LANGUAGE IN INDIA

Strength for Today and Bright Hope for Tomorrow

Volume 11 : 9 September 2011

ISSN 1930-2940

A Hybrid POS Tagger for Indian Languages

M. Mohamed Yoonus M.Sc., M.Phil., PGDNLP and Samar Sinha, M.A., M.Phil.

Abstract

This paper describes the work on building Part-of-Speech (POS) tagger for 12 Indian Languages using hybrid approach, and presents the performance of the tagger for each Indian language. Unlike the most of the previous POS taggers for Indian languages which are designed to annotate few languages, the present tagger called 'POS Tagger' is an attempt to facilitate annotation of several Indian languages following a computational approach. The POS Tagger is trained on 80K to 85K tagged corpora for each language from the LDC-IL corpus. Finally, this paper highlights the performance of the tagger and the need of language specific resources required for obtaining optimal result.

1 Introduction

The basic objective of Natural Language Processing (henceforth, NLP) is to facilitate human-machine interaction through the means of natural human language. Research on NLP has focused on various intermediate tasks that make partial sense of language structure without requiring complete understanding which, in turn, contributes to develop a successful system. Part-Of-Speech (henceforth, POS) tagging is one of the processes in which grammatical categories are assigned to each token in its context from a given set of tags called POS tagset. It serves wide number of applications like speech synthesis, and recognition, information extraction, partial parsing, machine translation, lexicography, Word Sense Disambiguation (WSD), question-answering etc.

Although various automatic POS taggers have been developed worldwide using linguistic rules, stochastic models and hybrid approaches, but each approach has its own merits and demerits. In this context, Indian languages further present a challenge in developing an automatic POS tagger as the languages are highly inflectional and morphologically rich. Hence, we need to consider text processing prior to POS tagging in order to achieve high performance, more reliability, and to incorporate most of the Indian languages into a single framework of POS annotation.

2 POS Tagger for Indian Languages: An Overview

In the last few decades of NLP initiatives on Indian languages and in India, different research groups working on various languages have developed POS tagger for respective or a group of Indian languages. In this paper, a brief summary of POS tagger for Indian languages is provided language family-wise as overview information.

Assamese POS tagger [8] has been developed using HMM and provides an average tagging accuracy of 87%. A word based hybrid model (Dandapat 2004) for Bengali POS tagging uses the HMM in which probabilities of words are updated using both tagged as well as untagged corpus. In the case of untagged corpus the Expectation Maximization algorithm has been used to update the probabilities. Another tagger for Bengali [10] follows an approach suitable for morphologically rich languages in a poor resource scenario. For Gujarati, machine learning algorithm has been developed [11] following the CRF model in which the features given to CRF are considered with respect to the linguistic aspects of Gujarati. Scutt and Brants (1998) has developed a POS tagger for Hindi based on the HMM. This tagger, however, fails to account for the language specific features and context to address the partial free word order characteristics of Hindi. The other POS tagger developed by Aniket Dalal, et.al (Aniket Dalal, 2006) is based on Maximum Entropy Model. In this POS tagger for Hindi, the main POS tagging features are word based context, one level suffix and dictionary-based features.

In addition to these taggers, a simple HMM based POS tagger [12] for Hindi employs a naïve (longest suffix matching) stemmer as a pre-processor and achieves reasonably good accuracy of 91.57%. Unlike for other languages, Punjabi has an online POS tagger developed by AGLSoft [21]. But it is not efficient to tag large size corpora. The TnT POS Tagger for Nepali [18] has an accuracy of 56% for unknown words and 97% for known words. Along with it, Unitag by Andrew Hardie [19] is designed for POS-tagging of Nepali text. Sajjad and Schmid [26] reports that the existing Trigram and Tag (TnT) and PENN Treebank for Urdu has an accuracy of 93.40% and 93.02%, respectively.

Malayalam POS tagger [14] is designed to capture finer morphological analysis; and consequently, generates the most suitable POS tag using statistical approach. It has an accuracy rate of 80% for the sequence generated automatically for the test case. SVM based POS tagger for Malayalam has also been developed [15]. Tamil Morpheme Components based POS Tagging [22] has an overall accuracy of 95.92%. Similarly, POS Tagging for Tamil using Linear Programming [24] provides an overall accuracy of 95.63%. Apart from these two approaches, the hybrid POS tagger using HMM and a rule based system is also developed for Tamil [23].

