Journal of information and communication convergence engineering
http://jicce.org

J. Inf. Commun. Converg. Eng. 18(4): 207-215, Dec. 2020 Regular paper

A Semi-supervised Learning of HMM to Build a POS Tagger for a Low Resourced Language

Sagarika Pattnaik[1*], Ajit Kumar Nayak[2], and Srikanta Patnaik[3]

[1]Department of Computer Science and Engineering, SOA, Deemed to be University, Bhubaneswar, 751030, India
[2]Department of Computer Science and Information Technology, SOA, Deemed to be University, Bhubaneswar, 751030, India
[3]Department of Computer Science and Engineering, SOA, Deemed to be University, Bhubaneswar, 751030, India

Abstract

Part of speech (POS) tagging is an indispensable part of major NLP models. Its progress can be perceived on number of languages around the globe especially with respect to European languages. But considering Indian Languages, it has not got a major breakthrough due lack of supporting tools and resources. Particularly for Odia language it has not marked its dominancy yet. With a motive to make the language Odia fit into different NLP operations, this paper makes an attempt to develop a POS tagger for the said language on a HMM (Hidden Markov Model) platform. The tagger judiciously considers bigram HMM with dynamic Viterbi algorithm to give an output annotated text with maximum accuracy. The model is experimented on a corpus belonging to tourism domain accounting to a size of approximately 0.2 million tokens. With the proportion of training and testing as 3:1, the proposed model exhibits satisfactory result irrespective of limited training size.

Index Terms: HMM, NLP, Odia, POS tagger

I. INTRODUCTION

Part-of-speech (POS) tagging or grammatical tagging, an integral part of significant natural language processing (NLP) models, is an active research topic at present. It is the process of labeling words or tokens of a sentence to their appropriate lexical category, such as noun or pronoun, based on its definition and context [1]. A POS tag of a word not only defines it, but also provides information about its surrounding lexical categories. If a word is tagged as a noun, this indicates that its preceding word may be an adjective or determiner. Its application is noticed in various language processing activities such as automatic text summarization (extracting the significant words), named entity recognition (key feature in identifying entities), and machine translation (to obtain the correct meaning without ambiguity) [2]. Solving ambiguity by considering the morphological complexities of various languages has been a major issue. The process is also limited by the availability of required resources, such as a sufficient training corpus and linguistic information related to the language. Thus, varied techniques have been attempted for the process, such as rule-based and stochastic methods that comprise methods including support vector machine (SVM), HMM, and maximum entropy (ME) [3, 4]. Rule-based methods have good accuracy and were attempted first [5]; their implementation requires deep linguistic knowledge and a large set of rules that is difficult to achieve. Second rule-based methods are not generalized methods; they are language dependent. Therefore, researchers have moved on to statistical methods. These methods have been tested in various languages and have shown their efficacy in solving the problem [6-9]. As far as Indian languages are concerned, especially the Odia language, very few NLP activities

Received 02 July 2020, **Revised** 25 November 2020, **Accepted** 22 December 2020
***Corresponding Author** Sagarika Pattnaik (E-mail:sagarika.pari@gmail.com, Tel:+91674-2553540)
Department of Computer Science and Engineering, SOA, Deemed to be University,Bhubaneswar, 751030, India.

Open Access https://doi.org/10.6109/jicce.2020.18.4.207 print ISSN: 2234-8255 online ISSN: 2234-8883

are found. An efficient automatic POS tagger is needed for the present scenario. Odia, which has been declared as one of the classical languages of India and spoken by 45 million people, including Odisha and its neighboring regions, is in a computationally dormant stage. It is necessary to carry out various NLP activities to bring Odia into the digital platform. POS tagging is the fundamental step in this path. The proposed work suggests an HMM-based POS tagger integrated with a dynamic Viterbi algorithm for the Odia language. The model adopts a supervised bigram approach. The corpus of the tourism genre has been considered for implementing the model. This work can be regarded as a novel attempt.

Section II does a brief literature survey on the works done related to the proposed model. Section III discuses the morphology of the language, an essential element to be considered in the development of any POS model. Section IV gives a comprehensive description of the technique adopted by the model. Section V does a detailed discussion of the implementation of the proposed model with its result analysis and finally the work is concluded in section VI with a future direction.

II. LITERATURE SURVEY

Though POS tag ging has started with primitive approaches like rule based [10], it soon adopted statistical means [5] like ME, SVM, HMM etc. or a combination of both [11], giving a promising result. HMM a statistical method has been a befitting approach for several languages of the globe. A classical POS tagger for English languageusing untagged corpus for training and adopting HMM [12] have been reported of giving a performance of 96%. The model is evaluated on Brown corpus. The flexibility of the model has been tested on French language. Similarly the HMM approach has been witnessed for Myanmar language [13] achieving an accuracy of 96.56%.

