Workshop on Indian Language and Data: Resources and Evaluation

Workshop Programme

21 May 2012

08:30-08:40 – Welcome by Workshop Chairs

08:40-08:55 – Inaugural Address by Mrs. Swaran Lata, Head, TDIL, Dept of IT, Govt of India

08:55-09:10 – Address by Dr. Khalid Choukri, ELDA CEO

0910-09:45 – Keynote Lecture by Prof Pushpak Bhattacharyya, Dept of CSE, IIT Bombay.

09:45-10:30 – Paper Session I
Chairperson: Sobha L

- Somnath Chandra, Swaran Lata and Swati Arora, *Standardization of POS Tag Set for Indian Languages based on XML Internationalization best practices guidelines*
- Ankush Gupta and Kiran Pala, *A Generic and Robust Algorithm for Paragraph Alignment and its Impact on Sentence Alignment in Parallel Corpora*
- Malarkodi C.S and Sobha Lalitha Devi, *A Deeper Look into Features for NE Resolution in Indian Languages*

10:30 – 11:00 Coffee break + Poster Session
Chairperson: Monojit Choudhury

- Akilandeswari A, Bakiyavathi T and Sobha Lalitha Devi, *'atu' Difficult Pronominal in Tamil*
- Subhash Chandra, *Restructuring of Painian Morphological Rules for Computer processing of Sanskrit Nominal Inflections*
- H. Mamata Devi, Th. Keat Singh, Bindia L and Vijay Kumar, *On the Development of Manipuri-Hindi Parallel Corpus*
- Madhav Gopal, *Annotating Bundeli Corpus Using the BIS POS Tagset*
- Madhav Gopal and Girish Nath Jha, *Developing Sanskrit Corpora Based on the National Standard: Issues and Challenges*
- Ajit Kumar and Vishal Goyal, *Practical Approach For Developing Hindi-Punjabi Parallel Corpus*
- Sachin Kumar, Girish Nath Jha and Sobha Lalitha Devi, *Challenges in Developing Named Entity Recognition System for Sanskrit*
- Swaran Lata and Swati Arora, *Exploratory Analysis of Punjabi Tones in relation to orthographic characters: A Case Study*
- Diwakar Mishra, Kalika Bali and Girish Nath Jha, *Grapheme-to-Phoneme converter for Sanskrit Speech Synthesis*
- Aparna Mukherjee and Alok Dadhekar, *Phonetic Dictionary for Indian English*
- Sibansu Mukhapadyay, Tirthankar Dasgupta and Anupam Basu, *Development of an Online Repository of Bangla Literary Texts and its Ontological Representation for Advance Search Options*
- Kumar Nripendra Pathak, *Challenges in Sanskrit-Hindi Adjective Mapping*

- Nikhil Priyatam Pattisapu, Srikanth Reddy Vadepally and Vasudeva Varma, *Hindi Web Page Collection tagged with Tourism Health and Miscellaneous*
- Arulmozi S, Balasubramanian G and Rajendran S, *Treatment of Tamil Deverbal Nouns in BIS Tagset*
- Silvia Staurengo, *TschwaneLex Suite (5.0.0.414) Software to Create Italian-Hindi and Hindi-Italian Terminological Database on Food, Nutrition, Biotechnologies and Safety on Nutrition: a Case Study.*

11:00 – 12:00 – Paper Session II
Chairperson: Kalika Bali

- Shahid Mushtaq Bhat and Richa Srishti, *Building Large Scale POS Annotated Corpus for Hindi & Urdu*
- Vijay Sundar Ram R, Bakiyavathi T, Sindhuja Gopalan, Amudha K and Sobha Lalitha Devi, *Tamil Clause Boundary Identification: Annotation and Evaluation*
- Manjira Sinha, Tirthankar Dasgupta and Anupam Basu, *A Complex Network Analysis of Syllables in Bangla through SyllableNet*
- Pinkey Nainwani, *Blurring the demarcation between Machine Assisted Translation (MAT) and Machine Translation (MT): the case of English and Sindhi*

12:00-12:40 – Panel discussion on "*India and Europe - making a common cause in LTRs*"
Coordinator: Nicoletta Calzolari
Panelists - Kahlid Choukri, Joseph Mariani, Pushpak Bhattacharya, Swaran Lata, Monojit Choudhury, Zygmunt Vetulani, Dafydd Gibbon

12:40- 12:55 – Valedictory Address by Prof Nicoletta Calzolari, Director ILC-CNR, Italy

12:55-13:00 – Vote of Thanks

Editors

Girish Nath Jha	Jawaharlal Nehru University, New Delhi
Kalika Bali	Microsoft Research Lab India, Bangalore
Sobha L	AU-KBC Research Centre, Anna University, Chennai

Workshop Organizers/Organizing Committee

Girish Nath Jha	Jawaharlal Nehru University, New Delhi
Kalika Bali	Microsoft Research Lab India, Bangalore
Sobha L	AU-KBC Research Centre, Anna University, Chennai

Workshop Programme Committee

A. Kumaran	Microsoft Research Lab India, Bangalore
A. G. Ramakrishnan	IISc Bangalore
Amba Kulkarni	University of Hyderabad
Dafydd Gibbon	Universitat Bielefeld, Germany
Dipti Mishra Sharma	IIIT, Hyderabad
Girish Nath Jha	Jawaharlal Nehru University, New Delhi
Joseph Mariani	LIMSI-CNRS, France
Kalika Bali	Microsoft Research Lab India, Bangalore
Khalid Choukri	ELRA, France
Monojit Choudhury	Microsoft Research Lab India, Bangalore
Nicoletta Calzolari	ILC-CNR, Pisa, Italy
Niladri Shekhar Dash	ISI Kolkata
Shivaji Bandhopadhyah	Jadavpur University, Kolkata
Sobha L	AU-KBC Research Centre, Anna University
Soma Paul	IIIT, Hyderabad
Umamaheshwar Rao	University of Hyderabad

Table of contents

Author Index

Introduction

WILDRE – the first 'Workshop on Indian Language Data: Resources and Evaluation' is being organized in Istanbul, Turkey on 21st May, 2012 under the LREC platform. India has a huge linguistic diversity and has seen concerted efforts from the Indian government and industry towards developing language resources. European Language Resource Association (ELRA) and its associate organizations have been very active and successful in addressing the challenges and opportunities related to language resource creation and evaluation. It is therefore a great opportunity for resource creators of Indian languages to showcase their work on this platform and also to interact and learn from those involved in similar initiatives all over the world.

The broader objectives of the WILDRE is

- To map the status of Indian Language Resources
- To investigate challenges related to creating and sharing various levels of language resources
- To promote a dialogue between language resource developers and users
- To provide opportunity for researchers from India to collaborate with researchers from other parts of the world

The call for papers received a good response from the Indian language technology community. Out of 34 full papers received for review, we selected 24 for presentation in the workshop (7 for oral and 17 as posters).

Standardization of POS Tag Set for Indian Languages based on XML Internationalization best practices guidelines

Swaran Lata, Somnath Chandra, Prashant Verma and Swati Arora
Department of Information Technology
Ministry of Communications & Information Technology, Govt. of India
6 CGO Complex, Lodhi Road, New Delhi 110003
E-mail: slata@mit.gov.in, schandra@mit.gov.in, Verma@w3.org, Arora@w3.org

Abstract:
This paper presents a universal Parts of Speech (POS) tag set using W3C XML framework covering the major Indian Languages. The present work attempts to develop a common national framework for POS tag-set for Indian languages to enable a reusable and extendable architecture that would be useful for development of Web based Indian Language technologies such as Machine Translation , Cross-lingual Information Access and other Natural Language Processing technologies. The present POS tag schema has been developed for 13 Indian languages and being extended for all 22 constitutionally recognized Indian Languages. The POS schema has been developed using international standards e.g. metadata as per ISO 12620:1999, schema as per W3C XML internationalization guidelines and one to one mapping labels used 13 Indian languages.

1. Introduction:

Parts of Speech tagging is one the key building block for developing Natural Language Processing applications. A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags. The early efforts of POS tag set development was based on Latin based languages that lead to the development of POS structures such as Upenn, Brown and C5 [1]-[3] which were mostly flat in nature. The hierarchical structure of POS tag set was first demonstrated under the EAGLES recommendations for morpho-syntactic annotation of corpora (Leech and Wilson, 1996) to develop a common tag-set guideline for several European languages [4].

In India, several efforts have been made for development of POS schema for Natural Language Processing applications in Indian Languages. Some of efforts are (i) POS structure by Central Institute of Indian Languages (CIIL) , Mysore , (ii) POS schema developed by IIIT Hyderabad. These POS structures are mostly flat in nature, capturing only coarse-level categories and are linked to Language Specific technology development. Thus, these POS structures could not be reused and non-extensible for other Indian Languages. Another disadvantage that has been observed is that these flat POS schema have not been developed in XML format, thus the use of these schema are limited to the stand-alone applications. To overcome the difficulties of the flat POS schema, first attempt of development of Hierarchical POS schema was reported in by Bhaskaran et.al [5]. However, the structure does not have the backward compatibility of the earlier POS schemas of CIIL Mysore and IIIT Hyderabad.

In order to overcome the lacunae and shortcomings of the existing POS schemas, Dept of Information Technology, Govt. of India has developed a common, hierarchical, reusable and extensible POS schema for all 22 constitutionally recognized Indian Languages. The present schema development has been completed for 13 major Indian Languages and would soon be extended for 22 Indian Languages. The schema is based on W3C XML Internationalization best practices, used ISO 639-3 for Language identification, ISO 12620:1999 as metadata definition and one to one mapping table for all the labels used in POS schema.

The paper is organized as follows. Section 2 describes the comparison of existing POS schema for Indian Languages and how common framework for the present XML based POS schema has been developed using all the features of the present schemas to achieve seamless compatibility. In Section 3, we have described the one-to one mapping table of 13 Indian Languages to have the common framework. The XML based schema using the ISO Language Tag and Metadata standard has been described in section 4. Finally the conclusion and future plan is drawn in Section 5.

2. Development of Common Framework for POS schema in Indian Languages.

It has been mentioned that a slew of the POS schemas are presently exist for Indian Languages. The schemas developed by CIIL and IIIT Hyderabad are flat in nature and that proposed by Bhaskaran et-al are hierarchical.

A comparison of the existing POS schemas is elucidated in Table 1 below:

Table 1: Comparison of Existing POS schemas

CIIL	**IIIT-H**	**Bhaskaran etal**	
Structure : Flat	**Structure : Flat**	**Structure : Hierarchical**	
NN (Common Noun)	NN (Common Noun)	**Noun (N)**	Common (C)
NNP (Proper Noun)	NNP (Proper Noun)		Proper (P)
NC (Noun Compound)	*C (for all compounds)		Verbal (V)
NAB (abstract Noun)			Spatiotemporal (ST)
CRD (Cardinal No.)	QC (Cardinal No.)		
ORD (Ordinal No.)	QO (Ordinal No.)		
PRP (Personal Pronoun)	PRP (Pronoun)	**Pronoun (P)**	Pronominal (PR)
PRI (Indefinite Pronoun)			Reflexive (RF)
PRR (Reflexive Pronoun)			Reciprocal (RC)
PRL (Relative Pronoun)			Relative (RL)

PDP (Demonstrative)			Wh (WH)
VF (Verb Finite Main)	VF (Verb Finite Main)	**Verb (V)**	Main(M)
VNF (Verb Non-Finite adverbial and adjectival)	VNF (Verb Non-Finite adverbial and adjectival		
VAX (Verb Auxiliary)	VAUX (Verb Auxiliary)		
VNN (Gerund/Verb non-finite nominal)	VNN (Gerund/Verb non-finite nominal)		
VINF (Verb Infinitive)	VINF (Verb Infinitive)		Auxiliary (A)
VCC (Verb Causative)			
VCD (Verb Double Causative)			
	JJ (Adjectives)		
ADD (Adjective Declinable)		** Radically Different from CIIL and IIIT Hyderabad Tag sets are placed in Table 2	
ADI (Adjective Indeclinable)			
IND (Indeclinable)			
QOT (Quotative)			
RDP (Reduplication)			
FWD (Loan Word)			
IDM (Idiom)			
PRO (Proverb)			
CL (Classifier)			
SYM (Special)			

It has been observed that, there are significant differences in the above POS schema. To minimize such differences, and to ensure backward compatibility, Dept of Information Technology has proposed the common framework of POS schema as defined in Table 2 below:

Table 2: Proposed Schema for Common Framework of POS in Indian Languages

S.No.	English
Noun Block	Noun
	common
	Proper
	Verbal
	Nloc
Pronoun Block	Pronoun
	Personal

	Reflexive
	Reciprocal
	Relative
	Wh-words
	Indefinite
Demonstrative Block	Demonstrative
	Deictic
	Relative
	Wh-words
	Indefinite
Verb Block	Verb
	Auxiliary Verb
	Main Verb
	Finite
	Infinitive
	Gerund
	Non-Finite
	Participle Noun
Adjective Block	Adjective
Adverb Block	Adverb
Post Position Block	Post Position
Conjunction Block	Conjunction
	Co-ordinator
	Subordinator
	Quotative
Particles Block	Particles
	Default
	Classifier
	Interjection
	Negation
	Intensifier
Quantifier Block	Quantifiers
	General
	Cardinals
	Ordinals
Residual Block	Residuals
	Foreign word

	Symbol
	Unknown
	Punctuation
	Echo-words

The above structure has taken into account the features of both the existing flat and hierarchical schema structures and has been agreed upon by linguists and language experts for developing NLP applications in Indian languages

3. One to One Mapping Table for Labels in POS Schema

In order to develop common framework of XML based POS schema in all 22 Indian Languages, it is necessary that labels defined in POS schema for English to have one to one mapping for Indian Languages. The XML schema needs to have a complete tree structure as depicted in Fig1. Below:

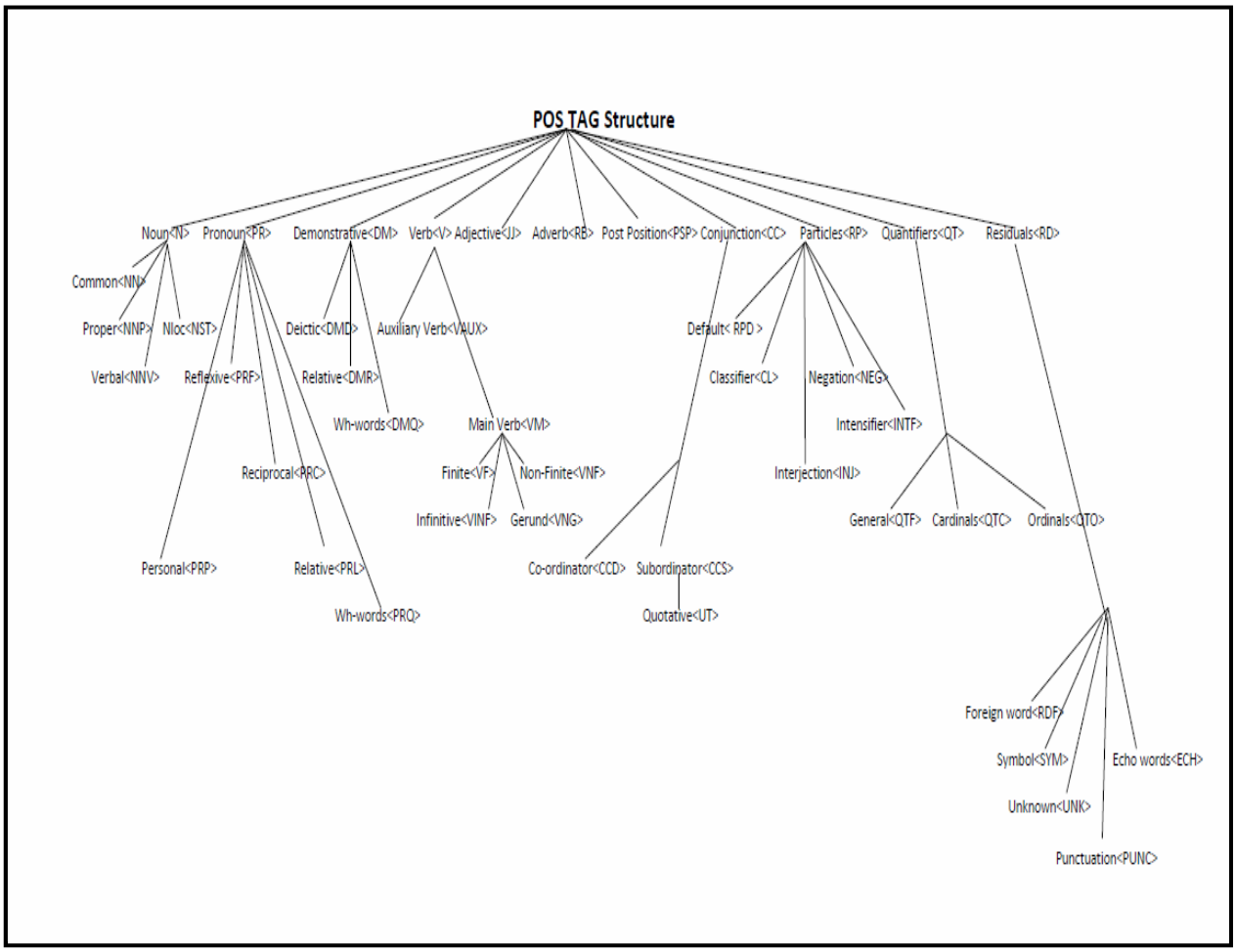
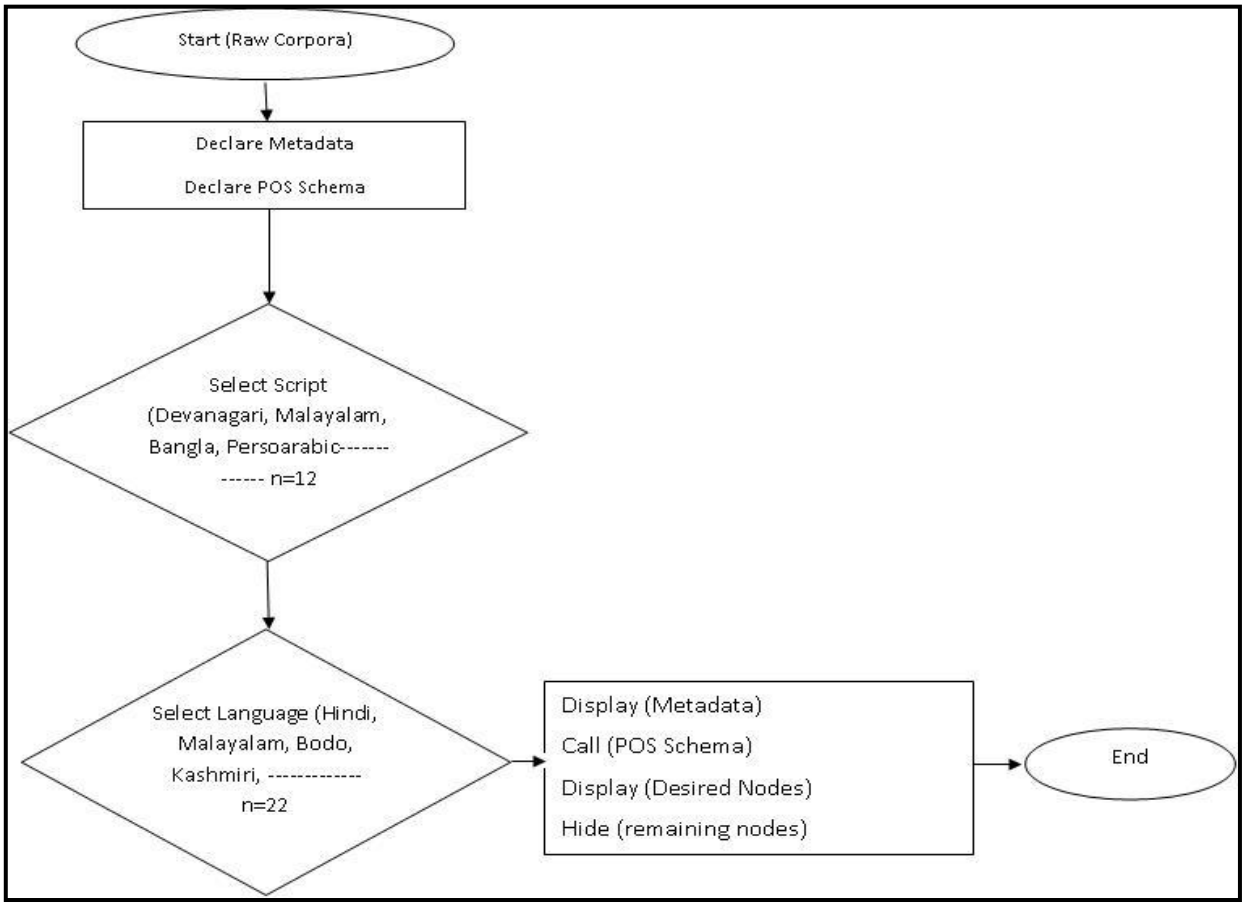


Fig1. Tree POS Schema structure

The Common XML schema would select a particular Indian Language by and the Schema then needs to be transformed into POS schema for that particular language. The language specific POS schema could be enabled by making a particular branch of the tree structure 'off'. It is schematically represented in Fig 2. Below:



Draft version of one to one mapping table to incorporate such facility in the XML schema as shown in **Annexure I**.

Similar one to one Mapping tables have also been generated for Assamese, Bodo, Kashmiri (Urdu script) , Marathi ,Malayalam and Konkani etc also shown in **Annexure I**.

4. XML POS schema for Indian Languages

To make the common POS schema for Indian Languages completely interoperable, extensible and web enabled, W3C XML Internationalization best practices guidelines [6]-[8] and ISO Metadata standard [9] are adopted in the above framework. The set of W3C internationalization guidelines that are adopted are elaborated in Table 4 below:

XML Best practices	Tag
Defining markup for natural language labelling	Xml:lang -defined for the root element of your document, and for any element where a change of language may occur.
Defining mark-up to specify text direction	Its:dir -attribute is defined for the root element of your document, and for any element that has text content.
Indicating which elements and attributes should be translated	its:translateRule -element to address this requirement.
Providing information related to text segmentation	Ita:within Text Rule -elements to indicate which elements should be treated as either part of their parents, or as a nested but independent run of text.
Defining markup for unique identifiers	xml:id -elements with translatable content can be associated with a unique identifier.

The draft Common POS Schema based on the above best practices is the architecture defined in section 3 as given in Annexure II. It is evident from the XML based schema as shown in Annexure II that ; (i) it Supports multilingual documents and Unicode (ii) It allows developers to add extra information to a format without breaking applications. Further, the tree structure of XML documents allows documents to be compared and aggregated efficiently element by element and is easier to convert data between different data types.(iii)This XML schema helps annotators to select their script and language/languages in order to get the XML scheme based on their requirements.

5. Conclusions:

The common unified XML based POS schema for Indian Languages based on W3C Internationalization best practices have been formulated. The schema has been developed to take into account the NLP requirements for Web based services in Indian Languages. The present schema would further be validated by linguists and would be evolved towards a national standard by Bureau of Indian Standards

6. References:

[1] Cloeren, J. (1999) Tagsets. In *Syntactic Wordclass Tagging*, ed. Hans van Halteren, Dordrecht: Kluwer Academic. Hardie, A. (2004). The Computational Analysis of Morpho-syntactic Categories in Urdu. PhD Thesis submitted to Lancaster University.

[2] Greene, B.B. and Rubin, G.M. (1981). Automatic grammatical tagging of English. Providence, R.I.:Department of Linguistics, Brown University.

[3] Garside, R. (1987) The CLAWS word-tagging system. In *The Computational Analysis of English*, ed. Garside, Leech and Sampson, London: Longman.

[4] Leech, G and Wilson, A. (1996), Recommendations for the Morpho-syntactic Annotation of Corpora. EAGLES Report EAG-TCWG-MAC/R.