For morphologically rich Tibeto-Burman languages like Manipuri, a morphologically driven POS tagger [16] is developed; however, the accuracy is limited due to lack of morphologically defined rules. On the other hand, CRF and SVM approaches based POS tagger for Manipuri [17] provide a promising result of 72.04% and 74.38%, respectively.

It is important to note that these POS taggers use different POS tagsets, which are developed to cater specific needs of the individual project. In other words, the basis of each POS tagset is different. Similarly, different computational approaches are used on these POS tagsets yielding different performance result. Such lack of a common basis for POS tagsets along with the different computational approaches which are used in developing POS tagger, cumulatively, has made the basis of comparison heterogeneous. Consequently, it serves in creating an unequal ground to access the performance of an approach as well as of the computational approach across Indian languages.

This paper primarily attempts to address such an issue regarding Indian languages. We have designed a tagger for labelling POS called POS Tagger, primarily for the twelve Indian languages following the POS tagset based on the ILPOSTS Framework. In this paper, we present the performance of the POS Tagger based on the hybrid approach.

3 Training Data Preparation

In the preparation of the training data, we have used in-house developed POS AnnTool v0.3, a manual annotation tool, and Simple Pattern Matching Tagger (SPMT) Tool. The former is used to annotate the 10K corpora and the latter is used subsequently for annotating 70K – 75K data following pattern matching and partial manual annotation. The size of the training corpus, therefore, is 80K to 85K for twelve Indian languages. These stages are described in detail.

3.1 Stage 1

With the help of the POS AnnTool v0.3, minimum of two annotators in each language annotated the randomly sampled data text containing approximately 10K words. In the process of annotation, the annotators are advised not to discuss the issues so that the mutual decisions do not influence the assignment of tags. Later on, the Inter Annotator Agreement based on the disagreement on the assigned tags is carried on to examine the variations in tags assigned among the annotators [7]. The 10K words annotated corpus is sanctified as a Gold Standard (GS) Corpus.

3.2 Stage 2

The GS 10K tokens are trained on untagged corpus of 25K using SPMT Tool. We observed that an approximately 30% of tokens are tagged and the remaining tokens are untagged. In this stage too, the untagged tokens are manually tagged and validated.

3.3 Stage 3

After stage 2, the size of the training data increases to 35K, which is trained on untagged corpus of 50K. The result shows that an approximately 60% to 65% of tokens are tagged and the remaining tokens are left untagged. Once again, the untagged tokens are manually annotated and validated.

In general, this process can be iterated until good reasonable training corpora for developing a POS tagger can be satisfied. The structure of training data development process is shown in Fig 1.

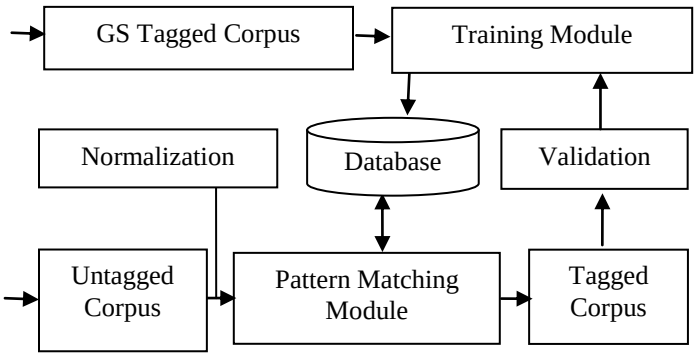

Fig 1: Training Data Development Structure

4 POS Tagging Issues

A POS annotation process encounters several issues regarding normalization, ambiguity and unknown words among others. In our POS Tagger, we have incorporated the modules to facilitate tagging. Some of these issues that are incorporated are discussed in detail providing illustrations from Indian languages.

4.1 Normalization

A process of organising data to tokens from a given corpus is called normalisation. In Indian languages, normalisation plays an important role since a wide variety of scripts and orthographic conventions and practices are followed which also differ language-wise as well as within categories in a language. The tagging algorithm, hence, needs to be designed to handle such cases optimally. For example in Nepali, भा'थ्यो (bhA'thyO) or भा'-थ्यो is contracted form of भएको थियो. The contracted form भा is a contraction of a participial भएको (bhaEkO) which is different from a dubitative particle भा [7].

In such cases, apostrophe is not considered as a delimiter in the process of normalization in the concerned languages. As a result, it also retains single quote marker as it is in the text. To resolve the issue of normalization in these languages, single quote marker is normalized when an apostrophe comes with boundaries of token.