A tagger based on same principles for Indonesian language has been recorded [14]. It exploits the linguistic features of the language and is experimented on a corpus size of 15000 tokens. It has got an accuracy of 96.5%. Another POS tagger for the same language [15] has also been developed using HMM and affix tree and have got higher performance. Similarly Cahyani et al. [16] developed a tagger for Indonesian language using both bigram and trigram HMM with Viterbi decoding. The bigram has a greater performance of 77.56% than trigram approach having 61.67% as accuracy result. Indonesian manually tagged language corpus is used as the knowledge base for the system. A POS tagger for seven prominent languages Serbian, Czech, Hungarian, Estonian, Romanian, Slovene and English developed on the same platform is reported [17]. They have adopted a TNT trigram tagger and achieved satisfactory result. An Arabic POS tagger based on HMM [18] for doing linguistic level sequencing and a morphological analyzer doing a word level analysis to make use of the derivational nature of the language is recognized. A corpus has been developed using texts from old books Hijri for the experiment and has given a promising result of 96%. Another POS tagger for Arabic language using Quran corpus [19] has also been identified of giving a promising result.

Similarly for Indian languages HMM has marked its existence. A POS model for Hindi has been developed [20] adopting a simple H idden Markov approach with longest suffix matching stemmeras a pre-processor. The model has been reported of giving an accuracy of 93.12%. Another Hindi tagger [11] using hybrid approach has given an accuracy of 89.9%. A combination of rule based and HMM has been adopted for building the model. It is evaluated over a corpus size of 80,000 words. For Kannada a POS tagger [21] has been modeled considering CRF and HMM and claimed of achieving 84.58% and 79.9% accuracy respectively. A bigram HMM POS tagger with dynamic Viterbi decoding for Punjabi language [22] has been reported of giving 90.11% accuracy. The training has been conducted on 20,000 annotated words and the testing on 26, 479 words or tokens. Similarly other Indian languages like Bengali, Tamil have also been explored in this field giving a satisfactory result [6].

Considering the language Odia on which the proposed model has been implemented reports limited work. POS taggers developed for the said language has been approached with techniques like ANN (Artificial Neural Network) and SVM [23, 24] giving an accuracy of 81% and 82% respectively. The taggers have been developed on a training size of 10,000 words. A small tag set of five tags have been considered. Another instance of POS tagger for Odia based on SVM and CRF++ (Conditional Random Field) has also been reported [25] suggesting that SVM performs better than CRF++. Still Odia needs a better robust and transparent model with a deeper categorization of the tokens to their lexical class.

The literature survey concludes that the performance of the tagger based on same methodology varies in accordance with the morphology of the language. Another major issue observed for Indian languages especially Odia language is data sparseness or lack of sufficient resource to carry out research activities. This limitation has ended up in making Odia language lag behind in the race. Thus the language has to be computationally explored and has been the instinct of motivation to carry out the proposed work.

III. MORPHOLOGY OF ODIA LANGUAGE

Every language has its unique morphology and script and accordingly the NLP operations vary. The syntactic struc-

ture, the chronology of the words and their lexical classes are the key features in determining the performance of a computational model [5]. This section describes the essential features of the Odia language that play a significant role in determining the suitability and efficiency of the proposed model. The said language has a rich morphology [26, 27, 28] i.e. the number of word forms per lexeme in Odia is more compared to English. It is also agglutinative in nature. Unlike English the sentences comprises of postpositions instead of prepositions and are mostly attached to the noun/ pronoun word. Few exist separately.

e.g. ଗତକାଲିଠାରୁ, ମା'ଠାରୁ, ପାହାଡ଼ତଳେ.

Transliterated: gatakāliṭhāru, mā'ṭhāru, pāhāḍataḷe.
The postposition

"ଠାରୁ" ("ṭhāru")

isattached with the main word whereas the postposition

"ତଳେ" (taḷe)

is separate.
The suffixes and prefixes play a major role in identifying some major lexical categories like noun and verb.
e.g.

"ମାନଙ୍କୁ" ("mānaṅku")

suffix in the noun word

"ରାଜାମାନଙ୍କୁ" ("rājāmānaṅku")

and

"ଥାଏ" ("thāe")

suffix in the verb word

"ବସିଥାଏ" ("basithāe")

• The transliterated form of the Odia words are in bracket.

The script does not comprise of capital letters to distinguish proper nouns. Unlike English the order "subject object verb" (SOV) is followed [29]. There are no identifiers to identify genders.Thus these few examples of syntactical featureshave a major impact in the design and performance of a POS tagger. To have complete lexical knowledge of the language is a cumbersome task and statistical approach like HMM is one of the means to achieve the goal and is explained in the following section.

IV. HIDDEN MARKOV MODEL (HMM)

HMM is an effective statistical sequential model on which a POS tagger can be built. It meticulously exploits lexical knowledge acquired from input Odia training corpora. It gives a label to each unit (word or token) in the sequence and maps the series of observations into a chronology of befitting classes. Thus HMM enumerates the probabilistic distribution over each possible sequence and chooses the leading label sequence. The modeled system is assumed to be a Markov process and predicts the sequence of labels for each unit in the input observable sequence whose states are hidden [30]. Thus an HMM is built on the following components:

- A set of sequence of observations w_1 $w_2 \ldots w_k$ drawn from a vocabulary V.
- A set of states $t_1 t_2 \ldots t_N$ that are hidden i.e. the tag set considered.
- Emission probability

The probability of occurrence of a word depends only on its own tag, not on its neighboring tags or words.

$$P(w_1^n / t_1^n) \approx \Pi_{i=1}^{n} P(w_i / t_i) \quad (1)$$

Thus (1) gives a set of probable tags $t_i$for an observed event i.e a word or a token w_i with their emission probability values. It gives the probability of emission of an event w_i given a state $t_{i..}$. The value of i varies from 1 to n the length of the sequence.

