===
Language in India www.languageinindia.com ISSN 1930-2940 Vol. 17:1 January 2017
===

An Experiment with the CRF++ Parts of Speech (POS) Tagger for Odia

Pitambar Behera, M.A., B.Ed., M.Phil., Ph.D.

===

Abstract

This research work presents a probability-based CRF++ parts of speech (POS) tagger for Odia language. A corpus of approximately 600k tokens has been annotated manually in the Indian Languages Corpora Initiative (ILCI) project for Odia. The whole Odia corpus has been annotated based on the Bureau of Indian Standards (BIS) tagset developed by the DIT, govt. of India with some modifications under the ILCI. The tagger has been trained and tested with 2, 36, 793 and 1, 28, 646 tokens respectively. It provides 94.39% accuracy in the domain of seen data and 88.87% in the unseen dataset in precision and recall measures. In addition, this study further conducts an IA (inter-annotator) agreement, an error analysis to figure out salient erroneous labels committed by the automatic tagger and provides various suggestions to improve its efficiency. Furthermore, this study also provides the user-interface architecture and its functionalities.

Key words: Indo-Aryan language, Odia, BIS, ILCI, POS tagger, CRF++, NLP.

Overview

Parts of Speech (POS) tagging, as well as annotation or labelling task (Mitkov, 2003) is the method of assigning a grammatical category label for each token based on the linguistic and contextual information within a sentence. There are several approaches and methods for POS annotation task out of which rule-based, statistical and hybrid methods are salient.

Indian languages have always been quite challenging for both linguistics and NLP owing to the fact that they are diverse and multiple in nature and morphologically richer; including some other unique features. India has been the homeland for five diverse language families, namely, the Indo-Aryan, Dravidian, Austro-Asiatic, Tibeto-Burman, and the Andamanese (Abbi, 2001, pp. 24).

Odia Language

Odia /oɽɪɑ/ is recently declared as the sixth classical language (Pattanayak and Prushty, 2013; Jha et al., 2014) in India and belongs to the IA group but is resource-poor in terms of the availability of the corpus for any NLP task. Odia, probably the only IA language, exhibits some of the features in line with the Dravidian group. The Dravidian features that are observed in Odia are the occurrence of complementizer /bolɪ/ post-verbally, agglutination, 'not allowing participial agreement' and the curved shape of the alphabets etc. This could be ascribed to its geographical location as it is located in a belt where both the IA and Dravidian families converge from both the sides. Therefore, it can vehemently be stated that Odia is a 'typologically-syntactically disturbed' IA language (Patnaik, 2014) as it has both the IA and Dravidian features. However, Boulton (2003) has provided a historical foundation that Odisha had been the homeland for the aboriginal and Dravidian tribes for years. Thus, it can be deduced that there could be the possibility that Odia may have loaned some of the linguistic features from the Dravidian families.

Literature Review of Parts of Speech (POS) Annotation in Indian Languages

This section partly draws from Antony and Soman (2011) and provides some of the updated research in the areas of POS annotation.

- **Odia Neural Network Tagger**

Das and Patnaik (2014) have proposed a Single Neural Network-based POS tagger for Odia language. The tagger labels the input data on the basis of voting on the output of all single-neuron taggers. Errors have been corrected with the 'forward propagation' method and then the corrected outputs have been transferred by the 'feed-forward technique'. It is reported that the tagger has an accuracy rate of 81%.

- **Odia SVM Taggers**

Das et al., (2015) have developed an SVM Tagger with the application of a training dataset of 10k word tokens and have reported 82% accuracy which is a nominal one percent increase in the accuracy rate from the earlier reported Neural Network tagger. The tagset they adhered for the manual labelling consists of only five tags. They have considered the POS features and affixes as they play a pivotal role in morphology. For efficient functioning of the POS tagger, they have applied a set of lexicon consisting of around 200 words. Ojha et al (2015) have reported an Odia POS tagger the accuracy of which ranges between 88 to 93.7%. A corpus of 90k has been annotated using the BIS POS Annotation guideline. The training and testing data sets applied for developing this tagger are 90k and 2k respectively. Behera (2015, 2016) has reported an SVM-based POS tagger which is trained with around 236k word tokens and tested with 128k word tokens. All the data have been annotated adhering to the guideline developed for Odia under the BIS Annotation Schema. The features for SVM have been selected taking into consideration the word, POS, ambiguity and unknown words. The tagger is reported to have the highest number of accuracy so far.

- **Odia CRF Tagger**

Behera (2015) has reported a CRF-based POS tagger which is trained with around 236k word tokens and tested with 128k word tokens. Ojha et al (2015) have reported an Odia POS tagger the accuracy of which ranges between 82 to 86.7%. A corpus of 90k has been annotated using the BIS POS annotation guideline. The training and testing data sets applied for developing this tagger are 90k and 2k respectively. Unigram feature template has been selected during training period.

- **Sambalpuri POS Taggers**

Sambalpuri is a less-resourced Eastern Indo-Aryan language with a population amounting to approximately 11 million. Behera et al. (2015) have reported two statistical POS taggers (SVM=83% & CRF++=71.56) for Sambalpuri. Both the taggers are trained and tested with approximately 80k and 13k word tokens respectively. A corpus of around 121k word tokens is collected from a blog which is the only source of data so far and converted into Unicode. The whole corpus is labelled adapting the BIS (Bureau of Indian Standards) annotation scheme devised for Odia under the ILCI (Indian Languages Corpora Initiative) Corpora Project.

- **Rule-based POS Tagger for Sanskrit**

Chandrashekhar[1] (2002-2007) has developed a POS tagger for Sanskrit language with the application of a rule-based method as part of his doctoral research. He has made a robust POS tagset which contains fine-level 134 tags. Out of them, 65 are word-level tags, 43 are feature sub-tags, 25 are punctuation tags and one tag is UN for annotating unknown words.[2]

- **Stochastic POS Tagger for Sanskrit**

Oliver Hellwig (2009) developed a Sanskrit stochastic tagger which is a tagger for un-pre-processed Sanskrit text. The tagger exploits a Markov model for tokenization task and conducts POS annotation task applying a Hidden Markov model. A huge manually annotated corpus of approximately 1,500k words was applied for training the system.[3] It is a freeware software available under a permissive license and standalone application (Hellwig, 2009). Tiwary (2015) has developed an SVM-based POS tagger for Sanskrit which provides an accuracy of 82%. The tagger has been trained and tested with 34k and 28k word tokens respectively.