4.2 Ambiguity

In computational linguistics, ambiguity refers to a state where there is a choice of tag to a given token. Interestingly, it is observed that ambiguity varies from language to language and also from corpus to corpus. Although the most of the words in a language are unambiguous and can be tagged straight forwardly but there is also a good size of words that are ambiguous. Consider the following sentence in Tamil and its corresponding POS tag.

1. இன்று\NC என்ன\DWH கிழமை\NC ?\PU (in'Ru en'n'a kiZamai?)
'What day is today?'

2. அவர்\PPR இன்று\ALC வரலாம்\V (avar in'Ru varalAm)
'He may come today.'

In (1) and (2), இன்று (in'Ru) has a lexical ambiguity either as an NC (Noun Common) or an ALC (Adverb Location) depending on its context. To resolve such ambiguity, the POS Tagger incorporates context based rules to disambiguate them.

4.3 Unknown Words

One of the issues that a POS tagger encounters frequently in tagging new corpus is respect to new tokens that do not exist in the training data. Such tokens are generally known as unknown words. In our POS Tagger, we have tried to resolve the issue using context driven rules to tag them.

5 POS Tagger: Procedure and Architecture

POS tagger involves basically two tasks: learning or training task and tagging task. The former task is also classified into base-level learning and context-rule learning. In developing the POS Tagger, first, we have trained the validated tagged data into base-level training module. This module generates a database that provides statistics regarding the frequency and the status of ambiguity associated with the input data (i.e. token with its POS tag).

On the basis of the database generated by the base-level module, an n-gram table is created. The learning algorithm, further, generates context-rules for disambiguation following the n-gram tables. Later on, this table is utilized by the POS Tagger to assign the appropriate tag.

In the tagging module, the input for the tagging algorithm is a token and the output is a POS tagged token. While assigning the appropriate tag to a token, the tagging process follows the following procedural steps:

1. Text normalisation and Tokenisation
2. Non-ambiguous tokens are assigned a POS tag through pattern matching method.
3. Ambiguous tokens are assigned the most appropriate tag based on the context-rules for disambiguation.
4. A common list containing names of common and important person, place, months, days, etc. is prepared for Indian languages but following language specific script. This list is provided to the system to tag the untagged tokens.
5. The remaining untagged words are assigned tags following bi-gram, tri-gram and penta-gram.

In the initial step, POS Tagger normalizes the untagged corpus and the information is updated in the metadata of the file normalized. It is largely carried out to avoid the repetition of the normalisation process of the input file. In the following step, POS tags are assigned to the known words of the file normalised. Finally, the system assigns an appropriate tag to the unknown tokens. The flowchart can be illustrated as below.