• Transition probability

It gives the probability of occurrence of a tag $t_i$given its previous tag t_{i-1}. Probability of a tag t_i is dependent only on its previous tag t_{i-1} (bigram) not on the entire previous tag sequence.

$$P(t_1^n) \approx \Pi_{i=1}^{n} P(t_i / t_{i-1}) \quad (2)$$

Thus (2) derives a set of transition probabilities from the training corpus at a click of an observable event.

Finally the best tag sequence T^* for the observed word sequence is equated as:

$$T^* = ar\ gmax P(T/W)$$

$$\approx ar\ gmax \prod_{i=1}^{n} P(w_i / t_i) P(t_i / t_{i-1}) \quad (3)$$

The experimented HMM model proceeds from left to right of the given observable word sequence. Fig. 1 gives a structural representation of the equations adopted by HMM.

To the basic HMM a dynamic Viterbi algorithm when applied for decoding finds the best tag sequence. It computes the leading path without specifying all paths explicitly.

$$V(t,i) = max : V(t_{t-1}, i-1) P(t_i / t_{i-1}) \cdot P(w_i / t_i) \quad (4)$$

where $V(t, i)$ is the probable Viterbi value generated at a state i.

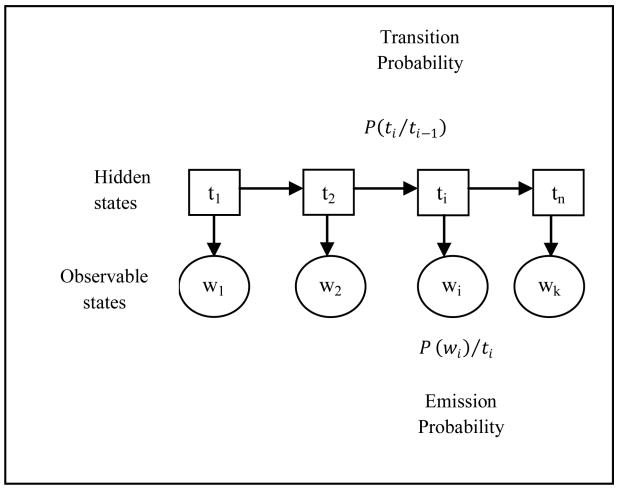

Fig. 1. The HMM structure.

$V(t_{t-1}, i-1)$ is the previous Viterbi value.

If the number of states for a token is more the beam search algorithm in combination with Viterbi reduces the complexity of the algorithm. The number of states is maintained according to the predefined beam size, other states which contribute a low probability value are not carried forward. In our proposed model considering the statistic of occurrence of state for a token, the beam size is kept at 2. Thus the efficiency of predicting tag sequence by HMM can be exploited for building a promising POS tagger for Odia text.

V. EXPERIMENT

This section covers the algorithm adopted for building the model. It briefly describes the corpus and the tag set considered for evaluation.

A. Corpus Information

The tag set adopted for the proposed model has a flat structure with eleven significant tags. The considered tags are set with reference to Unified POS tag set for Indian Languages [31]. Due to the lack of sufficient annotated training data only major tags are considered to get an unambiguous precise result. Table 1 gives a clear description of the considered tags.The considered tags are categorized into open class and closed class [5]. Open class do not have a fixed membership, there are chances of new members to be coined. Like new words are continuously coined under the noun tag (open class). Contrarily closedclass has fixed membership i.e. there is a least chance of new conjunctions to be formed than the predefined ones.

Of the total corpus available 65% is considered for training and 35% is considered for testing. The corpus used for the experiment is Odia monolingual text corpus. It comprises of approximately 0.2 million words. The tagged texts belong to the tourism domain and are collected from TDIL (Technology Development for Indian Languages, govt. of India) [31] for training and testing. Under the project, initiated by the DeitY, Govt. of India, Jawaharlal Nehru University, New Delhi have collected corpus in Hindi as the source language and trans-

Table 1. Tag description

Tags	Description	Example	Transliteration	English transformation
		Open class		
NN	Noun	କଟକ, ଚକ, ଜିନିସ	kaṭaka, chaka, jinisa	Cuttack, wheel, thing
VV	Verb	ଖେଳୁଛି, ପଢିଲି	kheḷuchhi, paḍhili	playing, read
JJ	Adjective	ଗରମ, ଲାଜୁଆ	garama, lājuā	hot, shy
RB	Adverb	ଧୀରେ, ଶୀଘ୍ର	dhire, sighra	slow, fast
		Closed class		
PR	Pronoun	ମୁଁ, ତୁ	mũ, tu	I, you
QT	Quantifier	୧, ଏକ, ସାତଟି	1, eka, sātaṭi	1, one, seven
QW	Question word	କିଏ, କଣ	kie, kaṇa	who, what
RP	Particle	ନାହିଁ, ହିଁ, ନ	nāhĩ, hĩ, na	Negative words and words that cannot becategorized properly to a particular class. Found mainly in association with verbs.
CC	Conjunction	ମଧ୍ୟ, କିନ୍ତୁ	madhẏa, kintu	also, but
PUNC	Punctuation	| , \ ? ( ) { }	The Odia Full stop (|) has a different sign. All other punctuation marks are same as English.	
PSP	Postposition	ଉପରେ, ତଳେ	upare, taḷe	above, below

lated it into Odia as the target language. The chart in Fig. 2 shows the overall distribution of major tags in the corpus.