- **Hindi POS-Taggers**

The year 2006 witnessed three different POS taggers for Hindi based on morphology driven, ME, and CRF++ approaches. There have already been two attempts for POS tagger developments in 2008 based on HMM approaches proposed by Shrivastava and Bhattacharyya. A POS annotation for Hindi Corpus has been proposed by Nidhi and Amit Mishra (2011). A POS tagger algorithm for Hindi was proposed by Ray et al. (2003). Ojha et al. (2015) have reported a Hindi CRF++ POS tagger the accuracy of which ranges between 82 to 86.7% and an SVM tagger with an accuracy ranging from 88% to 93.7%. A corpus of 90k has been annotated using the BIS POS Annotation guideline. The training and testing data sets applied for developing these taggers are 90k and 2k respectively.

[1] http://sanskrit.jnu.ac.in/post/post.jsp
[2] http://sanskrit.jnu.ac.in/post/post.jsp
[3] http://www.indsenz.com/int/index.php?content=sanskrit_tagger

- **POS-Taggers for Bengali**

Dandapat et al. (2007) have proposed two stochastic POS taggers using HMM and Maximum Entropy (ME) approaches. Further, Ekbal developed a POS tagger for Bengali applying Conditional Random Fields (CRF++). Ekbal and Bandyopadhyay (2008) developed another POS tagger applying SVM algorithm. An Unsupervised Parts-of-Speech Tagger for the Bengali language was proposed by Hammad Ali in 2010. Chakrabarti (2011) proposed A Layered Parts of Speech Tagging for Bengali.

- **Tamil POS Taggers**

Vasu has proposed a Tamil POS tagger based on Lexical phonological approach. Another POS tagger was prepared by Ganesan based on CIIL Corpus and tagset. Selvam & Natarajan (2009) have made an improvement over a rule-based morphological analysis and POS Tagging in Tamil. Dhanalakshmi et al. (2009) have prepared two POS taggers for Tamil using their own tagset.

- **POS Taggers for Punjabi**

A Panjabi POS tagger was developed by Singh et al. (2008) applying the rule-based approach. The fine-grained tagset applied by them during manual annotation contains about 630 tags. Only handwritten heuristic rules are exploited considering the contextual information for disambiguating the POS category of a given word. Employing the rule-based disambiguation approach, a database was created to store the rules. In addition, a separate database was designed for marking verbal operator. As has been reported the system provides 80.29% accuracy including unknown words and 88.86% by excluding unknown words.

- **POS Taggers for Telugu**

There are three POS taggers in Telugu language which are based on Rule-based, transformation-based learning and Maximum Entropy-based approaches. A corpus size of 12k word tokens was used for training the transformation-based learning and Maximum Entropy-based models. The existing accuracy of the Telugu POS tagger was later improved by a voting algorithm by Rama Sree, R.J. and Kusuma Kumari P. in 2007.

- **POS Taggers for Malayalam**

Manju et al. (2009) have proposed a statistical Hidden Markov Model (HMM) based POS tagger. A labelled corpus of approximately 1,400 tokens were generated with the application of a morphological analyzer and the tagger was trained applying the HMM algorithm. The performance of the developed POS Tagger is about 90%. The second POS tagger is based on Support Vector Machine (SVM) algorithms developed by Antony et al. (2010). They have proposed a new tagset called Amrita POS tagset and annotated a corpus size of 180,000 word tokens for training. The SVM-based tagger achieves 94%, around 4% improved result than the HMM-based tagger.

- **POS Tagger for Kannada**

Antony and Soman (2010) have proposed a statistical approach for developing an SVM POS tagger for Kannada. They have proposed a tagset which contains 30 tags in totality. The architecture of the proposed POS tagger is corpus-based and motivated by supervised machine learning approach. It was modeled applying SVM kernel. A corpus size of 54k word tokens was used for training the tagger.

- **Bhojpuri POS-Taggers**

Singh & Jha (2015) have developed an SVM-based POS tagger applying a training data set of approximately 89k. This tagger provides an accuracy of around 89-90%. Ojha et al. (2015) have reported a Bhojpuri CRF POS tagger the accuracy of which ranges between 82 to 86.7%. A corpus of 90k has been annotated using the BIS POS Annotation guideline. The training and testing data sets applied for developing this tagger are 90k and 2k respectively. Unigram feature template has been selected during training period.

Conditional Random Fields Model

CRF is a statistical tagging model based on probability developed by Charles Sutton. It is applied in the recognition of pattern, analysing regression, predicting structure, and so on. They are widely applied in the areas of computer vision (He et al., 2004), natural language applications or biological sequences (Lafferty et al., 2001), named entity recognition (Settles, 2004), shallow parsing, (Sha and Pereira, 2003) and so on. They are of ‘discriminative probabilistic undirected graphical model’. They have been designed as an alternative probabilistic model to the HMM. Where G is taken as a factor graph over y, then p (y|x) is a Conditional random field if the distribution factorizes according to G for any fixed x (Agarwal and Mani, 2011).

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{\Psi_A \in G} \exp \left\{ \sum_{k=1}^{K(A)} \lambda_{Ak} f_{Ak}(\mathbf{y}_A, \mathbf{x}_A) \right\}$$