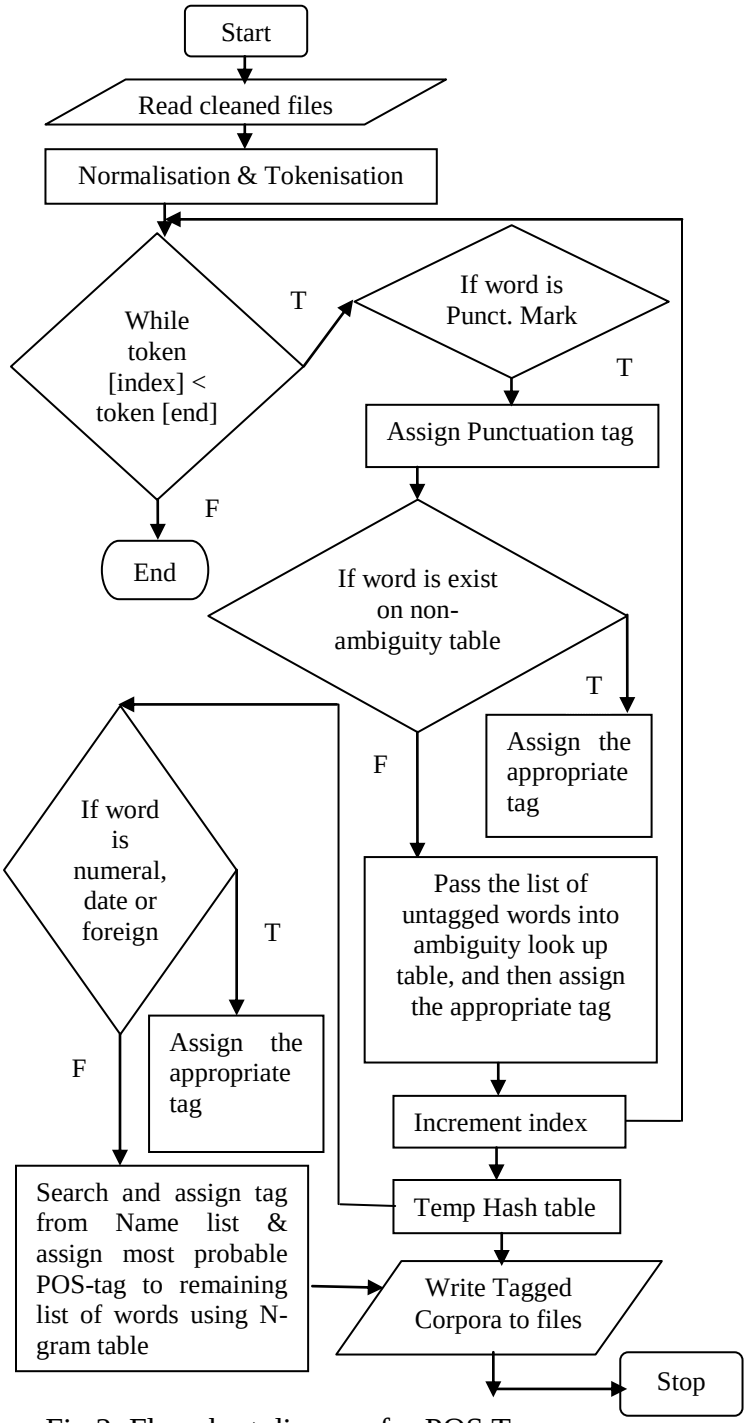

Fig 2: Flowchart diagram for POS Tagger

Our POS Tagger is built upon a hybrid approach that puts together the stochastic approach and the rule-based approach. The architecture of POS Tagger consists of three layers such as Data Layer (DL), Business Layer (BL) and Presentation Layer (PL). The system architecture is shown schematically in Fig 3.

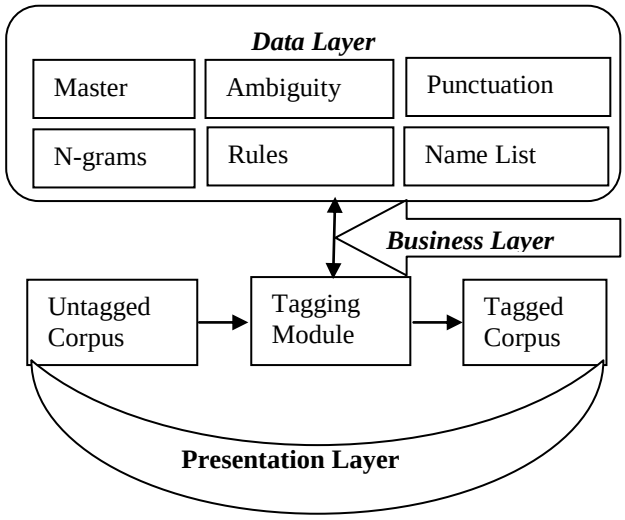

Fig 3: POS Tagger Architecture

The DL is prepared at the time training. It encapsulates all information related to data from the tagging module. The BL contains logic for retrieving persistent data from the DL and placing it into business objects. The PL gives graphical user interface (GUI) environment.

6 Experiment

6.1 Set-up

In this section, we are describing the POS annotation experiment carried out on twelve Indian languages in eight different scripts that each language uses using the POS Tagger written in C# using Microsoft Visual Studio 2008.

The LDC-IL tagset is a hierarchical tag set based on the EAGLES Guidelines which is designed to tag the maximum morpho-syntactic features of the Indian languages. It contains Category, Type and their Attributes. For this experiment, we have removed Attribute level from this tagset in order to test the efficiency of the POS tagger with respect to the Category and the Type levels. Such a strategy is designed to achieve objective of the larger research project of which this experiment is a part.

The size of tagset in each of the twelve languages is presented in Table 1 and their details in appendix 1.

The LDC-IL team has carried out manual POS tagged with help of the POS AnnTool v0.3. This manually tagged data is used as our training and test set data. The training set consists of tagged corpora of 70K to 75K and the test set consists of 10K of the GS corpus for 12 Indian languages.