The structure of the annotated sentence in the source corpus is of the form

htd24001 ନନ୍ଦାକୋଟ\N_NNP ପୀକ\N_NNP ଉତ୍ତରାଖଣ୍ଡାର \N_NNP ପିଥୌରାଗଡ଼\N_NNP ଜିଲ୍ଲାରେ\N_NN ଅବସ୍ଥିତ\JJ | \RD PUNC

Transliterated:

htd24001nandākoṭa\N_NNP pika\N_NNP uttarākhaṇḍaara\N_NNP pithaurāgada\N_NNP jillāre\N_NN abasthita\JJ |\RD_PUNC

Where htd24001 is the sentence-id, N_NNPis proper noun under noun category, N_NN is common noun under noun category, JJ is adjective and RD_PUNC is punctuation under residual tag category.

Without loss of generality we have grouped all sub tags of nouns to one noun, all sub tags of verbs to only verb and so on to keep the experiment simple as follows.

<ST> ନନ୍ଦାକୋଟ_NN ପୀକ_NN ଉତ୍ତରାଖଣ୍ଡାର_NN ପିଥୌରାଗଡ଼_NN ଜିଲ୍ଲାରେ_NN ଅବସ୍ଥିତ_JJ <END>

Transliterated:

<ST>nandākoṭa_NNpika_NNuttarākhaṇḍaara_NNpithaurāgada_NNjillāre_NNabasthita_JJ<END>

The symbols used for the tags are in accordance with Table 1.

B. Adopted Methodology

This section lays out the procedure followed by the proposed model in the form of an algorithm. It also coversthe implementation and evaluation of the model on the Odia test corpus.

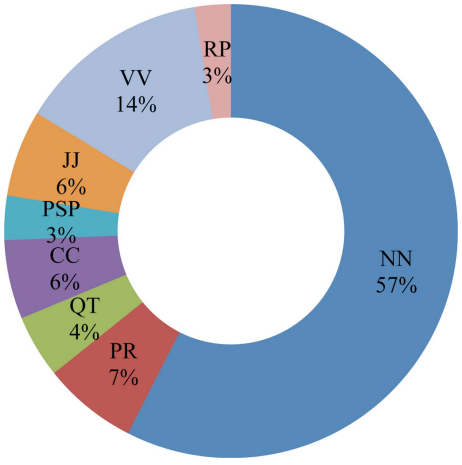

Fig. 2. Distribution of major tags in the corpus.

Algorithm1: HMM adopted for Odia POS tagger

Training Phase:

- *Input set of tagged sentences*
- *Generate all possible emission and transition probability values for each word w $_i$in the training set.*
  Emission Probability: $P(w_i/t_i)$
  Transition Probability: $P(t_i/t_{i-1})$

Testing Phase:

Input:

- Set of untagged sentences $S=\{s_1,s_2,s_3,\ldots,s_j\}$
- $P(w_i/t_i)$ *a set of derived tags for* w_i
- $P(t_i/t_{i-1})$ *a set of previous tag pairs for* t_i

Initialize: $V(t, i)$ = 1 and k=2 (beam size)
For each sentence s_j in the test sentence:
do
  Tokenize the sentence into words w_i.
For each w_i in s_j find the paths of length i-1 to each state:
do
  If $w_i \in$ training set:
Generate the set of values $P(w_i/t_i)$ and $P(T_i/t_{i-1})$ from the training set.
Else:
  tag the token as noun (i.e. t_i= NN) and
    $P(w_i/t_i)$ = *Average probability of* t_i *in the training corpus*
    $P(T_i/t_{i-1})$ values are derived from the training set.
    Compute and store the partial results V(t,i) as the path proceedsaccording to equation (4).
    - Consider the maximum value over each possible previous tag t_{i-1}.
    - Keep highest k probable paths.
      Proceed from left to right of the test sentence.
    - Choose the best path from k available computed paths to reach the final state.
    - Store a back trace to show from which state (i-1) state (i) came from.
    - Return tag sequence (T*) for the given word sequence by back tracing.

done
done

The algorithm traverses the given input sequence of objects or tokens from left to right. This statistical approach does a proper identification of the logical sequence of Odia words and their tags in the text. It takes into consideration the history associated with each word to predict its tag. Fig. 3 gives the pictorially representation of the concept adopted by the algorithm.

