracy gained in the third dataset of 83.34% Shambhabi [10] represents a POS tagger for Kannada language using second order Hidden Markov Model and Conditional Random Fields. This language is both agglutinative and morphologically rich, 25 tags are used. 51269 words are used for training and 2932 tokens for test data. over all count for known token using HMM is 79.91% and using CRF is 84.58%. PVS [11]. POS tagging and chunking for HINDI, BENGALI and TELUGU using conditional random field and Transformation based learning. Window size used for Hindi, Bengali and Telegu are 6, 6 and 4 respectively. The accuracy seemed to be lower in Telugu because of its agglutinative nature.

Revised Manuscript Received on April 07, 2019.

Aarti Singh, Department, of Computer Science, Banasthali Vidyapith, Rajasthan, India.
Nisheeth Joshi, Department, of Computer Science, Banasthali Vidyapith, Rajasthan, India.

Singha [12] POS tagger for Manipuri Language is developed using Stochastical model i.e. hidden markov mode and Viterbi algorithm for finding maximum probability. Tag set of 97 morphosyntactic categories of Manipuri is used and manually annotated accuracy obtained is 92%. Nakagawa [13] worked on Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines. Three features - POS context, word context and substring are used for predicting POS tag for unknown words. 50 POS tags are used and highest accuracy gained is 97.1% using Preceding and Succeeding POS tags. Barman [14] This paper presents POS tagging for Assamese language using CRF and TL. Two corpora are used, one is manually annotated corpus containing nearly 10000 words and another one is raw corpus. Accuracy gained by TBL is 87.17% and accuracy gained by CRF is 67.73%. TBL gives more accurate rate than CRF tagger. Kumar [15] This paper demonstrates POS tagging for Hindi Language. Hybrid (Rule based+ statistical) approach is used for tagging the tokens. In Statistical approach HMM model is used because of the ambiguity of the words. Three evaluation metrics are used. Final accuracy for Precision is 96.46%, for Recall is 90.13%, for F-measure is 93.17%. Paul [16] This paper presents a POS tagger for Nepali language using Hidden Markov Model.

Annotation tool SANCHAY is used which contain around 1,50,839 tagged words. Including generic attributes and language specific attribute. 42 tags are included in the tag set. Supervised training method is used and for implementing the system NLTK library is used. Accuracy obtained is 96%. Mishra [17] In this paper they worked on POS Tagging for Resource Poor Indian Languages Through Feature Projection. They present an approach for POS tagging without any labelled data. They use two data sets for their experiments. (1). ILCI (Indian Languages Corpora Initiative) parallel corpora. (2). Hindi Tree-bank. The algorithm which they followed has following step: Word Alignment, Feature Selection, Mapping File Creation, Creation of training model. They experiments on following language with following accuracies : Gujarati (Accuracy - 86.2, Train- 380K, test-6.5K), Punjabi (Accuracy -84.3 , Train-450K, test-9K), Urdu(Accuracy -85.6,Train- 450K, test- 7.5K), Konkani (Accuracy -76.9, Train- 380K,test- 7.5K), Bengali (Accuracy of 75.5, Train-380K, test-7.5K), Telugu (Accuracy of 74.2, Train-380K, test-7K), Marathi (Accuracy of 77.7, Train- 380K, test- 7K), Malayalam (Accuracy of 65.01 Train-380K, test- 6K). Gupta [18] In This paper they worked on advanced NLP learning resources in Indian languages Hindi and Urdu. Their research based on domain-specific platform such as health, tourism and agriculture corpora with 60k sentences. They developed stemmer, lemmatize, POS tagger, and MWE identifier and having following accuracy stemmer, lemmatize, POS tagger, and MWE identifier are 77.0, 86.8, 73.20, and 43.50% for Hindi and 74.0,85.4, 84.97 and 47.2% for Urdu, respectively. For tagged corpora using CRF++ tool.