TABLE 1
LDC-IL TAGSET SIZE

LDC-IL Tagset - Category and Subcategory	
Language	**Tagset Size**
Malayalam, Manipuri	37
Tamil	38
Assamese	39
Bengali	40
Hindi, Punjabi, Urdu	43
Bodo, Oriya	44
Gujarati, Nepali	50

6.2 Result

Our experiment with the hybrid POS Tagger has two sets of data. The training set has approximately 70K to 75K and the test set contains approximately 10K. In this particular experiment, we have merged the training and the test data, and the resultant data was equally divided into seven data sets. Of the seven data sets, the six data sets were used for training the POS Tagger, and the seventh data set was used to test the performance. Similar test was carried on all the data sets in which one of the data sets was used for the performance testing. Finally, the average result was calculated from the results obtained from the seven data sets. We have used the standard Information Retrieval (IR) metrics of Precision, Recall and F-Score to evaluate the system.

The Precision, Recall and F-score evaluation results as shown below in Table 2.

TABLE 2
EVALUATION OF THE SYSTEM FOR TWELVE LANGUAGES

S No	**Language**	**Precision**	**Recall**	**F-Score**
1	Assamese	83.65	98.21	90.35
2	Bengali	84.26	98.52	90.84
3	Bodo	83.32	97.94	90.04
4	Gujarati	84.45	98.47	90.92
5	Hindi	86.11	99.55	92.34
6	Malayalam	81.14	95.45	87.72
7	Manipuri	81.51	96.17	88.02
8	Nepali	86.17	98.65	91.99
9	Oriya	86.78	99.14	92.55
10	Punjabi	88.97	99.77	94.06
11	Tamil	81.41	95.66	87.96
12	Urdu	82.62	99.18	90.15

6.3 Error analysis

The performance error analysis provides information about the nature of error that the system makes. In our experiment, to ascertain the nature of error with respect to the POS tags that our hybrid POS Tagger assigns, we have used confusion matrix method. Table 3 shows the result for Punjabi data.

TABLE 3
ERROR ANALSIS RESULT (Punjabi)

Actual Tag	Assigned Tag	Percentage
NC	NP	0.12
NC	DAB	0.13
NC	JJ	0.33
NC	VM	0.41
NC	PP	0.38
NP	VC	0.62
NP	DAB	0.12
NP	JJ	0.24
NP	VM	0.12
NP	PP	0.1
NV	VM	0.1
NV	VA	0.09
NST	PP	0.11
PPR	DAB	0.46
DAB	PPR	0.17
JJ	NC	0.58
JJ	DAB	0.13
JJ	VM	0.1
JJ	PP	0.15
JQ	JINT	0.23
JINT	JQ	0.14
VM	NC	0.45
VM	NV	0.24
VM	VA	1.1
VM	PP	0.48
VA	VM	1.07
PP	CCD	0.28
CSB	CCD	0.28

The error figures can be reduced if we can find some mechanisms to handle the significant number of unknown words.

On the basis of the confusion matrix, it was found that the most of the errors occur with respect to Noun, Verb and Adjective categories in the twelve Indian languages. It is often the case that, in these languages, Common Noun and Proper Noun are often tagged reverse. The similar misappropriation of tag is witnessed between Main Verb and Auxiliary Verb, and Adjective and Noun.

7 Conclusion

The above experiment on twelve Indian language regarding POS tagging based on the LDC-IL POS tagset v0.3 using hybrid POS Tagger shows that the most frequent errors occur with respect to Noun, Verb and Adjective categories. On the computational front, it is also observed that due to the unknown tokens the time taken to assign the tag is more as the system undergoes through several modules to assign the most appropriate tag. However, its efficiency and accuracy increase as we increase the training data size.

In our future work, we would like to introduce a finer algorithm that reduces the margin of error regarding POS category tags. However, developing such an algorithm is not an easy task as the Indian languages do have free word ordering property. In other words, the n-gram statistics based on individual token as used in this experiment may not be adequate enough and reasonable to account for the property. Further, these facts point to develop an algorithm that accounts for such property of Indian languages with respect to the above mentioned categories. Similarly, to accelerate the computational speed, developing a Named Entity Recognizer (NER) and a module to identify category based on the morphological/morph syntactic cues of the unknown token is at the forefront of our endeavour to develop a generic toolkit for all Indian languages.

==========

Acknowledgement

We would like to thank Dr. L. Ramamoorthy, Dr. B. Mallikarjun and Er. M. Venkatesan for valuable advice and support. We would also like to thank A. Vadivel and LDC-IL team for valuable technical and academic suggestion.