[5] Bhaskaran et.al [2008] *A Common Parts-of-Speech Tag-set Framework for Indian Languages* Proc. LREC 2008

[6] Best Practices for XML Internationalization: *http://www.w3.org/TR/xml-i18n-bp/*
[7] Internationalization Tag Set (ITS) Version 1.0: http://www.w3.org/TR/2007/REC-its-20070403/
[8] XML Schema Requirements: http://www.w3.org/TR/1999/NOTE-xml-schema-req-19990215
[9] ISO 12620:1999, Terminology and other language and content resources — Specification of data categories and management of a Data Category Registry for language resources
[10] ISO 639-3, Language Codes: http://www.sil.org/iso639-3/codes.asp
[11] www.w3.org/2010/02/convapps/Papers/Position-Paper_-India-W3C_Workshop-PLS-final.pdf

Annexure I

Languages: Hindi, Punjabi, Urdu, Gujarati, Oriya, Bengali

S. No	English	Hindi	Punjabi	Urdu	Gujarati	Odiya	Bengali
1	Noun	*संज्ञा*	ਨਾਂਵ	اسم	સંજ્ઞા	ସଂଜ୍ଞା	বিশেষ্য
	common	जातिवाचक	ਆਮ	نکرہ	જાતિવાચક	ଜାତିବାଚକ	জাতিবাচক
	Proper	*व्यक्तिवाचक*	ਖਾਸ	معرفہ	વ્યક્તિવાચક	ବ୍ୟକ୍ତିବାଚକ	ব্যক্তিবাচক
	Verbal	क्रियामूलक / कृदंत	ਕਿਰਿਆਮੂਲਕ	حاصل مصدر	ક્રિયાવાચક	କ୍ରିୟାବାଚକ	ক্রিয়ামূলক
	Nloc	देश-काल सापेक्ष	ਸਥਿਤੀ ਸੂਚਕ	ظرف	સ્થાનવાચક	ଦେଶ-କାଳ ସାପେକ୍ଷ	স্থানবাচক
2	Pronoun	*सर्वनाम*	ਪੜਨਾਂਵ	ضمیر	સર્વનામ	ସର୍ବନାମ	সর্বনাম
	Personal	*व्यक्तिवाचक*	ਪੁਰਖਵਾਚੀ	ضمیر شخصی	પુરુષવાચક	ବ୍ୟକ୍ତିବାଚକ	ব্যক্তিবাচক
	Reflexive	निजवाचक	ਨਿਜਵਾਚੀ	ضمیر معکوسی	પ્રતિબિંબિત	ଆତ୍ମବାଚକ	আত্মবাচক
	Reciprocal	पारस्परिक	ਪਰਸਪਰੀ	ضمیر راجع	પરસ્પરવાચી	ପାରସ୍ପାରିକ	ব্যতিহার
	Relative	संबंध- वाचक	ਸੰਬੰਧਵਾਚੀ	ضمیر موصولہ	સાપેક્ષ	ସଂବନ୍ଧବାଚକ	সম্বন্ধবাচক
	Wh-words	प्रश्नवाचक	ਪ੍ਰਸ਼ਨਵਾਚੀ	ضمیر استفہامیہ	પ્રશ્નાર્થવાયક	ପ୍ରଶ୍ନବାଚକ	প্রশ্নবাচক
	Indefinite	अनिश्चयवाचक	NA	NA	અનિશ્ચિત સર્વનામ	NA	অনির্দেশ্য
3	Demonstrative	निश्चयवाचक/ संकेतवाचक	ਸੰਕੇਤਵਾਚੀ	اشارے	દર્શકો	ନିଶ୍ଚୟବାଚକ/ସଂକେତବାଚକ	নির্দেশক
	Deictic	निर्देशी	ਪ੍ਰਤੱਖ ਪ੍ਰਮਾਣਵਾਚੀ	اشارہ	ઉલ્લેખદર્શક		প্রত্যক্ষ নির্দেশক
	Relative	संबंधवाचक	ਸੰਬੰਧਵਾਚੀ	اشارہ موصول	સાપેક્ષ	ସଂବନ୍ଧବାଚକ	সম্বন্ধবাচক
	Wh-words	प्रश्नवाचक	ਪ੍ਰਸ਼ਨਵਾਚੀ	اشارہ استفہامیہ	પ્રશ્નવાચી	ପ୍ରଶ୍ନବାଚକ	প্রশ্নবাচক
	Indefinite	अनिश्चयवाचक	NA	NA	અનિશ્ચિત સર્વનામ	NA	অনির্দেশ্য
4	Verb	*क्रिया*	ਕਿਰਿਆ	فعل	આખ્યાત	କ୍ରିୟା	ক্রিয়া
	Auxiliary Verb	सहायक क्रिया	ਸਹਾਇਕ ਕਿਰਿਆ	فعل امدادی	સહાયકારી ક્રિયા	ସହାୟକ କ୍ରିୟା	গৌণ ক্রিয়া
	Main Verb	मुख्य क्रिया	ਮੁੱਖ ਕਿਰਿਆ	فعل لازم	મુખ્ય	ମୁଖ୍ୟ କ୍ରିୟା	মুখ্য ক্রিয়াপদ
	Finite	परिमित	ਕਾਲਕੀ	فعل محدود	પૂર્ણ	ପରିମିତ	সমাপিকা
	Infinitive	क्रियार्थक संज्ञा	ਅਮਿਤ	مصدر	હેત્વર્થ	ଅନନ୍ତ	অপূর্ণ ক্রিয়া
	Gerund	क्रियावाचक	ਕਿਰਿਆਵਾਚੀ	حاصل مصدر	વર્તમાનકૃદન્ત	କ୍ରିୟାବାଚକ	প্রযোজক ক্রিয়া
	Non-Finite	गैर-परिमित	ਅਕਾਲਕੀ	فعل غیر محدود	અપૂર્ણ	ଅପରିମିତ	অসমাপিকা
	Participle Noun	कृदंत परक नाम	NA	NA	NA	NA	ক্রিয়াজাত বিশেষ্য
5	Adjective	*विशेषण*	ਵਿਸ਼ੇਸ਼ਣ	صفت	વિશેષણ	ବିଶେଷଣ	বিশেষণ
6	Adverb	*क्रिया-विशेषण*	ਕਿਰਿਆ ਵਿਸ਼ੇਸ਼ਣ	متعلّق فعل	ક્રિયાવિશેષણ	କ୍ରିୟା-ବିଶେଷଣ	ক্রিয়া-বিশেষণ

7	Post Position	परसर्ग	ਸਬੰਧਕ	جار موخّر	અનુગો	ପରସର୍ଗ	পরসর্গ
8	Conjunction	योजक	ਯੋਜਕ	حرف عطف	સંયોજકો	ସଂଯୋଜକ	সংযোগমূলক
	Co-ordinator	समन्वयक	ਸਮਾਨ ਯੋਜਕ	حرف وصل	સહક્રિયાદર્શક	ସମନ୍ୱୟକ	সমন্বয়ক
	Subordinator	अधीनस्थ	ਅਧੀਨ ਯੋਜਕ	حرف تابع کننده	ગૌણક્રિયાદર્શક		শর্ত সংযোজক
	Quotative	उक्ति-वाचक	ਕਥਨਵਾਚੀ	حرف اقتباسی	NA	ଉକ୍ତିବାଚକ	উক্তিবাচক
9	Particles	अव्यय	ਨਿਪਾਤ	حالیہ/حرف پابند	નિપાત	ଅବ୍ୟୟ / ନିପାତ	অব্যয়
	Default	व्यतिक्रम	ਤਰੁਟੀਵਾਚਕ	حرف ڈیفالٹ	સ્વયંભૂ	ବ୍ୟତିକ୍ରମ	সাধারণ অব্যয়
	Classifier	वर्गीकारक	ਵਰਗੀਕ੍ਰਿਤ	حرف درجہ بند	NA	ବର୍ଗୀକାରକ	বর্গবাচক
	Interjection	विस्मयादिबोधक	ਵਿਸਮਕ	حرف فجائیہ	વિસ્મયઆદિ બોધક	ବିସ୍ମୟ ବୋଧକ	বিস্ময়াদিবোধক
	Negation	नकारात्मक	ਨਾਂਹਵਾਚੀ	حرف نہی	નકારદર્શક	ନିଷେଧାତ୍ମକ	নঞর্থক
	Intensifier	तीव्रक	ਤੀਬਰਤਾਵਾਚੀ	حرف تاکید	માત્રાસૂચક	ତୀବ୍ରତାବାଚକ	তীব্রতাবোধক
10	Quantifiers	संख्यावाची	ਸੰਖਿਆਵਾਚੀ	کمیت نما	પરિમાણસૂચકો	ସଂଖ୍ୟାବାଚୀ	পরিমাণবাচক
	General	सामान्य	ਸਧਾਰਨ	عمومی/ عام	સામાન્ય	ସାମାନ୍ୟ	সাধারণ
	Cardinals	गणनासूचक	ਗਿਣਤੀਸੂਚਕ	اعداد مطلق	સંખ્યાવાચક	ଗଣନାସୂଚକ	সংখ্যাবাচক
	Ordinals	क्रमसूचक	ਕ੍ਰਮਸੂਚਕ	ترتیبی اعداد	ક્રમવાચક	କ୍ରମସୂଚକ	ক্রমবাচক
11	Residuals	अवशेष	ਬਾਕੀ	باقی ماندہ	શેષ	ଅବଶେଷ	অবশিষ্ট পদ
	Foreign word	विदेशी शब्द	ਵਿਦੇਸ਼ੀ ਸ਼ਬਦ	بیرونی لفظ	પરદેશી શબ્દો	ବିଦେଶୀ ଶବ୍ଦ	বিদেশী শব্দ
	Symbol	प्रतीक	ਸੰਕੇਤ	علامت	સંકેત	ପ୍ରତୀକ	প্রতীক
	Unknown	अज्ञात	ਅਗਿਆਤ	نامعلوم	અજાણ્યા શબ્દો	ଅଜ୍ଞାତ	অজ্ঞাত
	Punctuation	विरामादि-चिह्न	ਵਿਸ਼ਰਾਮ ਚਿੰਨ੍ਹ	اوقاف	વિરામચિહ્નો	ବିରାମ ଚିହ୍ନ	যতিচিহ্ন
	Echowords	प्रतिध्वनि-शब्द	ਪ੍ਰਤਿਧੁਨੀ ਸ਼ਬਦ	گونج دار الفاظ	અનુરણનાત્મક	ପ୍ରତିଧ୍ୱନୀ	অনুকার শব্দ

Languages: Assamese, Bodo, Kashmiri (Urdu Script), Kashmiri (Hindi Script), Marathi

S.No	English	Hindi	Assamese	Bodo	Kashmiri	Kashmiri (Hindi)	Marathi
1	Noun	*संज्ञा*	বিশেষ্য	मुंमा	ناوُت	नावुत	नाम
	common	जातिवाचक	জাতিবাচক	फोलेर दिन्थिग्रा	عام	आम	सामान्य नाम
	Proper	*व्यक्तिवाचक*	ব্যক্তিবাচক	मुं दिन्थिग्रा	خاص	ख़ास	वशिष नाम
	Verbal	क्रियामूलक / कृदंत	ক্রিয়াবাচক	हाबा दिन्थिग्रा	کرٕاوِتأوی	क्रावतांव्य	धातुसाधति नाम
	Nloc	देश-काल सापेक्ष	স্থানবাচক	थावनि दिन्थिग्रा मुंमा	ناوتہٕ جایہٕ باو	नाव त जायि हाव	देश कालवाचक नाम
2	Pronoun	*सर्वनाम*	সর্বনাম	मुंराइ	پَرناوُت	पर नावुत	सर्वनाम
	Personal	*व्यक्तिवाचक*	ব্যক্তিবাচক	संबुं दिन्थिग्रा	شخصیأتی	शख्सियांती	पुरुषवाचक
	Reflexive	निजवाचक	আত্মবাচক	गाव दिन्थिग्रा	ماکوسی	माकूसी	आत्मवाचक
	Reciprocal	पारस्परिक	পাৰস্পৰিক	गावजों गाव सोमोन्दो	باہمی	बाहिमी/ बोहिमी	पारस्पारकि

	Relative	संबंध- वाचक	সম্বন্ধবাচক	সোমোন্দো দিন্থিগ্রা	رأبتأوى	রোবিতাংব্য	संबंधवाची
	Wh-words	प्रश्नवाचक	প্রশ্নবোধক সর্বনাম	সৌথি দিন্থিগ্রা	ک لفظ	ক-লফ়জ়	प्रश्नार्थक
	Indefinite	अनिश्चयवाचक					
3	Demonstrative	निश्चयवाच/ संकेतवाचक	নির্দেশবোধক	থাবনি দিন্থিগ্রা	باوَن پَرناوِتى	হাবন পরনাবুত্য	दर्शक
	Deictic	निर्देशी	প্রত্যক্ষ নির্দেশক	থি দিন্থিগ্রা	وأنيأوى	বোনযোব্য	
	Relative	सम्बन्ध वाचक	সম্বন্ধবাচক	সোমোন্দো দিন্থিগ্রা	رأبتأوى	রোবতাংত্য	संबंधवाच/ संबंधदर्शक
	Wh-words	प्रश्नवाचक	প্রশ্নবোধক অব্যয়	ম সৌথি দিন্থিগ্রা	ک لفظ	ক-লফ়জ়	प्रश्नार्थक
	Indefinite	अनिश्चयवाचक	NA	NA	NA	NA	NA
4	Verb	*क्रिया*	ক্রিয়া	থাइজা	كرَاوُت	ক্রাবুত	क्रियापद
	Auxiliary Verb	सहायक क्रिया	সহায়কাৰী ক্রিয়া	লেডাइ থাइজা	ڈکھہٕ کراوُت	डख क्रावुत	सहायकारी क्रियापद
	Main Verb	मुख्य क्रिया	মুখ্য ক্রিয়া	গুबै থাइজা	راے کراوُت	राय क्रावुत	मुख्य क्रियापद
	Finite	परिमित	সমাপিকা	জাফুंজা থাइজা	ہِشرِ باو	हिशर हाव	आख्यात क्रियारूप
	Infinitive	अनंत	অসমাপিকা	জাফুডি থাइজা	ہِشرِ کھاو	हिशर खाव	भाववाचक कृदंत
	Gerund	क्रियावाचक	নিমিত্তার্থক সংজ্ঞা	জাফুবায় থানায় দিন্থিগ্রা	کرَاوتہٕ ناوُت	क्राव त नावुत	वभिक्तक्षिम कृदंतरूप
	Non-Finite	गैर-परिमित	অসমাপিকা	জাফুডি থাइজা	نا ہِشرِ باو	ना हिशर हाव	आख्यातेतर क्रियारूप
	Participle Noun	कृदंत परक नाम	NA	NA	NA	NA	NA
5	Adjective	*विशेषण*	বিশেষণ	থাइলালি	باوُت	बावुत	वशिषण
6	Adverb	*क्रिया-विशेषण*	ক্রিয়া বিশেষণ	থাइজানি থাइলালি	لَگہٕ باَش	लग बांश	क्रियावशिषण
7	Post Position	परसर्ग	অনুসর্গ	সোদোব উন মহরথি	پوّت جاے	पोत जाय	अंत्यस्थान
8	Conjunction	योजक	সংযোজক	দাজাব মহরথি	واتّوَن	राटवन	उभयान्वयी अव्यय
	Co-ordinator	समन्वयक	সমন্বয়ক	লোগো মহর	واتّت	वाटत/ वाटथ	NA
	Subordinator	अधीनस्थ	NA	লেডাइ লোগো মহর	تحتوُن	तहतून	NA
	Quotative	उक्ति-वाचक	NA	মুंখ'থি	دَپَن نِشانہٕ	दपन निशान	उद्गारवाचक
9	Particles	अव्यय	আনুষংগিক অব্যয়	মহরথি	ٹوٹّہٕ وَنِتى	टोट वनत्य	अव्यय/ नपिात
	Default	व्यतिक्रम		গোরোন্থি	ڈِفالٹ	डिफाल्ट	सामान्य
	Classifier	वर्गीकारक	নির্দিষ্টতাবাচক সর্গ	থি দিন্থিগ্রা দাজাবদা	وَرگِہا	वरगहा	NA
	Interjection	विस्मयादिबोधक	বিস্ময়বোধক	সোমোনাংনায় দিন্থিগ্রা	ژ ھتّت	छटत/ छटथ	वसि्मयवाचक

	Negation	नकारात्मक	নঞার্থক	নङি দিন্থিগ্রা	نَہ کارٕی	नकांरय	नषिधात्मक
	Intensifier	तीव्रक		গুন দিন্থিগ্রা	شدت ہار	शदत हाव	तीव्रतावाचक
10	Quantifiers	संख्यावाची	পৰিমাণবাচক	বিবাং দিন্থিগ্রা	گرێند	ग्रेन्द	संख्यावाचक
	General	सामान्य	সাধাৰণ	सरासनस्रा	عمومی	अमूमी	सामन्य
	Cardinals	गणनासूचक	সংখ্যাবাচক	গুৱৈ বিসান	آنکونہٕ گرێند	ओकँवन ग्रेन्द	गणनावाचक
	Ordinals	क्रमसूचक	ক্রমবাচক সংখ্যাবাচক শব্দ	फारि बिसान	وٗنی گرێند	वेन्य ग्रेन्द	क्रमवाचक
11	Residuals	अवशेष	NA	आद्रा	باقیاٗتی	बाक़यांती	शेष
	Foreign word	विदेशी शब्द	বিদেশী শব্দ	गुबुन हादरारि सोदोब	غاٗر مُلکی لَفظ	गोर मुल्की लफ़ुज़	वदिशी शब्द
	Symbol	प्रतीक	প্রতীক	नेर्सोन	علامَت	अलामत	चन्हि
	Unknown	अज्ञात	অজ্ঞাত	मिथियि	اَزون	अज़ोन	अज्ञात
	Punctuation	विरामादि-चिह्न	যতি চিন	थाद' सिन खान्थि	لَہجوَن	लहजिवन	वरिामचन्हि
	Echowords	प्रतिध्वनि-शब्द	ধ্বন্যাত্মক শব্দ	रिंखां सोदोब	پوٗت دُنی لفظ	पोत देन्य लफ़ज़	नादानुकारी/ अभ्यस्त

Languages: Telugu, Malayalam, Tamil, Konkani

S.No.	English	Hindi	Telugu	Malayalam	Tamil	Konkani
1	Noun	*संज्ञा*	సంఞ	നാമം	பயெர்	नाम
	common	जातिवाचक	జతవచకం	സാമാന്യ നാമം	பொதுப் பயெர்	जातवाचक नाम
	Proper	*व्यक्तिवाचक*	వయకతవచకం	സംജ്ഞാ നാമം	சிறப்புப் பயெர்	व्यक्तीवाचक नाम
	Verbal	क्रियामूलक / कृदंत	కరయమూలకం	NA	தொழில் பயெர்	क्रियामूळक नाम
	Nloc	देश-काल सापेक्ष	దశ-కల సపకషకం	ആധാരിക നാമം	இடப் பயெர்	थळसापेक्ष-काळ- नाम
2	Pronoun	*सर्वनाम*	సరవనమం	സർവ്വനാമം	பதிலீடுப் பயெர்	सर्वनाम
	Personal	*व्यक्तिवाचक*	వయకతవచకం	പുരുഷ സർവ്വനാമം	மஒவிடப்பயெ	पुरूश सर्वनाम
	Reflexive	निजवाचक	ఆతమరథకం	നിചവാചി സർവ്വനാമം	தற்சுட்டுப் பதிலீடுப் பயெர்	आत्मवाचक सर्वनाम
	Reciprocal	पारस्परिक	పరసపరకం	സംബന്ധവാചി സർവ്വനാമം	பரஸ்பர பதிலீடுப் பயெர்	संबंदी सर्वनाम
	Relative	संबंध- वाचक	సంబంధ-వచకం	പാരസ്പിക സർവ്വനാമം	இணைப்பு பதிலீடுப் பயெர்	एकमेकी सर्वनाम
	Wh-words	प्रश्नवाचक	పశర నవచకం	ചോദ്യവാചി സർവ്വനാമം	வினாச் சொல்	प्रस्नार्थी सर्वनाम
	Indefinite	अनिश्चयवाचक		NA	சுட்டு	अनिश्चित सर्वनाम
3	Demonstrative	निश्चयवाचक/ संकेतवाचक	నరదశకవచకం	നിർദേശകം	நேர்ச்சுட்டு	दर्शक
	Deictic	निर्देशी	నరదషట	പ്രത്യക്ഷ സൂചകം	சுட்டு பதிலீடுப் பயெர்	दर्शक उतर

	Relative	संबंधवाचक	సంబంధ-వచకం	സംബന്ധവാചി നിർദേശകം	வினாச் சொல்	संबंदी दर्शक
	Wh-words	प्रश्नवाचक	ప్రశ్న నవచకం	ചോദ്യവാചി നിർദേശകം	வினை	प्रस्नार्थी दर्शक
	Indefinite	अनिश्चयवाचक	NA	NA	தூணை வினை	अनिश्चित सर्वनाम
4	Verb	*क्रिया*	కరయ	ക്രിയ	மூதன்மை வினை	क्रियापद
	Auxiliary Verb	सहायक क्रिया	సహాయక కరయ	സഹായക ക്രിയ	மூற்றூ வினை	पालवी क्रियापद Auxiliary Finite (पूर्ण पालवी क्रियापद) Auxiliary Non Finite (अपूर्ण पालवी क्रियापद)
	Main Verb	मुख्य क्रिया	ముఖయ కరయ	പ്രധാന ക്രിയ	கூறை எச்சம்	मुखेल क्रियापद
	Finite	परिमित	సమపక	പൂർണ്ണ ക്രിയ	வினைப் பயெர்	निश्चीत क्रियापद
	Infinitive	क्रियार्थक संज्ञा	తుముననరథకం	ക്രിയാരൂപം	வினை எச்சம்	सादारण रूप
	Gerund	क्रियावाचक	కరయవచకం	NA	பயெரடை	क्रियावाचक नाम
	Non-Finite	गैर-परिमित	అసమపక	അപൂർണ്ണ ക്രിയ	வினையடை	अनिश्चीत क्रियापद
	Participle Noun	कृदंत परक नाम	NA	NA	பின்னூரூபூ	NA
5	Adjective	*विशेषण*	వశషణం	നാമ വിശേഷണം	இணைப்பூச் சொல்	विशेशण
6	Adverb	*क्रिया-विशेषण*	కరయవశషణం	ക്രിയാ വിശേഷണം	இணை இணைப்பூச் சொல்	क्रियाविशेशण
7	Post Position	परसर्ग	పరసరగ	അനുപ്രയോഗം	சார்பூ இணைப்பூச் சொல்	संबंदी अव्यय
8	Conjunction	योजक	సముచఛయం	സമുച്ചയം	நிரப்பூ இடைச்சொல்	जोड अव्यय
	Co-ordinator	समन्वयक	సమనధకరణం	ഏകോപിത സമുച്ചയം	இடைச்சொல்	समानाधीकरण जोड अव्यय
	Subordinator	अधीनस्थ	వయధకరణం	ആശ്രയസൂചക സമുച്ചയം	மூன்னிரூப்பூ	आश्रीत जोड अव्यय
	Quotative	उक्ति-वाचक	అనుకరకం	ഉദ്ധാരണവാചി സമുച്ചയം	இனப்பிரிப்பூ ஒட்டூ	अवतरणअर्थी- उतर
9	Particles	अव्यय	అవయయం	നിപാദം	வியப்பிடைச் சொல்	अव्यय
	Default	व्यतिक्रम	వయతకరమం	സാമാന്യം	எதிர்மறை	सरभरस अव्यय
	Classifier	वर्गीकारक	వరగకరకం	വർഗ്ഗകം	மிகூவிப்பான்	वर्गक अव्यय
	Interjection	विस्मयादिबोधक	వసమయదబోధకం	വ്യാക്ഷേപകം	அளவையடை	उमाळी अव्यय
	Negation	नकारात्मक	నకరతమకం	നിഷേദം	பொதூ	न्हयकारी अव्यय
	Intensifier	तीव्रक	అతశయరథకం	തീവ്ര നിപാദം	எண்ணூப் பயெர்	तीव्रकारी अव्यय
10	Quantifiers	संख्यावाची	సంఖయవచకం	സംഖ്യാവാചി	எண்ணூ மூறைப் பயெர்	संख्यादर्शक
	General	सामान्य	సమనయం	പൊതുസംഖ്യാ വാചി	எஞ்சியவை	सामान्य

	Cardinals	गणनासूचक	గణనసూచకం	അടിസ്ഥാന സംഖ്യാവാചി	அயல் சொல்	संख्यावाचक
	Ordinals	क्रमसूचक	కరమసూచకం	കർമ്മവാചി	கூறியீடு	क्रमवाचक
11	Residuals	अवशेष	అవశషం	അവശിഷ്ടപദം	தரியாதது	हेर
	Foreign word	विदेशी शब्द	వదశ శబదం	അന്യഭാഷാപദം	நிறுத்தற்கூறியீடு	विदेशी
	Symbol	प्रतीक	సంకతం	ചിഹ്നം	இரட்டைக்கிளவி	कुरू
	Unknown	अज्ञात	అజఞత	ഇതരപദം	NA	अनवळखी
	Punctuation	विरामादि-चिह्न	వరమం	വിരാമ ചിഹ്നം	NA	विरामकूरू
	Echo-words	प्रतिध्वनि-शब्द	పరతధవన-శబంద	മാറ്റൊലിവാക്ക്	NA	पडसादी उतरां

Annexure II

Pos schema ()
{
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<file Desc>
<titleStmt>
<title>POS tag in multilingual language</title>
<script>..... </script>
<language>multilingual</language>
<label language>……..</label language>
<type>multimodal</type>

[Languages taken: Hindi, Bodo, Malyalam, Kashmiri, Assamese, Konkani, Gujarati]

----------Noun Block----------

<xs:element name="cat" POS cat="noun" hin-cat="*संज्ञा*" brx-cat="मुंमा" mal-cat="നാമം" kas-cat="ناوُت " asm-cat="বিশেষ্য" kok-cat="नाम" guj-cat"સંજ્ઞા" tag="N">

<xs:attribute name="type" subcat="common" hin-cat**="**जातिवाचक**"** brx-cat="फोलेर दिन्थिग्रा" mal-cat="സാമാന്യ നാമം" kas-cat="**عام** " asm-cat="জাতিবাচক" kok-cat="जातवाचक नाम" guj-cat"જાતિવાચક" tag="NN">

<xs:attribute name="type" subcat ="Proper" hin-cat="*व्यक्तिवाचक*" brx-cat="मुं दिन्थिग्रा" mal-cat="സംജ്ഞാ നാമം" kas-cat="خاص" asm-cat="ব্যক্তিবাচক" kok-cat="व्यक्तीवाचक नाम" guj-cat"વ્યક્તિવાચક" tag="NNP">

<xs:attribute name="type" subcat ="Verbal" hin-cat="क्रियामूलक" brx-cat="हाबा दिन्थिग्रा" kas-cat="کرِیاوِتأوی " asm-cat="ক্রিয়াবাচক" kok-cat="क्रियामूळक नाम" guj-cat"ક્રિયાવાચક" tag="NNV">

<xs:attribute name="type" subcat ="Nloc" hin-cat="देश-काल सापेक्ष" brx-cat="थावनि दिन्थिग्रा मुंमा" mal-cat="ആധാരിക നാമം" kas-cat="ناوتہٕ جایہٕ باو " asm-cat="স্থানবাচক" kok-cat="थळसापेक्ष-काळ- नाम" guj-cat"સ્થાનવાચક" tag="NST">

----------Pronoun Block----------

<xs:element name="cat" POS cat="Pronoun" hin-cat="*सर्वनाम*" brx-cat="मुंराइ" mal-cat="സർവ്വനാമം" kas-cat="پَرناوُت " asm-cat="সর্বনাম" kok-cat="सर्वनाम" guj-cat"સર્વનામ" tag="PR">

<xs:attribute name="type" subcat ="Personal" hin-cat="*व्यक्तिवाचक*" brx-cat="संबुं दिन्थिग्रा" mal-cat="പുരുഷ സർവ്വനാമം" kas-cat="شخصیأتی " asm-cat="ব্যক্তিবাচক" kok-cat="पुरूश सर्वनाम" guj-cat"પુરૂષવાચક" tag="PRP">

<xs:attribute name="type" subcat ="Reflexive" hin-cat="निजवाचक" brx-cat="गाव दिन्थिग्रा" mal-cat="നിചവാചി സർവ്വനാമം" kas-cat="ماکوسی " asm-cat="আত্মবাচক" kok-cat="आत्मवाचक सर्वनाम" guj-cat"પ્રતિબિંબિત" tag="PRF">

<xs:attribute name="type" subcat ="Reciprocal" hin-cat="पारस्परिक" brx-cat="गावजों गाव सोमोन्दो" mal-cat="സംബന്ധവാചി സർവ്വനാമം" kas-cat="بابمی " asm-cat="পাৱস্পৱিক" kok-cat="संबंदी सर्वनाम" guj-cat"પરસ્પરવાચી" tag="PRC">

<xs:attribute name="type" subcat ="Relative" hin-cat="सम्बन्ध वाचक" brx-cat="सोमोन्दो दिन्थिग्रा" mal-cat="പാരസ്പിക സർവ്വനാമം" kas-cat="رأبتأوی " asm-cat="সম্বন্ধবাচক" kok-cat="एकमेकी सर्वनाम" guj-cat"સાપેક્ષ" tag="PRL">

<xs:attribute name="type" subcat ="Wh-words" hin-cat="प्रश्नवाचक" brx-cat="सौंथि दिन्थिग्रा" mal-cat="ചോദ്യവാചി സർവ്വനാമം" kas-cat="ک لفظ " asm-cat="প্রশ্নবোধক সর্বনাম" kok-cat="प्रस्नार्थी सर्वनाम" guj-cat"પ્રશ્નાર્થવાચક" tag="PRQ">

----------Demonstrative Block----------

<xs:element name="cat" POS cat="Demonstrative" hin-cat="निष्चयवाचक**"** brx-cat="थावनि दिन्थिग्रा" mal-cat="നിർദേശകം" kas-cat="باوَن پَرناوِتی " asm-cat="নির্দেশবোধক" kok-cat="दर्शक" guj-cat"દર્શકો" tag="DM">