Input sentence:

ମହାନଦୀ ସୁନ୍ଦର ଦେଖାଯାଏ

Transliterated: mahānadisundaradekhājāe

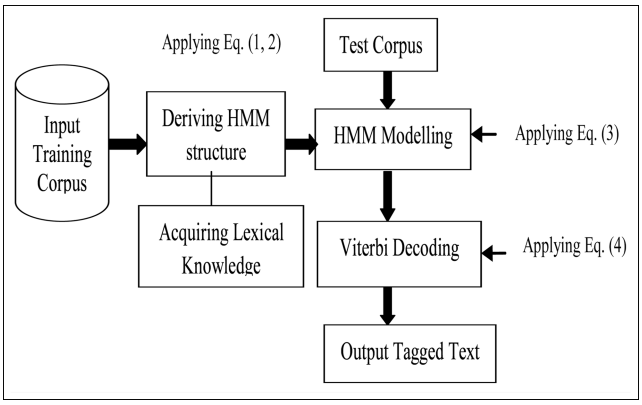

Fig. 3. Odia POS mode.

Output tagged sentence:

ମହାନଦୀ_NN ସୁନ୍ଦର_JJ ଦେଖାଯାଏ_VV

Transliterated: mahānadi_NNsundara_JJdekhājāe_VV

Illustration

The illustration for the stated algorithm will make a clear understanding of the functioning of the proposed model.

- Input sentence s_j

ମହାନଦୀ ସୁନ୍ଦର ଦେଖାଯାଏ

- Tokenized into three tokens w_1, w_2, w_3, where

w_1= ମହାନଦୀ, w_2= ସୁନ୍ଦର, w_3 = ଦେଖାଯାଏ

- The emission and transition probabilities of the tokens calculated from the training corpus are depicted in Table 2 and 3.

At the click of each observable event i.e. a token the set of Viterbi values according to (4) are calculated. The values are calculated for state i=1 to i=N, where N=3.

At state i=1, the previous Viterbi entry $V(t_{i-1}, i\text{-}1)$ is taken as 1 and Viterbi values generated with its corresponding tags for w_1 with reference to Table 2 and Table 3 is as follows: $(w_1/NN) = 1 \cdot 2\times10^{-1} \cdot 3\times10^{-2} = 6\times10^{-3}$ is the probability of w_1 to be tagged as noun. $(w_1/JJ) = 1 \cdot 3\times10^{-3} \cdot 2\times10^{-4} = 6\times10^{-7}$ is the probability of w_1 to be tagged as adjective.

Both the partial results $V(t, i)$ are stored and carried over to the next level, as coming to a decision at this point will not give the best tag sequence.

The probability values of the next token i.e. w_2 to be tagged with a suitable POS is calculated accordingly.

$(w_2/NN)=$

$$\max\begin{Bmatrix} 6\times10^{-3}.\ 1\times10^{-1}.\ 4\times10^{-4} = 24\times10^{-8} \\ 6\times10^{-7}.3\times10^{-2}\ .\ 4\times10^{-4} = 72\times10^{-13} \end{Bmatrix}$$

Table 2. Emission probability

	Tokens		
Tag	**w_1**	**w_2**	**w_3**
NN	3×10^{-2}	4×10^{-4}	
VV			4×10^{-2}
JJ	2×10^{-4}	5×10^{-1}	

Table 3. Transition probability

	t_i				
t_{i-1}	**<ST>**	**NN**	**VV**	**JJ**	**<END>**
<ST>		2×10^{-1}	4×10^{-5}	3×10^{-3}	
NN		1×10^{-1}	3×10^{-1}	1×10^{-2}	
VV		2×10^{-1}	1×10^{-1}	1×10^{-2}	1×10^{-1}
JJ		3×10^{-2}	2×10^{-2}	1×10^{-3}	
<END>					

$(w_2/JJ)=$

$$\max\begin{Bmatrix} 6\times10^{-3}.\ 1\times10^{-2}.\ 5\times10^{-1} = 3\times10^{-5} \\ 6\times10^{-7}.\ 1\times10^{-3}\ .\ 5\times10^{-1} = 3\times10^{-10} \end{Bmatrix}$$

The maximum values are considered and are stored as Viterbi entries $V(t,i)$ with a track of their previous level tag entry.

$(w_3/VV)=$

$$\max\begin{Bmatrix} 24\times10^{-8}.\ 3\times10^{-1}.\ 4\times10^{-2} = 2.88\times10^{-9} \\ 3\times10^{-5}.\ 2\times10^{-2}\ .\ 4\times10^{-2} = 24\times10^{-9} \end{Bmatrix}$$

The <END> will have an emission probability 1, as there are no different types of endings. The final entry will have the probability value 24×10^{-10}.

Fig. 4 gives a clear description of the concept. The values in the ellipse contain the tags generated for each token with their emission probability values. The solid boxes contain the Viterbi values retained for consideration. The final Viterbi path generated according to the algorithm are marked by solid lines.

By back tracing the final tag sequence generated is “NN JJ VV”.

After implementation of the model considering 35% of the available corpus for testing, the result is analyzed. Fig. 5 shows the performance of the model with the increase in training size. The horizontal axis of the graph shows the training data size in terms of the number of files where each file size counts approximately 11,000 words/tokens. The vertical axis shows the accuracy achieved corresponding to the training data size considered.