Methodology

Method of Corpora Collection

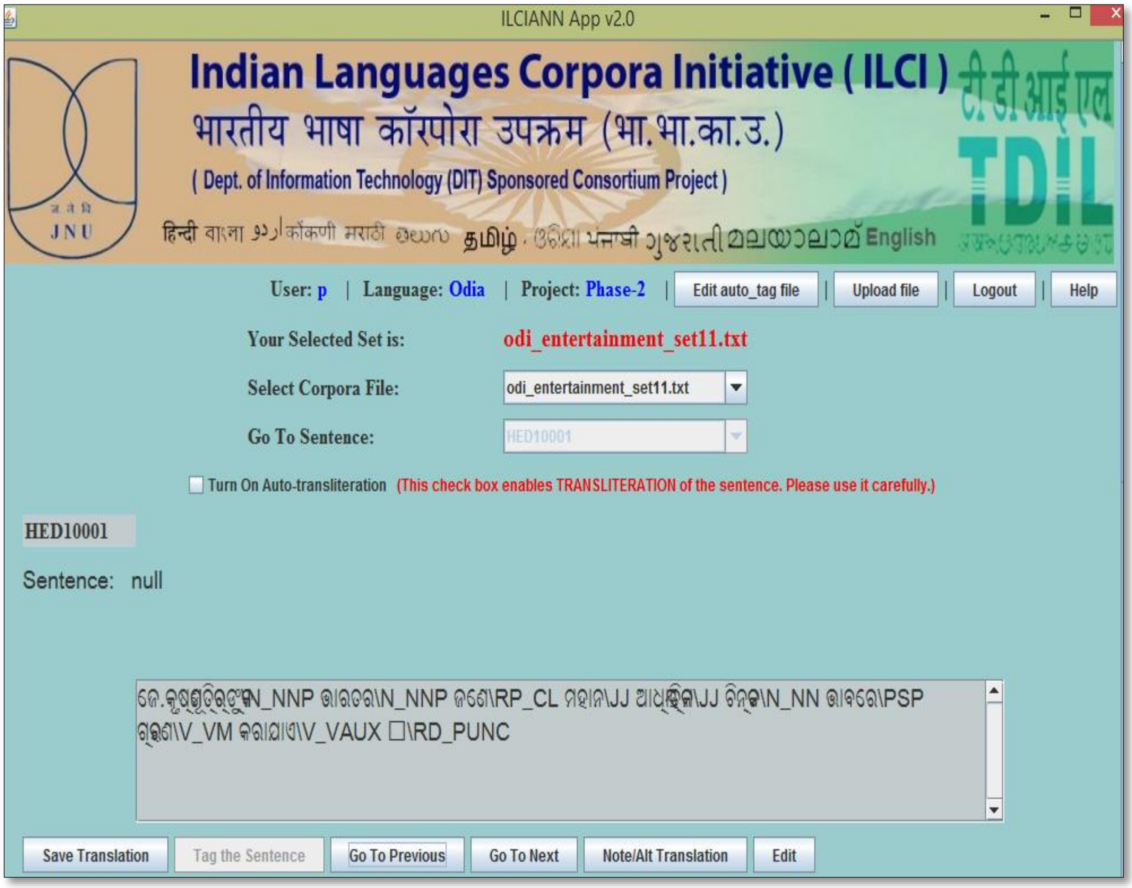
During the phase-I of the ILCI (Banerjee et al., 2013) project, 50k sentences corpora have been collected in Hindi and translated into 12 major Indian languages in the domains of health and tourism (Choudhary & Jha, 2011) including Odia. In the phase-II, the other scheduled languages have been incorporated and the domains also covered entertainment, agriculture, religion, literature and so on with another 50k sentences collected corpora.

Collected Corpora for Odia in the ILCI Project		
Phase-I: Parallel Corpora	Health	25k
	Tourism	25k
Phase-II: Monolingual Corpora	Entertainment, Agriculture, Religion, Literature	50k

Table 1. Domain-wise Distribution of the Collected Corpora

Method of Data Annotation

The BIS tagset is a hierarchical set designed by the POS Standardization Committee appointed by the DeitY, Govt. of India. It is a combination of both flat and hierarchical tags. It contains 11 top-level categories, 39 sub-type labels for annotation convention and examples. Under the ILCI Project, 50k corpus from the phase-I has been annotated online manually on the ILCI platform. Some of the data have further been annotated by a semi-automated tool named ILCIANN App v2.0 (Kumar et al., 2012) manually. The tool has a special feature of auto-edit tag list which automatically tags those tokens identical to the assigned token in the prescribed list.


ILCIANN App v2.0
Indian Languages Corpora Initiative (ILCI)
भारतीय भाषा कॉरपोरा उपक्रम (भा.भा.का.उ.)
(Dept. of Information Technology (DIT) Sponsored Consortium Project)
JNU
TDIL
English
Edit auto_tag file
Upload file
Logout
Help
Save Translation
Tag the Sentence
Go To Previous
Go To Next
Note/Alt Translation
Edit

User: p | Language: Odia | Project: Phase-2

Your Selected Set is: odi_entertainment_set11.txt

Select Corpora File: odi_entertainment_set11.txt

Go To Sentence: HED10001

□ Turn On Auto-transliteration (This check box enables TRANSLITERATION of the sentence. Please use it carefully.)

HED10001

Sentence: null

ଜେ.କୃଷ୍ଣମୂର୍ତ୍ତି\N_NNP ଭାରତର\N_NNP ଜଣେ\RP_CL ମହାନ\JJ ଆଧ୍ୟାତ୍ମିକ\JJ ଚିନ୍ତକ\N_NN ଭାବରେ\PSP ଗ୍ରହଣ\V_VM କରାଯାଏ\V_VAUX □\RD_PUNC

Fig. 1. ILCIANN App v. 2.0

Stages of POS Annotation

The POS annotation work is a painstaking enterprise which comprises of the following stages (see fig. 2). Firstly, one needs to have a huge amount of machine readable corpus (preferred encoding is UTF-8 or ASCII). During this process, both automatic and manual collections are conducted. In the former collection a software assists in crawling the data online which is called as a Data Crawler and the said data is sanitized applying a sanitizer. In the latter, human annotators assist in collecting and filtering the data. During pre-processing, data are sanitized either automatically or manually. At the annotation level, corpora are to be labelled adhering a tagset (both supervised and unsupervised approaches are in practice). After that, the corpus selected for training is to be tokenized using an automatic tokenizer or manually; although the former is preferred. After the stage of tokenization, there are still some errors that persist and need to be eradicated which are normalized during the normalization stage. With a normalized corpus, the data is ready for the perusal of training, testing and evaluation. After these three important processes, one needs to conduct an error analysis to figure out the issues in automatic annotation of various categories.