<xs:attribute name="type" subcat =" Deictic" hin-cat="" brx-cat="थि दिन्थिग्रा" mal-cat="പ്രത്യക്ഷ സൂചകം" kas-cat="وأنیاُوی " asm-cat="প্রত্যক্ষ নির্দেশক" kok-cat="" guj-cat"ઉલ્લેખદર્શક" tag="DMD">

<xs:attribute name="type" subcat ="Relative" hin-cat="सम्बन्ध वाचक" brx-cat="सोमोन्दो दिन्थिग्रा" mal-cat="സംബന്ധവാചി നിർദേശകം" kas-cat="رأبتاُوی " asm-cat="সম্বন্ধবাচক" kok-cat ="संबंदी दर्शक" guj-cat"સાપેક્ષ" tag="DMR">

<xs:attribute name="type" subcat ="Wh-words" hin-cat="प्रश्नवाचक" brx-cat="म सोंथि दिन्थिग्रा" mal-cat="ചോദ്യവാചി നിർദേശകം" kas-cat="ک لفظ " asm-cat="প্রশ্নবোধক অব্যয়" kok-cat="प्रस्नार्थी दर्शक" guj-cat"પ્રશ્નવાચી" tag="DMQ">

----------Verb Block----------

<xs:element name="cat" POS cat="Verb" hin-cat="*क्रिया*" brx-cat="थाइजा" mal-cat="ക്രിയ" kas-cat="کرٲوُت" asm-cat="ক্রিয়া" kok-cat="क्रियापद" guj-cat"આખ્યાત" tag="V">

<xs:attribute name="type" subcat ="Auxiliary Verb" hin-cat="सहायक क्रिया" brx-cat="लेडाइ थाइजा" mal-cat="സഹായക ക്രിയ" kas-cat="ڈکھہٕ کراوُت " asm-cat="সহায়কাৰী ক্রিয়া" kok-cat="पालवी क्रियापद" guj-cat"" tag="VAUX">

<xs:attribute name="type" subcat ="Main Verb" hin-cat="मुख्य क्रिया" brx-cat="गुबै थाइजा" mal-cat="പ്രധാന ക്രിയ" kas-cat="راے کراوُت " asm-cat="মুখ্য ক্রিয়া" kok-cat="मुखेल क्रियापद" guj-cat"મુખ્ય" tag="VM">

<xs:attribute name="subtype" subcat ="Finite" hin-cat="परिमित" brx-cat="जाफुंजा थाइजा" mal-cat="പൂർണ്ണ ക്രിയ" kas-cat="بِشرِ باو " asm-cat="সমাপিকা" kok-cat="निश्चीत क्रियापद" guj-cat"પૂર્ણ" tag="VF">

<xs:attribute name="subtype" subcat ="Infinitive" hin-cat="अनंत" brx-cat="जाफुङि थाइजा" mal-cat="ക്രിയാരൂപം" kas-cat="بِشرِ کھاو " asm-cat="অসমাপিকা" kok-cat="सादारण रूप" guj-cat"હેત્વર્થ" tag="VINF">

<xs:attribute name="subtype" subcat ="Gerund" hin-cat="क्रियावाचक" brx-cat="जाफुबाय थानाय दिन्थिग्रा" kas-cat="کرٲوتہٕ ناوُت " asm-cat="নিমিত্তার্থক সংজ্ঞা" kok-cat="क्रियावाचक नाम" guj-cat"વર્તમાનકૃદન્ત" tag="VNG">

<xs:attribute name="subtype" subcat ="Non-Finite" hin-cat="गैर परिमित" brx-cat="जाफुङि थाइजा" mal-cat="അപൂർണ്ണ ക്രിയ" kas-cat="نا بِشرِ باو " asm-cat="অসমাপিকা" kok-cat="अनिश्चीत क्रियापद" guj-cat"અપૂર્ણ" tag="VNF">

----------Adjective Block----------

<xs:element name="cat" POS cat="Adjective" hin-cat="*विशेक्षण*" brx-cat="थाइलालि" mal-cat="നാമ വിശേഷണം" kas-cat="باوُت" asm-cat="বিশেষণ" kok-cat="विशेशण" guj-cat"વિશેષણ" tag="JJ">

----------Adverb Block----------

<xs:element name="cat" POS cat="Adverb" hin-cat="*क्रिया विशेक्षण*" brx-cat="थाइजानि थाइलालि" mal-cat="ക്രിയാ വിശേഷണം" kas-cat="لَگہٕ بأش " asm-cat="ক্রিয়া বিশেষণ" kok-cat="क्रियाविशेशण" guj-cat"ક્રિયાવિશેષણ" tag="RB">

----------Post Position Block----------

<xs:element name="cat" POS cat="Post Position" hin-cat="परसर्ग" brx-cat="सोदोब उन महरथि" mal-cat="അനുപ്രയോഗം" kas-cat="پوّت جاے " asm-cat="অনুসর্গ" kok-cat="संबंदी अव्यय" guj-cat"અનુગો" tag="PSP">

----------Conjunction Block----------

<xs:element name="cat" POS cat="Conjunction" hin-cat="योजक" brx-cat="दाजाब महरथि" mal-cat="സമുച്ചയം" kas-cat= "واتُوَن" asm-cat="সংযোজক" kok-cat="जोड अव्यय" guj-cat"સંયોજકો" tag="CC">

<xs:attribute name="type" subcat ="Co-ordinator" hin-cat="समन्वयक" brx-cat="लोगो महर" mal-cat="ഏകോപിത സമുച്ചയം" kas-cat="واتُت " asm-cat="সমন্বয়ক" kok-cat="समानाधीकरण जोड अव्यय" guj-cat"સહક્રિયાદર્શક" tag="CCD">

<xs:attribute name="type" subcat ="Subordinator" hin-cat="" brx-cat="लेडाइ लोगो महर" mal-cat="ആശ്രയസൂചക സമുച്ചയം" kas-cat="تحتوُن " asm-cat="" kok-cat="आश्रीत जोड अव्यय" guj-cat"ગૌણક્રિયાદર્શક" tag="CCS">

<xs:attribute name="subtype" subcat ="Quotative" hin-cat="उक्ति-वाचक" mal-cat="ഉദ്ധാരണവാചി സമുച്ചയം" brx-cat="मुंख'थि" kas-cat= "دَپَن نِشانہٕ " asm-cat="" kok-cat="अवतरणअर्थी- उतर" guj-cat"" tag="UT">

----------Particles Block----------

<xs:element name="cat" POS cat="Particles" hin-cat="अव्यय" brx-cat="महरथि" mal-cat="നിപാദം" kas-cat="ٹوٹہٕ وَنٹی " asm-cat="আনুষংগিক অব্যয়" kok-cat="अव्यय" guj-cat"નિપાત" tag="RP">

<xs:attribute name="type" subcat ="Default" hin-cat="व्यतिक्रम" brx-cat="गोरोन्थि" mal-cat="സാമാന്യം" kas-cat="ڈِفالٹ" asm-cat="" kok-cat="सरभरस अव्यय" guj-cat"સ્વયંભૂ" tag="RPD">

<xs:attribute name="type" subcat ="Classifier" hin-cat="वर्गीकारक" brx-cat="थि दिन्थिग्रा दाजाबदा" mal-cat="വർഗ്ഗകം" kas-cat="وَرگٕہا " asm-cat="निर্দিষ্টতাবাচক সর্গ" kok-cat="वर्गक अव्यय" guj-cat"" tag="CL">

<xs:attribute name="type" subcat ="Interjection" hin-cat="विस्मयादिबोधक" brx-cat="सोमोनांनाय दिन्थिग्रा" mal-cat="വ്യാക്ഷേപകം" kas-cat="ژھٹُت " asm-cat="বিস্ময়বোধক" kok-cat="उमाळी अव्यय" guj-cat"" tag="INJ">

<xs:attribute name="type" subcat ="Negation" hin-cat="नकारात्मक" brx-cat="नडि दिन्थिग्रा" mal-cat="നിഷേധം" kas-cat="نَہ کأری " asm-cat="নঞর্থক" kok-cat="न्हयकारी अव्यय" guj-cat"નકારદર્શક" tag="NEG">

<xs:attribute name="type" subcat ="Intensifier" hin-cat="तीव्रक" brx-cat="गुन दिन्थिग्रा" mal-cat="തീവ്ര നിപാദം" kas-cat="شدت بار " asm-cat="" kok-cat="तीव्रकारी अव्यय" guj-cat"માત્રાસૂચક" tag="INTF">

----------Quantifiers Block----------

<xs:element name="cat" POS cat="Quantifiers" hin-cat="संख्यावाची" brx-cat="बिबां दिन्थिग्रा" mal-cat="സംഖ്യാവാചി i" kas-cat="گرِنٛد " asm-cat="পৰিমাণবাচক" kok-cat="संख्यादर्शक" guj-cat"પરિમાણસૂચકો" tag="QT">

<xs:attribute name="type" subcat ="General" hin-cat="सामान्य" brx-cat="सरासनसा" mal-cat="പൊതുസംഖ്യാവാചി" kas-cat="عمومی " asm-cat="সাধাৰণ" kok-cat="सामान्य" guj-cat"સામાન્ય" tag="QTF">

<xs:attribute name="type" subcat ="Cardinals" hin-cat="गणनासूचक" brx-cat="गुबै बिसान" mal-cat="അടിസ്ഥാന സംഖ്യാവാചി" kas-cat="آنکونہٕ گرِنٛد " asm-cat="সংখ্যাবাচক" kok-cat="संख्यावाचक" guj-cat"સંખ્યાવાચક" tag="QTC">

<xs:attribute name="type" subcat ="Ordinals" hin-cat="क्रमसूचक" brx-cat="फारि बिसान" mal-cat="കർമ്മവാചി" kas-cat="وُنی گرِنٛد " asm-cat="ক্রমবাচক সংখ্যাবাচক শব্দ" kok-cat="क्रमवाचक" guj-cat"ક્રમવાચક" tag="QTO">

----------Residuals Block----------

<xs:element name="cat" POS cat="Residuals" hin-cat="अवशेष" brx-cat="आद्रा" mal-cat="അവശിഷ്ടപദം" kas-cat="باقیأتی " asm-cat="" kok-cat="हेर" guj-cat"શેષ" tag="RD">

<xs:attribute name="type" subcat ="Foreign word" hin-cat="विदेशी शब्द" brx-cat="गुबुन हादरारि सोदोब" mal-cat="അന്യഭാഷാപദം" kas-cat="غأر مُلکی لَفظ " asm-cat="বিদেশী শব্দ" kok-cat="विदेशी" guj-cat"પરદેશી શબ્દો" tag="RDF">

<xs:attribute name="type" subcat ="Symbol" hin-cat="प्रतीक" brx-cat="नेर्सोन" mal-cat="ചിഹ്നം" kas-cat="علامَت " asm-cat="প্রতীক" ki="कुरू" guj-cat"સંકેત" tag="SYM">

<xs:attribute name="type" subcat ="Unknown" hin-cat="अज्ञात" brx-cat="मिथियि" mal-cat="ഇതരപദം" kas-cat="اَزون" asm-cat="অজ্ঞাত" kok-cat="अनवळखी" guj-cat"અજાણ્યા શબ્દો" tag="UNK">

<xs:attribute name="type" subcat ="Punctuation" hin-cat="विरामादि-चिह्न" brx-cat="थाद,,,,' सिन खान्थि" mal-cat="വിരാമ ചിഹ്നം" kas-cat="لَہجِوَن " asm-cat="যতি চিন" kok-cat="विरामकूरू" guj-cat"વિરામચિહ્નો" tag="PUNC">

<xs:attribute name="type" subcat ="Echowords" hin-cat="प्रतिध्वनि-शब्द" brx-cat="रिंखां सोदोब" mal-cat="മാറ്റൊലിവാക്ക്" kas-cat="پوٚت دُنی لفظ " asm-cat="ধ্বন্যাত্মক শব্দ" kok-cat="पडसादी उतरां" guj-cat"અનુરણનાત્મક" tag="ECH">

</xs:attribute>

</xs:element>

</xs:schema>

}

A Generic and Robust Algorithm for Paragraph Alignment and its Impact on Sentence Alignment in Parallel Corpora

Ankush Gupta and Kiran Pala

Language Technologies Research Centre
IIIT-Hyderabad, Hyderabad, India.
ankush.gupta@research.iiit.ac.in
kiran.pala@research.iiit.ac.in

Abstract

In this paper, we describe an accurate, robust and language-independent algorithm to align paragraphs with their translations in a parallel bilingual corpus. The paragraph alignment is tested on 998 anchors (combination of 7 books) of English-Hindi language pair of Gyan-Nidhi corpus and achieved a precision of 86.86% and a recall of 82.03%. We describe the improvement in performance and automation of text alignment tasks by integrating our paragraph alignment algorithm in existing sentence aligner framework. This experiment carried out with 471 sentences on paragraph aligned parallel corpus, achieved a precision of 94.67% and a recall of 90.44%. Using our algorithm results in a significant improvement of 16.03% in Precision and 23.99% in Recall of aligned sentences as compared to when unaligned paragraphs are given as input to the sentence aligner.

1. Introduction

Parallel corpora offer a rich source of additional information about language (Matsumoto et al., 2003). Aligned parallel corpora is not only used for tasks such as bilingual lexicography (Klavans and Tzoukermann, 1990; Warwick and Russell, 1990; Giguet and Luquet, 2005), building systems for statistical machine translation (Brown et al., 1993; Vogel and Tribble, 2002; Yamada and Knight, 2001; Philipp, 2005), computer-assisted revision of translation (Jutras, 2000) but also in other language processing applications such as multilingual information retrieval (Kwok, 2001) and word sense disambiguation (Lonsdale et al., 1994). Alignment is the first stage in extracting structural information and statistical parameters from bilingual corpora. Only after aligning parallel corpus, further analyses such as phrase and word alignment, bilingual terminology extraction can be performed.

Manual alignment of parallel corpus is a labour-intensive, time-consuming and expensive task. Aligning a parallel corpus at paragraph level means taking each paragraph of the source language and aligning it to an equivalent translation in the target language. The task is not trivial because many times a single paragraph in one language is translated as two or more paragraphs in other language or two or more paragraphs in one language are aligned to two or more paragraphs in other language.

The algorithm proposed in this paper automatize the existing sentence aligner for English and Hindi language pairs (Chaudary et al., 2008) and improves its performance by upto 16.03%(Precision) and 23.99%(Recall). The results reported for English-Hindi sentence alignment in Chaudary et al. (2008) are by using manually aligned paragraphs. The goal of our research is to automate this task without a drop in the accuracy of sentence alignment.

This algorithm is motivated by the desire to develop for the research community a robust and language-independent paragraph alignment system which uses lexical resources easily available for most language pairs, thereby increasing its applicability. Building on this, we can do alignment at the sentence and word level with much higher accuracy.

2. Motivation

Not much work has been done on paragraph alignment, specifically on a diverse language pair like English-Hindi. Gale and Church (1991) use a two step process to align sentences. First paragraphs are aligned, and then sentences within a paragraph are aligned. In the corpus they have used, the boundaries between the paragraphs are usually clearly marked, which is not the case with our dataset. They found a threefold degradation in performance of sentence alignment when paragraph boundaries were removed. Hence, paragraph alignment is an important step and the difficulty of the problem depends on the language pair and the dataset.

Several algorithms for sentence alignment have been proposed, which can be broadly classified into three groups: (a) **Length-based** (b) **Lexicon-based**, and (c) **Hybrid Algorithms**. We explored whether the existing sentence alignment techniques can be used to align paragraphs.

(a) **Length-based algorithms** align sentences according to their length. Brown et al. (1991) uses word count as the sentence length and assumes prior alignment of paragraphs, whereas Gale and Church (1991) uses character to measure length and require corpus-dependent anchor points. These two works on sentence alignment show that length information alone is sufficient to produce surprisingly good results for aligning bilingual texts written in two closely related languages such as French-English and English-German. But it is quite a different case when we consider bilingual text from diverse language families such as English-Hindi. As stated in Singh and Husain (2005) "*Hindi is distant from English in terms of morphology. The vibhaktis of Hindi can adversely affect the performance of sentence length (especially word count) as well as word correspondence based algorithms.*" English is a fixed

English Paragraph	Hindi Paragraph
That very night, when the Brahmin returned, the mouse came out of its hole, stood up on its tail, joined its tiny paws and, with tears in its beady, black eyes, cried: 'Oh Good Master!, You have blessed me with the power of speech. Please listen now to my tale of sorrow!' 'Sorrow?' exlaimed the Brahmin in utter surprise, for he expected the mouse would have been delighted to talk as humans do. 'What sorrow?' the Brahmin asked gently, 'could a little mouse possibly have?' 'Dear Father!' cried the mouse. 'I came to you as a starving mouse, and you have starved yourself to feed me! But now that I am a fat and healthy mouse, when the cats catch sight of me, they tease me and chase me, and long to eat me, for they know that I will make a juicy meal. I fear, oh Father, that one day, they will catch me and kill me! I beg you, Father, make me a cat, so I can live without fear for the rest of my life'. The kind-hearted Brahmin felt sorry for the little mouse. He sprinkled a few drops of holy water on its head and lo and behold! the little mouse was changed into a beautiful cat!	उसी रात ब्राह्मण के लौटते ही चूहा बिल से निकल कर अपनी पूंछ के बल खडा हो गया। फिर उसने अपने छोटे पंजों को जोडकर चमकीली काली आंखों में आंसू लिए प्रार्थना की, ' हे भगवन्, आपने मुझे बोलने की शक्ति दी है। अब मेरी व्यथा की कथा सुनने की कृपा करें।' 'व्यथा' शब्द मात्र ही ब्राह्मण को चौंकाने वाला था। उसके अनुसार तो मनुष्यों की तरह बोलकर उस चूहे को अति प्रसन्न होना चाहिए था। फिर भी उसने धीरे से पूछा, 'एक छोटे से चूहे को भला क्या दु:ख हो सकता है?' इस पर चूहे ने याचना की, 'हे स्वामी, मैं आपके पास एक भूखे चूहे की तरह आया। आपने खुद को भूखा रख मुझे खिलाया। अब मैं एक मोटा-तगडा चूहा बन गया हूं। बिल्लियां, मुझे देखते ही चिढाती हैं और खदेडती हैं। मैं उनके लिए एक स्वादिष्ट भोजन बन चुका हूं। मुझे डर है कि एक दिन वे मुझे पकडकर मार देंगी। अत: हे स्वामी, मेरी आपसे याचना है कि मुझे बिल्ली बना दीजिये, ताकि बाकी का जीवन मैं निडर होकर बिता सकूं।' यह सुनते ही दयालू ब्राह्मण दुखी हो गया। और चूहे के माथे पर उसने गंगाजल छिडक दिया। देखते ही देखते वह चूहा एक सुंदर बिल्ली बन गया।

Table 1: Many-to-Many (3-to-2) Paragraph Alignment

word order language while Hindi is a comparatively free word order language (Ananthakrishnan et al., 2007). For sentence length based alignment, this doesn't matter since they don't take the word order into account. However, Melamed (1996) algorithm is sensitive to word order. It states "*how it will fare with languages that are less closely related, which have even more word order variation. This is an open question*"

In addition, the corpus we have used does not contain the literal translation of the source language. The translators have translated the gist of the source language paragraph into the target language paragraph which sometimes results in a large amount of omissions in the translation. So the length ratio of the English and the Hindi paragraphs varies considerably making length based sentence alignment algorithms not apt for the paragraph alignment task. To verify this, we calculated the length ratio of manually aligned English and Hindi paragraphs and it varies from 0.375 to 10.0. Another weakness of the pure length-based strategy is its susceptibility to long stretches of passages with roughly similar lengths. According to Wu and Xia (1995) "*In such a situation, two slight perturbations may cause the entire stretch of passages between the perturbations to be misaligned. These perturbations can easily arise from a number of cases, including slight omissions or mismatches in the original parallel texts, a 1-for-2 translation pair preceding or following the stretch of passages*". The problem is made more difficult because a paragraph in one language may correspond to multiple paragraphs in the other; worse yet, sometimes several paragraphs content is distributed across multiple translated paragraphs. Table 1 shows three English paragraphs aligned to two Hindi paragraphs. To develop a robust paragraph alignment algorithm, matching the passages lexical content is required, rather than relying on pure length criteria.

(b) **Lexicon-based algorithms** (Xiaoyi, 2006; Li et al., 2010; Chen, 1993; Melamed, 1996; Melamed, 1997; Utsuro et al., 1994; Kay and Roscheisen, 1993; Warwick et al., 1989; Mayers et al., 1998; Haruno and Yamazaki, 1996) use lexical information from source and translation lexicons to determine the alignment and are usually more robust than length-based algorithms.

(c) **Hybrid algorithms** (Simard et al., 1993; Simard and Plamondon, 1998; Wu, 1994; Moore, 2002; Varga et al., 2005) combine length and lexical information to take advantage of both. According to Singh and Husain (2005) "*An algorithm based on cognates (Simard et al., 1993)*

is likely to work better for English-French or English-German than for English-Hindi, because there are fewer cognates for English-Hindi. It won't be without a basis to say that Hindi is more distant from English than is German. English and German belong to the Indo-Germanic branch whereas Hindi belongs to the Indo-Aryan branch."

With this motivation, we propose a generic and robust algorithm for aligning paragraphs and test its performance on a distinct language pair such as English-Hindi.

The rest of the paper is organized as follows: Section 3 discuss the tools and resources (3.1) used and various modules (3.2) in an integrated framework for paragraph and sentence alignment. Section 4 describes the algorithm for Paragraph Alignment. Section 5 shows the experimental results. In Section 6, we do an error analysis and highlight some of the advantages of our algorithm; and Section 7 is the conclusion.

3. Architecture

3.1. Tools and Resources

3.1.1. English Sentence Splitter

This program checks candidates to see if they are valid sentence boundaries. Its input is a text file, and its output is another text file where each text line corresponds to one sentence. It requires a honorifics file as an argument which must contain honorifics, not abbreviations. The program detects abbreviations using regular expressions. It was able to split 97.02% of the sentences correctly when tested on a dataset of 471 sentences.

3.1.2. English Porter Stemmer

The Porter Stemming algorithm (Porter, 1980) is a process for removing the commoner morphological and inflexional endings from words in English.

3.1.3. Bilingual Parallel Corpora

We have used GyanNidhi parallel corpus (Arora et al., 2003) for our experiments. GyanNidhi is the first attempt at digitizing a corpus which is parallel in multiple Indian Languages. For our experiments, the source language is English and the target language into which the text is translated is Hindi. For this experiment non-aligned English-Hindi parallel corpus is taken. The paragraphs are numbered according to book number, page number and paragraph number information. For example, the paragraph notation is : EN-1000-0006-3 [where EN stands for English, 1000 is the book number, 6 is the page number and 3 is the paragraph number]. Similar notation scheme is used for Hindi text.

3.1.4. Lexicon Preparation

English-Hindi shabdanjali dictionary[1] is used to prepare an enriched lexicon. It contains about 24,013 distinct English words with their corresponding Hindi translation(s). English (Miller, 1995) and Hindi Wordnet (Jha et al., 2001) are used to enhance the number of words in the lexicon of both the languages. The final lexicon contains 47,240 distinct English and 48,394 Hindi words. Some of the sample entries from the lexicon are shown in Table 2.

English Entry	Hindi Entry
allegation	आरोप/ इल्ज़ाम/ इल्जाम
allegedly	कथित रूप से
allocate	निर्धारित करना/ नियत करना/ निश्चित करना/ वितरण करना/ वितरित करना/ विभाजित करना/ तक़सीम करना/ हिस्से करना/ भाग करना/ आवंटन/ आबंटन/ शेयर करना/ संविभाजित करना ...
election	चुनाव/ इन्तख़ाब/ इंतख़ाब/ इन्तखाब/ इंतखाब/ चुनाई/ वरण/ चयन/ अधिवाचन/ निर्वाचन
fashion	फैशन (कार्य प्रणाली)/ अंदाज/ कार्य विधि/ कायदा/ रीति/ रीत/ तरीका/ विधि/ अंदाज़/ शैली/ तर्ज/ कायदा/ आचरण/ व्यवहार/ बर्ताव/ रंग-धंग/ बात-व्यवहार/ सलीका/ अचार/ चाल-चलन/ चाल/ सलीका/ तौर-तरीका/ आचार/ चाल-धाल/ ....
probably	संभवतः/ शायद/ सम्भव/ मुमकिन/ संभाव्य/ संभावित/ संभव/ सम्भाव्य/ सम्भावित/ ....

Table 2: Sample Entries from English-Hindi Lexicon

3.2. Modules

The architecture of the framework (our paragraph alignment algorithm integrated with existing sentence alignment algorithms) is explained in Figure 1.

- **Preprocessor Module**- The preprocessor takes raw data from GyanNidhi corpus as input and cleans the text by removing the unwanted characters and tags.

- **Seed Anchors Module**- Seed Anchors are the paragraphs which are aligned manually after a certain interval. In our experiments, the interval is set as 20 empirically. So, about 5% of the total paragraphs are aligned by hand. If the alignment algorithm makes an error, this modules makes sure that the error is not propagated to the later alignments. The paragraph alignment algorithm can work even without this module but with lesser efficiency depending on the dataset size and the quality of the translations.

- **Paragraph Aligner Module**- The paragraph aligner, takes the preprocessed data and seed anchors and aligns the paragraphs between each seed anchor. The functionality of this module is discussed in detail in Section 4.

[1] http://ltrc.iiit.ac.in

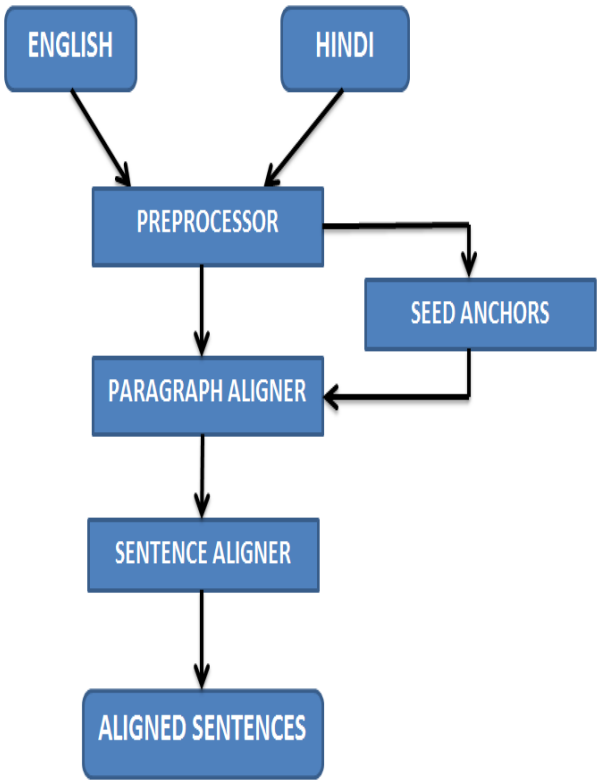


Figure 1: Architecture of Paragraph-Sentence Aligner framework

- **Sentence Aligner Module** - The aligned paragraphs are given as input to the existing sentence aligners. The output is the aligned sentences.