III. PURPOSED METHODOLOGY

A. *TRIGRAM TAGGER:*

In language modelling assigning the probability to whole sequence of word P ($w_1 \ldots w_n$) such as P ($w_i \ldots\ldots\ldots w_n$) = P(w_i / $w_{i-1} \ldots w_{i-n}$) this is known as N-Gram modelling. N-Gram use only n-1 word of prior context to assign the tag. There are unigram, bigram, trigram mode etc. We will only discuss trigram model. Which is based on markov mode it is simply a stochastic process in which distribution of future state is depend on present state. To implement Trigram Tagger following equation is used:

$$P(t/w) = \text{argmax}\ (P\ (t_i / t_{i-2}, t_{i-1}) * P\ (w_i / t_i)) \quad \ldots\ldots\ldots 1$$

Here in equation (1) calculate the probability of tag given word is calculated by finding the maximum probability of tag transition probability multiply by, word-tag probability (Likelihood Probability). Now equation for tag transition probability and word-tag probability (Likelihood Probability) as follow.

Calculate Tag - Tag probability (Transition Probability):

$$P\ (t_i / t_{i-2}, t_{i-1}) = \frac{\text{count}\ (t_i, t_{i-2}, t_{i-1})}{\text{count}\ (t_{i-2}, t_{i-1})} \quad \ldots\ldots 2$$

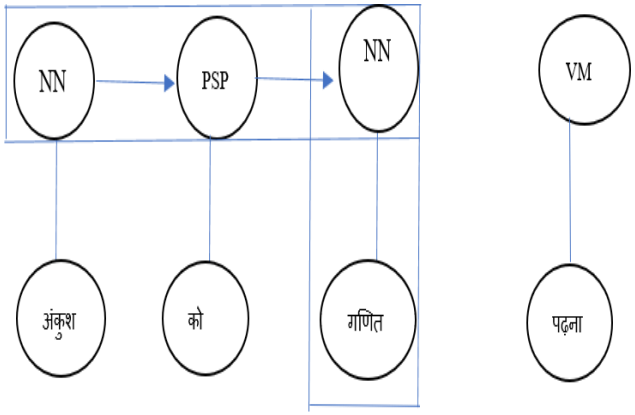
In equation (2) transition probability of trigram tagger. It counts the probability of prior two tag to calculate the current tag probability and divide by probability count of prior two tag found together in corpus.

Calculate word tag probability (Likelihood Probability)

$$P\ (w_i / t_i) = \frac{\text{count}\ (t, w)}{\text{count}(t)} \quad \ldots\ldots\ldots\ldots 3$$

In likelihood Probability to calculate tag probability given word is done by counts the word tag Probability and divide by the probability count of tag.

Example of Trigram tagger



Trigram model works when the word गणित will be tagged as NN. So according to equation (2) and (3) we compute equation (1).

Now according to equation (2)

$$P\ (NN / PSP, NN) = \frac{\text{count}\ (NN, PSP, NN)}{\text{count}\ (PSP, NN)} \quad \ldots\ldots 2$$

It counts probability of NN
given probability of PSP, NN together.
Now according to equation (3)

$$P\ (\text{गणित} / NN) = \frac{\text{count}\ (NN, \text{गणित})}{\text{count}\ (NN)} \quad \ldots\ldots\ldots\ldots 3$$

It counts probability of गणित given probability of NN.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

By equation (2) and (3) we calculated Transition and Likelihood probability and put in equation (1) to compute the trigram model.

P (NN / गणित) = argmax (P (NN / PSP, NN) * P (गणित/ NN)

Implementing Trigram in following way:

1. **Input**: a sentence to tagged
2. Tokenize the sentence.
3. Load tag transition (tag-tag) probability in tt array
   // tt[0], tt[1], tt[2]
4. Load likelihood s(word-tag) probability in wt array
   //wt[0] =word-tag probability, wt[1]=tag wt[2]=word
5. For i: = w_1 to w_n // (w_i……w_n Tokenize words)
6. C = 0
7. If w_i has a unique tag in wt array
   Then (i) ti = wt[1]
   (ii) break
8. For each wt [1] of w_i
   Search wt [1] in tt array
   If tt [2] = wt [1] then
   Pr [c]=tt [0]. tt[1] . tt[2]
   Pr[c]= pr[c] . wt[2]
   C++
9. t[i] = max (pr [c])
10. for i = 1 to n
11. print w_i + "/" + t_i

B. HMM Tagger:

Hidden Markov model (HMM) it calculates forward and backward probability of tag in a given sequence such as:

$$P(t_i/w_i) = P(t_i/t_{i-1}).\ P(t_{i+1}/t_i).\ P(w_i/t_i) \ \ldots\ldots\ldots 4$$