==========

References

[1] Brill Eric,"*Transformation-Based Error-Driven Learning and Natural Language Processing:* A Case Study in Part of Speech Tagging. Computational Linguistics, vol. 21, pp. 543--565, 1995.

[2] D. Jurafsky and J.H. Martin, "Chapter 8: *Word classes and Part-Of-Speech Tagging*", Speech and Language Processing, Prentice Hall, 2000.

[3]http://www.au-kbc.org/research_areas/nlp/ projects/postagger.html

[4] Xunlei Rose Hu and Eric Atwell, "*A Survey of Machine Learning Approaches to Analysis of Large Corpora*",School of Computing, University of Leeds, *U.K. LS2 9JT*

[5] A. Voutilainen, *The Oxford handbook of computational linguistics*, ch. 11: Part-of-Speech Tagging, pp. 219-232. Oxford University Press, 2005.

[6] R. Garside and N. Smith, "*A hybrid grammatical tagger: Claws4*," in Corpus Annotation: Linguistic Information from Computer Text Corpora, pp. 102-121, Longman, 1997.

[7] Mallikarjun B, Mohamed Yoonus M, Samar Sinha and Vadivel A, "*Indian Languages and Part-of-Speech Annotation*", CIIL Publication No.598, in press.

[8] Navanath Saharia, Dhrubajyoti Das, Utpal Sharma and Jugal Kalita, "*Part of Speech Tagger for Assamese Text*", Proceedings of the ACL-IJCNLP 2009 Conference Short Papers, pages 33–36, Suntec, Singapore, 4 August 2009.

[9] Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu, "*A Hybrid Model for Part-of-Speech tagging and its application to Bengali*", Transaction on Engineering, Computing and Technology VI Dec 2004.

[10] Dandapat S., Sarkar S. and Basu A. "*Automatic Part-of-Speech Tagging for Bengali: An approach for Morphologically Rich Languages in a Poor Resource Scenario*", In Proceedings of the Association of Computational Linguistics (ACL 2007), Prague, Czech Republic. 221-224,2007

[11] Chirag Patel and Karthik Gali "*Part-Of-Speech Tagging for Gujarati Using Conditional Random Fields*", Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages, 2008

[12] Manish Shrivastava, Pushpak Bhattacharyya, "*Hindi POS Tagger Using Naive Stemming: Harnessing Morphological Information without Extensive Linguistic Knowledge*", International Conference on Natural language processing, 2008

[13] Smriti Singh, Kuhoo Gupta, Manish Shrivastava, and Pushpak Bhattacharyya, "*Morphological richness offsets resource demand – experiences in constructing a pos tagger for hindi*", In Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pages 779–786, Sydney, Australia, July. Association for Computational Linguistics, 2006.

[14] Manju K & Soumya S, "*Parts Of Speech Tagger for Malayalam*", A project report on Master of Technology in Computer and Information Science, Cochin University of Science & Technology, Kochi, May 2009.

[15] Antony P.J, Santhanu P. Mohan and Soman K.P, "*SVM Based Part of Speech Tagger for Malayalam*", itc, pp.339-341, International Conference on Recent Trends in Information, Telecommunication and Computing, 2010.

[16] Thoudam Doren Singh, Sivaji Bandyopadhyay, "*Morphology Driven Manipuri POS Tagger*", In proceeding of IJCNLP NLPLPL 2008, IIIT Hyderabad pp 91-97. 2008.

[17] Thoudam, Asif and Bandyopadhyay, "*Manipuri POS Tagging using CRF and SVM: A Language Independent Approach*", International Conference on Natural language processing, 2008.

[18] Bal Krishna Bal and Madan Puraskar Pustakalaya, "*Parts of Speech Tagger for Nepali*", Regional Conference on Localized ICT Development and Dissemination across Asia.PAN 7 Localization Project.12th- 16th January, 2009, Novotel Hotel, Vientiane, Laos

[19] http://www.ling.lancs.ac.uk/staff/hardie

[20] Linda Van Guilder, "*Automated Part of Speech Tagging: A Brief Overview*", Handout for LING361, Georgetown University, Fall 1995.