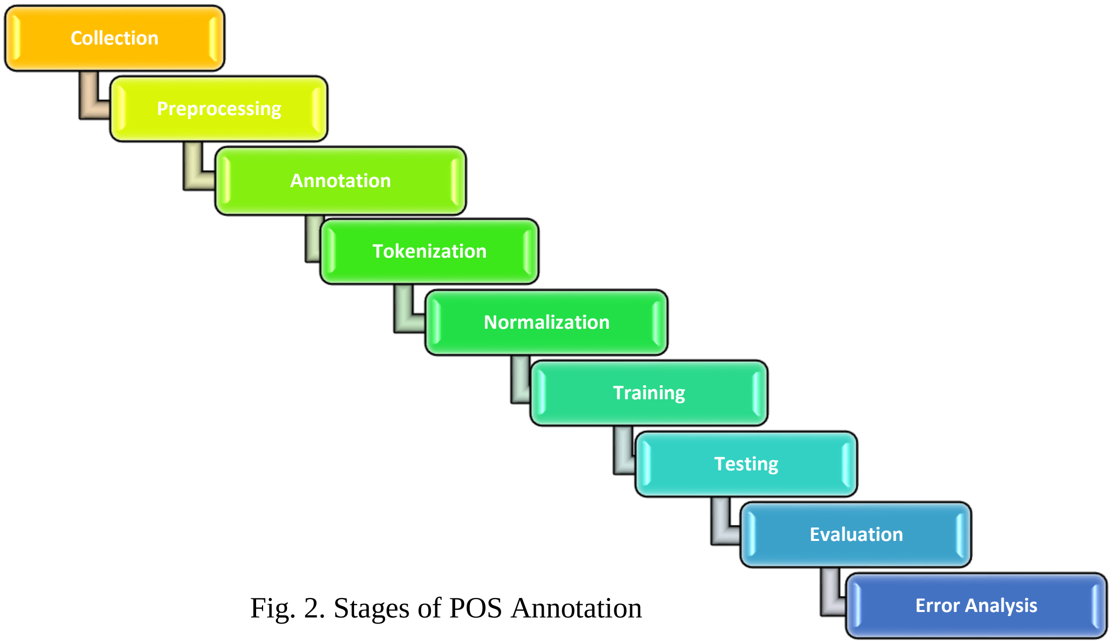


Fig. 2. Stages of POS Annotation

Distribution of Corpus

Training Data Sets				**Testing Data Sets**	
	Domains	Tokens seen	Tokens unseen	Tokens seen	Tokens unseen
I Phase	Health	46, 785	46, 785	15, 935	32, 691
	Tourism	30, 987	30, 987	15, 442	14, 407
II Phase	Entertainment	13, 834	30, 929	13, 834	18, 463
	Agriculture	29, 470	29, 470	29, 470	17, 885
	Literature	20, 633	98, 622	20, 633	45, 200
Total		1, 41, 709	2, 36, 793	95, 314	1, 28, 646

Table 2. Domain-wise Distribution of Training and Testing Data sets

Evaluation

Inter-annotator Agreement

The tabulated data (see table 3) demonstrates the fact that the average accuracy of the CRF++ IA judgment is 90.95%. Furthermore, the total accuracy of the tokens where all the annotators have agreed with a consensus is 90.44. The cases where all of them have disagreed account for 8.56%. The cases of POS where the annotators have largely disagreed are common nouns, adjectives, proper, coordinating and subordinating conjunctions, and deictic and indefinite demonstratives. This could be ascribed to the fact that there are ambiguity issues, multiword expressions, foreign and unknown words, difficult linguistics, the gapping in the lexicon etc.

models	**CRF++ Evaluation**		
annotators	**ANN 1**	**ANN 2**	**ANN 3**
accuracy	91.34	90.61	90.9
average	90.95%		
all agree	90.44%		
all disagree	8.56%		

Table 3. The Inter Annotator Agreement

Statistical Evaluation

So far as the evaluation in the domain of seen data is concerned (see table 4), the error-prone categories are verbal nouns, indefinite, interrogative and reciprocal pronouns, interrogative demonstrative, gerundival, non-finite and main verbs, interjections, foreign and echo words. As far as the unseen domain is concerned, on the other hand, the most frequent erroneous grammatical categories are reciprocal pronouns, demonstratives, gerundival, finite, non-finite and infinitive verbs, cardinals, unknown words, classifiers and adjectives. The most common error-prone POS categories are reciprocal pronouns, demonstratives, gerundival, non-finite and main verbs which is suggestive of the fact that Odia has an agglutinated nominal morphology and some its traces can also be observed from the verbal morphology as well. Case markers, post-positions, classifiers and affixes alternate with all the elements that can potentially be under a determiner phrase (DP) such as demonstratives, pronouns, quantifiers, adjectives, nouns etc. In addition, some of them also agglutinate with verbs; especially classifiers.

Accuracy per Part-of-Speech for the Odia CRF++ Tagger as Precision and Recall

			Results on Seen data		**On Unseen data**	
Id	**Description**	**Tag**	**Recall**	**Precision**	**Recall**	**Precision**
1	Common Noun	N_NN	98.70	90.25	96.43	79.78
2	Proper Noun	N_NNP	81.48	95.09	54.30	80.87
3	Spatial-temporal Nouns	N_NST	93.92	96.32	87.70	97.61
4	Verbal Noun	N_NNV	29.87	94.23	47.43	97.36
5	Personal Pronoun	PR_PRP	97.35	93.13	96.70	97.31
6	Reflexive Pronoun	PR_PRF	98.39	99.45	94.09	99.75
7	Relative Pronoun	PR_PRL	74.46	94.59	87.36	97.64
8	Reciprocal Pronoun	PR_PRC	50	100	61.53	100
9	Interrogative Pronoun	PR_PRQ	28.81	70.83	92.85	76.47
10	Indefinite Pronoun	PR_PRI	81.81	45	87.03	100
11	Deictic Demonstrative	DM_DMD	94.18	97.90	96.60	99.79
12	Relative Demonstrative	DM_DMR	92.14	95.62	93.69	100
13	Interrogative Demonstrative	DM_DMQ	68.42	67.70	75.59	100
14	Indefinite Demonstrative	DM_DMI	93.36	97.53	94.57	98.33
15	Main Verb	V_VM	77.21	91.27	72.07	89.77
16	Finite Verb	V_VM_VF	98.22	98.23	94.86	93.51
17	Non-finite Verb	V_VM_VNF	69.36	90.78	73.99	93.02
18	Infinitive Verb	V_VM_VINF	86.82	91.54	79.01	99.34
19	Gerundive Verb	V_VM_VNG	70.62	97.15	71.83	98.28
20	Auxiliary Verb	V_VAUX	90.68	98.50	85.45	97.91
21	Adjective	JJ	89.96	95.48	68.19	90.69
22	Adverb	RB	82.84	92.78	80.28	89.87
23	Postposition	PSP	96.03	97.26	93.91	97.39