In this model state represent as tag and observation state represent as word. The transition probability depends on state which consist the probability of tag. In the eq. (4) P(ti/ti-1) the probability of a current tag given the previous tag and P (ti+1/ti) is the probability of the future tag given the current tag. This transition probability of tag is calculated as:

$$P(t_i / t_{i-1}) = \frac{\text{count}(t_{i-1}, t_i)}{\text{count}(t_{i-1})} \ldots\ldots\ldots 5$$

In eq. (5) probability of two tag together seen and divided by previous tag independent in corpus.

HMM capture the context to tag the sentence correctly as one tag is use to predict the other tag for example in grammar rules adjective (JJ) will be followed by a common noun (NN) and not by a postposition (PSP) or a pronoun (PRP).

पुरानी	किताब	पुरानी	ने	पुरानी	वह
JJ	NN	JJ	PSP	JJ	PR
Prob (0.6161)		Prob (0.00107)		Prob (0.00153385)	

Fig 1. Tag transition probabilities

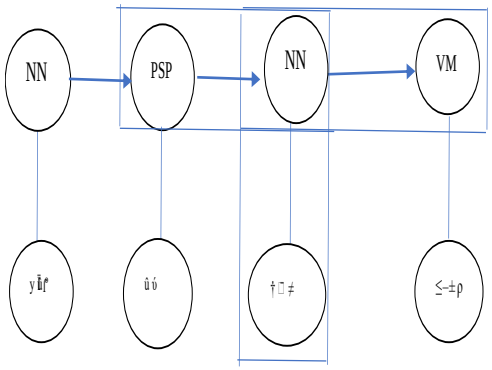
So, the P(JJ|NN) is more probability. Then the P(JJ|PSP) and P(JJ|PRP).

In eq. (4) P(wi/ti) is word likelihood probability. Calculating probability of word given tag. This word likelihood probability is calculated as count the probability of word and tag together divide by count the probability of tag alone in the corpus.

To calculate likelihood probability of word given tag:

$$P(w_i / t_i) = \frac{\text{count}(t_i, w_i)}{\text{count}(t_i)} \ldots\ldots\ldots 6$$

Example of HMM tagger



As the HMM has feature to capture the context forward and backword. So, in this example it tag गणित as NN by capture forward and backward probability of tags.

Algorithm for HMM tagger is as follows:

1. **Input:** a sentence to tagged
2. Tokenize the sentence
3. Load tag transition (tag-tag) probability in tt array.
   // tt[0], tt[1], tt[2]
4. Load likelihood (word-tag) probability in wt array.
   //wt[0] = word-tag , wt[1] = tag, wt [2] = word
5. For i = w_i to w_n // (w_1……w_n) tokenize words
6. c= 0
7. If w_i has a unique tag in wt array
   Then (i) ti = wt [1]
   (ii) break
8. For each wt [1] of w_i
   Search wt [1] in tt array
   If tt [1] = wt [1] then
   Pr [c]=tt [0]. tt [1]. tt [2]
   Pr[c]=pr [c]. wt[2]
   c ++
9. t[i] = max (pr[c])
10. For i = 1 to n
11. print w_i + "/" + t_i

IV. EVALUATION

We evaluated our system using on 500 sentences using accuracy as evaluation measure. It is computing as No. of correct tags identified divided by the total number of tags.

This is shown in equation 8.

$$\text{Accuracy} = \frac{\text{No. of correct POS tag assigned by the system}}{\text{No of POS tags in the text}} \ldots.8$$

In all these 500 sentences had 6000 words. Out these 6000 words, trigram tagger was able to correctly tag 5823 words and HMM tagger was correctly identified were 5897 tags. Thus, the accuracy of trigram and HMM tagger was 97.05% and 98.28% respectively.

V. CONCLUSION

In this paper we have shown the development of two POS tagger on Hindi. These were based on Markov Model. The two algorithms implemented were trigram-based approach and HMM based approach. Among these, trigram tagger gave an accuracy of 97.06% and HMM tagger gave an accuracy of 98.28%.