4. Algorithm

Given an English and Hindi Paragraph file and a list of few manually aligned anchors (seed anchors), the task is to automatically align the paragraphs between each seed anchor. First of all, English paragraphs are split into sentences using the sentence splitter and Hindi paragraphs are split using '।' and '?' as delimiters. Then, sentences are processed by replacing characters like {'} {,} {(} {.} {)} {;} {!} {?} with spaces. Four indexed lists are constructed by considering first (SA1) and second (SA2) seed anchor :

1. **First English List (FEL)** : List containing words present in first unaligned (next to SA1) English paragraph. Algorithm 2 describes the construction of FEL.
2. **Second English List (SEL)** : List containing words present in second unaligned (next to next to SA1) English paragraph.
3. **First Hindi List (FHL)** : List containing words present in first unaligned (next to SA1) Hindi paragraph. Construction of FHL is explained in Algorithm 3.
4. **Second Hindi List (SHL)** : List containing words present in second unaligned (next to next to SA1) Hindi paragraph.

Heuristics(H) (defined in Section 4.1.) are computed using these 4 indexed lists and the lexicon (created in Section 3.1.4.) and paragraphs are aligned using Algorithm 4.
The pseudo-code of entire Paragraph Alignment method is described in Algorithm 1.

Algorithm 1 Paragraph Alignment Algorithm

Input : English Paragraph file, Hindi Paragraph file, Seed Anchors, Stop word list for English (source language), English-Hindi lexicon
Output : Aligned English-Hindi Paragraphs
Algorithm :
– Split English and Hindi Paragraphs into sentences
– Replace characters {'} {,} {(} {.} {)} {;} {!} {?} with space
– Construct four indexed lists : FEL, SEL, FHL and SHL (Algorithm 2, 3)
– Compute Heuristics(H) (Section 4.1.)
– Align paragraphs (Algorithm 4)

Algorithm 2 Algorithm to Construct FEL

$P_1 : First\ unaligned\ English\ Paragraph$
$n_1 : number\ of\ sentences\ (P_1)$
for $i = 1$ to $n_1 - 2$ **do**
  for $j = i$ to $j = i + 2$ **do**
    for all $word_k$ such that $word_k \in sentence_j$ **do**
      if $word_k \notin stopword - list$ **then**
        if $word_k \in lexicon$ **then**
          $Add\ word_k\ to\ FEL_i$
        else
          $word_s = stemmer(word_k)$
          if $word_s \in lexicon$ **then**
            $Add\ word_s\ to\ FEL_i$
          end if
        end if
      end if
    end for
  end for
end for

Algorithm 3 Algorithm to Construct FHL

$P_1 : First\ unaligned\ Hindi\ Paragraph$
$n_1 : number\ of\ sentences\ (P_1)$
for $i = 1$ to $n_1 - 2$ **do**
  for $j = i$ to $j = i + 2$ **do**
    for all $word_k$ such that $word_k \in sentence_j$ **do**
      $Add\ word_k\ to\ FHL_i$
    end for
  end for
end for

The 0^{th} index of FEL contains the words (stopwords are removed) present in 1^{st}, 2^{nd} and 3^{rd} sentences of the English paragraph next to Seed Anchor1 (SA1) (word should be present in the lexicon), 1^{st} index of FEL contains the words of 2^{nd}, 3^{rd} and 4^{th} sentences and so on. Similar distribution is followed for SEL, FHL and SHL. While con-

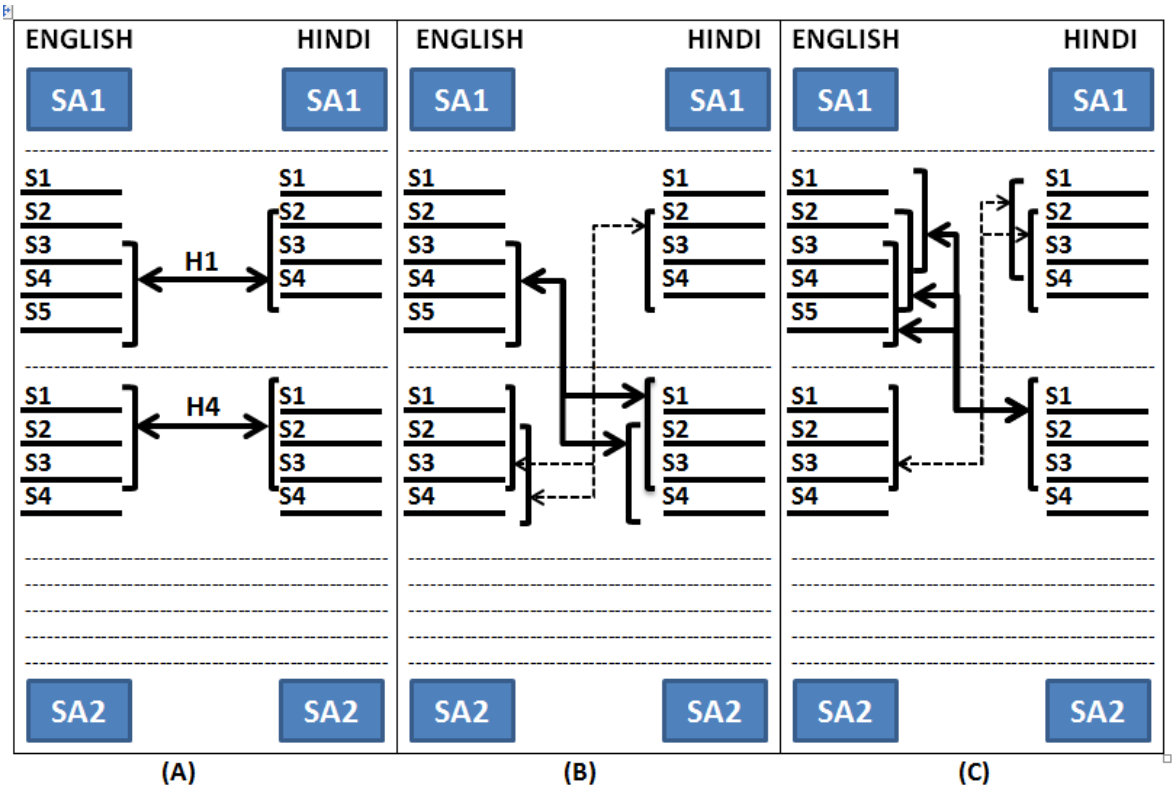


Figure 2: Heuristics: (A) explains heuristics H1 and H4; (B) explains H2 and H3(dotted) and (C) explains H5(dotted) and H6.

structing FHL and SHL, we avoid the computation of stem as it makes the algorithm very slow.

4.1. Heuristics (H)

Lists of English and Hindi words (FEL, SEL, FHL, SHL) and English-Hindi bilingual lexicon (Section 3.1.4.) are used to compute following six heuristics (Figure 2):

- Calculate the number of words present in last three sentences of first English unaligned paragraph which have their corresponding translation (using English-Hindi lexicon) in last three sentences of first Hindi unaligned paragraph. To do a normalization, divide it by the total number of words present in last three sentences of first English unaligned paragraph.

$$H1 = \frac{FEL_{last-index} \cap FHL_{last-index}}{length(FEL_{last-index})} \quad (1)$$

  We look at the translations of each word of FEL in the lexicon and check if any of the translation is present in FHL.

  This heuristic guides the algorithm when to stop expanding the current unaligned English and Hindi paragraphs.

  Many times a sentence in source language is translated as two or more sentences in target language or vice-versa. To handle this issue, we match sentences in groups of three instead of sentence-by-sentence.

- Words present in last three sentences of first English unaligned paragraph are matched with all pairs of three consecutive sentences of second Hindi unaligned paragraph. Divide it by the number of words present in last three sentences of first English unaligned paragraph and take the maximum value.

$$H2 = \forall_i \; max \; \frac{FEL_{last-index} \cap SHL_{i^{th}-index}}{length(FEL_{last-index})} \quad (2)$$

  The translation of last three sentences of English unaligned paragraph might be present anywhere in second Hindi unaligned paragraph. Hence, all pairs[2] of sentences are considered to calculate H2.

- All pairs of three consecutive sentences of second English unaligned paragraph are matched with last three sentences of first Hindi unaligned paragraph. Divide it by the number of words present in the corresponding sentences of second English unaligned paragraph and take the maximum value.

$$H3 = \forall_i \; max \; \frac{SEL_{i^{th}-index} \cap FHL_{last-index}}{length(SEL_{i^{th}-index})} \quad (3)$$

  This heuristic takes care of the cases when translation of a part of current unaligned Hindi paragraph is present in next unaligned English paragraph.

[2]Pairs consist of Sentences (1,2,3), (2,3,4), (3,4,5), ...

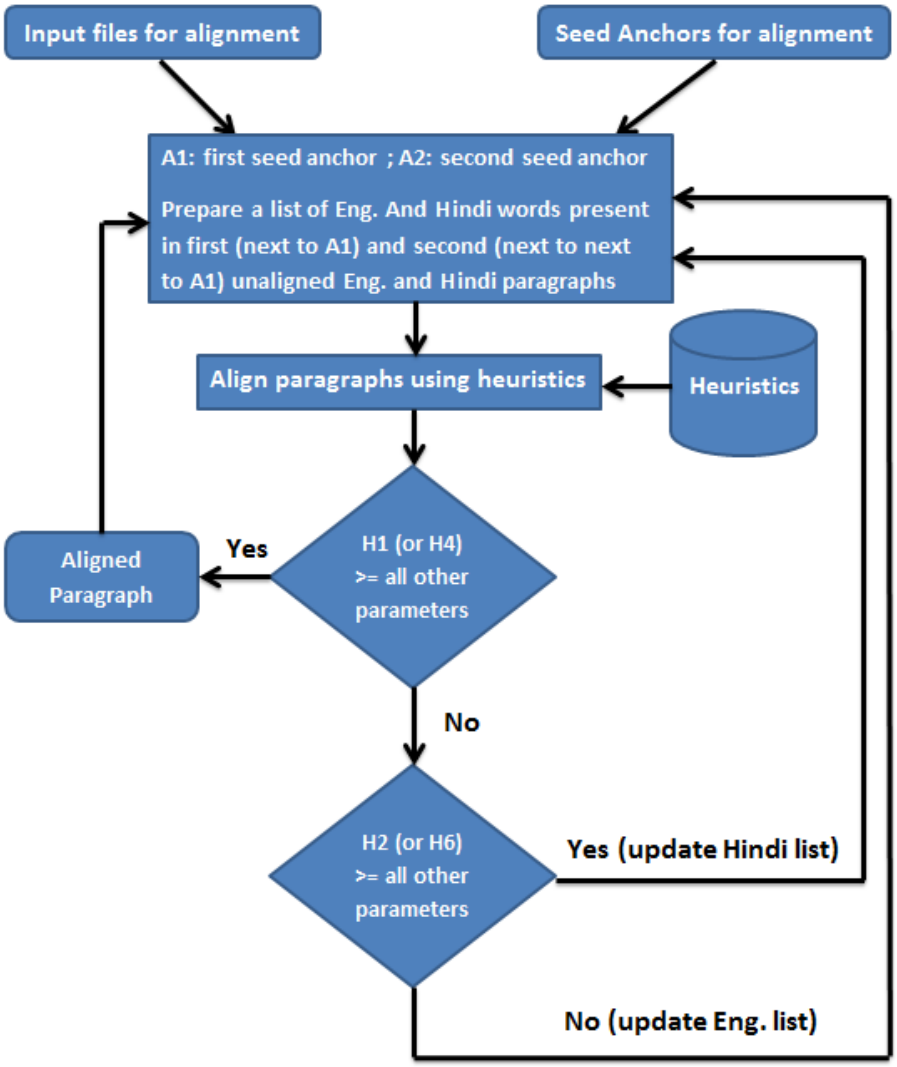


Figure 3: Paragraph Alignment Algorithm

- Calculate the number of matches between the words present in top three sentences of second English unaligned paragraph and the words present in top three sentences of second Hindi unaligned paragraph. Divide it by the number of words present in top three sentences of second English unaligned paragraph.

$$H4 = \frac{SEL_{0^{th}-index} \cap SHL_{0^{th}-index}}{length(SEL_{0^{th}-index})} \quad (4)$$

  Besides serving similar purpose as H1, this heuristic also handle issues of deletion or insertion in the text. Sometimes the translation of current unaligned English (or Hindi) paragraph might not be present in the corpus. In that case, to avoid propagating the error, we stop the expansion of current paragraphs at this stage.

- Words in top three sentences of second English unaligned paragraph are matched with all pairs of three consecutive sentences of first Hindi unaligned paragraph. Divide it by the number of words present in the top three sentences of second English unaligned paragraph and take the maximum value.

$$H5 = \forall_i \; max \; \frac{SEL_{0^{th}-index} \cap FHL_{i^{th}-index}}{length(SEL_{0^{th}-index})} \quad (5)$$

  This heuristic takes care of the cases when translation of a part of next unaligned English paragraph is present in current unaligned Hindi paragraph (Similar to H3).

- All pairs of three consecutive sentences of first English unaligned paragraph are matched with top three sentences of second Hindi unaligned paragraph. Divide it by the number of words present in corresponding sentences of first English unaligned paragraph and take the maximum value.

$$H6 = \forall_i \; max \; \frac{FEL_{i^{th}-index} \cap SHL_{0^{th}-index}}{length(FEL_{i^{th}-index})} \quad (6)$$

This heuristic takes care of the cases when translation of a part of next unaligned Hindi paragraph is present in current unaligned English paragraph (Similar to H2).

Algorithm 4 Aligning Paragraphs using Heuristics

if $H1(or H4) \geq (H2, H3, H4, H5, H6)$ **then**
Consider the paragraphs as aligned and upgrade them to seed anchors (SA1).
else if $H2(or H6) \geq (H1, H3, H4, H5, H6)$ **then**
Expand the first Hindi unaligned paragraph and update FHL and SHL
else if $H3(or H5) \geq (H1, H2, H4, H5, H6)$ **then**
Expand the first English unaligned paragraph and update FEL and SEL
end if

5. Results

The paragraph alignment technique is tested on a data set of 7 different books from GyanNidhi corpus, including diverse texts. A total of 998 English anchors are used for Testing and 48 [4.8%] are used as seed anchors. The ouput of the paragraph alignment technique is evaluated against manually aligned output. We achieved a precision of 86.86% and a recall of 82.03%.

To test the effectiveness of the algorithm, we integrated it into an existing sentence aligner framework for English-Hindi (Chaudary et al., 2008). Three evaluation measures are used :

$$\textbf{Accuracy} = \frac{Number\ of\ aligned\ Sentences}{Total\ number\ of\ Sentences} \quad (7)$$

$$\textbf{Precision} = \frac{Number\ of\ correctly\ aligned\ Sentences}{Total\ number\ of\ aligned\ Sentences} \quad (8)$$

$$\textbf{Recall} = \frac{Number\ of\ correctly\ aligned\ Sentences}{Total\ number\ of\ Sentences\ in\ source} \quad (9)$$

Using paragraph alignment results in an improvement of **11.04%** in Accuracy, **16.03%** in Precision and **23.99%** in Recall. The results are shown in Table 3. [SA - Sentence Aligner, PA - Paragraph Aligner]

We also experimented using Gale and Church (Gale and Church, 1991) sentence alignment algorithm[3] which is a language-independent length-based algorithm. When no paragraph boundaries were given, only 3 sentences were correctly aligned. In Gale and Church (1991), first paragraphs are aligned and then sentences within paragraphs are aligned. When only manually aligned paragraphs (count=6) were given as paragraph boundaries, 39 sentences were correctly aligned. After running our paragraph alignment algorithm, correctly aligned sentences increased to 297 which is a significant improvement. Table 3 shows that lexicon-based algorithms work much better than length-based algorithms for English-Hindi.

Some of the paragraphs aligned by the paragraph alignment algorithm are shown in Table 4.

[3] www.cse.unt.edu/~rada/wa

6. Discussion / Error-Analysis

One of the potential advantages of the proposed paragraph alignment algorithm is that it corrects itself if it makes an error in alignment. For example: EN-1000-0010-5 | HI-1000-0010-5:HI-1000-0012-1 and EN-1000-0012-1 | HI-1000-0012-2 are the correct manually aligned anchors. The algorithm makes an error while aligning EN-1000-0010-5 | HI-1000-0010-5 but it corrects itself in the next alignment as EN-1000-0012-1 | HI-1000-0012-1:HI-1000-0012-2 to prevent the error from propagating further. If the correct alignment is 2-to-2, sometimes our algorithm aligns them as separate 1-to-1 alignments and vice-versa. So, we took a window of 2 while matching to see the deviation in the incorrect aligned paragraphs and got a recall of 98.9%, highlighting less deviation.

As Hindi is morphologically a very rich language, one word can have several correct ways of writing. Though many variations are already there in the lexicon but still sometimes the text contains a word which is not present in the lexicon. For example: Hindi text contains "iMjina" [इंजिन] (engine) while the lexicon contains "iMjana" [इंजन] (engine), so these two do not get matched. Sometimes two words in English have a single word as a translation in Hindi, eg: "necessities of life" is translated as "jIvanopayogI" [जीवनोपयोगी], "Yoga Maya" as "yogamAyA" [योगमाया], "cooking gallery" as "rasoIGara" [रसोईघर].

As we are considering the root form of only English word, some times words do not match because the lexicon has only Hindi translations in root form. So, "praWAoM" [प्रथाओं] is not in lexicon but the root form "praWA" [प्रथा] is present. The reason behind not calculating the root form of Hindi word is that it makes the algorithm very slow. So we did a preprocessing and stored the root forms of the Hindi words in a separate file before running the algorithm so that we do not have to calculate the root form each time we run the algorithm. There was a slight increase in precision from 86.86% to 87.6% and recall from 82.03% to 83.85%. We have tested our algorithm on a domain-independent dataset. If we add domain specific linguistic cues to the lexicon, the accuracy is expected to increase.

Another advantage of the algorithm is that in one pass, it creates one-to-one, one-to-many, many-to-one and many-to-many alignments. As we avoid the use of complex resources like chunker, pos tagger, parser and named entity recognizer which are difficult to get for most of the languages, the algorithm can be easily applied to other language pairs. Because we use minimal resources, the alignment computation is fast and therefore practical for application to large collections of text.

7. Conclusion

We have described an accurate, robust and language-independent algorithm for paragraph alignment which combines the use of simple heuristics and resources like bilingual lexicon and stemmer for source language. This unique approach gives high precision and recall even for distinct language pair like English and Hindi and shows a significant improvement in sentence alignment when integrated with existing sentence aligners. The algorithm is

SA Algorithm	Procedure	Sentences	Aligned	Correct	Accuracy	Precision	Recall
Chaudary et al. (2008)	Only SA	471	398	313	84.5	78.64	66.45
	First PA, then SA	471	450	426	**95.54**	**94.67**	**90.44**
Gale and Church (1991)	Only SA	471	471	39	100	8.28	8.28
	First PA, then SA	471	471	297	100	63.05	63.05

Table 3: Results of Sentence Alignment

English Paragraph	Hindi Paragraph
The object turned out to be a big meteorite. Uttama was delighted. He had never seen anything like it on sea or land before. Despite its journey in space and stay in water, it had retained its shape and colour.	यह एक बडे आकार का उल्का पिंड था। उत्तम बहुत खुश हुआ। उसने ऐसी कोई चीज कभी पहले नहीं देखी थी - न समुद्र में और न जमीन पर। अंतरिक्ष यात्रा और पानी में रहने पर भी इस चीज का रंग और आकार नहीं बदला था।
The stand-still alert ended. Uttama was ordered to surface. He immediately telephoned his friend, Professor Maruthi of the Stellar School in the Kavalur Observatory complex and informed him about the meteorite. Professor Maruthi was very excited. The meteorite was the largest he had ever heard of. Receiving permission to examine it Professor Maruthi began conducting tests on the cosmic relic.	ठहरे रहने की चेतावनी खत्म हो गयी थी। उत्तम ने ऊपर जाने का आदेश दिया। पहुंचते ही उसने अपने मित्र कावालूर बेधशाला क्षेत्र में स्थित तारामंडल स्कूल के प्रोफेसर मारुति को टेलीफोन किया और इस उल्का पिंड के बारे में उन्हें बताया। प्रोफेसर मारुति बहुत उत्साह में आ गये थे। अब तक उन्होंने जितने भी उल्का पिंडों के बारे में सुना था, यह उन सबसे बडा था। इसका परीक्षण करने की अनुमति मिलते ही प्रोफेसर मारुति ने अंतरिक्ष के इस अवशेष पर परीक्षण करना शुरु कर दिया।
As layer after layer of filmy material was removed, a clear pattern emerged, looking like 10101 which Professor Maruthi suggested was a binary code for 21. And 21 could stand for the 21 cm. radio frequency of hydrogen in space.	इस पर जमी बाहरी तहों को उतारने के बाद एक स्पष्ट आकृति सामने आयी जो 10101 जैसे दिख रही थी। प्रोफेसर ने बताया कि यह 21 का दिवचर प्रणाली का रुप है। और 21 का अर्थ अंतरिक्ष में हाइड्रोजन की 21 सेंटीमीटर रेडियों आवृत्ति है।
Just then, there was a call from the Medical Research Council. Dr. Danwantri, who headed the Biochemistry Department spoke, 'I understand that you are planning to send a message to outer space. I would like to make a suggestion.' Dr. Danwantri explained that he was keen to get new information on the structure and working of the human brain. He wondered if it might be possible to encode questions on this which might elicit an answer from intelligent beings who were well wishers far out in the distant depths of space.	तभी आयुर्विजान अनुसंधान परिषद की ओर से एक संदेश मिला। जीव रसायन विभाग के अध्यक्ष डाक्टर धनवंतरी कह रहे थे। 'मेरे ख्याल से आप बाह्य अंतरिक्ष में संदेश भेजने की तैयारी कर रहे हैं। मेरा एक सुझाव है।' डाक्टर धनवंतरी ने समझाया कि वे मानव-मस्तिष्क की संरचना और कार्यविधि के बारे में नयी जानकारी पाना चाहते है। काश यह संभव होता कि इस पर संकेतिक प्रश्न का उत्तर अंतरिक्ष की गहराईयों में बैठे उन समझदार प्राणियों से मिल पाता जो हमारे शुभचिंतक है।

Table 4: Output of Paragraph Alignment Algorithm

parallelizable as paragraphs between seed anchors can be aligned parallely. The paragraph aligned parallel corpora will facilitate to improve the sentence alignment as well as the development of word alignment tools and it can be further used to enhance the statistical MT systems.

8. Acknowledgements

We would like to thank Dr. Sriram Venkatapathy, Dr. Dipti Misra Sharma and Anusaaraka Lab from LTRC, IIIT Hyderabad for helpful discussions and pointers during the course of this work.

9. References

R. Ananthakrishnan, P. Bhattacharya, M. Sasikumar, and R. M. Shah. 2007. Some issues in automatic evaluation of english-hindi mt: more bleus for bleu. In *Proceedings of 5th International Conference on Natural Language Processing(ICON-07)*, Hyderabad,India.

K. K. Arora, S. Arora, V. Gugnani, V. N. Shukla, and S. S. Agarwal. 2003. Gyannidhi: A parallel corpus for indian languages including nepali. In *Proceedings of Information Technology: Challenges and Prospects (ITPC-2003)*, Kathmandu, Nepal, May.

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning sentences in parallel corpora. In *Proceedings of the 29th Annual Meeting of the ACL (1991)*, pages 169–176.

Peter F. Brown, V. Della Pietra, S. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. In *Computational Linguistics 19,2*, pages 263–311.

S. Chaudary, K. Pala, L. Kodavali, and K. Singhal. 2008. Enhancing effectiveness of sentence alignment in parallel corpora : Using mt & heuristics. In *Proceedings of 6th International Conference on Natural Language Processing(ICON-08)*.

Stanley F. Chen. 1993. Aligning sentences in bilingual corpora using lexical information. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 9–16, Columbia, Ohio, USA, June. Association for Computational Linguistics.

William A. Gale and Keneth W. Church. 1991. A program for aligning sentences in bilingual corpora. In *Proceedings of the 29th Annual Meeting of the ACL*, pages 177–184.

Emmanuel Giguet and Pierre-Sylvain Luquet. 2005. Multilingual lexical database generation from parallel texts with endogenous resources. In *PAPILLON-2005 Workshop on Multilingual Lexical Databases.*, Chiang Rai, Thailand, December.

M. Haruno and T. Yamazaki. 1996. High-performance bilingual text alignment using statistical and dictionary information. In *Proceedings of the 34th conference of the Association for Computational Linguistics*, pages 131–138, Santa Cruz, California.

S. Jha, D. Narayan, P. Pande, and P. Bhattacharyya. 2001. A wordnet for hindi. In *International Workshop on Lexical Resources in Natural Language Processing*, Hyderabad, India, January.

J-M. Jutras. 2000. An automatic reviser: The transcheck system. In *Proceedings of Applied Natural Language Processing*, pages 127–134.

M. Kay and M. Roscheisen. 1993. Text translation alignment. In *Computational Linguistics, 19(1)*, pages 75–102.

J. Klavans and E. Tzoukermann. 1990. The bicord system. In *COLING-90*, pages 174–179, Helsinki, Finland.

K.L. Kwok. 2001. Ntcir-2 chinese, cross-language retrieval experiments using pircs. In *Proceedings of the Second NTCIR Workshop Meeting*, pages 14–20, National Institute of Informatics, Japan.

Peng Li, Maosong Sun, and Ping Xue. 2010. Fast-champollion: A fast and robust sentence alignment algorithm. In *23rd International Conference on Computational Linguistics: Posters*, pages 710–718, Beijing. Association for Computational Linguistics.

D. W. Lonsdale, T. Mitamura, and E. Nyberg, 1994. *Acquisition of large lexicons for practical knowledge-based MT.*, pages 251–283. Vol. 9(3-4) edition.

Y. Matsumoto, H. Ishimoto, and T. Utsuro. 2003. Structural matching of parallel texts. In *Proceedings of the 31st Annual Meeting of the ACL*, pages 23–30.

A. Mayers, R. Grishman, and M. Kosaka. 1998. A multilingual procedure for dictionary-based sentence alignment. In *Proceedings of the Third Conference of the Association for Machine Translation in the Americas on Machine Translation and the Information Soup*.

I. Dan Melamed. 1996. A geometric approach to mapping bitext correspondence. In *Proceedings of the First Conference on Empirical Methods in Natural Language Processing*, pages 1–12.

I. Dan Melamed. 1997. A portable algorithm for mapping bitext correspondence. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 305–312, Madrid, Spain.

G. Miller, 1995. *WordNet:A Lexical Database for English.*, pages 39–41. Vol. 38, no. 11 edition.

Robert C. Moore. 2002. Fast and accurate sentence alignment of bilingual corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas on Machine Translation: From Research to Real Users*, pages 135–144, London, UK. Springer-Verlag.

K. Philipp. 2005. Europarl: A parallel corpus for statistical machine translation. In *Machine Translation Summit*, Phuket, Thailand.

M.F. Porter. 1980. An algorithm for suffix stripping. In *Program, 14*, pages 130–137.

Michel Simard and P. Plamondon. 1998. Bilingual sentence alignment: Balancing robustness and accuracy. In *Machine Translation 13(1)*, pages 59–80.