24	Coordinating Conjunction	CC_CCD	95.16	98.08	98.40	98.35
25	Subordinating Conjunction	CC_CCS	86.76	90.34	96.82	96.69
26	Default Particle	RP_RPD	99.07	97.08	98.95	98.06
27	Interjection	RP_INJ	44.44	88.88	82.35	100
28	Intensifier	RP_INTF	91.73	95.68	80.80	91.40
29	Negative Particle	RP_NEG	98.73	93.98	99.10	99.66
30	Classifiers	RP_CL	81.84	98.79	69.04	99.75
31	Foreign Words	RD_RDF	69.76	93.75	0	0
32	Symbols	RD_SYM	99.50	98.29	99.63	99.63
33	Punctuations	RD_PUNC	99.82	100	99.57	99.99
34	Unknown Words	RD_UNK	0	0	17.50	27.60
35	Echo-words	RD_ECH	5.26	100	0	0
36	Default Quantifier	QT_QTF	93.69	92.43	89.95	95.95
37	Cardinal Quantifier	QT_QTC	88.37	98.66	77.00	97.63
38	Ordinal Quantifier	QT_QTO	89.26	95.75	85.68	99.40
Total			94.39		88.87	

Table 4. Accuracy per POS Category in Seen & Unseen Domains with Precision & Recall

Error Analysis

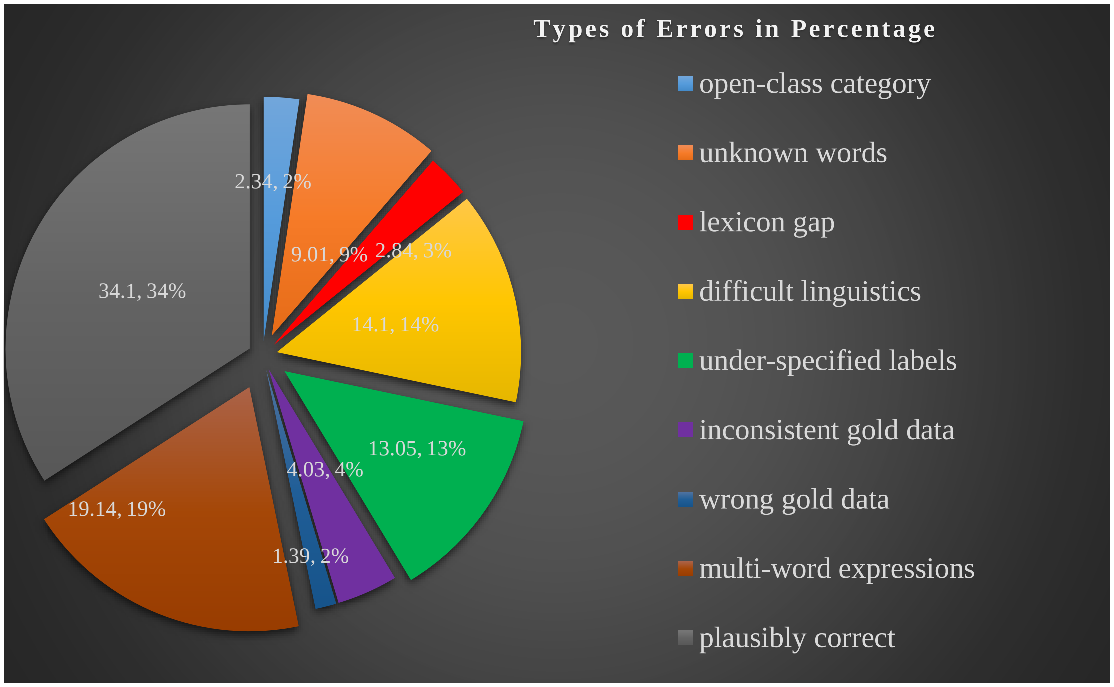


Fig. 3. Distribution of Errors

The chart demonstrates the categorization of errors into nine broader levels. Out of them, the most frequently occurring errors are the plausibly correct ones that refer to the categories that are inconsistently annotated by the annotators even if they are correct in both the training and gold files. On the contrary, the less-frequent errors are the wrong gold data.

Architecture of the POS Tagger

The present POS tagger (see fig. 4) is soon going to be hosted on the official website of the Special Centre Sanskrit Studies[4], Jawaharlal Nehru University, New Delhi, India.

- **Pre-processor**

The job of the pre-processor is to figure and filter out any unwanted linguistic or extra-linguistic elements present in the input text. In case if it figures out so it can either discard the said element or leaves it un-corrected. For example, if it finds non-specified characters like the unwanted punctuations within the token or half-finished letters or any other 'control characters', it leaves them as they are by labeling with the default tag.

[4] Sanskrit.jnu.ac.in

- Input token:

ମୃ?ତଙ୍କ

- Output token

ମୃ?ତଙ୍କ N_NN

- **Tokenization**

After the pre-processing stage, the next step that the tool approaches to is tokenization. The tool tokenizes the input data encoded in a sentence-by-sentence fashion. Furthermore, it tokenizes the given input data wherever it finds two tokens separated by a white space. Thus, white spaces are considered to be the identification indicators for token boundary detection.

Thereafter, it converts the file with sentences into token-by-token fashion. The tokenizer used in the tool is the Java Class Tokenizer.

- **The CRF++ Toolkit**

Thirdly, the Tool forwards the process of actions to the CRF++ Toolkit which runs with the CRF algorithm. It accesses the model and input files and executes them. Thereafter, it assigns a grammatical label identifying the probable tag for the given input token based on its previous learning and provides the output. When the user selects the CRF tag button, the toolkit starts processing the data based on its earlier training.

- **The POS-tagged Output**

Obviously, the quality of the output text is solely based on the efficient decoding by the tagger based on the training data. For making the tagger more efficient, one needs to focus much on the training stage. The output generated by the tagger is in a token-by-token fashion in each line as exemplified in the following example. It primarily depends upon the input file as to what will be the probable best output of the input data. For example,

ମୋର PR_PRP

ନାମ N_NN

ପିତାମ୍ବର N_NNP

| RD_PUNC

- **The De-tokenizer**

The tokenizer tokenizes each linguistic element into individual token while the de-tokenizer detokenizes them into the reverse order. So the tokenizer and the de-tokenizer are contrary to each other. Thus, the de-tokenizer converts the tagged output text into its tokenized forms; separating each token and tag with a white-space. Thereafter, the tool provides the final output. For instance,

ମୋର\PR_PRP ନାମ\N_NN ପିତାମ୍ବର\N_NNP ।\RD_PUNC

Suggested Solutions for the Statistical Tagger

Behera (2015) has proposed different approaches for the efficient functioning of a statistical tagger in terms of quality, reliability, and efficiency. They are formulating heuristic linguistic rules, the data approach and words sense disambiguation. Another approach could be added which is the application of a stemmer or lemmatizer. The only approach which has been applied and verified in this study is the data approach.