[21] http://punjabi.aglsoft.com/?show=tagger

[22] S. Lakshmana Pandian and T.V. Geetha, "*Morpheme Components based Part of Speech tagging*", International Conference on Natural language processing, 2008

[23] Arulmozhi. P, Sobha L, "*A hybrid POS Tagger for a Relatively Free Word Order Language*", In proceedings of MSPIL-2006, Indian Institute of Technology, Bombay. Pp 79-85, 2006

[24] Dhanalahsmi V, Anand Kumar, Shivapratap G, Soman KP and Rajendran S, "*Tamil POS Tagging using Linear Programming*", International Journal of Recent Trends in Engineering, Vol. 1, Vol. 2, May 2009

[25] Ahmed Muaz, Aasim Ali and Sarmad Hussain, "*Analysis and Development of Urdu POS Tagged Corpus*", Proceedings of the 7th Workshop on Asian Language Resources, ACL-IJCNLP 2009, pages 24–31, Suntec, Singapore, 6-7 August 2009.

[26] Sajjad H and Schmid H, "*Tagging Urdu Text with Parts Of Speech: A Tagger Comparison*", 12th Conference of the European chapter of the association for computational Linguistics, 2009

[27] Jesse Liberty and Donald Xie, "*Programming C# 3.0",* O'Reilly 5th edition, 2007.

==========

APPENDIX (LDC-IL Tagset without Attributes)

Assamese	Bengali	Bodo	Gujarati	Hindi	Malayalam
NC	NC	NC	NC	NC	NC
NP	NP	NP	NP	NP	NP
NV	NV	NV	NV	NV	NV
NST	NST	NST	NST	NST	PPR
PPR	PPR	PPR	PPR	PPR	PRF
PRF	PRF	PRF	PRF	PRF	PRC
PRC	PRC	PRC	PRC	PRC	PWH
PRL	PRL	PRL	PRL	PRL	DAB
PWH	PWH	PWH	PWH	PWH	DWH
DAB	DAB	DAB	DAB	DAB	JJ
DRL	DRL	DRL	DRL	DRL	JQ
DWH	DWH	DWH	DWH	DWH	JINT
JJ	JJ	JJ	JJ	JJ	V
JQ	JQ	JQ	JQ	JQ	AMN
JINT	JINT	JINT	JINT	JINT	ALC
VM	VM	VM	VM	VM	LRL
VA	VA	AMN	VA	VA	LV
AMN	AMN	ALC	AMN	AMN	LN
ALC	ALC	LV	ALC	PP	LC
PP	LV	LC	LPR	CCD	PP

Manipuri	Nepali	Oriya	Punjabi	Tamil	Urdu
NC	NC	NC	NC	NC	NC
NP	NP	NP	NP	NP	NP
NV	NV	NV	NV	NV	NV
NST	NST	NST	NST	PPR	NST
PPR	PPR	PPR	PPR	PRF	PPR
PRF	PRF	PRF	PRF	PRC	PRF
PRC	PRC	PRC	PRC	PWH	PRC
PWH	PRL	PRL	PRL	DAB	PRL
DAB	PWH	PWH	PWH	DWH	PWH
DWH	DAB	DAB	DAB	JJ	DAB
JJ	DRL	DRL	DRL	JQ	DRL
JQ	DWH	DWH	DWH	JINT	DWH
JINT	JJ	JJ	JJ	V	JJ
V	JQ	JQ	JQ	AMN	JQ
AMN	JINT	JINT	JINT	ALC	JINT
LRL	VM	VM	VM	LNF	VM
LV	VA	VA	VA	LRL	VA
LC	AMN	AMN	AMN	LV	AMN
CCD	LPRF	LRL	PP	LN	PP
CSB	LPFV	LV	CCD	PP	CCD

CCD	LC	PP	LPS	CSB	CCD
CSB	PP	CCD	LFU	CIN	CSB
CIN	CCD	CSB	PPC	CAGR	CIN
CAGR	CSB	CIN	PPNC	CEMP	CAGR
CDLIM	CIN	CAGR	CCD	CTOP	CCON
CDED	CAGR	CEPM	CSB	CDLIM	CDLIM
CDUB	CCL	CTOP	CIN	CHON	CX
CSIM	CSIM	CDLIM	CAGR	CDED	NUMR
CX	CINT	CDED	CEMP	CEXCL	NUMS
NUMR	CX	CEXCL	CTOP	CINT	NUMC
NUMS	NUMR	CTERM	CDLIM	CDUB	NUMO
NUMC	NUMS	CDUB	CHON	CSIM	RDP
NUMO	NUMC	CSIM	CNEG	CX	RDF
RDP	NUMO	CX	EXCL	NUMR	RDS
RDF	RDP	NUMR	CTERM	NUMS	UNK
RDS	RDF	NUMS	CDUB	NUMC	PU
UNK	RDS	NUMC	CSIM	NUMO	
PU	UNK	NUMO	CINCL	RDP	
	PU	RDP	CCOM	RDF	
		RDF	CX	RDS	
		RDS	NUMR	UNK	
		UNK	NUMS	PU	
		PU	NUMC		
			NUMO		
			RDP		
			RDF		
			RDS		
			UNK		
			PU		