Michel Simard, George F. Foster, and Pierre Isabelle. 1993. Using cognates to align sentences in bilingual corpora. In *Proceedings of the 1993 Conference of the Centre for Advanced Studies on Collaborative Research*, pages 1071–1082. IBM Press.

Anil Kumar Singh and Samar Husain. 2005. Comparison, selection and use of sentence alignment algorithms for new language pairs. In *Proceedings of the ACL Workshop on Building and Using Parallel Texts*, pages 99–106, Ann Arbor, Michigan. Association for Computational Linguistics.

Takehito Utsuro, Hiroshi Ikeda, Masaya Yamane, Yuji Matsumoto, and Makoto Nagao. 1994. Bilingual text matching using bilingual dictionary and statistics. In *Proceedings of the 15th Conference on Computational Linguistics*, pages 1076–1082, Moristown, NJ, USA. Association for Computational Linguistics.

D. Varga, L. Nmeth, P. Halcsy, A. Kornai, V. Trn, and Nagy

V. 2005. Parallel corpora for medium density languages. In *Proceedings of the RANLP 2005*, pages 590–596.

S. Vogel and A. Tribble. 2002. Improving statistical machine translation for a speech-to-speech translation task. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP-02)*.

S. Warwick and G. Russell. 1990. Bilingual concordancing and bilingual lexicography. In *EURALEX 4th International Congree*, Malaga, Spain.

S. Warwick, R. Catizone, and R. Graham. 1989. Deriving translation data from bilingual texts. In *Proceedings of the First International Lexical Acquisition Workshop*, Detroit.

D. Wu and X. Xia, 1995. *Large-Scale Automatic Extraction of an English-Chinese Translation Lexicon.*, pages 285–313. Vol. 9 edition.

Dekai Wu. 1994. Aligning a parallel english-chinese corpus statistically with lexical criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 80–87, Las Cruces, New Mexico, USA, June. Association for Computational Linguistics.

Ma Xiaoyi. 2006. Champollion: A robust parallel text sentence aligner. In *Proceedings of LREC-2006: Fifth International Conference on Language Resources and Evaluation*, pages 489–492.

K. Yamada and K. Knight. 2001. A syntax-based approach to statistical machine translation. In *Proceedings of the Conference of the Association for Computational Linguistics*, pages 523–530.

A Deeper Look into Features for NE Resolution in Indian Languages

Malarkodi, C. S. and Sobha, Lalitha Devi
AU-KBC Research Centre,
MIT Campus of Anna University,
Chennai, India
{csmalarkodi, sobha}@au-kbc.org

Abstract

Named Entity Recognition (NER) is the task of identifying and classifying proper nouns such as names of person, organization, location, etc. NER is used in various applications namely information extraction, question-answering, cross-lingual information access and query processing. Named Entity identification is mostly done for a specific domain and a particular language. In this paper, we developed NER for different Indian languages by using machine learning technique. Here we identify the common minimal features in English and various Indian languages of different language family (Bengali, Marathi, Punjabi, Tamil and Telugu). Further we used language dependent features to improve the system performance. The main goal of our task is to develop the NER with basic features and attain high performance. Conditional Random Fields (CRF) is used to build the training model for Indian languages. We trained CRF with few basic features and yielded encouraging results for both language generic and language specific system

1. Introduction

NER refers to the recognition and classification of proper nouns in the given document. It plays a vital role in Natural Language Processing (NLP) and Information Extraction (IE), molecular biology and bio-informatics. In English, proper nouns are specified in capital letters. Since the capitalization concept is not in Indian languages it is hard to identify the named entities. In this paper we developed the NER to recognize and categorize the named entities across various Indian languages like Bengali, Marathi, Punjabi, Tamil and Telugu. Unlike English, NE dictionaries and gazetteer list for Indian languages are not available on the web. The main advantage of our system is to identify the language independent and dependent aspects among Indian languages with only minimal set of features. We used CRF to label named entities and build language model for all languages. Section 2 describes the previous works done on NER and section 3 explains the challenges in NER. Section 4 explains our NER Tag set and section 5 gives brief introduction about the machine learning approach we used and section 6 describes the features implemented. Section 7 & 8 explains the experiments and results. The paper is concluded in section 9.

2. Related Works

Many research works **have been** done on NER across Foreign and Indian languages. Sasidhar, et al. (2011) made a survey on NER and explained various approaches used for NE identification. Some of them are decision trees, Support Vector Machine (SVM), Maximum Entropy Model, Hidden Markov Model (HMM), and CRF. In general all these categories come under either rule-based system or machine learning technique. In olden days rule-based systems were widely used for NER task. With the help of Gazetteer list and linguistic rules named entities are recognized. Vast amount of grammatical knowledge is required to generate linguistic constraints in that system. Even though this approach is easy to handle, if the NE is not found in the gazetteer list it is difficult to get good result.
HMM is a generative model and considered as a dynamic Bayesian network. Each state has a probability distribution over the possible output tokens. Therefore the sequence of tokens generated by an HMM gives some information about the sequence of states. Since the current label depends on the previous one it suffers from dependency problem. It also needs a large training corpus. Maximum Entropy Markov Model (MEMM) is a graphical model for sequence labeling that combines features of hidden Markov models (HMMs) and maximum entropy (MaxEnt) models. It overcomes the limitation of multiple feature and long term dependency suffered by HMM. As the probability transition leaving any given state must sum to one, it bias towards states with outgoing transitions. Hence the label bias problem occurred in MEMM. A conditional random field (CRF) is a statistical modeling method and a type of discriminative undirected probabilistic graphical model. It overcomes the label bias problem in MEMM.
In earlier days (1990s) NER was implemented for journalistic articles and military reports. Nowadays it is widely used for web blogs and biological domains. Since identification of named entities is the essential task to extract knowledgeable information from the documents, NER system has been added to the Message Understanding Conference (MUC-6). NER may be domain specific or language independent to all domains. NER for biological domain can be used to identify the name of medicine or type of gene products in the document. Biological NER has been developed to identify gene, protein, chemical, cell and organism names using rule-based system in English (Narayanaswamy, et al. 2003). NER has been applied to extract and identify names from e-mail by using CRF and five tags are designed to label named entities. The basic features like capitalization, lexical value, lower value, first & last names as dictionary features and some e-mail features were used (Minkov, et al. 2005).
A slightly modified version of HMM called 'Nymble' to identify names in Spanish and English text obtained 90% accuracy (Bikel, et al. 1997). A maximum entropy

approach to the NER task, where NER not only made use of local context within a sentence, but also made use of other occurrences of each word within the same document to extract useful features (global features). Such global features enhance the performance of NER. Two systems are represented as follows: a system ME1 that does not make use of any external knowledge base other than the training data, and a system ME2 that makes use of additional features derived from name lists. The features applied are First-word, Token Information, Lexicon features of previous and next token, Out of vocabulary, Dictionaries, suffixes and prefixes. They subdivided each class name into 4 sub-classes, i.e., N begin, N continue, N end, and N unique (Chieu, et al. 2002).

A hybrid system that applies maximum entropy model, language specific rules and gazetteers to the task of NER in Indian languages designed for the IJCNLP NERSSEAL shared task. They used different features like static word (previous and next word), context lists, dynamic NE tag, First word, Contains digit, numerical word, suffix, prefix, root information, pos as different features and got good results for Telugu, Hindi, Bengali, Oriya, Urdu (Srihari, et al. 2010). Wu, et al. (2006) presented a machine learning-based NE system by using SVMs. By integrating with rich feature set and the proposed mask method, high performance result is obtained. Different from previous SVM-based NE systems, this method achieved higher performance and efficiency which is improved by working with linear kernel. For real time processing usage the NE system can extract 2000 tokens per second.

Ekbal & Bandyopadhyay (2009) had developed Named Entity Recognition (NER) systems for two leading Indian languages, namely Bengali and Hindi using the Conditional Random Field (CRF) framework. The system makes use of different types of contextual information along with a variety of features that are helpful in predicting the different named entity (NE) classes. They have considered only the tags that denote person names, location names, organization names, number expressions, time expressions and measurement expressions. Evaluation results in overall f-score values of 81.15% for Bengali and 78.29% for Hindi for the test sets. 10-fold cross validation tests yield f-score values of 83.89% for Bengali and 80.93% for Hindi. They also described a pattern-directed shallow parsing approach for NER in Bengali by using linguistic features along with the same set of lexical contextual patterns.

(Vijayakrishna et al. 2008) worked on Domain focused Tamil Named Entity Recognizer for Tourism domain using CRF. It handles nested tagging of named entities with a hierarchical tag set containing 106 tags. They considered root of words, POS, combined word and POS, Dictionary of named entities as features to build the system. They obtain the f-score values of 81.79% for level 1, 83.77% for level 2 and 75.77 for level 3.

The NER system (Gali et al. 2008) build for NERSSEAL-2008 shared task which combines the machine learning techniques with language specific heuristics. The system has been tested on five languages such as Telugu, Hindi, Bengali, Urdu and Oriya using CRF followed by post processing which involves some heuristics.

Bhattarcharya et al. (2010) developed an approach of harnessing the global characteristics of the corpus for Hindi Named Entity Identification using information measures, distributional similarity, lexicon, term co-occurrence and language cues. They described that combining the global characteristics with the local contexts improves the accuracy; and with a very significant amount when the train and test corpus are not from same domain or similar genre. They also introduced a new scoring function, which is quite competitive with the best measure and better than other well known information measures.

The work proposed by Kumar, et al. (2011) to identify the NEs present in under-resourced Indian languages (Hindi and Marathi) using the NEs present in English, which is a high resourced language. The identified NEs are then utilized for the formation of multilingual document clusters using the Bisecting k-means clustering algorithm. They didn't make use of any non-English linguistic tools or resources such as Word Net, Part-Of-Speech tagger, bilingual dictionaries, etc., which makes the proposed approach completely language-independent. The system is evaluated using F-score, Purity and Normalized Mutual Information measures and the results obtained are encouraging.

3. Challenges in NER

For several decades more research work has been done on NER, yet some challenges exist in European languages. Indian languages belong to several language families, the major ones being the Indo-European languages, Indo-Aryan and the Dravidian languages. Bengali, Marathi, Punjabi are an Indo-Aryan languages. Tamil and Telugu belong to Dravidian Languages. The problems need to be resolve in Indian languages are discussed below.

Agglutination

All Dravidian languages including Tamil and Telugu have agglutinative nature. Case Markers attach as postpositions to proper or common nouns to form a single word.

Example 1

Koyilil vituwi ullathu
NN+PSP NN VM
(there is hostel in the temple)
in the above example, the case marker "il" suffixed to the common noun "koyil" .

Example 2

Kerala mannar marthandavarmanukku
NN NN NNP+PSP
therivikkappattathu
VM VM
(informed to the king marthandavarman of kerela)
Where the case marker "kku" attached to the proper noun marthandavarman.

Example 3

1948lo warangaltho saha hyderabadlo bharatha
QC+PSP NN+PSP RP NNP NN
desamlo kalisi poyindi
NN+PSP VM VAUX
(In 1948 Hyderabad along with Warangal merged in

India)
The Telugu case marker 'lo' and “tho” which denotes the preposition “in” and “with” attach to the year 1948 and the proper noun “warangal”.
As the case markers suffixed to the noun increases, the number of Nes in the training corpus also increases. So it is difficult for the machine to learn distinct NE patterns.

Ambiguity

In comparison with English, Indian languages suffer more due to the ambiguity that exists between common and proper nouns. For example the common names such as roja (rose) in Tamil, thamarai (lotus), malar (flower) in dictionaries can also be the names of person and in such cases it will be considered as NE.

Nested Entities

In some cases if the proper noun is considered separately, it belongs to one NE type but if we consider the same NE as a nested entity it belongs to another NE type. For instance,

kanchi sankaracharyar
NNP NNP
(kanchi sankaracharya)

“kanchi” refers to a location individually, in the case of nested entity it refers to the person “sankaracharyar” who lives in “kanchi”.

andal sannathi
NNP NN
(andal Temple)

If we take andal as a single entity it refers to a Person , where as the common noun sannathi followed by andal it refers to the location.

entiaar bhavan
NNP NN
(entiaar bhavan)

As a proper noun entiaar refers to Person name, if we consider it as nested entity it refers to FACILITY.

Capitalization

In English and some other European languages Capitalization is considered as the important feature to identify proper noun. It plays a major role in NE identification. Unlike English capitalization concept is not found in Indian languages.

Insufficient Resources

Like English, Indian languages are not resource rich language. For English named entities list, dictionaries and corpus of required size are available on the net. Sufficient NE resources for Indian languages are not available on the web. NLP tools such as POS taggers, morph analyzers are not efficient for Indian languages.

4. Our NER Tagset

We used the 22 second level tagset from Indian Language Named Entities tagset. The Named entity hierarchy is divided into three major classes; Entity Name, Time and Numerical expressions. The Name hierarchy has eleven attributes. Numeral Expression and time have four and three attributes respectively. Entities may be referenced in a text by their name, indicated by a common noun or noun phrase or represented by a pronoun. Person, organization, Location, Facilities, Cuisines, Locomotives, Artifact, Entertainment, Organisms, Plants and Diseases are the eleven types of Named entities. Numerical expressions are categorized as Distance, Money, Quantity and Count. Time, Year, Month, Date, Day, Period and Special day are considered as Time expressions.

5. Our approach

CRF is one of the machine learning techniques used for building probabilistic models to segment and label sequential data. (Lafferty et al. 2001), define Conditional Random Fields as follows: “Let G = (V,E) be a graph such that Y = (Yv)vεV so that Y is indexed by the vertices of G. Then (X,Y) is a conditional random field in case, when conditioned on X, the random variables Yv obey the Markov property with respect to the graph: $p(Y_v|X,Y_w,w\neq v) = p(Y_v|X,Y_w,w\sim v)$ where w~v means that w and v are neighbors in G”. Here X denotes a sentence and Y denotes the label sequence. The label sequence y which maximizes the likelihood probability $p\Theta(y|x)$ will be considered as the correct sequence, while testing for new sentence x with CRF model Θ. We used CRF++ (Taku, 2005), an open source toolkit for linear chain CRF. The information provided in the training file was taken by this tool to build the language model. Named entities are tagged in the test file by using this model build by CRF. Here we consider window of 5 words for analysis.

6. Features for CRFs

Capitalization, suffix, prefix, dictionary patterns, first word, frequent word, digit are the widely used feature for NE identification. These features vary for each language. The main goal of our engine is to identify the language dependent and independent aspects in Indian languages with only basic features such as word, pos and chunk. Our features are common and similar for all languages.

6.1 Language Independent Features

Dictionary patterns and gazetteer list varies for every languages and it contains only the frequent named entities. Our features are generic and compatible for all Indian Languages .We provide only minimal consistent features for the CRF engine to build the language model. Language Independent features used in our system are as follows.

- Word
- Part of speech tag
- Combination of word and POS
- Chunk

Language independent features are identified linguistically. These features are generic in all Indian Languages and the explanation is given as follows

Word

Our document contains both the NE and non NE tokens regarding to tourism domain. This will recognize the context in which a name entity will occur.

Part of speech (POS)
POS depends on the syntax and its relationship with its adjacent words. Most of all named entities are proper nouns, except some NE types like date, year, money etc. It represents whether the noun is proper or common noun. We consider the pos of current word and pos of word adjacent to the current word. To annotate the POS in the corpus we used Brill tagger.

Combination of word and POS
Current word combined with POS of preceding two words and current word combined with succeeding two words is taken as features.

Chunk
In linguistics chunking can be used to cluster each word with phrase level. Since named entities occur in noun-phrase chunking we take chunk of current, previous and next word as one of the feature.

6.2 Language Dependent Features

Further to improve the system performance, we analyzed the pos tags in Indian languages and used the most frequent pos tag that occur in the adjacent position of NE token for language dependent features. Along with word, POS and chunk, the most frequent POS of the preceding token of named entity is also considered.

Our data represents outer most tag of named entities for Indian languages. Our corpus contains 50k-70k words for training and 10k-20k words for testing in all six languages. Since we identify the independent and dependent features for Indian languages, we used two training and testing sets for each language. Tourism corpus were preprocessed and divided into 80% for training and 20% for testing. Along with training data, features generated in the template file were given to CRF for learning the NE patterns to build the language model. By using this model, CRF can identify and classify the NE type and produce the output.

To identify the most frequent POS tags we analyzed the tourism corpus of 1L to 50k words for all languages. Our NER system identified this language dependent feature automatically. The frequently occur POS in all the languages are almost same for both 50k and 1L corpus. Most frequently occurring POS tags in the preceding and succeeding position of named entities across Indian languages are identified automatically and their details are shown in the table.

Language	Word size	Frequent POS in preceding position of NE	Frequent POS in succeeding position of NE
English	1,0000	IN=4063 DT=2081 NN=881	IN=1813 DT=2081 NN=881
	50000	IN=1213 DT=658 NN=395	IN=572 NN=447 CC=403
Tamil	1,07193	NN=3697 VM=2390 PSP=1111	NN=7299 QC=1237 NNC=1162
	50000	NN=1501 VM=1017 PSP=559	NN=4143 VM=1401 QC=671
Punjabi	50000	NN=2435 VM=1189 PSP=445	NN=1565 VM=714 PSP=332
	15000	NN=656 VM=505 PSP=108	NN=621 VM=308 JJ=114
Bengali	73000	NN=384 JJ=125 VM=155	NN=414 VM=132 JJ=129
	50000	NN=292 VM=130 JJ=86	NN=505 VM=157 JJ=167
Marathi	129227	PSP=1976 NN=1140 CC=596	PSP=2267 NN=1094 CC=325
	50000	PSP=519 NN=325 CC=239	PSP=1232 NN=400 CC=158
Telugu	117016	NN=903 CC=811 PRP=379	NN=1094 RP=548 PRP=543
	50000	NN=445 PRP=5144 JJ=263	NN=9359 VM=5144 JJ=3797

Table 1: Frequent POS

7. Experiments

The named entities are tagged in the tourism corpus collected from various on-line tourism sites. The NE tagged corpuses are available in the following languages: Bengali, English, Marathi, Punjabi, Tamil and Telugu. For tourism domain in languages like Bengali, English, Marathi, Punjabi and Tamil we used 70k-90k word corpus and 50k word corpus for Telugu. For each language, this corpus is divided into training and testing sets and they consist of 80% and 20% of the whole data. Total number of named entities used in the corpus for the six languages is shown in the table 2. CRF is trained with the training data to build the language model by using the features we provided. Named entities in the Test data are identified and results are evaluated manually.

S.no	Language	Total no of NE
1	English	12645
2	Tamil	18685
3	Punjabi	7062
4	Bengali	3270
5	Marathi	6318
6	Telugu	10316

Table 2: NE Details

8. Results & Discussions

Experimental results are evaluated using precision, recall and f-measure. The evaluation results obtain for language independent is shown in table 3 and language dependent is presented in table 4 respectively. NER engine achieved above 65% to 90% as precision and above 50% to 70% as f-measure except Telugu. For Bengali we obtain an accuracy of 91% as precision and 83% as f-measure. Since we want to improve the recall, we identified the language dependent features for six languages by using the most frequently occurred POS tags in preceding position of NE.

The results we obtained for language dependent features shown the major improvement in recall score for all languages except Bengali. The recall and f-measure scores of language dependent system show that their results were much better than language independent system. The mean value for language dependent system attain above 60% to 75% for English, Tamil, Marathi, Bengali and Punjabi and 35% for Telugu. Evaluation of NE tags for six languages are shown in table 5. "P, R and F-M" denote the precision, recall and f-measure for each NE tags.

S.no	Language	Precision	Recall	F-measure
1	English	81.93	65.56	72.83
2	Tamil	74.89	54.86	63.33
3	Punjabi	77.83	37.59	50.70
4	Bengali	91.78	75.75	83.00
5	Marathi	68.04	43.99	53.44
6	Telugu	39.32	27.96	32.68

Table 3: Independent Features

S.no	Language	Precision	Recall	F-measure
1	English	80.90	74.80	77.73
2	Tamil	77.07	65.89	71.04
3	Punjabi	73.96	52.66	61.52
4	Bengali	83.00	72.58	77.44
5	Marathi	63.60	66.24	64.89
6	Telugu	39.21	33.21	35.97

Table 4: Dependent Features

S.no	Language	PERSON & LOCATION			NUMERICAL EXPRESSIONS			TIME EXPRESSIONS			Others		
		P	R	F-M	P	R	F-M	P	R	F-M	P	R	F-M
1	English	88.79	86.03	87.39	88.10	88.43	88.26	69.76	59.52	64.23	52.50	47.36	49.80
2	Tamil	92.82	77.92	84.72	50.00	37.72	43.00	93.66	83.88	88.50	62.50	32.03	42.35
3	Punjabi	89.56	64.09	74.72	33.33	75.00	46.15	39.72	34.52	36.94	71.42	15.30	25.21
4	Bengali	89.06	76.90	82.54	83.83	55.00	64.42	58.82	66.66	62.50	89.79	61.11	72.72
5	Marathi	80.84	81.84	81.34	68.22	73.18	70.61	45.28	47.05	46.15	35.81	41.73	38.54
6	Telugu	65.89	40.46	50.13	49.57	42.44	45.73	36.69	23.12	28.36	30.08	23.61	26.45

Table 5: Evaluation of NE Tags for Dependent Features

Numerical expressions include Distance, Quantity, Count and Money. Month, Day, Year, Period and Time comes under Time expressions. NE tags such as Artifact, Organisms, Facilities, Locomotives, Organization, Plants and Materials are mapped as 'others' category. Person and Location tags obtain higher accuracy than other categories.

8.1 Discussions

To improve the recall level, we applied language dependent features. From the results we observed that all languages except Bengali obtain high recall rate than language independent scores. Our NER engine can identify the NE and non NE words with the accuracy of more than 80% for all the languages. Due to data sparsity and discrepancy in tagging we got fewer score in Telugu than other languages. From the results it is clear that due to language dependent features recall and f-measure are improved for all languages except Bengali. Since the named entities in Bengali corpus are less ambiguous, primary features in language independent system are enough to obtain efficient score. The language dependent features we added generate more false positives than true positives. So the language dependent features impede the data and affect the results for Bengali. In future we should analyze the language dependent features which lead to fewer score than language independent. With dependent features NER engine can identify more named entities which results in high recall and f-measure for languages English, Marathi, Punjabi, Tamil and Telugu. Though the language dependent system find more number of named entities, but the recognition of NE type is not accurate in some NEs which affects the precision, while in comparison with language independent system.

For error analysis we want to discuss some confusion exists among NE themselves. In some cases entities considered as LOCATION are recognized as PERSON and vice versa.

For instance, "sri thesigan thirumaligai" (sri thesikan mahal) 'sri thesigan' is a name of the PERSON, since it is followed by the proper noun 'mahal' it is considered as LOCATION. But the machine can recognize 'sri thesigan' as a PERSON and 'mahal' as LOCATION.

In English an entity "Kanagasabhai hall" is considered as LOCATION, since the word 'kanagasabhai' indicates the person name machine tagged 'kanagasabhai' as PERSON and hall as LOCATION. As nested entity

"Alappuzha beach" belong to ENTERTAINMENT but the machine recognized 'Alappuzha' as LOCATION and 'beach' as ENTERTAINMENT.

Consider another example "avani urchavam" (avani festival) avani is one of the Tamil month, as it is followed by festival it is considered as ENTERTAINMENT. But the machine tagged 'avani' as MONTH and festival as ENTERTAINMENT.

There may be ambiguity between PLACE and ORGANIZATION tags. For example in Telugu "thirumala thirupathi thevasthanam" (thirumala thitumathi devasthanam) as a nested entity it refers to ORGANIZATION. But the machine recognizes 'thirumala thirupathi' as LOCATION and 'thevasthanam' as ORGANIZATION.

Sometimes confusion occurs between Special day and PERSON. "sri ramanujar jayanthi" which indicates the birthday of sri ramanujar and it is considered as special day but the system classified sri ramanujar as PERSON.

These problems can be resolved by finding the patterns of common noun followed by named entities, and using the NE dictionaries. We proved that with basic minimal features our NER engine can achieve high accuracy. Despite of these ambiguities without using NE dictionaries and additional features such as prefix& suffix pattern, presence of digits, or any smoothing techniques we attain efficient score for all the languages.

9. Conclusion

In the paper we showed that with the basic minimal features our system is able to obtain efficient results for different languages. The language independent and dependent features we identified are compatible and simple for all languages. Our NER system is developed efficiently so that it can handle different NE tags and unknown named entities. Even we doesn't make use of extra features and smoothing or post processing techniques our NER engine performed well for all languages. We plan to increase our corpus size and extend our work towards other Indian languages. We are developing novel language dependent features and accomplishment of such features can improve our NER system significantly.

10. References

Bikel, D.M., Miller, S., Schwartz, R., Weischedel., R. (1997). Nymble: A high-performance learning name-finder. *In proceedings of Fifth Conference on Applied Natural Language Processing*, pp. 194-201.

Chieu, H.L., Ng, H.T. (2002). Named Entity Recognition: A Maximum Entropy Approach Using Global Information. *In proceedings of 19th international conference on Computational linguistics.* 1, Association for Computational Linguistics, Stroudsburg, PA, USA, pp. 1-7.

Ekbal, A., Bandyopadhyay, S. (2009). A Conditional Random Field Approach for Named Entity Recognition in Bengali and Hindi. *Linguistic Issues in Language Technology*, 2(1), pp.1-44.

Gali, k., Surana, H., Vaidya, A., Shishtla, P., Sharma, D.M. (2008). Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition. *In Proceedings of the workshop on Workshop on NER for South and South East Asian Languages, IJCNLP-08*, Hyderabad, India.

kumar, K.N., Santosh, G.S.K., Varma, V. (2011). A Language-Independent Approach to Identify the Named Entities in under-resourced languages and Clustering Multilingual Documents. *International Conference on Multilingual and Multimodal Information Access Evaluation*, University of Amsterdam, Netherlands

Lafferty, J., McCallum, A., Pereira, F. (2001). Conditional Random Fields for segmenting and labeling sequence data. *In the proceedings of ICML-01*, pp. 282-289.

Minkov, E., Wang, R.C., Cohen, W.W. (2005). Extracting Personal Names from Email: Applying Named Entity Recognition to Informal Text. *In proceedings of Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics Stroudsburg, PA, USA, pp. 443-450.

Narayanaswamy, M., Ravikumar, K.E., Shanker, V.K. (2003). A Biological Named Entity Recognizer. *In proceedings of Pacific Symposium on Biocomputing*, 8, pp. 427-438.

Sasidhar, B., Yohan, P.M., Babu, V.A., Govarhan, A. (2011). A Survey on Named Entity Recognition in Indian Languages with particular reference to Telugu. *J. International Journal of Computer Science Issues*, 8, pp. 438-443.