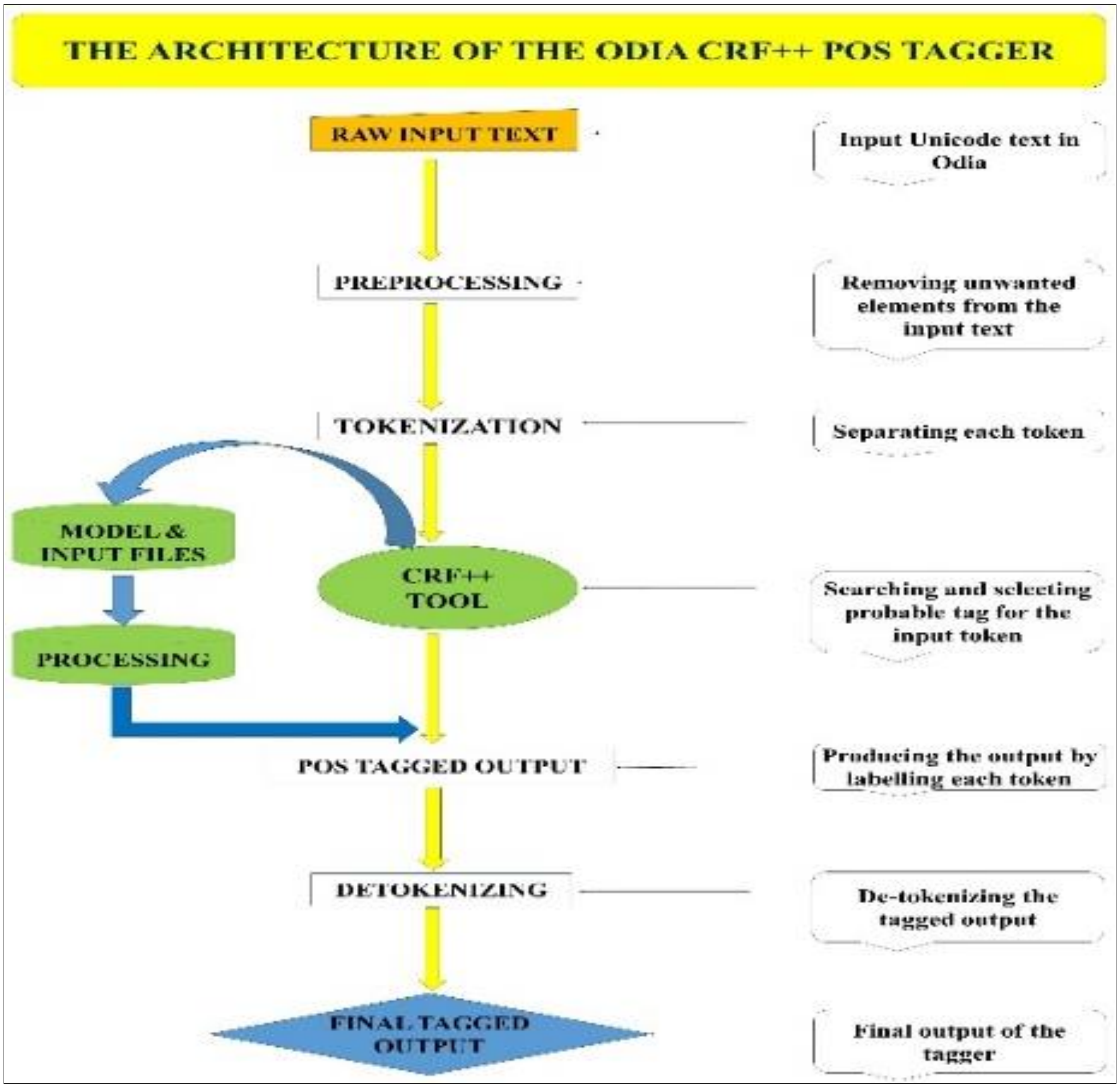

THE ARCHITECTURE OF THE ODIA CRF++ POS TAGGER
RAW INPUT TEXT
Input Unicode text in Odia
PREPROCESSING
Removing unwanted elements from the input text
TOKENIZATION
Separating each token
MODEL & INPUT FILES
PROCESSING
CRF++ TOOL
Searching and selecting probable tag for the input token
POS TAGGED OUTPUT
Producing the output by labelling each token
DETOKENIZING
De-tokenizing the tagged output
FINAL TAGGED OUTPUT
Final output of the tagger

Fig. 4. Architecture of the CRF++ Odia POS Tagger

Formulation of Heuristic Rules

One of the methods for improving the performance of the tagger could be to formulate linguistic rules by observing the erroneous patterns that the tagger provides. The encoding of these linguistic rules to the statistical taggers invariably makes it hybrid in nature.

The CRF++ tagger annotates the data based on the probability occurrences of the given input token. For instance, if a given token contains 13 times proper noun label and 8 times common noun label {N_NN (8) and N_NNP (13)} in the training data, the CRF algorithm labels the token with the higher frequent tag i.e. proper noun in this case. Thus, it can be stated that it annotates the input data taking into consideration the frequency of occurrences in the whole training data. Therefore, this makes the CRF++ tagger performs less accurately in comparison to the SVM model. To increase the efficiency and performance, a hybrid approach has been proposed which will be an amalgam of both the statistical and linguistically rule-driven. For developing hand-crafted rules the contextual features (the following and the preceding tags or tokens) of a given word have been given due consideration. Some of the rules are as follows:

- When /ɔt̪ɪrɪkt̪ɔ/ precedes a noun phrase, it needs to be tagged as an adjective. When it follows a noun phrase, it can be tagged as a postposition.
- When /pɑkʰɑ pɑkʰɪ/ occurs before a prenominal cardinal, it is tagged as an adverb since it is used in the sense of ‘approximately’. If it is used as a modifier to noun just preceding it, it has been tagged as an adjective.
- Whenever the word “/t̪ɔ/ is preceded by conjunct words”, it can be annotated as a conjunct. Otherwise, it is a particle by default.
- When /bʰɑbɔre/ is preceded by an adjective, it is an adverb or a noun. Or else, it is a post-position.
- Whenever spatio-temporal nouns (having the tag of N_NST) carry the genitive marker /-rɔ/ they are to be annotated as adjectives (JJ).
- When the word /ɟe/ is used as the complementizer augmenting a following subordinate clause, it is tagged as a subordinating conjunction.