To validate our model K-fold cross validation is carried out. For a machine learning technique with limited corpus size validation is an essential step. It ensures that the corpus considered is a balanced data and all of it takes part in the training and testing process. The result of our validationpro-

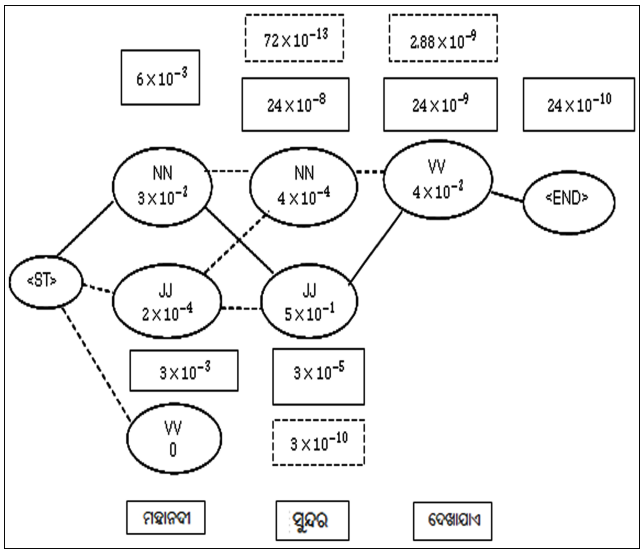

Fig. 4. The Viterbi path derived for tag prediction.

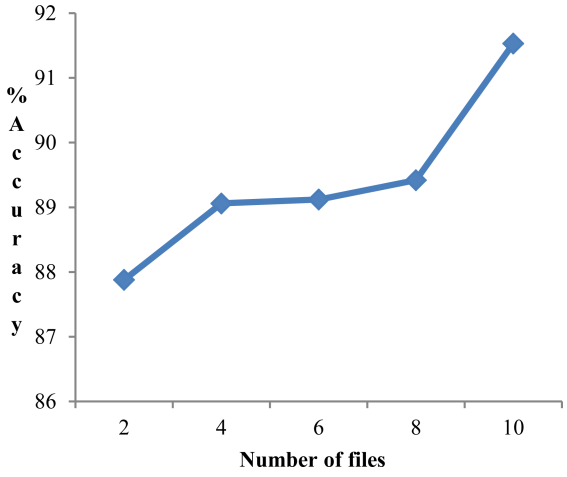

Fig. 5. Accuracy % with the increase in training size.

Table 4. % Accuracy at each iteration

	Average % accuracy	Variance	Skewness
Iteration 1	89.52		
Iteration 2	89.90	0.16	0.21
Iteration 3	90.34		

cess is depicted in Table 4.

The small value of variance signifies that the % accuracy obtained are not far away from the mean value and the skewness value that is near to zero suggests that the corpus considered is balanced with minor inconsistencies.

Table 5 shows the tagging accuracy for each tag and mistags done by the model in the form of a confusion matrix. It depicts the ambiguity levels in tags.

The F score value obtained for each tag is represented in Table 6. It is a better way of evaluation of the proposed model as it takes both precision and recall into consideration.

The results obtained are clearly analyzed in section C.

C. Results Analysis

It can be observed from Fig. 5 that with the increase intraining data size the tagger is able to perform with a higher accuracy. The tagging accuracy for tags noun, pronoun, verb, postposition and conjunction has got a satisfactory result as can be seen from the confusion matrix Table 5. But, for the tags adjective, adverb and particle the accuracy level is low. These tags have a high intensity of ambiguity that couldn't be solved due to unavailability of sufficient training data. Most of the adverbs are wrongly tagged as adjective. This could be resolved by the availability of sufficient training-data and removing the prevailing few inconsistencies. The accuracy level for question words has got a moderate value. Most of the question words are wrongly tagged as pronoun. The inconsistency in the training data adds up to the error percentage existing in the model. The computed F score valuefor each tag (Table 6) shows a satisfactory result. Overall,

Table 5. Confusion matrix in percentage

	NN	VV	JJ	CC	PR	QT	QW	RP	RB	PUNC	PSP
NN	**96.54**	0.421	1.91	0.114	0.20	0.25	0	0.155	0	0	0.362
VV	13.2	**83.87**	2.6	0.04	0.025	0	0	0.038	0	0	0.23
JJ	42.35	0.615	**51.66**	0	0.48	0.524	0	0.365	0	0	0
CC	3.62	0.133	0	**92.87**	1.93	0.23	0	0.66	0	0	0.53
PR	3.81	0	0	1.59	**92.46**	0	0.72	0.15	0.48	0	0.78
QT	7.5	0	4.07	0.11	0.388	**85.87**	0.15	1.9	0	0	0
QW	0	0	0	0	25	0	**75**	0	0	0	0
RP	13.36	0.74	5.84	7.89	1.80	1.74	0	**67.37**	0.18	0	1.05
RB	4.35	1.35	9.35	4.35	6.7	0	0	4.35	**52.17**	0	17.4
PUNC	0	0	0	0	0	0	0	0	0	**100**	0
PSP	10.98	0.14	0	0.19	0.67	0	0	1.93	0.193	0	**85.87**

Table 6. F score value obtained for each tag

Tag	F score
NN	92.44
VV	90.08
JJ	62.19
CC	92.69
PR	92.90
QT	89.79
QW	45
RP	75.35
RB	36.08
PUNC	100
PSP	88.24

the highest accuracy achieved by the tagger with available maximum size training data was 91.53%.