CIN	LIPFV	LC	CSB	LC	CSB
CAGR	PP	PP	CIN	CCD	CIN
CTERM	CCD	CCD	CAGR	CSB	CAGR
CDELIM	CSB	CSB	CEMP	CIN	CEMP
CCUM	CIN	CIN	CTOP	CX	CTOP
CDED	CCL	CCL	CDLIM	CARG	CDLIM
CX	CEMP	CAGR	CHON	CCON	CHON
NUMR	CEVID	CEMP	CDED	CDLIM	CDED
NUMS	CDLIM	CDLIM	CEXCL	NUMR	CEXCL
NUMC	CHON	CEXCL	CDUB	NUMS	CINT
NUMO	CDED	CTERM	CINT	NUMC	CDUB
RDP	CCLU	CDUB	CSIM	NUMO	CSIM
RDF	CTERM	CSIM	CX	RDP	CX
RDS	CDUB	CX	NUMR	RDF	NUMR
UNK	CSIM	NUMR	NUMS	RDS	NUMS
PU	CCOM	NUMS	NUMC	UNK	NUMC
	CPRT	NUMC	NUMO	PU	NUMO
	CACCD	NUMO	RDP		RDP
	CDLNK	RDP	RDF		RDF
	CX	RDF	RDS		RDS
	NUMR	RDS	UNK		UNK
	NUMS	UNK	PU		PU
	NUMC	PU			
	NUMO				
	RDP				
	RDF				
	RDS				
	UNK				
	PU				

Description

Noun (N)	**Nominal Modifier(J)**	(Dis)Agreement (CAGR)	Comparative (CCOM)	Others (CX)
Common(NC)	Adjective (JJ)	Emphatic (CEMP)	Classifier (CCL)	**Adverb (A)**
Proper(NP)	Quantifier (JQ)	Topic(CTOP)	Interrogative(CINT)	Manner (AMN)
Verbal(NV)	Intensifier (JINT)	Delimiting (CDLIM)	Dedative(CDED)	Location(ALC)
Spatio-temporal (NST)	**Verb (V)**	Honorific (CHON)	Confirmative (CCON)	**Numeral (NUM)**
Pronoun (P)	Main Verb (VM)	Negative (CNEG)	Cumulative(CCUM)	Real (NUMR)
Pronominal (PPR)	Auxiliary Verb (VA)	Exclusive(CEXCL)	Evidential (CEVID)	Serial (NUMS)
Reflexive (PRF)	**Particle (C)**	Terminative (CTERM)	Clusive (CCLU)	Calendric (NUMC)
Reciprocal (PRC)	Co-ordinating(CCD)	Dubitative (CDUB)	Partitive (CPRT)	Ordinal (NUMO)
Relative (PRL)	Subordinating (CSB)	Similative (CSIM)	Accordance (CACCD)	**Reduplication (RDP)**
Wh-pronoun (PWH)	Interjection (CIN)	Inclusive (CINCL)	Discourse-linking (CDLNK)	

Demonstrative (D)	**Postposition (PP)**	Verbal (LV)	Future (LFU)	**Residual (RD)**
Absolutive (DAB)	Case (PPC)	Nominal (LN)	Perfective(LPFV)	Foreign Word RDF)
Relative Demonstrative (DRL)	Non-Case (PPNC)	Conditional (LC)	Perfect(LPRF)	Symbol (RDS)
Wh-demonstrative (DWH)	**Participle (L)**	Present (LPR)	Imperfective(LIPFV)	**Unknown (UNK)**
	Relative (LRL)	Past (LPS)	Infinite (LNF)	**Punctuation (PU)**

==========

M. Mohamed Yoonus, M.Sc., M.Phil., PGDNLP
Lecturer cum Resource Person
LDC-IL Project,
Central Institute of Indian Languages
Mysore 570006
Karnataka, India
yoonussoft@gmail.com

Samar Sinha, M.A., M.Phil.
Senior Lecturer cum Junior Research Officer
LDC-IL Project
Central Institute of Indian Languages
Mysore 570006
Karnataka, India
samarsinha@gmail.com