The Data Approach

The graphical representation (see fig. 5) demonstrates the fact that the accuracy rate of the POS tagger increases with the increase in the number of the tokens. With each evaluation, results were evaluated and error analysis has been conducted manually. Based on the rule judgments of the human evaluator, corrections have been made. Initially, the accuracy rate has been evaluated manually, but the final three evaluations have been conducted automatically. At the first stage with a training data size of approximately 56k tokens the rate of accuracy was around 83.34%. With 86k the tagger provided 86% correct output, with 130k the accuracy rose to 91.22% and with 200k it further increased to 92.11%.

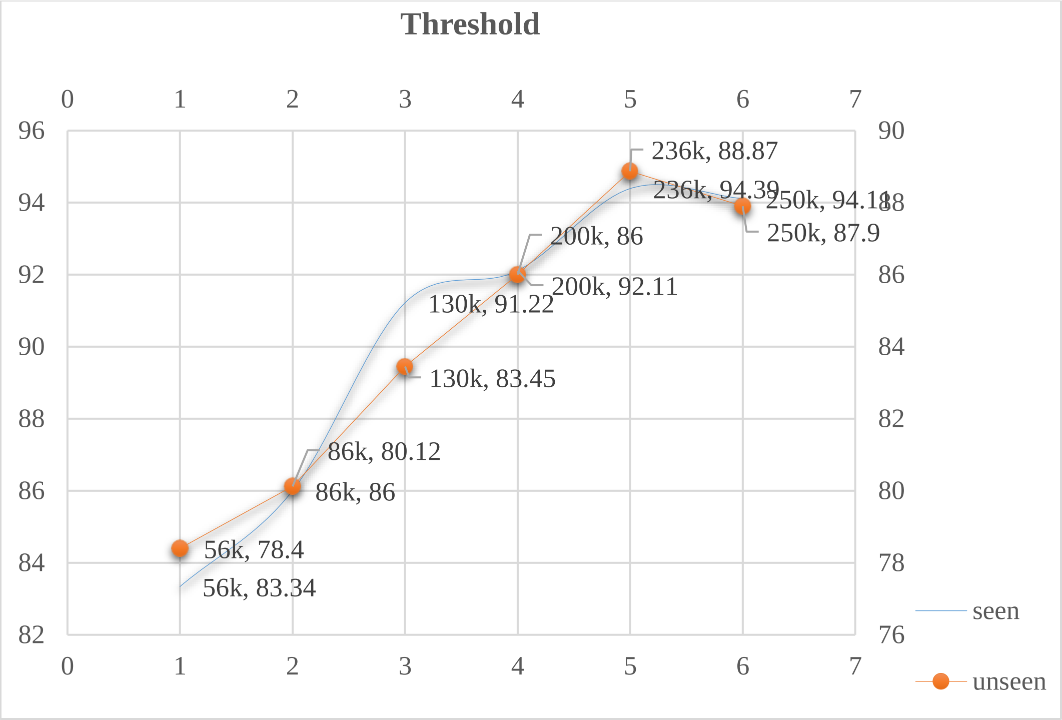


Fig. 5 Development of the Accuracy Rate during the Evaluation Period

When tested with a training data size of 236k which is the threshold, it rose to 94.39. When tested with the addition of another 14k corpus the accuracy dipped to 94.11%. On the other hand, when the tagger has been tested with the unseen data, the accuracy decreases to 88.87 because of a number of unknown and ambiguous words found by the taggers.

Word Sense Disambiguation (WSD)

It is often quite difficult to decide as to which annotation label is best suitable for a particular word even within a given context. When there is ambiguity or confusion, the context along with the linguistic intuition has been given utmost importance for deciding the tag of a given word.

"Categorial ambiguity arises when a particular word form can, in different instances, represent different grammatical categories" (De Rose, 1990). The ambiguity also arises when a particular word form has different tags at the same kind of contexts. This sub-section presents a couple of specimens of the grammatical categories that can easily be confused and instructions on how to disambiguate them. Furthermore, it is noteworthy to mention that in this section only the lexical ambiguities (token-wise and label-wise) have been addressed.

CC_CCD or QT_QTF (coordinator or general quantifier)

When /ɑʊ/ and /ɑhʊrɪ/ are used as coordinators conjoining words, phrases, and clauses, they are tagged as coordinators. Example,

mʊ̃\PR_PRP ɑʊ\CC_CCD mo\PR_PRP bʰɑɪ\N_NN ‘I and my brother’

When they are used as prenominal modifiers, they are tagged as general quantifiers. For Example:

ɑʊ\QT_QTF ekɔ\QT_QTC ‘another one’
ɑhʊrɪ\QT_QTF d̪ʊɪʈɪ\QT_QTC ‘another two’
/d̪ eɪ/: (PSP or V_VM_VNF)

It can both be used as a postposition and a non-finite verb. When it is used after common and proper nouns, and postpositions, it is a postposition. However, it is not clear as to which occurrence has to be a postposition and non-finite verb as the difference is quite blurred since the selectional features apply to both the tags. For instance,

penʈɪ\N_NN d̪eɪ\V_VM_VNF ɟɑɔ\V_VM_VF “go after giving me the pen”
ɟɔŋɔlɔ\N_NN d̪eɪ\PSP ɟɑɔ\V_VM_VF nɑ\RP_NEG “don’t go through the forest”

Conclusion

The tagger erroneously annotates the data specifically with respect to reciprocal pronouns, demonstratives, gerundival, non-finite and main verbs, foreign, unknown and echo words (Behera, 2015). One of the main reasons of the inaccuracy is that Odia has agglutinative nominal morphology and inflectional verb morphology. The performance of the model can be enhanced by introducing tools like NER (Singh et al., 2008), discourse anaphora resolver, a morph analyser, morph synthesizer, WSD or by converting it into a hybrid tagger formulating hand-crafted linguistic rules. It can potentially be applied for developing chunker, parser, MT and other such NLP tools in Odia.

Acknowledgements

I acknowledge the ILCI Project, led by JNU under the leadership of Prof. Girish Nath Jha, funded by the TDIL, DeitY, for providing the Odia corpus. I further acknowledge the ILCI Group for providing the ILCI ANN standalone semi-automated annotation tool. In addition, I also acknowledge the POS annotation work carried out for Odia at the University of Hyderabad and Ravenshaw University, Cuttack under the leadeship of Prof. Panchanan Mohanty and Dr. Bijaylaxmi Das respectively.

===

References

Abbi, A. (2001). A Manual of Linguistic Fieldwork and Structures of Indian languages (Vol. 17). Lincom Europa.