VI. CONCLUSION AND FUTURE WORK

The proposed tagger can be considered as a promising model achieving a satisfactory result. The error in the tagger can be accounted to ambiguity nature of the tags and to some extent to the existing inconsistencies in the training data which needs revision by the linguists. Still with the present performance level the proposed model can be considered novel with respect to Odia language and a contribution to the society. Unavailability of sufficient bench mark training Odia corpus required for developing statistical models is a major issue that needs to be solved. The model can be enhanced by addition of linguistic features, giving it hybrid attire and a direction for future research. The model has been proposed with an aim of having its application in the development of an auto text summarizer for Odia language.

REFERENCES

[1] T. Siddiqui and U. S. Tiwari, *Natural Language Processing and Information Retrieval*, Oxford University Press, pp.77-88, 2008.

[2] D. N. Mehta and N. Desai, "A survey on part-of-speech tagging of Indian languages," In *1st International Conference on Computing, Communication, Electrical, Electronics, Devices and Signal Processing*, vol. 34, 2011. DOI: 10.13140/RG.2.1.3595. .2481.

[3] F. Md. Hasan, U. Naushad, and K. Mumit, "Comparison of different POS tagging techniques (N-Gram, HMM and Brill's tagger) for Bangla," *Advances and Innovations in Systems, Computing sciences and Software engineering*, Springer, Dordrecht, pp.121-126, 2007. DOI: 10.1007/978-1-4020-6264-3_23.

[4] S. G. Kanakaraddi and S. S. Nandyal, "Survey on parts of speech tagger techniques," *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, Coimbatore, pp. 1-6, 2018. DOI: 10.1109/ICCTCT.2018.8550884.

[5] D. Jurafsky and J. H. Martin, *Speech and Language Processing: An introduction to speech recognition*, computational linguistics and natural language processing, 2nd Edition, 2008.

[6] P. J. Antony, A.V. Vidyapeetham, and K.P.Soman, "Parts of speech tagging for Indian Languages: A Literature Survey," *International Journal of Computer Applications,*vol. 34, no. 8, pp. 0975–8887, 2011.

[7] H. Amiri, F. Raja, M. Sarmadi, S.Tasharofi, H. Hojjat, and F. Oroumchian, "A survey of part of speech tagging in Persian," *Data base Research Group*, 2007.

[8] J. H. Kim and J. Seo, "A Hidden markov model imbedding multiword units for Part-of-Speech tagging," *Journal of Electrical Engineering and Information Science*, vol. 2, no. 6, pp. 7-13, 1997.

[9] D. Modi, N. Nain, and M. Nehra, "Part-of-speech tagging for Hindi corpus in poor Resource Scenario," *Journal of Multimedia Information System*, vol. 5, no. 3, pp. 147-154, 2018. DOI: 10.9717/JMIS.2018.5.3. 147.

[10] E. Brill, "A simple rule-based part of speech tagger," *Proceedings of the workshop on Speech and Natural Language*, Association for Computational Linguistics, 1992. DOI: 10.3115/974499. 974526.

[11] K. Mohnot, N. Bansal, S.P. Singh and A. Kumar, "Hybrid approach for part of speech tagger for Hindi language," *International Journal of Computer Technology and Electronics Engineering (IJCTEE)* 4.1, pp. 25-30 2014.

[12] J. Kupiec, "Robust part-of-speech tagging using a hidden Markov model," *Computer Speech & Language,* vol. 6, no.3,pp.225-242, 1992.DOI: 10.1016/0885-2308(92)90019-Z.

[13] K. K. Zin and N. L.Thein, "Part of speech tagging for Myanmar using hidden Markov model," *2009 International Conference on the Current Trends in Information Technology (CTIT)*, pp. 1-6,2009. DOI: 10.1109/CTIT.2009.5423133.

[14] A. F. Wicaksono and A. Purwarianti, "HMM based part-of-speech tagger for bahasa Indonesia," In *Fourth International Malindo Workshop*, Jakarta. Aug. 2010.

[15] U. Afini and C. Supriyanto, "Morphology analysis for Hidden Markov Model based Indonesian part-of-speech tagger," in *2017 1st International Conference on Informatics and Computational Sciences (ICICoS)*, pp. 237-240, Nov. 2017. DOI: 10.1109/ICICOS.2017.8276368.

[16] D. E. Cahyani and M. J. Vindiyanto, "Indonesian Part of Speech tagging using Hidden markov model – Ngram& Viterbi," in *2019 4th International Conference on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, Indonesia, pp. 353-358, 2019. DOI: 10.1109/ICITISEE48480.2019.9003989.

[17] Z. Agic, M. Tadic, and Z. Dovedan, "Investigating Language Independence in HMM PoS/MSD-Tagging," *ITI 2008 - 30th International Conference on Information Technology Interfaces*, Dubrovnik, pp. 657-662, 2008. DOI: 0.1109/ITI.2008.4588489.

[18] Y. O. M. ElHadj, I. A. Al-Sughayeir, and A. M. Al-Ansari, "Arabic part-of-speech tagging using the sentence structure," In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt, Apr. 2009.