Sobha, L., Vijayakrishna, R. (2008). Domain focused Named Entity for Tamil using Conditional Random Fields. *In proceedings of IJNLP-08 workshop on NER for South and South East Asian Languages,* Hyderabad, India, pp. 59-66

Srihari, R., Niu, C., Yu, L. (2000). A Hybrid Approach for Named Entity Recognition in Indian Languages. *In proceedings of 6th Applied Natural Language Conference*, pp. 247-254

Taku, k. (2005). CRF++, an open source toolkit for CRF, *http://crfpp.sourceforge.net*

Wu, Y., Fan, T., Lee, Y., and Yen, S. (2006). Extracting Named Entities Using Support Vector Machines. *In proceedings of KDLL*, pp. 91-103

‘atu’ Difficult Pronominal in Tamil

Akilandeswari, A., Bakiyavathi, T., and Sobha, Lalitha Devi
AU-KBC Research Centre,
MIT Campus of Anna University,
Chennai, India
{akila, bakiya, sobha}@au-kbc.org

Abstract

The paper presents a detailed analysis of 'atu' in Tamil, which is an equivalent of 'it' in English. ‘atu’ has many roles such as third person neuter pronoun, emphatic and as nominalizer. In this paper we are considering ‘atu’ in one particular construction, where the relative participle verb is suffixed with ‘atu’ in a multiple embedded sentence. In this form ‘atu’ can be anaphoric and non-anaphoric. In this paper we give a detailed analysis of ‘atu’ in the above construction. Using the analysis we identify the anaphoric and non anaphoric ‘atu’ and also the antecedent of the anaphoric ‘atu’ using CRFs.

1. Introduction

Anaphora resolution is the problem of resolving references to previous references in discourse. These references are usually noun phrases. The process of finding the antecedent for an anaphor is anaphora resolution. The most widespread type of anaphora is the pronominal anaphora. In this paper, we analyzed in detail ‘atu’ which is similar to ‘it’ in English and it can be anaphoric or non anaphoric.

‘atu’ is a third person, singular, neuter pronoun in Tamil. But this also has many grammatical roles such as an emphatic and nominalizer. When ‘atu’ is suffixed with a relative participial (RP) non- finite verb form it nominalizes the verb into a noun according to contemporary linguistic theories. Consider the following RP form of the verb ‘vaa’ ‘come’ example .

1. vaa + nth + a -> vantha
   Come [root]+ past(pst) + RP -> who came

When the third person pronoun ‘atu’ gets suffixed to the RP verb the following form is obtained. All third person pronouns (avan ‘he’, aval ‘she’ and atu ‘it’) can be suffixed to the RP form of the verb.

2. vaa+ nth+a+atu -> *vantha**atu***
come+ pst+RP+ pronoun -> It that came

In this study we consider ‘atu’ which occurs when it is suffixed to a non-finite RP form of the verb. Consider the following examples for anaphoric and non-anaphoric ‘atu’.

3. <Ant>**meenakshi urvalam** </Ant> neRRu
Meenakshi procession yesterday
<Ana>*vant**atu***<Ana> azhakaka Irunthathu.
come+pst+Pronoun beautiful be+pst+n .

‘The Meenakshi procession that came yesterday was beautiful.’

In sentence (3) the pronoun ‘atu’ is agglutinated to the non-finite verb ‘vanta’ ‘come’. Here ‘atu’ refers to Meenakshi Procession. Hence ‘meenakshi urvalam’ is the antecedent of ‘***atu***’. It agrees in person, number, and gender features of ‘atu’. The antecedent is also the subject of the sentence. We can substitute the antecedent ‘meenakshi urvalam’ for ‘atu’ in the above sentence and the meaning of the sentence remains the same. The substituted sentence is given below.

3 (a). neRRu **vanta Meenakshi urvalam** azhakaka irunthathu.

The Meenakshi procession that came yesterday was beautiful

4. intha azhakaana viittai <Non-**Ana**> *kattiy**atu***
This beautiful house-acc built+RP+Pronoun
</Non-Ana> antha periyavar.
the oldman.

‘This is a beautiful house which was built by the old man’

In sentence (4) ‘atu’ refers to the verb itself. It can be seen that the pronoun refers to the construction of the building and not the building. We consider this as non anaphoric ‘atu’.

The suffixed form of the verb was considered as nominalization in contemporary linguistics. From the above examples and analysis it is evident that it is not nominalization but it is a suffixation of pronoun. ‘atu’ in this form behaves as an anaphora and it has all the features of anaphora. This has motivated us to further explore on this and identify the anaphoric and non anaphoric ‘atu’ in this construction.

2. Previous Work:

Anaphora resolution is an area well researched for many languages in the last few decades. There are several

approaches used in resolving anaphors such as rule based, knowledge based and machine learning. One of the early works in pronominal resolution is by Hobb's naive approach, which relies on semantic information (Hobbs, J, 1978). Carter with Wilkas' common sense inference theory came up with a system (Carter. D, 1987). Carbonell and Brown's introduced an approach of combining the multiple knowledge system (Carbonell.J. G. & Brown.R.D, 1988). The initial approaches, where broadly classified as knowledge poor and rich approach.

Syntax based approach by Hobb (naive approach), centering theory based approaches (Joshi, A. K.& Kuhn. S, 1979; Joshi, A. K.& Weinstein.S, 1981) and factor/indicator based approach such as Lappin and Leass' method of identifying the antecedent using a set of salience factors and weights associated to it (Lafferty et al, 2001). This approach requires deep syntactic analysis. Ruslan Mitkov introduced two approaches based on set of indicators, MOA (Mitkov's Original Approach) and MARS (Mitkov's Anaphora Resolution System) (Mitkov.R, 1998). These indicators return a value based on certain aspects of the context in which the anaphor and the possible antecedent can occur. The return values range from -1 to 2. MOA does not make use of syntactic analysis, whereas MARS system makes use of shallow dependency analysis.

There are very few works done in anaphora resolution with respect to Indian Languages. Some of the works done are VASISTH a rule based system which works with shallow parsing and exploits the rich morphology in Indian languages for identifying the antecedent for anaphors (Sobha.L & Patnaik.B.N, 1999). (Sobha.L, 2007) used salience measure for resolving pronominals in Tamil. (Murthi.N.K.N, 2007) have looked into the anaphora resolution in Tamil, using Machine Learning technique: Linear Regression and compared it with salience factors. Dhar worked on "A method for pronominal anaphora resolution in Bengali (Dhar.A. & Garain.U, 2008; Sobha.L & Pralayankar.P, 2008) worked on "Algorithm for Anaphor Resolution in Sanskrit". Resolving Pronominal Anaphora in Hindi Using Hobb's Algorithm was done by Kamalesh Dutta. Identification of anaphoric and non-anaphoric 'atu' is not attempted till now and this work is first of its kind. Previous papers which are discussed about Tamil Anaphora, not specifically discussed about 'atu'. But it was used some semisupervised algorithms for pronominal anaphoras like he(avan), she(aval).

Tamil is a Dravidian language which is morphologically rich and word order free. As discussed earlier 'atu' has many roles such as anaphoric (pronominal), nominalizer and emphatic markers. So identification of 'atu' as a pronominal is a difficult task. Here we consider the construction where 'atu' is suffixed with the non-finite RP verb in the relative participle clause. This paper has the followings sections: Analysis of 'atu', methodology, algorithm, results and discussion, and conclusion.

3. 'atu' as Anaphoric and Non-Anaphoric

3.1 Anaphoric 'atu'

The most widespread type of anaphora is the pronominal anaphora. 'atu' can occur independently in the text. It always refers to non-human, inanimate object or events.. 'atu', a pronoun which is marked inside a relative participle clause within a larger sentence. 'atu' is pronoun because it relates the relative clause to the noun that it modifies. Here we had taken 'atu' and its suffixes.

5. ariviyal maanavarkaLukku nuNNiya aRivu
Science students+dat in-depth knowledge
<Ant> intha puththakam **</Ant>** **<Ana>** *kotuthth**atu***
this book gave+pst+RP+pronoun
</Ana> anaivarukkum theriyum.
everybody knows.

'Everybody knows that this book gave in-depth knowledge to science students'

'atu' in *kotuthtatu* is a pronoun, which is anaphoric and the antecedent is 'intha puththakam'**.** The antecedent is agreeing in person, number and gender features of 'atu' and is the subject of the sentence.

6. Cuciinthram koovilil uLLa <Ant> piramaaNta
Cuciinthram temple-loc be big
Anjaneeyar </Ant> ulakap pukazh <Ana>
Anjaneyar world famous
*peRR**atu*** </Ana> akum.
got+pst+RP+pronoun become.

'The world famous big Anjaneyar which is in cuciinthram temple.'

'atu' in 'peRRatu' is a pronoun, which is anaphoric and the antecedent is 'piramaaNta Anjaneeyar'. The antecedent is agreeing in person, number and gender features of 'atu' and is also the subject of the sentence.

7. ore kallilirunwu cethuukkappatta wUNkalil
one stone-loc-abl sculpt pillar-loc
iwuponru <Ant>vithavithamaana ocaikal</Ant>
like this different sounds
<Ana>*varuv**atu***</Ana> accariyam
come+pst+RP+pronoun surprise
warum vicaayam akum.
give matter become.

'The pillars which is sculpted from a single stone which will give different sounds is a surprise.'

'atu' in varuvatu is a pronoun, which is anaphoric and the antecedent is vithavithamaana ocaikal. The antecedent is agreeing in person, number and gender features of 'atu' and is the subject of the sentence.

From the above examples if is evident that 'atu' when occurring with the a RP non-finite verb behaves like a

pronoun and has antecedent as the subject of the clause or sentence.

3.2 Non-Anaphoric 'atu'

'atu' can be non-anaphoric when it is suffixed with a RP verb form. Consider the following sentences.

8. oru kai pirakalaathanai
One hand pirakalaathan+acc
<non-ana> *aacirvathipp**atu*** </non-ana> polavum
bless+pst+RP+pronoun , like
oru kai apaya muththiraiyaiyum
one hand apaya muththirai+acc+-um
kaattukiRathu
showing

'It is shown as though one hand kept as blessing Prahalathan and the other showing the danger symbol'
In sentence (8) 'atu' refers to the RP verb to which it is suffixed to, thus giving an emphatic meaning to the verb. Here it is not referring to any noun and we consider this as non-anaphoric.

9.appoothu vettaikku <non-ana>*vanth**atu***<non-ana> pol
At that time hunt+dat come+pst+RP+pronoun like
SrIrakam_perumaaL vara avarathu azhakil
Srirankam_perumaal come his beauty+loc
kamalavalli mayankinaar.
kamlavalli fell-3sh.

'At that time SriRankapperumal came like for hunting, kamalavalli fell on his beauty'

Here in sentence (9) the pronominal 'atu' refers to 'vantha' the RP verb to which it is suffixed. Here also the pronoun is not in the anaphoric form.

From the above analysis it is evident that there are two types of 'atu', anaphoric and non anaphoric when attached to relative participle verb.

4. Methodology

Here we identify the anaphoric and non anaphoric 'atu' and also the antecedent of the anaphoric 'atu' using CRFs. For the present work we have taken the Tourism domain corpus form the web. A total of 6586 sentences with RP clause constructions are taken for the analysis. The corpus is preprocessed with syntactic information such as POS, NP and VP chunk and clause boundary. To identify 'atu' we tagged the corpus for clause boundary and marked the 'RP' marker to the RP verb forms. The corpus is also tagged for anaphor and its antecedents using annotation tool. We are dividing this task into three:
1**.** Annotation of anaphors and Antecedents with suitable index in the corpus.
2. Identifying Anaphoric and Non-Anaphoric and the antecedent for the anaphoric 'atu' in the corpus.

4.1 Annotation of the Corpus

The first step is to annotate the corpus with anaphor, non-anaphor and antecedents with index. For that we used a annotation tool, PALinkA. We used PALinka customized for Indian languages for the tagging purpose.

PALinkA is abbreviated as Perspicuous and Adjustable Links Annotator, a Annotation tool. It is a language independent tool, written in java. It is tested on French, Romaninan Spanish, Tamil and other Indian Languages, also it is user friendly, we can annotate by selecting the markables and click on it. The input file to PALinkA has to be a well-formed XML file and the produced output is also a well-formed XML.

We used PALinka tool for annotating the anaphor and antecedent. The preprocessed files with all syntactic information to be annotated should be in XML format. We considered both anaphor and antecedent as markables. For annotations, first anaphor and antecedent should be marked as markables and if it is anaphoric, link these two markables. Finally all the possible anaphor and antecedents are tagged with index. After annotation, these XML files are converted to column format files which are required for the machine learning system.

4.2 Identifying the Anaphoric and Non-Anaphoric 'atu'

The second step is to identify the anaphoric and non-anaphoric pronouns and to find antecedents. The training corpus contains 109 anaphoric pronouns and its antecedents and 41 non-anaphoric pronouns. The features identified for training the system are given below.

Features for anaphoric and non anaphoric 'atu' identification

1. RP marker (the word is anaphor or not, which is marked as anaphor)
2. Post Positional following the RP for non anaphoric
3. Adjective following the RP for anaphoric
4. Verb following the RP for anaphoric
5. Noun following the RP for anaphoric
6. Part of speech tag
7. NP chunk
8. Clause Marking (clause start /end)

Features for 'atu' resolution identification
1. Part of speech tag
2. NP chunk
3. Position of the noun phrase in the sentence
4. Person, number, gender
5. Root word category
6. Suffixes in the antecedent
7. clause (clause start /end)
8. Position of the noun phrase in the sentence (FirstNP, MiddleNP, ImmediateNP)
9. Case marker of NP phrase

5. Conditional Random Fields

CRF++ is toolkit designed for generic purpose and are applied to a variety of NLP tasks. The machine learning method of CRFs was chosen to do our experiments,

because of its flexibility to build linguistic rules. CRFs can contain number of feature functions. The advantage of CRFs is that it can model not only sequential data, but also non-linear data. Since the task of extraction of antecedents is syntactic and semantic task, CRFs is appropriate for this purpose.

CRFs is an undirected graphical model, for labeling and segmenting structured data, such as sequences, trees and lattices, where the conditional probabilities of the output are maximized for a given input sequence [9]. CRF++ (Kudo, 2005), an open source toolkit for linear chain CRFs, was used for the experiments performed. For our task, CRFs works in two stages, namely training and testing. In the training stage, the training data and template file is given as input. The training data is preprocessed and manually tagged. The template file contains the rules to learn the features from the training data. Taking these two files as input CRFs training module generates a model file, which holds the extracted features. In the next phase, testing is done by supplying test data and feature model to the CRFs test engine. The CRFs test module takes these two files as input and generates the antecedent.

6. Figures & Tables

6.1 Figures

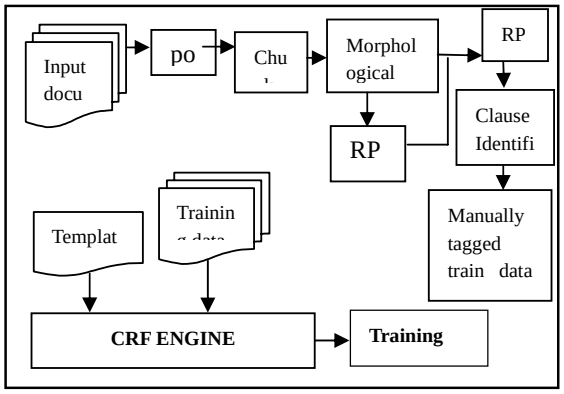


Figure 1: System flow to training the data

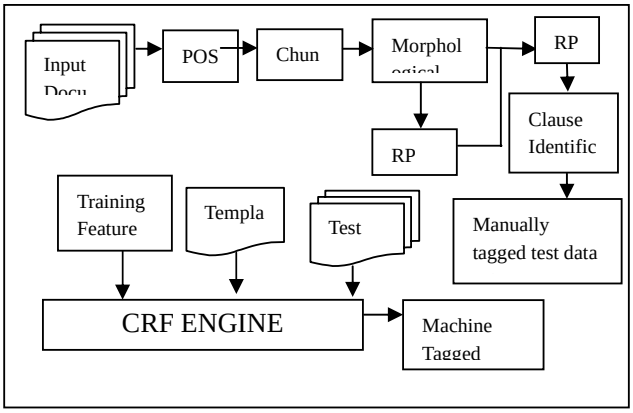


Figure 2: System flow to test the data

The algorithm is as given below

Step1: Input the Tourism corpus
Step2: Preprocess the corpus, tagged with POS,NP-VP Chunk, Morph information
(root word category, gender, number, person, suffix, case) and Clause
Information.
Step3: Mark the Anaphor 'atu' using "RP" marker (relative participle) which is
in relative participle clause of a sentence.
Step4: Tag the Anaphor and the Antecedent in PALinkA tool.
Step5: Identify the features and given to CRF
Step6: Train and test the Train corpus and Test corpus with the features in CRF

6.2 Results and Discussion

S.no	Anaphors & Non-Anaphors	Testing	Correctly tagged	Uncorrectly Tagged	Totally system tagged	Precision	Recall
1	No of Anaphors	31	26	23	49	53%	83.87%
2	No of Non-Anaphors	182	151	73	224	67.49%	82.96%
3	Total	213	177	96	273	64.83%	83.09%

Table 1: Results for Anaphoric and Non-Anaphoric

S.no	Total Anaphors	Anaphors system tagged	Antecedents Tagged correctly	Antecedents Tagged incorrectly	Precision	Recall
1	30	23	20	3	86..95%	66.66%

Table 2: Results for 'atu' Resolution

Here in this table, the total number of Anaphors is 30. The total number of anaphoric instances were 30. The system correctly identified antecedent for 20 anaphors. And for 2 anaphors the system wrongly tagged the antecedent. Though we could not test in large corpus, the data we used for testing showed that the features identified works well for any data.

7. Conclusion and Future Work

This work considered 'atu' in the RP construction for resolution. From the analysis we confirmed that when pronoun 'atu' is suffixed to the RP non finite verb , it functions as pronominal. And resolution of such pronominals is very much necessary for all natural language processing applications.

8. References

Carbonell, J. G. and Brown, R. D. (1988). Anaphora resolution: A multi-strategy approach. In: *12th International Conference on Computational Linguistics*, 96--101.

Carter, D. (1987). Interpreting anaphors in natural language texts. Chisester: Ellis Horwood ltd.

Dhar, A. and Garain, U (2008). A method for pronominal anaphora resolution in Bengali In: *proceedings 6th Int. Conf. on Natural Language Processing (ICON),* Pune, India, December

Hobbs, J. (1978). Resolving pronoun references. Lingua 44, 339--352.

Jha, G.N., Sobha, L, Mishra, D., Singh, S.K., Pralayankar, P. (2008). Anaphors in Sanskrit In: *Proceedings Second Workshop on Anaphora Resolution* Johansson, C.(Ed.)

Joshi, A. K. and Kuhn, S. (1979). Centered logic: The role of entity centered sentence representation in natural language inferencing. *In: International Joint Conference on Artificial Intelligence.*

Joshi, A. K. and Weinstein. S (1981). Control of inference: Role of some aspects of discourse structure - centering. *In: International Joint Conference on Artificial Intelligence*, pp. 385--387.

Lappin, S. and Leass, H. J. (1994). An algorithm for pronominal anaphora resolution. Computational Linguistics 20 (4), 535--561.

Lafferty, J., McCallum, A. and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: *18th International Conference on Machine Learning*, pp .282--289. Morgan Kaufmann, San Francisco, USA,

Mitkov, R. (1998). Robust pronoun resolution with limited knowledge. In: *17th International Conference on Computational Linguistics* (COLING' 98/ACL'98), Montreal, Canada, pp. 869--875

Murthi, N.K.N., Sobha, L., Muthukumari, B. (2007) Pronominal Resolution in Tamil Using Machine Learning Approach The First Workshop on Anaphora Resolution (WAR I), Ed Christer Johansson, Cambridge Scholars Publishing, 15 Angerton Gardens, Newcastle, NE5 2JA, UK pp.39-50

Orasan, C. (2003). PALinkA: a highly customizable tool for discourse annotation. In: *proceedings 4th SIGdial Workshop on Discourse and Dialog*, Sapporo, Japan, 5 – 6 July, pp. 39 – 43

Sobha, L., Patnaik, B.N. (1999). VASISTH- An Anaphora Resolution System Unpublished Doctoral dissertation. Mahatma Gandhi University, Kottayam, Kerala

Sobha, L., Pralayankar, P. (2008). Algorithm for Anaphor Resolution in Sanskrit In: *Proceedings 2nd Sanskrit Computational Linguistics Symposium*, Brown University, USA

Sobha, L. (2007). Resolution of Pronominals in Tamil, Computing Theory and Application, The IEEE Computer Society Press, Los Alamitos, CA, pp. 475-79.

Restructuring of Pāṇinian Morphological Rules for Computer Processing of Sanskrit Nominal Inflections

Subhash Chandra
Special Centre for Sanskrit Studies
Jawaharlal Nehru University, New Delhi
E-mail: subhash.chandra@cdac.in

Abstract

Sanskrit is morphologically rich language. Pāṅini is known for his Sanskrit grammar, particularly in his formulation of the 3,959 rules of Sanskrit morphology, syntax and semantics described in Aṣṭādhyāī (AD). The rules have been set out, much in the way of a mathematical function, to define the basic elements of the language including sentence structure, vowels, consonants, nouns, and verbs. The paper presents a developed Morphological Analyzer for Sanskrit Nominal Inflections which is based on restructuring of Pāṇinian morphological rules. Restructuring of morphological rules is performed by reordering the Pāṇinian morphological rules for computational purpose. Author proposed three steps for Sanskrit nominal morphological analysis. There are two parts of the system. One is recognition of nominal inflections from Sanskrit Texts and other is analysis of nominal inflections. The recognition of all nominal inflections is done in Sanskrit texts with the help of relational databases. Then the nominal word is sliced into a sequence of morphemes and stem. Finally detailed information including stem, suffix, case and number of those nominal words are provided. Evaluation report shows that the accuracy of the system is 85%. It is also found that the system fails to provide 15% correct results where 5% errors are generated due to improper recognition of nominal words, 8% errors for incorrect morphological analysis of nominal words and rest 2% errors occur because of unsuccessful analysis. Online version of the system is available which runs on an Apache Tomcat Server.

Keywords: Sanskrit Morphology, Morphological Analyzer, Nominal Inflection in Sanskrit, Paṇinian Morphological Rules, Language resources for basic NLP.

1. Introduction

Aṣṭādhyāyī (AD) is a grammar defining the structure and syntax of the Sanskrit language. In 3959 aphorisms, or sutras (rules), Paṇini has described the structure of Sanskrit completely. The developed morphological analyzer for Sanskrit nominal inflections is based on engineering of Paṇinian morphological rules in restructuring because he gave the rules for Morphology generation. Restructuring is performed by reordering the Paṇinian morphological rules to analysis of derived morphological forms for computational purpose. The study of the structure and form of words in languages or a language, including inflection, derivation, and the formation of compounds is called morphology. In Sanskrit, a syntactic unit is called pada. Cordona (1988) posits the formula for Sanskrit sentence (N-En) p…(V-Ev)p (Cardona, 1998). In the Sanskrit pada can be nominal (*subanta*) or verbal (*tiṅanta*) (Chandra and Jha, 2011; Jha, 2004, Chandra, 2010). These forms are formed by inflecting the stems and hence they are part of Sanskrit inflectional morphology. The derivational morphology in Sanskrit studies primary forms (*kṛdanta*) and secondary forms (*taddhitānta*), compounds (*samāsa*), feminine forms (*strīpratyayānta*) (Chandra and Jha, 2011, Chandra, 2010) etc. Sanskrit morphology can be divided into two major categories- nominal (consisting of primary and secondary derivations of nouns including feminine forms and compounds and their inflected forms called padas) and verbal (consisting all verbal forms). Sanskrit has approximately 2014 verb roots (including *kandvādi* (a group of Sanskrit verb roots), classified in 10 groups (gaṇas), the derived verb forms can have 12 derivational suffixes. These can have *ātmanepadī* (verbs referring to the activity for the self) and *prasmaipadī* (verbs referring to the activity for others) (Jha, 2004). A single verb root may have approximately 2190 (tense, aspect, number etc.) morphological forms (Mishra and Jha 2005). Sanskrit nominal inflectional morphology is of two types, Primary *kṛdanta* (roots forms that end with *kṛt* suffixes) and secondary *taddhitānta* (noun forms that end with *taddhita* suffixes listed by Pāṇini). Secondary nominal inflectional morphology may be of following types like- *samāsānta* (compound nouns), *strīpratyayānta* (feminine forms) etc. They can also include *upasargas* (prefix) and *avyayas* (indeclinables) etc. According to Pāṇini, there are 21 nominal suffixes (seven *vibhaktis* and combination of three numbers = 21), which are attached to the nominal bases (*prātipadika*) according to syntactic category, gender and ending of the base character of the base (Chandra and Jha, 2011; Mishra and Jha 2005).

In a Sanskrit sentence, all non-verbal categories are nominal inflections (*subanta-padas*) which makes it essential to analyze these padas before any other computer processing (Chandra and Jha, 2011). Sanskrit subanta forms can be potentially very complex. Pāṇini has listed the nominal suffixes (*sup* suffixes) *su, au, jas,am, auṭ, śas, ṭā, bhyām, bhis, ṅe, bhyām, bhyas, ṅasi, bhyām, bhyas, ṅas, os, ām, ṅi, os, sup.* These suffixes are the sets of three as- (*su, au, jas*) (*am, auṭ, śas*) (*ṭā, bhyām, bhis*)

(*ṅe, bhyām, bhyas*) (*ṅasi, bhyām, bhyas*) (*ṅas, os, ām*) (*ṅi, os, sup*) for singular, dual and plural (Sharma, 2003) respectively. These suffixes are added to the *prātipadikas* (any meaningful form of a word, which is neither a root nor an inflected forms) to obtain inflected forms (*subanta padas*). *prātipadikas* are of two types: primitive and derived. The close list of primitive bases are stored in *gaṇapāṭha* (collection of bases with similar forms) while the latter are formed by adding the derivational suffixes.

The main objective of this work is to create a collection of resources and tools for Sanskrit morphology. It is very helpful for machine translation system for Sanskrit to other languages and self-reading and understanding Sanskrit language. It is helpful in most conceivable applications of natural language processing, in particular machine translation. We emphasize the methodological analysis based on reverse engineering implementation of Pṇṇini's Sanskrit Grammar. In this case reverse engineering process is the reversing of Pāṇinian morphology formation process. Pāṇini has described the rules for morphology forms (*shabda-roopa*) generation in the order of base + suffix = inflected forms (*prātipadika + sup = subanta-pada*) for example, राम + सु = रामः [*IAST: rāma + su = rāmaḥ*] but this system does analysis in the order of inflected forms = base + suffix + case + number (*subanta-pada = prātipadika + sup + vibhakti + vachana*) for example, रामः = राम + सु, प्रथमा, एकवचन [IAST: *rāmaḥ = rāma + su, prathamā, ekavacana*] of those inflected forms from the Sanskrit text. Morphological analysis is the key to analysis of Sanskrit- a language extremely compact due to its rich morphology. On the surface level, Pāṇini, like his predecessors, seems to have defined rules in a forward looking generative fashion which makes reverse analysis necessary for parsing. At times, the method followed in this R&D looks ad-hoc and un-Pāṇinian, but in reality it derives from the Pāṇini's formulations. Since parsing inflections is the first basic step towards complete analysis, the present work has relevance for any larger system that may evolve in future.

2. Background

Recent advancements in this field for building useful analyzers have been in last 15-20 years. The field has become more applied now than being just concerned with academic research.