Antony, P. J., Mohan, S. P., & Soman, K. P. (2010, March). SVM Based Part of Speech Tagger for Malayalam. In Recent Trends in Information, Telecommunication and Computing (ITC), 2010 International Conference on (pp. 339-341). IEEE.

____________ & Soman, K. P. (2011). Parts of speech tagging for Indian languages: a literature survey. *International Journal of Computer Applications (0975-8887), 34*(8), 22-29.

Behera, P., Ojha, A. K. & Jha, G. N. (2015). Issues and Challenges in Developing Statistical POS Taggers for Sambalpuri, In *Proceedings of LTC-2015, Poland, Springer Verlag*.

Behera, P. (2015). Odia Parts of Speech Tagging Corpus: Suitability of Statistical Models. M.Phil. Dissertation, Centre for Linguistics, Jawaharlal Nehru University, New Delhi, India.

Behera, P., Singh, R. & Jha, G. N. (2016). Evaluation of Anuvadaksh (EILMT) English-Odia Machine-assisted Translation Tool. In *Proceedings of WILDRE-3 (LREC-2016),* Portoroz, Slovenia.

Behera, P. (2016). Evaluation of SVM-based automatic parts of speech tagger for Odia. In *Proceedings of WILDRE-3 (LREC-2016),* Portoroz, Slovenia.

Behera, P., Maurya, N., Pandey, V. & Banerjee, E. (2016a). Dealing with Linguistic Divergences in English-Bhojpuri Machine Translation. In *Proceedings of WSSANLP-6, COLING-2016*, Osaka, Japan.

Behera, P., Muzaffar, S., Ojha, A. Ku. & Jha, G. N. (2016b). The IMAGACT4ALL Ontology of Animated Images: Implications for Theoretical and Machine Translation of Action Verbs from English-Indian Languages. In *Proceedings of WSSANLP-6, COLING-2016*, Osaka, Japan.

Bharati, A., Chaitanya, V., & Sangal, R. (2004). Natural language Processing: A Paninian Perspective, Prentice Hall of India Private Limited, New Delhi.

Brill, Eric (1992) “A simple rule-based part of speech tagger”. In the Proceedings of the Workshop on Speech and Natural Language (HLT-91), Morristown, NJ, USA: Association for Computational Linguistics. Pp. 112-116.

Choudhary, N. and Jha, G. N. 2011. Creating Multilinugal Parallel Corpora in Indian Languages,” Poznan: 5th Human Language Technology Conference Proceedings, Springer, 527-538.

Das, B. R., & Patnaik, S. (2014). A novel approach for Odia part of speech tagging using artificial neural network. In Proceedings of the International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2013, pp. 147-154. Springer International Publishing.

Das, B. R., Sahoo, S., Panda, C. S., & Patnaik, S. (2015). Part of speech tagging in Odia using support vector machine. Procedia Computer Science, Volume 48, 2015, pp. 507-512, ISSN 1877-0509.

Dey, G. & Maringanti, H. B. 2014. Paninian Framework for Odia Language Processing.

De Rose, Steven J. 1990. Stochastic Methods for Resolution of Grammatical Category Ambiguity in Inflected and Uninflected Languages. Ph.D. Dissertation. Providence, RI: Brown University Department of Cognitive and Linguistic Sciences.

Hellwig, O. (2009). Sanskrittagger: A stochastic lexical and pos tagger for sanskrit. In *Sanskrit Computational Linguistics* (pp. 266-277). Springer Berlin Heidelberg.

Jena, I., Chaudhury, S., Chaudhry, H., & Sharma, D. M. (2011). Developing Oriya morphological analyzer using Lt-toolbox. In *Information Systems for Indian Languages* (pp. 124-129). Springer Berlin Heidelberg.

Jha, G. N., Hellan L., Beermann, D., Singh, S., Behera, P. & Banerjee, E. (2014). Indian Languages on the TypeCraft Platform– The Case of Hindi and Odia, In *Proceedings of WILDRE-2014 (ISBN: 978-2-9517408-8-4), LREC: Rekyavijk, Iceland*.

Lafferty, J., McCallum, A., & Pereira, F. (2001, June). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML* (Vol. 1, pp. 282-289).

Kumar, R., Kaushik, S., Nainwani, P., Banerjee, E., Hadke, S., & Jha, G. N. (2012, July). Using the ILCI annotation tool for pos annotation: A case of hindi. In *13th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing 2012)*.

Mohapatra, R., & Hembram, L. (2010). Morph-Synthesizer for Oriya Language-A Computational Approach. *Language in India*, *10*, 205-211.

Muzaffar, S., Behera, P., & Jha, G. N. (2016). A Pāniniān Framework for Analyzing Case Marker Errors in English-Urdu Machine Translation. *Procedia Computer Science*, *96*, 502-510.

Ojha, A. K., Behera, P., Singh, S., & Jha, G. N. (2015). Training & Evaluation of POS Taggers in Indo-Aryan Languages: A Case of Hindi, Odia and Bhojpuri. In *the proceedings of 7th Language & Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics* (pp. 524-529).

Patnaik, B. N. (2014). Oriya as a Typologically Disturbed Language and Some Related Matters.

Pattanayak, D. P. and Prushty, S. K. (2013). Classical Odia Language, KIS Foundation, Bhubaneswar. Retrieved on 06.06.15 from http://www.orissalinks.com/odia/classical1.pdf

Ruslan Mitkov. 2003. The Oxford Handbook of Computational Linguistics. New York: Oxford University Press.

Singh, S., Gupta, K., Shrivastava, M., & Bhattacharyya, P. (2006, July). Morphological richness offsets resource demand-experiences in constructing a POS tagger for Hindi. In *Proceedings of the COLING/ACL on Main conference poster sessions* (pp. 779-786). Association for Computational Linguistics.

Singh, S., & Jha, G. N. (2015, August). Statistical Tagger for Bhojpuri (employing Support Vector Machine). In *Advances in Computing, Communications and Informatics (ICACCI), 2015* (pp. 1524-1529). IEEE.

Tiwary, A. (2015). Building a Statistical tagger for Sanskrit. In WILDRE-3, LREC-2016.

==

Pitambar Behera, M.A., B.Ed., M.Phil., Ph.D.
Centre for Linguistics
School of Language, Literature and Culture Studies
Jawaharlal Nehru University
New Delhi-110067
India
pitamb38llh@jnu.ac.in
pitambarbehera2@gmail.com