[19] A. M. Alashqar, "A comparative study on Arabic POS tagging using Quran corpus," In *2012 8th International Conference on Informatics and Systems (INFOS)*, IEEE, May. 2012.

[20] M. Shrivastava and P. Bhattacharyya, "Hindi pos tagger using naive stemming: Harnessing morphological information without extensive linguistic knowledge," *International Conference on NLP (ICON08)*, Pune, India, 2008.

[21] R. B.Shambhavi and R. K. P, "Kannada Part-Of-Speech tagging with probabilistic classifiers," *International Journal of Computer Applications 48.17,* pp. 26-30, 2012. DOI:10.5120/7442-0452.

[22] S. K. Sharma and G. S. Lehal, "Using hidden markov model to

improve the accuracy of Punjabi POS tagger," In *2011 IEEE International Conference on Computer Science and Automation Engineering,*IEEE, vol. 2, pp. 697-701, Jun. 2011. DOI: 10.1109/ CSAE.2011.5952600.

[23] B. R. Das and S. Patnaik, "A novel approach for Odia part of speech tagging using Artificial Neural Network," *Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA)*, Springer, 2014. DOI:10.1007/ 978-3-319-02931-3_18.

[24] B. R. Das, S. Sahoo, C. S. Panda, and S. Patnaik, "Part of speech tagging in Odia using Support Vector machine," *Procedia Computer Science 48*, pp. 507-512, 2015. DOI:10.1016/j.procs. 2015.04.127.

[25] B. Pitambar, "Odia Parts of Speech Tagging Corpora: Suitability of Statistical Models," *M.Phil. Diss. Jawaharlal Nehru University New Delhi*, India, Jul. 2015.

[26] S. Pattnaik and A.K. Nayak, "An Automatic Summarizer for a Low-Resourced Language," *Advanced Computing and Intelligent Engineering*, Singapore, pp. 285-295, 2020. DOI:10.1007/978-981-15-1081-6_24.

[27] K. C. Pradhan, B. K. Hota, and B. Pradhan, *Saraswat Byabaharika Odia Byakarana*, Styanarayan Book Store, fifth edition 2006.

[28] S. C. Mohanty. *Applied Odia Grammar*, A.k. Mishra Publishers private Limited, first edition 2015.ISBN: 978-93-82550-38-9.

[29] B. P. Mahapatra, *Prachalita Odia BhasaraEkaByakarana, published by Pitambar Mishra*, Vidyapuri, Cuttack, First Edition, Mar. 2007.

[30] M. Padró and L. Padró, "Developing competitive HMM POS taggers using small training corpora," In *International Conference on Natural Language Processing (in Spain)*, Springer, Berlin, Heidelberg, pp.127-136, Oct. 2004. DOI:10.1007/978-3-540-30228-5_12.

[31] Indian Language Technology Proliferation and Deployment Center [Internet], Accessed 27th, Jun. 2020.Available: http://tdil-dc.in.

SagarikaPattnaik

received her Bachelor in Engineering in Computer Science and Engineering from Seemanta Engineering College (North Orissa University), Orissa, India in 2001. She received her Master's in Computer Science& Informatics from ITER (S'O'A Deemed to be University) Bhubaneswar, Orissa, India in 2013. She is presently working as a lecturer in Computer Science at P.N. Autonomous College, Khordha, Orissa, India. Her research interests include natural Language processing, artificial intelligence, communications, and network and wireless sensor networks.

Ajit Kumar Nayak

is the professor and HoD of the Department of Computer Science and Information Technology, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, Odisha. He graduated in Electrical Engineering from the Institution of Engineers, India in 1994, and obtained M. Tech. and Ph. D. degrees in Computer Science from UtkalUniversity in 2001 and 2010, respectively. His research interests include computer networking, ad hoc & sensor networks, machine learning, natural language computing, and speech and image processing. He has published about 55 research papers in various journals and conferences. He has also co-authored a book, Computer Network Simulation using NS2, CRC Press. He has participated as an organizing member of several conferences and workshops at the national and international levels.

Srikanta Patnaik

is a Professor in the Department of Computer Science and Engineering, Siksha 'O' Anusandhan Deemed to be University, Bhubaneswar, India. He has received his Ph. D. (Engineering) on Computational Intelligence from Jadavpur University, India in 1999. He has supervised more than 25 Ph. D. Theses and 60 Master theses in the area of Computational Intelligence, Machine Learning, Soft Computing Applications and Re-Engineering. Dr. Patnaik has published around 100 research papers in international journals and conference proceedings. He is author of 2 text books and 42 edited volumes and few invited book chapters, published by leading international publisher like Springer-Verlag, Kluwer Academic, etc. Dr. Patnaik is the Editors-in-Chief of International Journal of Information and Communication Technology and International Journal of Computational Vision and Robotics published from Inderscience Publishing House, England and also Editors-in-chief of Book Series on "Modeling and Optimization in Science and Technology" published from Springer, Germany. He is also Associate Editor of International Journal of Telemedicine and Clinical Practices (IJTMCP) and International Journal of Granular Computing, Rough Sets and Intelligent Systems (IJGCRSIS)