2.1 Morphological Analyzers for Non Indian Languages

A morphological analyzer is a very important component for Natural Language Processing (NLP). Therefore morph-analyzers have been developed for most of the common languages. Out of those analyzers the PC-KIMMO is the one development is the field of morphology which is an implementation for microcomputers of a program called KIMMO after its inventor Kimmo Koskenniemi (1983). Other system called CLAWS (Constituent Likelihood Automatic Word-tagging System), a POS tagging software for English text has been continuously developed by University Centre for Computer Corpus Research on Language (UCREL) in early 1980s (Garside,1987). Kenneth R. Beesley (1998) at the Xerox Research centre Europe, chemin de Maupertuis, France has developed an Arabic Morphological Analysis and Generation using Xerox Finite-State Technology (Kenneth R. Beesley, 1998). Basis Technology Corporation has been developed a "Comprehensive Morphological Analysis of Chinese, Japanese and Korean Text". It is very important tool for Chinese, Japanese and Korean (CJK) for critical analysis CJK text such as segmentation, lemmatization, noun decompounding, part-of-speech tagging, sentence boundary detection, and base noun phrase analysis. Natural Language Processing Group, University Autonoma de Madrid has developed the ARIES Natural Language tools make up a lexical platform for the Spanish language. These tools can be integrated into NLP applications. They include: a large Spanish lexicon, lexicon maintenance, access tools and morphological analyzer and generator (Jose, 1997). For a couple of year department of Computer Science of the University of Plovdiv has developed a system for machine dictionaries: a lexical database for the Bulgarian language and a morphological processor (Krushkov, 2000). A French Morphological Analyzer (FMA) has been developed under the American and French Research on the Treasury of the French Language (ARTFL) Project. FMA allows entering one or more French words (lower case only, no punctuation) at the prompt and returns the context-free morphological analysis for each (Lun, 1983).

The Computer-assisted morphological analysis of ancient Greek by University of California is an automated morphological analysis of ancient Greek had its origin in a practical need rather than a theoretical concern for natural language parsing (David, 1973). Latin Parser and Translator have been developed by Adam McLean which a Visual Basic is programming which translating from Latin into English. Turkish Morphological Analyzer has been developed using the two-level transducer technology developed by Xerox. It is implemented using Stuttgart Finite State Transducer Tools (SFST) and uses a lexicon based on (but heavily modified) the word list from Zemberek spell checker. It is distributed under the GNU General Public License (GPL). It can process near about 900 forms per second. This implementation of Turkish uses about 30,000 Turkish root words (Cagri, 2010).

2.2 Morphological Analyzers for Indian Languages

Work related to Sanskrit informatics is carried out by CDAC (Pune), CDAC Hyderabad and Bangalore, Special Centre for Sanskrit Studies, Jawaharlal Nehru University, New Delhi, Vanashtali Vidyapeeth, Rajasthan, Rastriya Sanskrit Vidyapeeth Tirupathi, Lal Bahadur Shastri Rastriya Sanskrit Vidyapeeth, New Delhi, Academy of

Sanskrit Research, Melkote, Mysore, IIT Mumbai and University of Hyderabad. These organizations have been developed basic tools for Sanskrit.

In India Morphological analyzer for Sanskrit, Telugu, Hindi, Marathi, Kannada and Punjabi have been developed by Akshara Bharathi group (2001) at Indian Institute of Technology, Kanpur, India and University of Hyderabad, Hyderabad, India (funded by Ministry of Information Technology, India) and claim for the 95% coverage for Telugu (for arbitrary text in modern standard Telugu) and 88% coverage for Hindi. The Resource Centre for Indian Languages Technology solution Indian Institute Technology Guwahati has developed two morphological analyzers for Assamese and Manipuri (Sirajul, 2004). They have been used both in the spell checker and OCR systems. Both the morphological analyzers use the technique of stemming where in the affixes are either deleted or added to arrive at the root words. Kannada Morphological Analyzer and Generator is student project of Department of CSE, R V College of Engineering, Bangalore, Yahoo! India Software Development Center, Bangalore and National Instruments India, Bangalore. The morphological analyzer and generation tool for the South Indian language of Kannada language using paradigm approach (Shambhavi, 2011). ILTR-Oriya,Utkal University, Vani Vihar, Bhubaneswar, Orissa, India has been developed a Oriya Morphological Analyzer (OMA). The major contents on which the OMA has been built up are Pronoun Morphology (PM), Inflectional Morphology (IM) and Derivational Morphology (DM) (Mohanty, Santi and Das, 2011). Punjabi Morphological Analyzer and Generator (Gill, Lehal and Joshi, 2007) are developed by Advanced Center for Punjabi and are used by one users of Software Informer. The most popular version of this product among our users is 1.0.

Telugu Morphological Analyser has been introduced by Uma Maheshwar Rao, G., Amba P. Kulkarni, Christopher M (2011). Department of Sanskrit Studies, University of Hyderabad has been worked on Sanskrit Morphology (Kulkarni and Shukl, 2009). Special Centre for Sanskrit Studies is deeply involved in the field of computational Sanskrit for developing tools for Sanskrit (Bhadra, Singh, Kumar, Subhash, Agrawal, Chandrasekhar, Mishra and Jha, 2008; Chandra, 2010; Jha, Agrawal, Subhash, Mishra, Mani, Mishra, Bhadra and Singh, 2009). Current Morphological Analyser is rule based which is able to identify Sanskrit verbs forms in Sanskrit texts and analyze nominal inflections.

3. Research Methodology

All rules are stored in a specific format which is accessed by the machine during analysis. The surface analysis of the string is done in the reverse – from the largest chunk to the smallest (8 characters to 1 character).

Sanskrit *subantas* are a combination of stems with affixes in the constrained rule based environment. The largest possible chunks are of 8 characters and the smallest is 1 character long. The rules have been created keeping this in mind. Approximate 1500 rules have been generated for analysis which is stored in the database in the Unicode Devnagari format. A sample of rules is given below-

1. वान्=वत् + सु प्रथमा एकवचन [IAST: vān=vat + su prathamā ekavacana]
2. ◌िद्या=◌िद्या + सु, प्रथमा, एकवचन [IAST: idyā=idyā + su prathamā ekavacana]
3. ◌िद्ये=◌िद्या + औ/औट्, प्रथमा/द्वितीया, द्विवचन [IAST: idye=idyā + au/auṭ prathamā/dvitīyā dvivacana]
4. ◌िद्याः=◌िद्या + जस्/शस् प्रथमा/द्वितीया बहुवचन [IAST: idyāḥ=idyā + jas/śas prathamā/dvitīyā bahuvacana]
5. ◌िद्याम्=◌िद्या + अम् द्वितीया, एकवचन [IAST: idyām=idyā + am dvitīyā ekavacana]
6. ◌ायै=◌ा+ङे चतुर्थी एकवचन [IAST: āyai=ā + ṅe caturthī ekavacana]
7. यै=◌ि + ङे चतुर्थी एकवचन [IAST: yai=i + ṅe caturthī ekavacana]
8. भ्यः= + भ्यस्, चतुर्थी/पञ्चमी, बहुवचन [IAST: bhyaḥ= + bhyas caturthī/pañcamī bahuvacana]
9. ◌ेः=◌ि + ङसि पञ्चमी/षष्ठी एकवचन [IAST: eḥ=i + ṅasi/ṅas pa'camī/ ṣaṣṭhī ekavacana]
10. र्ता=र्तृ+सु, प्रथमा, एकवचन [IAST: rtā=rtṛ + su prathamā ekavacana]

Each rule mentioned above consists of the following processes –

```
Surface analysis, measuring the string
Splitting the string assuming the largest possible residue from the affix to the smallest
Removing the residue and assuming its original form from the rule
Adding a substring if needed to rebuild the stem
```

The rules have been formed as per the following template

```
END_STRING_TO_BE_SEARCHED[1] =
STRING_TO_REPLACE[1]_TO_OBTAIN_THE
STEM + AFFIX, MORPH_INFO
```

The system measures the input string and appropriately splits it starting with 8 characters downward to 1 character and matches each rule as mentioned above to see if the string to the left of '=' symbol matches the split substring. If the match is found, then that part is removed from the input and the substring to the right of '=' symbol is added to obtain the stem. The information to the right of '+' symbol is the affix and pertinent morph information. The displayed result is of the following form –

```
INPUT_STRING = STEM + AFFIX, MORPH
INFO
```

For example if the input is भगवान् [IAST: *bhagavān*] then it will be split from 8 characters downward to 4 when the first rule will be matched and will return the following - वान्=वत् + सु प्रथमा एकवचन [IAST: *vān=t + su prathamā ekavacana*]

Following the rule above, the input भगवान् (*bhagavān*) will lose वान् (*vān*) and will get वत् (*vat*) from the right of the '=' symbol to obtain the rebuilt stem भगवत् (*bhagavat*). The displayed result, as per the rule above wilt us be the following –

भगवान् = भगवत् + सु, प्रथमा एकवचन. Where भगवान् [IAST: *bhagvān*] is morphological forms of Sanskrit, भगवत् [IAST: *bhagvat*] is stem (*prātipadika*), सु [IAST: *su*] is suffix for Nominative case singular number, प्रथमा [IAST: *prathamā*] is case (nominative) and एकवचन[IAST: *ekavacana*] is number information.

This system has been developed for *subanta-padas* (nominal) in Sanskrit text. This system is not able to do analysis of secondary derived (*taddhitānta*, *samāsānta, strīpratyayānta* and *kṛdanta pada*). However, if any *taddhitānta*, *samāsānta, strīpratyayānta* and *kṛdanta pada* occur then the system split nominal suffix only and give the analysis. For example, पठितव्यम् = पठितव्यम् [पठितव्य + अम्, द्वितीया, एकवचन], नीलोत्पलम् = नीलोत्पलम् [नीलोत्पल + अम्, द्वितीया, एकवचन, मूषिका = मूषिका [मूषिका + सु, प्रथमा, एकवचन] [IAST: *paṭhitavyam = paṭhitavyam [paṭhitavya + am, dvitīyā, ekavacana], nīlotpalam = nīlotpalam [nīlotpala + am, dvitīyā, ekavacana, mūṣikā = mūṣikā [mūṣikā + su, prathamā, ekavacana*]

Overall system can understand through fig no. 1.

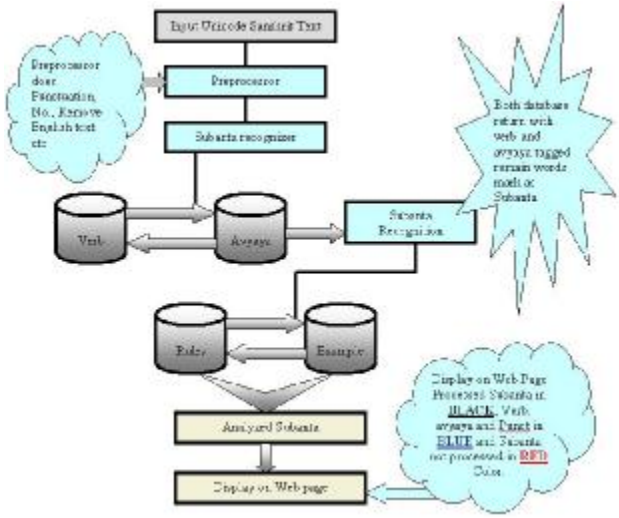


Figure 1: System Module for Sanskrit Nominal Morphology

4. Result and Discussion

It is an attempt to computational analysis of Noun Phrase inflections for Sanskrit by way of Pāṇini's morphological rule system, appropriate databases and example-bases. The system developed is an online system runs on Apache Tomcat platform using Java servlet. One of the major goals of this work is to simplify Sanskrit text for self-reading, understanding, and also for any Machine (Aided) Translation from Sanskrit to other languages. System prints result as output in three colors-Black, Blue and Red - subanta-pada with analysis in Black color non-subanta with tag in Blue color and unanalyzed subanta with '_SUBANTA' tag in Red color. This system has been developed according to Pāṇinian formulation. This method analyses only non-joint (sandhi-rahita) Sanskrit text in Devanagari script. The system fully depends on the rule base, example base and a database of other linguistic resources. It recognizes nominal inflectional morphology by marking all non-nominal inflectional morphology in the sentence by checking the verb and avyaya database. At this point, the system takes Sanskrit text in UTF-8 format (in a text area). Input and output are shown below:

Input:
सम्राजः अशोकस्य जन्मदिनम् आचार्यमाणम् आसीत् । तत्र अधीनाः नृपाः भागं गृहीतवन्तः आसन् । [IAST: *samrājaḥ aśokasya janmadinam ācāryamāṇam āsīt . Tatra adhīnāḥ nṛpāḥ bhāgaṃ gṛhītavantaḥ āsan* .]

Output:
सम्राजः [सम्राज+सु , प्रथमा एकवचन] अशोकस्य [अशोक+ ङस्, षष्ठी, एकवचन] जन्मदिनम् [जन्मदिन (पुल्लिङ्ग) + अम्, द्वितीया, एकवचन] आचार्यमाणम् [आचार्यमन्+अम्, द्वितीया, एकवचन] [आसीत्_VERB] [।_PUNCT] [तत्र_AV] अधीनाः [अधीन (पुल्लिङ्ग) + जस्, प्रथमा, बहुवचन] नृपाः [नृप (पुल्लिङ्ग) + जस्, प्रथमा, बहुवचन] भागं [भाग (पुल्लिङ्ग) + अम्, द्वितीया, एकवचन] गृहीतवन्तः [गृहीतवत्+जस्, प्रथमा, बहुवचन] [आसन्_VERB] [।_PUNCT]
[IAST: *{ samrājaḥ [samrāja+su , prathamā ekavacana] aśokasya [aśoka+ ṅas, ṣaṣṭhī, ekavacana] janmadinam [janmadina (pulliṅga) + am, dvitīyā, ekavacana] ācāryamāṇam [ācāryaman+am, dvitīyā, ekavacana] [āsīt_VERB] [._PUNCT] [tatra_AV] adhīnāḥ [adhīna (pulliṅga) + jas, prathamā, bahuvacana] nṛpāḥ [nṛpa (pulliṅga) + jas, prathamā, bahuvacana] bhāgaḥ [bhāga (pulliṅga) + am, dvitīyā, ekavacana] gṛhītavantaḥ [gṛhītavat+jas, prathamā, bahuvacana] [āsan_VERB] [._PUNCT]*].

This System has following recognition limitations in recognition of *subanta*:

- At present, only 90,000 primary verb forms in the verb database, which are commonly found in Sanskrit literature. Though it is very unlikely that ordinary Sanskrit literature will overshoot this list, yet the system is likely to start processing a verb as *subanta* if not found in the database
- The system wrongly marks a verb with *upasarga* or derived by other derivational process as subanta as it is not found in the verb database. A separate module has been developed for this which tag all Sanskrit verbs. The benefits of that research will also help this system in improving performance

- This work assumes initial sandhi processing, without which, some results may turn out to be incorrect. Therefore the next release of the system includes the capability to preprocess for sandhi joins as well for better results
- Only 519 *avyayas* listed in our avyaya database. The system is likely to start processing an avyaya as *subanta*, if it is not found in *avyaya* database.
- Some forms ending in primary affixes look like *subanta* but they are originally *avyayas*, for example, पठितुम्, गत्त्वा, आदाय, विहस्य [IAST: *paṭhitum, gattvā, ādāya, vihasya* etc.]. In this condition system recognizes incorrectly and processes them as subanta.
- Many *subantas* (basically *śtṛ pratyayānta* in Locative singular) look like verbs these *subantas* wrongly recognizes as verbs, for example, भवति, गच्छति, पठति, चलति [IAST: *bhavati, gacchati, paṭhati, calati* etc.].

The system has the following analysis limitations

- Same forms are available in the dual of nominative and accusative cases, for example, रामौ [IAST: *rāmau*] dual of instrumental, dative and ablative cases, रामाभ्याम [IAST: *rāmābhyām*] plural of dative and ablative cases, रामेभ्यः [IAST: *rāmebhyaḥ*] dual of genitive and locative cases, रामयोः [IAST: *rāmayoḥ*]. In neuter gender as well, the nominative and accusative singular forms may be identical as in the example, पुस्तकम [IAST:*pustakam*] (1-1 and 2-1). In such cases, the system gives all possible results
- Some *kṛdanta* forms (generally *lyap, tumun,* and *ktvā* suffix ending) look like subanta (for example - विहस्य, पठित्वा, गत्वा, पठितुम्, गन्तुम्, नेतुम्, प्रदाय, विहाय etc.[IAST: *vihasya, paṭhitvā, gatvā, paṭhitum, gantum, netum, pradāya, vihāya*]. In such cases, the system may give wrong results
- This system does not have gender information for *prātipadikas*, nor does it attempt to guess the gender.

The system has been tested on 10 separate files (collected from Sanskrit magazines, Pañcatantra story and other resources) and did an analysis of the correct and incorrect results. Evaluation report shows that the accuracy of the system is 85%. It is also found that the system fails to provide 15% correct results where 5% errors are generated due to improper recognition of nominal words, 8% errors for incorrect morphological analysis of nominal words and rest 2% errors occur because of unsuccessful analysis. The present system is available online at http://sanskrit.jnu.ac.in/subanta/rsubanta.jsp.

5. References

Bharati Akshar, Sangal Rajeev, Bendere S M, Kumar Pavan, Aishwarya (2001). Unsupervised Improvement of Morphological Analyzer for Inflectionally Rich Languages, *Proceedings of NLPRS-2001*, Tokyo.

Cagri Coltekin (2010). A Freely Available Morphological Analyzer for Turkish. *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010)*, Valletta, Malta.

Cardona, George (1998). *Pãnini, His Work and its Traditions*, vol i. Motilal Banarasidas, New Delhi.

Chandra Subhash and Jha Girish Nath (2011). *Computer Processing Sanskrit Nominal Inflections: Methods and Implementation*. Cambridge Scholars Publishing (CSP), Newcastle upon Tyne,UK.

Chandra, Subhash (2010). Automatic Nominal Morphological Recognizer and Analyzer for Sanskrit: Method and Implementation. *Language in India*. 10:2.

Choudhury, Sirajul Islam, Singh , Leihaorambam Sarbajit, Borgohain, Samir and Das, Pradip Kumar Das (2004). Morphological Analyzer for Manipuri: Design and Implementation. *Lecture Notes in Computer Science*, Volume 3285/2004, 123-129.

David W. Packard (1973). Computer-assisted morphological analysis of ancient Greek. *Proceedings of the 5th conference on Computational linguistics* - Volume 2.

Garside, R (1987). *The CLAWS Word-tagging System.* In: R. Garside, G. Leech and G. Sampson (eds), The Computational Analysis of English: A Corpus-based Approach. London: Longman.

Jha Girish Nath (2004). The System of Panini. *Language in India*, Volume 4:2.

Jose C.Gonzalez, Jose M. Goni, Amalio Nieto, Antonio Moreno, and Carlos A. Iglesias (1997). The ARIES Toolbox: a continuing R+D Effort. *Proceedings of the International Workshop on Spanish Language Processing Technologies, SNLP-97*, Santa Fe, New Mexico, USA.

Kenneth R. Beesley (1998). Arabic morphology using only finite-state operations. *Proceedings of the Workshop on Computational Approaches to Semitic Languages*, Stroudsburg, PA, USA.

Kimmo Koskenniemi (1983). Two-level Morphology: A General Computational Model for Word form Recognition and Production. *Publication No: 11, Department of General Linguistics*, University of Helsinki.

Krushkov Hr.: Automatic Morphological Processing of Bulgarian Proper Nouns. *Journal TAL*, Vol 41, No. 3.

Kulkarni, Amba and Shukl, Devanand (2009). Sanskrit Morphological Analyser: Some Issues. Festscrift volume of Bh. Krishnamoorty, *Indian Linguistics*, Vol 70, Nos 1-4.

M S Gill, G S Lehal, and S S Joshi (2007), A full-form lexicon based morphological analysis and generation tool for Punjabi. *International Journal of Systemics, Cybernetics and Informatics*, pp. 38-47.

Mishra Sudhir K, Jha Girish Nath (2005). Identifying Verb Inflections in Sanskrit Morphology. *Proceedings of SIMPLE05,* IIT Kharagpur.

Rao, Uma Maheshwar, G., Kulkarni, Amba P., Christopher M. (2011). Telugu Morphological Analyser, *Proceedings of International Telugu Internet Conference*, Milpitas, California, USA

S. Lun. (1983). A Two Level Morphological Analysis of

French, *Texas Linguistic Forum* 22, 13

S. Mohanty, P. K.Santi ,K.P. Das Adhikary (2004) Analysis and Design of Oriya Morphological Analyzer: Some Tests with OriNet. *Proceeding of symposium on Indian Morphology, phonology and Language Engineering*, IIT Kharagpur.

Shambhavi. B. R, Ramakanth Kumar P, Srividya K, Jyothi B J, Spoorti Kundargi, Varsha Shastri G. (2011). Kannada Morphological Analyzer and Generator Using Trie. *International Journal of Computer Science and Network Security*, VOL.11 No.

Sharma, Ram, Nath (2003). *The Astadhyayi of Panini*. Vol I. Munshiram Manoharlal Publishers Pvt. Ltd., New Delhi.

On the Development of Manipuri-Hindi Parallel Corpus

H. Mamata Devi[1], Th. Keat Singh[1], Bindia L[1], Vijay Kumar[2]

[1]Department of Computer Science, Manipur University

[2]TDIL Programme, DIT, GoI, New Delhi

E-mail: mamata_dh @rediffmail.com, thokchom.keat@gmail.com, bindia.laipubam@gmail.com, vkumar@mit.gov.in

Abstract

A sentence aligned parallel corpus is a useful resource for Cross Lingual Information Retrieval, Machine Translation and Computational Linguistics. This paper describes the development of a sentence aligned Manipuri-Hindi parallel corpus and a Corpus Tool. The corpus contains 30,000 pairs of aligned Manipuri-Hindi sentences. The Manipuri sentences were manually translated to its corresponding Hindi sentences. The contents of the corpus were collected from different domain and sub domains so as to represent a balance corpus. The Corpus Tool consists of a Corpus Manager, a Statistical Text Analyzer and a Concordancer. It can work on both mono-lingual and multi-lingual corpus in two different data format, i.e. ISCII and UTF8 and hence, is adaptable to other Indian languages. The paper also describes the functionality and features of the Corpus Tool with the results generated by it from a sample file.

1. Introduction

A *parallel corpus* is a collection of texts, each of which is translated into one or more language other than the original. It contains translation correspondence between the source text and the target text at different levels of the constituents. It is a crucial resource for extracting translation knowledge in machine translation systems (MT). The Manipuri-Hindi Parallel Corpus is the first initiative in the development of Parallel Corpora between Manipuri and Hindi.

To assist both language engineer and linguists in their work of extracting knowledge from corpora a variety of analysis has been carried out (Bharati et al., 2002; Arora et al., 2003; Dash et al., 2002, Kumar et al., 2007). This paper describes the development of a Manipuri-Hindi parallel corpus and a Corpus Tool consisting of a corpus manager, a statistical analyser and a concordance.

The rest of the paper is structured as follows: Section 2 describes the design of parallel corpus, Section 3 describes the design and development of Corpus Tool, which consists of Corpus Manager, Statistical Text Analyzer and concordancer, and finally Section 4 gives conclusion.

2. Parallel Corpus Design

A basic *Manipuri-Hindi Sentence Aligned Parallel* corpus has been developed at the Department of Computer Science, Manipur University, by manually translating Manipuri texts into Hindi. It is a precise and medium-scale corpus containing 30,000 aligned pairs of manually translated sentences.

2.1 Category Determination and Selection of Texts

Thirty thousand pairs of manually translated sentences of *Manipuri-Hindi Parallel* corpus has been selected from three different domains (approx. 10,000 sentence-pairs for each domain) viz. *Generic, Tourism and Health*. Each of these domains is further categorized to cover all possible aspects of Manipuri language, so as to create a balance corpus. In the generic domain, any type of text other than tourism and health domains are included. So it can be considered as general. Texts in the generic domain are divided into *Imaginative text* and *Informative text*. *Imaginative* texts are collected from *Literature* category. The *Literature* category includes novels, short stories, essays, travelogues, folk tales, textbooks, etc. while *Informative* texts are collected from newspapers, official documents and Government and NGO leaflets.

Texts of tourism and health domains are informative. Texts in the tourism domain are categorized into Arts and Culture, Festivals, Food, Handloom and Handicraft, Infrastructure, People, etc. They are collected from newspapers, books, leaflets, etc. A part of the tourism domain is collected from *English-Hindi* parallel corpus of IIT, Bombay as there was no enough material on tourism domain in Manipuri language at the time of creating this corpus. English sentences of the *English-Hindi* parallel corpus of IIT, Bombay are translated into Manipuri sentences thereby making *Manipuri-Hindi* parallel sentences.

The health domain (see table no.1) consists of texts collected from books, journals, newspapers, leaflets, etc. covering various categories such as Physical health, Health

policy, Health education, HIV AIDS, Health programs, Medicinal plants, Mental health, Public health, etc.

Domain	No. of files	No. of sentences
Generic	88	10993
Health	156	9136
Tourism	118	11937
Total	362	32066

Table 1: Domain-wise distribution of sentences

2.2 Sampling of texts

The random sampling method was used to collect the texts of this corpus. Such a sampling ensures the corpus to be balanced and representative of the language.

The text samples are collected randomly from books, magazines, leaflets and newspapers. The books that have the widest reception are award winning books and many such books are also included in the corpus. Magazines, journals and newspapers are more informative about language reception as they may be brought and read by a wider cross-section of the community than books. The daily papers that are selected for this purpose include *The Poknapham, the Sangai Express, the Huyen Lanpao* and *the Matamgi Yaikairol.*

2.3 Selection Procedure and Methods

Text samples normally consist of a continuous stretch of discourse within the whole. A convenient breakpoint, e.g. the end of a section or a chapter was found in all the text samples. Such selection procedure will not fragment the high level discourse units. Samples were taken randomly from the beginning, middle or end of the book. One sample was taken from each selected book, however rarely two or more samples were selected from a few books.

2.4 Writers of Texts

For broader representativeness, writings from both sexes of highly reputed authors, little known writers as well as young writers are included in the corpus. The corpus is broadly heterogonous in nature as the materials are from various sources and disciplines.

The books in this corpus include Sahitya Academy award winning books, the novels in the college syllabi and the text books in High school syllabi because these books have wide reception.

2.5 Translation

The text is manually translated into Hindi by skilled translators. The following guidelines were implemented for Human translation from Manipuri to Hindi:

1. Each Manipuri sentence is translated into a single Hindi sentence, so the corpus is sentence aligned.
2. Manipuri is a morphologically rich Tibeto-Burman language; hence, it may be difficult to find exact translation of Manipuri sentences into Hindi sentences of Indo-Aryan language. In this case, a translation close to the original sentence is preferred and selected.
3. To obtain natural Hindi translation, supplement, deletion, replacement, and paraphrase is made when necessary. If a translation is very long, word order is changed or commas are inserted because the priority is to get the naturalness of Hindi language. Therefore, in certain cases, a Manipuri sentence is translated into two Hindi sentences. In such case, the Manipuri sentence is aligned with the two Hindi sentences.
4. The translated Hindi texts are reviewed by a professor and a lecturer of the Department of Hindi, Manipur University.
5. All the text files are developed using Microsoft notepad and then stored in UTF-8 format.