In future we would like to train more POS taggers based on some ML approaches like Decision Trees and Naïve Bayes. We also plan to develop POS tagger based on neural models, specifically recurrent neural networks and Long Short Term Memory (LSTM).

REFERENCES

1. J. Nisheeth, D. Hemant and M. Iti, “HMM based POS tagger for Hindi”, International Conference on Artificial Intelligence, Soft Computing, pp. 341-349, 2013.
2. M. Deepa and N. Neeta, "Part-of-Speech Tagging of Hindi Corpus Using Rule-Based Method.", the International Conference on Recent Cognizance in Wireless Communication & Image Processing, pp. 241-241,2016.
3. E. Asif, H. Rejwanul and B. Sivaji, "Maximum entropy based bengali part of speech tagging.", A. Gelbukh (Ed.), Advances in Natural Language Processing and Applications, Research in Computing Science (RCS), pp 67-78, 2008.
4. D. Sandipan, S. Sudeshna and B. Anupam, "A Hybrid Model for Part-of-Speech Tagging and its Application to Bengali.", International conference on computational intelligence, pp 1-4, 2004.
5. S. Smriti, G. Kuhoo, S. Manish and B. Pushpak, . "Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi.", COLING/ACL on Main conference poster sessions. Association for Computational Linguistic, pp. 779-786, 2006.
6. D. Sandipan and S. Sudeshna, "Part of speech tagging for bengali with hidden markov model.", the NLPAI Machine Learning Competition, 2006
7. E. Asif, H. Rejwanul and B. Sivaji, "Bengali part of speech tagging using conditional random field.", seventh International Symposium on Natural Language Processing, pp. 131-136, 2007
8. S. Jyoti, J. Nisheeth and M. Iti, Part of speech tagging of Marathi text using trigram method." International Journal of Advanced Information Technology (IJAIT), Vol, pp 35-41,2003
9. P. H.B, P. A.S and P. B.V, "Part-of-Speech Tagger for Marathi Language using Limited Training Corpora.", International Journal of Computer Applications, pp. 33-37, 2014
10. P.K. Ramakanth and R. B. Shambhavi, "Kannada part-of-speech tagging with probabilistic classifiers.", international journal of computer applications, vol, pp. 26-30, 2012
11. PVS. Avinesh and G. Karthik, "Part-of-speech tagging and chunking using conditional random fields and transformation based learning.", Shallow Parsing for South Asian Languages, pp. 21-28, 2007
12. S. Kh Raju, P. Bipul Syam, and S. Kh Dhiren, "Part of Speech Tagging in Manipuri with Hidden Markov Model." International Journal of Computer Science Issue, vol, pp. 146-149, 2012.
13. N. Tetsuji, K. Taku and M. Yuji, "Unknown Word Guessing and Part-of-Speech Tagging Using Support Vector Machines.", NLPRS, 2001
14. B. Anup Kumar, S. Jumi and S. Shikhar Kr, "POS Tagging of Assamese Language and Performance Analysis of CRF++ and fnTBL Approaches.", UKSim 15th International Conference on Computer Modelling and Simulation, pp. 476-479,2013.
15. Kumar, Rajesh, and Sayar Singh Shekhawat. "PARTS OF SPEECH TAGGING FOR HINDI LANGUAGES USING HMM.", INTERNATIONAL JOURNAL OF SCIENTIFIC RESEARCH, 2018.
16. P. Abhijit, P. Bipul Syam and S. Sun ita, . "Hidden Markov model-based part of speech tagging for Nepali language.", International Symposium on Advanced Computing and Communication (ISACC), 2015.
17. M. Pruthwik, M.Vandan and S. Dipti Misra, . "POS Tagging For Resource Poor Indian Languages Through Feature Projection.", 2018.
18. G. Vaishali, J. Nisheeth and M. Iti, "Advanced Machine Learning Techniques in Natural Language Processing for Indian Languages.", Smart Techniques for a Smarter Planet, pp. 117-144,2019.

AUTHORS PROFILE

Aarti Singh is a student at Banasthali Vidyapith. Her areas of interest are natural language processing and machine translation.

Nisheeth Joshi is an Associate Professor at Banasthali Vidyapith, India. He primiraliy works in Machine Translation, Information Retrieval, Cognitive Computing. He has over 12 years of teaching experimence.