2.6 Data Entry

Optical Character Recognition (OCR) system for Manipuri script is not available at the time of creating the corpus. Texts were thus entered manually using the Microsoft notepad with the following features:

a. Each Manipuri sentence is given a unique number.
b. The same number is written for the corresponding Hindi sentence to indicate that both Manipuri and Hindi sentences are aligned.
c. If a Manipuri sentence is translated into two Hindi sentences, the sentence number is given to the first Hindi sentence and the following second Hindi sentence will not carry sentence number indicating that these two Hindi (target) sentences correspond to the single Manipuri (source) sentence.
d. No sentence numbers are given to the titles and the sub-titles.

2.7 Cleaning of Corpus

The types of errors found at the time of cleaning the corpus are (i) Insertion error (ii) Deletion error (iii)

Substitution error (iv) Transposition error (v) Run-on error (vi) Split-Word error (vii) Sentence joining error (viii) Repeated Words/Paragraph/Sentence (ix) Punctuation error (x) Grammatical error

Insertion error occurs when at least one unintended character is inserted in the desired word. Deletion error occurs when at least one unintended character is deleted from the desired word. Substitution error occurs when an unintended character is substituted by another character. Transposition error occurs when two adjacent characters are transposed by mistake. Run-on error occurs when there is a space missing between two or more valid words. Split-Word error is opposite of Run-on error. It occurs when some extra space is inserted between parts of a word. This error can be corrected by removing the extra space. In some files, same texts were found to be entered in two different files. In such cases, one of the files is removed. The spellings of the words of the source language (Manipuri) in the corpus are corrected according to spellings of the words written in the book. The spelling variations of the words are preserved.

The errors are corrected at three levels, first by the typists, second by the translators and third by the corpus tool developed during the creation of parallel corpus.

2.8 Corpus Storage Structure

The *Manipuri-Hindi Parallel* corpus consists of text files and their corresponding meta-information files. The meta-information includes details such as title of the book or newspaper or magazine, name of the author or editor, category of text, publisher and distributors, year of publication, language of text, place of creation, etc. The text file contains sampled text data from the above mentioned three domains. These files are arranged and stored in a hierarchical tree structure.

Each text file is named using a convention. The filename gives information such as domain, source of the text, category of the text, sub-category, date of publication, etc. The file naming convention adopted is as follows:

(Domain)_(Source)_(Category)_(Sub-category)_(Date of Publication).

Ex: G_B_Lit_Sc_2008 for a book.

3. Corpus Tool

During the development of the *Manipuri-Hindi Parallel* corpus, a set of tools, collectively referred as *Corpus Tool*, is also developed. The corpus tool can work both in ISCII (Indian Standards, 1992) as well as UTF8 (Unicode Standards) data format. The features provided by this Corpus Tool are: viewing and updating of corpus text files, segregation of Manipuri and Hindi texts and saving them individually, concordance of word or words from a single file or from a set of files, automatic detection of Mono-lingual and Bilingual corpus text data, statistical information on sentences, words, characters, clusters, etc.

The Corpus Tool is implemented using the Java programming language. The Corpus Tool consists of the following modules: Preprocessor, Sinnalemba: Corpus manager, Kup-yengba: Statistical text analyzer, Concomu: Concordancer

The block diagram of the Corpus Tool is shown in Figure 1.

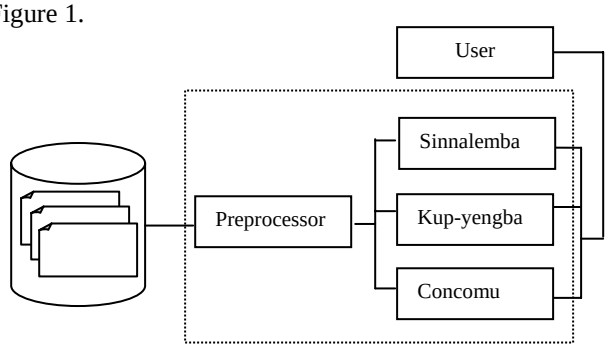


Figure 1: Block diagram of Corpus tool

3.1 Sinnalemba: Corpus Manager

The Corpus Manager provides the facilities for viewing and modification of corpus text files. The features provided by the manager are as follows:

a. Viewing of both *Monolingual* as well as *Parallel* corpus text files.
b. Viewing and exporting of texts in their constituent languages in case of Parallel corpus files.
c. Modification and updating of corpus text files.
d. Identification of sentence mis-alignment in parallel corpus text files.
e. Conversion of ISCII text files into UTF8 text files.

3.2 Kup-yengba: Statistical Text Analyzer

Kup-yengba, the Statistical Text Analyzer processes an input text file or a set of input text files and generates a variety of statistical information. The statistical information generated by this tool from the sample file “Shanti Thiba KhongchAt” is illustrated in tables 2, 3, 4 and in figures 3, 4 and 5.

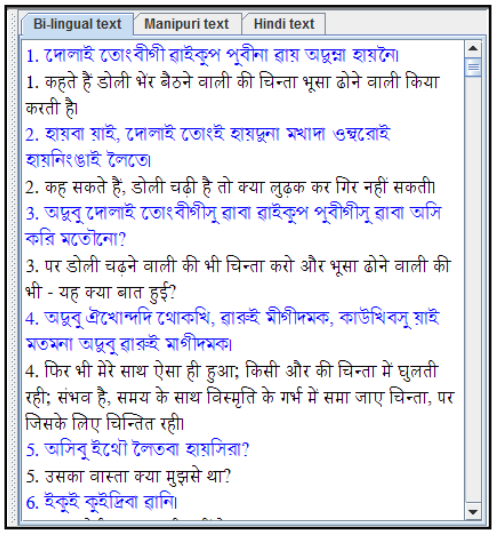


Figure 2: Sinnalemba displaying a corpus text file

Sentence	Manipuri	Hindi
Total	1159	1142
Maximum length	42	50
Minimum length	1	1
Average length	6.93	10.77

Table 2: Sentence statistics

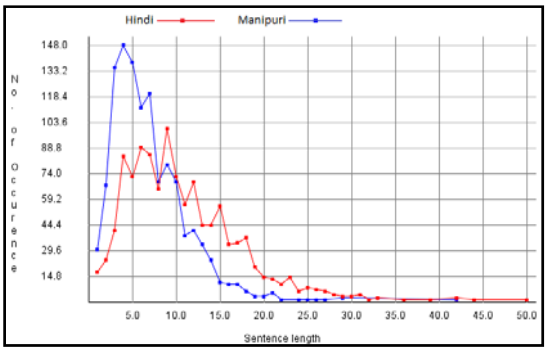


Figure 3: Sentence vs. no. of occurrence

The line graph of sentence length vs. number of occurrence of Manipuri and Hindi sentences in the sample generated by the tool is shown in Figure 3.

Word	Manipuri	Hindi
Total	8034	12307
Unique	4446	2742
Maximum length	18	16
Minimum length	1	1
Average length	5.77	3.58

Table 3: Word statistic

The word statistics of Manipuri and Hindi generated by the tool from the sample text file are shown in Table 3 and Table 4.

Manipuri	Frequency	Hindi	Frequency
মাগী	83	के	406
ঐনা	56	से	265
মানা	48	में	264
অমা	44	की	240
য়াম্না	40	है	222

Table 4: Ten most frequently used words

Table 4 shows the first ten most frequently used Manipuri and Hindi words along with their frequency of occurrences in the sample text file.

A type-token ratio can be used to measure the size of vocabulary of the parallel corpus and to know how many new types will be found as the size of the corpus is increased. Figure 5 shows a type-token ratio vs. token curves of Hindi and Manipuri texts in the sample text generated by Kup-yengba.

3.3 Concomu, the concordancer

Concomu, (Concordancer developed at Manipur University) can find concordance of word or words from monolingual as well as parallel corpus. The word to be concordanced can be selected from a list of unique words or typed explicitly.

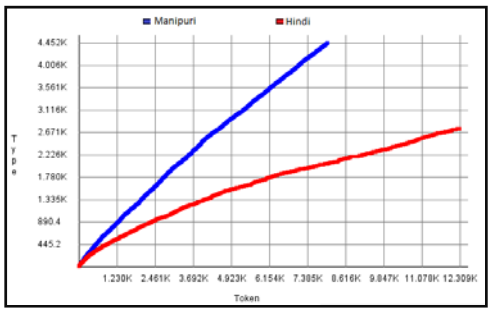


Figure 4: Type vs. Token curve

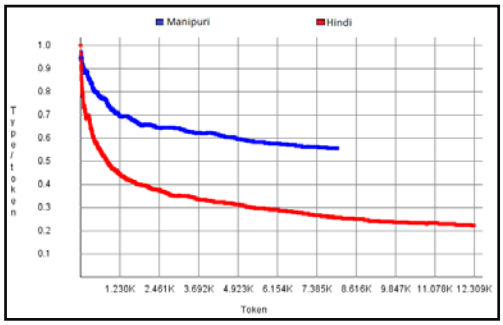


Figure 5: Type-token vs Token graph

LNo.	Filename	Word on left	Keyword	Word on right
10	G_B54_L...	নহাগী লেদর জেকেট	অমা	লীতলি
69	G_B54_L...	ঐখোয়গী লৈকায়দা ইংখোল	অমা	লৈদুনা য়ূম শারে
9	G_B54_L...	শুসিলা কৌবী নুপী	অমা	মশাদা এইদস পক্লে
28	G_B54_L...	চত্পসু নত্তবা নুপী	অমা	এইদস পকলে হায়বদগী
57	G_B54_L...	ফ্রেম চনবা ফোটো	অমা	পুদুনা থোবরুগা হায়-
60	G_B54_L...	ডাক্তর ঐহাক ইন্টভিয়ু	অমা	গৌহাটিদা চত্কদবনি অদুবু
64	G_B54_L...	ঐগী টেকেত এক্সট্রা	অমা	য়াওরি
82	G_B54_L...	চীংনা নাকল	অমা	লোকচপনা নাকল অমা
82	G_B54_L...	অমা লোকচপনা নাকল	অমা	লৈবা থেক খোয়
95	G_B54_L...	অচৌবা পায়বা মী	অমা	শকখঙদবশিংনা নোংমৈনা কাপশিল্লকপদ
28	G_B54_L...	অনী, লৈশাবী	অমা	য়াওই
29	G_B54_L...	দোজ হেন্দুনা পাখং	অমা	রান্ধরে
51	G_B54_L...	. মাগী ইঞ্জিন	অমা	লৈবদুনা মখোয়গী গাড়ী
131	G_B54_L...	. পাকখ্রবা লমপাক	অমা	লৈ

Figure 6: Concomu showing concordance of "অমা"

Figure 6 shows the concordance of a keyword selected from the Index list in KWIC display format. Additional information can be viewed by clicking on it.

4. Conclusion

The sentence aligned Manipuri-Hindi parallel corpus can be used as translation memory, raw and generalize database for machine translation systems for Manipuri to Hindi and Hindi to Manipuri. A rule based spelling variant generators can be developed which can also be used as a preprocessor in Information Retrieval, Cross Language Information Retrieval, and many NLP applications.The Corpus Tool provides important and useful information that would help lexicographers in many ways. Kup-yengba would be of great help for comparison and statistical analysis of corpora. The Concomu, the concordancer can be used for course-material preparation as well as for in-class concordancing tasks. The tool can also be of great help to native language users, researchers, academicians, teachers, students, scholars, language learners, etc.

Acknowledgements

We would like to thank Department of Information Technology, Government of India for the financial support.

References

Arora, K., Arora, S., Gugnani, Shukla, V.N. and Agrawal, S.S. (2003). Gyan Nidhi: A Parallel Corpus for Indian Languages including Nepali, *ITPC*, Kathmandu, Nepal,tdil.mit.gov.in/Jan_issue%202005/12-cdac%20noida.pdf.

Bharati, A., Rao, K. P., Sangal, R. and Bendre, S.M. (2002). Basic Statistical Analysis of Corpus and Cross Comparision among Corpora, *Proceedings of International Conference on Natural Language Processing, ICON-2002*, pp.121-129, ltrc.iiit.net/publications/technical reports/tr022/camera-187.pdf.

Dash, N.S. and Chaudhuri, B.B. (2002). Corpus generation and text processing, *International Journal of Dravidian Linguistics* Volume 31 No. 1, Page 24-44

Indian Standard: Indian Script Code for Information Interchange – ISCII. *Electronic Information and Planning*, Feb, 1992.

Kumar, G.B., Murthy, K.N. and Chaudhuri, B.B. (2007). Statistical Analyses of Telugu Text Corpora, *International Journal of Dravidian Languages (IJDL)*, Vol. 36, No. 2.

Marjorie K.M. Chan (2002). Concordancers and Concordances: Tools for Chinese Language Teaching and Research, *Journal of the Chinese Language Teachers Association*, Volume 37 No.2, Page 1-55.

Unicode Standards: http://www.unicode.org

Annotating Bundeli Corpus Using the BIS POS Tagset

Madhav Gopal
Centre for Linguistics, SLL & CS
Jawaharlal Nehru University, New Delhi
mgopalt@gmail.com

Abstract

Bundeli is an Indo-Aryan language, spoken mainly in the southern districts of Uttar Pradesh and northern districts of Madhya Pradesh. Despite having a large number of native speakers, the language terribly lacks language resources, in terms of corpus, language technology tools, guidelines, standards etc.; and this is partially because of its being a non-scheduled language of India and partially because of lack of an interested research community. Unlike Braj and Awadhi, Bundeli has never been a medium of literary expression, and consequently it lacks sufficient written texts. In this research an attempt is being made to develop its corpus and other computational resources to keep this language at a par with its other counterparts in the region and also to save this from possible extinction. Considering the fact that a digital corpus, after it was tagged at the POS level, could become an indispensable resource for various NLP tasks, machine learning, cognitive linguistics, comparative linguistics and theoretical linguistics, the development of annotated Bundeli corpus is unavoidable. Typologically it is close to Hindi (sparsely described as a variety of Hindi), but it is significantly different from Hindi. This paper introduces the scheme of corpus annotation for this less resourced language, using the BIS POS tagset, a standard tagset for tagging all the Indian languages. The creation of Bundeli corpus will also be discussed briefly.

Keywords: Bundeli, POS tagging, BIS POS tagset, corpus.

1. Introduction

Bundeli or Bundelkhandi derives its name from Bundelkhand, the land of Bundelas, a region extending south of the river Yamuna in the plains, hills and picturesque valleys of central India (Jaiswal, 1962). According to the 2001 Census of the Government of India, there are at least 49 dialects of Hindi with speakers varying from around 11,000 for Khairari to more than 257 million for standard Hindi. In this census Bundeli has been enumerated as a mother tongue of more than 3 million people under the cover term of "Hindi". These 3 million plus people are spread in the southwest marginal districts of Uttar Pradesh and wide northern area of Madhya Pradesh. According to the Grierson's (1968) account Bundeli, a major language in Madhya Pradesh is bounded on the east by the Bagheli dialect of eastern Hindi, on the north and north west by the closely related Kanauji and Braj Bhasha dialects of western Hindi and in Hamirpur by the Tirhari form of the Bagheli spoken on the south bank of the Yamuna, on the south west by various dialects of Rajasthani the most important of which is Malavi and on the south by Marathi. It merges gradually without any district boundary line through some mixed dialects into eastern Hindi, Kanauji, Braj Bhasha, and Rajasthani but there it is merging into Marathi although there are some broken dialects, which are mechanical mixtures of the two languages.

1.1 Linguistic Characteristics of Bundeli

Like other Indo-Aryan languages, Bundeli is an SOV language. However, like other Indian languages word order is not very rigid; this can be said to be relatively free word order. It has number and gender agreement on verb and its subject. For example:

(1) *gadel* *pani* *piyat* *hai*
boy.sg.mas.nom water drink.fin aux.prs.sg.mas

"The boy is drinking water."

(2) *gadyal pani piyat hin*
boy.pl.mas.nom water drink.fin aux.prs.pl.mas
"The boys are drinking water."

Modifier and modified also require agreement in terms of gender, for example *kaluva ghwar* (black horse) vs. *kalui ghori* (black mare). Unlike Sanskrit it has only singular and plural forms. Bundeli has two basic forms of adjectives and nouns, which may be loosely termed as form 1 and form 2. For instance, *ghwar* 'horse', *gay* 'cow', *piyar* 'yellow', *lal* 'red'etc are the form 1 while *ghorwa*, *gaiwa*, *piyarkawa*, *lalkawa* are their respective form 2. The suffixes attached to these words may be considered some kind of affixal particles with different kinds of linguistics functions like specificity, definiteness, focus etc. like in Magahi (Alok, 2010). These particles are not found in Hindi. Jaiswal (1962) in his book has given a beautiful linguistic description of Bundeli.

Socio-politically, there is a high demand in the region to include Bundeli among the scheduled languages of the Indian Constitution. The language is today used in daily lives of the people of the region. It is spoken in small towns and villages by more than 3 million people. In the digital world nothing is available in Bundeli except a multilingual dictionary of Bundeli compiled by the author in a course project under the supervision of Dr. Girish Nath Jha (hosted at http://sanskrit.jnu.ac.in/student_projects/lexicon.jsp?lexicon=bundeli). Due to the influence of Hindi as its being the medium of education, the educated class of Bundeli speakers tends to speak in Hindi instead of Bundeli, so there is a danger of its being extinct in course of time. In such a situation language technology can prove to be a boon for saving this language. The availability of various kinds of information in the language over internet can change the attitude of the people towards a language in which they are more comfortable with.

1.2 Corpus Collection of Bundeli

As we already discussed it is a highly less resource language, its corpus collection is a great challenge before us. We have got a few texts of folk literature, short stories, epics and magazines. Its digital presence is almost nil. We plan to search more Bundeli texts and manuscripts for their digitization. The young generation using computers can be encouraged to write blogs in their mother tongue and also the local writers of the language can be provided with some incentive like establishing some rewards and honors to write in their language.

Parallel corpus can also be developed for Bundeli. There is a big initiative going on for the development of parallel corpora for scheduled languages of India funded by the central government of India (Choudhary and Jha, 2011). Basing the Hindi corpus as source data by the process of translation a useful parallel corpora can easily be created. If not the central government then the state governments of U.P. and M.P. can jointly fund such initiatives of developing corpus of the language.

So far, we have found very useful information about the significant literature of Bundeli. Narmada Prasad Gupta (2001) in his monograph has provided detailed information on Bundeli texts. Among these are folk tales like Rani ki Chaturai (edited by Shiv Sahay Chaturvedi), Nak Chuma Dai Chali (edited by Narmada Prasad Gupta), Bundelkhand ki Gramya Kahaniyan (edited by Shiv Sahay Chaturvedi) etc. Other folk songs include Diwari, Karasdev, Gahanai, Rachhare, Lametara, Devigeet, and Savan Rai. Apart from this the famous epic entitled the Alha Khand (c. 12th Century) is a poetic work in Bundeli which consists of a number of ballads describing the brave acts of two Banaphar Rajput heroes, Alha and Udal. This work has been entirely handed down by oral tradition and presently exists in many recensions, which differ from one another both in language and subject matter (Wikipedia). The digitization of these texts can make a good beginning in the work at hand.

2. POS Tagging and Its Purpose

POS tagging (or morphosyntactic tagging) is the process of assigning to each word in a running text a label which indicates the status of that word within some system of categorizing the words of that language according to their morphological and/or syntactic properties (Hardie,

2003). Part-of-Speech Tagging is the process of assigning to each word the correct tag (part of speech) in the context of the sentence. Putting it another way we can say that POS Tagging is the process of identifying lexical category of a word according to its function in the sentence. While POS tagging is not a new research topic, it is, indeed, a new field as far as Bundeli is concerned. For natural language processing tasks, annotated corpus of a language has a great importance. Annotated corpora serve as an important tool for such well-known NLP tasks as POS- Tagger, Phrase Chunker, Parser, Structural Transfer, Word Sense Disambiguation (WSD), etc. POS tagging is also referred to as morpho-syntactic tagging or annotation especially when features like number, gender, person, tense etc are also added. Unfortunately, so far there have been no such annotated corpora available for Bundeli NLP tasks.

3. The BIS POS Tagset

This is a national standard tagset for Indian languages that has been recently designed under the banner of Bureau of Indian Standards by the Indian Languages Corpora Initiative (ILCI) group. This tagset has 11 categories at the top level. The categories at the top level have further subtype level 1 and subtype level 2. The standard which has been followed in this tagset takes care of the linguistic richness of Indian languages. This is a hierarchical tagset and allows annotation of major categories along with their types an

d subtypes. The hierarchy of tags is directly related to the granularity of linguistic information. Deeper the hierarchy, more fine grained would be the linguistic information. In this framework the granularity of the POS has been kept at a coarser level. Thus, the hierarchy for most POS categories is only of two levels. The maximum depth for the POS tags is three levels so far. Most of the categories of this tagset seem to have been adapted either from the MSRI or the ILMT tagset. For morphological analysis it will take help from Morphological Analyzer, so morpho-syntactic features are not included in the tagset.

4. Annotation of the Bundeli Corpus

To make the data useful for NLP research it is necessary to tag it with a standard tagset. The following description may serve as mini-guidelines for the POS tagging of Bundeli. The emphasis would on specific features of Bundeli which are different from Hindi and can be source of problems in tagging the text.

4.1 Noun (N)

In the BIS scheme, the top level category of noun has four subtypes at level 1: common, proper, verbal and noun location (out of these four Bundeli does not have verbal noun).

4.1.1 Common Noun (NN)

Common nouns in this tagset are the words that belong to the types of common noun (person, place or a thing), abstract noun (emotions, ideas etc), collective noun (group of things, animals, or persons), countable and non-countable nouns. (In the examples in this section, I provide every token tagged; the relevant one has been put in bold.)

व /PRP हमा /PRP **भाई /NN** आय /VM । /PUNC
"He is my brother."

4.1.2 Proper Noun (NNP)

When the word denotes a specific name of a person, place, shop, institution, date, day, month, species, etc., or whatever is considered to be a name would be marked as proper noun. If the word is of some other category, but is used as a proper noun in a context; should be marked as proper noun.

□□□□ **/NNP** हैदराबाद **/NNP** जात /VM है /VAUX । /PUNC
"Mohan is going to Hyderabad."

4.1.3 Nloc (space and time) (NST)

The fourth subtype under the category noun is Nloc. This category has been included to register the distinctive nature of some of the locational nouns which also function as part of complex postpositions. In Bundeli indeclinables like *agé, pache* and *pahile* could be labeled as noun location.

जौं /DMR आदमी /NN **हुंआ /NST** ठाड़ /VM है /VAUX व /PRP हमा /PRP भाई /NN आय /VM । /PUNC

"The man standing over there is my brother."

व /PRP परधान /NN के /PSP **पाछे /PSP** नहीं /NEG चलत /VM । /PUNC
"He does not dog to the Village chief."

4.2 Pronoun (PR)

The pronoun category is divided in 5 subtypes: personal, reflexive, relative, reciprocal, and wh-word.

4.2.1 Personal pronoun (PRP)

Personal pronouns are those which encode person feature in themselves. The words like *main, tay, va, tum* etc. Fall in this category.

तुम /PRP घरै **/NN** जात /VM हौ /VAUX । /PUNC
"You are going home."

4.2.2 Reflexive pronoun (PRF)

A reflexive pronoun is a pronoun that is preceded by the noun or pronoun to which it refers (its antecedent). In Bundeli, the sense of the reflexive pronoun is expresed by the words like *swayam, apan, niji* and *khud*. *swayam* and *khud* are an emphatic reflexives, *niji* a possessive reflexive and *apan* and *khud* serve as both reflexive proper and possessive reflexive.

भइया /NN गाँवै /NN □□□□□□ **/PRF** जात /VM है /VAUX । /PUNC
"The brother is going to the village himself."

4.2.3 Relative pronoun (PRL)

A relative pronoun is a pronoun that links two clauses into a single complex clause.

जेहि /PRL जाँयक /VM होय /VAUX व /PPR चला /VM जाय /VAUX । /PUNC
"Whoever wants to go, please go."

4.2.4 Reciprocal pronoun (PRC)

Reciprocity is expressed by the words like apas and ekdusar. They are always followed by some postposition.

उँ /PPR **आपस /PRC म** /PSP **झगड़ा** /NN **न** /NEG **करिहैं** /VM । /PUNC
"They will not quarrel among themselves."

4.2.5 Wh-word (PRQ)

Wh- Pronouns like *ko* (who), *ka* (what), *kahan* (where) etc. fall in this category.

तुम /PPR कहाँ **/PRQ** जात /VM हौ /VAUX । /PUNC
"Where are you going?"

4.3 Demonstrative (DM)

The next top level category is of demonstrative. Demonstratives have the same form of the pronouns, but distributionally they are different from the pronouns as they are always followed by a noun, adjective or another pronoun. In this category only deictic, relative and wh-word subtypes fall.

4.3.1 Deictic (DMD)

Deictics are mainly personal pronouns. Bundeli doesn't differentiate between demonstrative pronouns and third person pronouns.

व **/DMD** गदेल /NN भूँखा /JJ है /VM । /PUNC
"That child is hungry."

4.3.2 Relative (DMR)

Relative demonstratives are non-distinguishable from relative pronouns, except for that a demonstrative is followed by a noun, pronoun or adjective. In DRL distance attribute is absent.

जौं /DMR आदमी /NN हुंआ /NST ठाड़ /VM है /VAUX व /PRP हमा /PRP भाई /NN आय /VM । /PUNC
"The man standing over there is my brother."

4.3.3 Wh-word (DMQ)

Wh demonstratives are non-distinguishable from wh pronouns, except for that a demonstrative is followed by a noun, pronoun or adjective. The change in the morphological form is not found

कौं /DMQ गदेल /NN आय /CL भूँखा /JJ है /VM ?/PUNC
"Which child is hungry?"

4.4 Verb (V)

The category of verb is somewhat complicated in this framework. It has main and auxiliary divisions under subtype level 1 and finite, non-finite, infinitive and gerund divisions under subtype level 2. Like Hindi verbs, Bundeli verbs are complicated. They often appear in a group of words containing the verb of predication along with light verbs and auxiliaries.

4.4.1 Main (VM)

We apply the main verb tag for all the forms that express the main predication of the sentence. We do not use distinct tags for finite and non-finite as Bundeli, like Hindi, does not have enough information at the word level. Following are some examples:

सुषमा /NNP मदरसा /NNP जाथि **/VM** । /PUNC
जेहि /PRL जाँयक /VINF होय /VAUX व /PPR **चला/VM** जाय /VAUX । /PUNC

4.4.2 Auxiliary (VAUX)

In Bundeli, like in Hindi, the auxiliary verbs concatenate with either the verbal root or verbal inflected forms and they serve to signal distinctions of tense, aspect, mood and voice.

सुषमा /NNP मदरसा /NNP जात /VM रही **/VAUX है /VAUX** । /PUNC

4.5 Adjective

An adjective modifies a noun. Though adjectives are not always followed by nouns in Bundeli, it can be used as a predicate too. The first kind is called an attributive adjective and the second type is called a Predicative adjective. An adjective can function as a noun if not followed by a modified noun; in that case it is called an absolute adjective. When they are used with their modified item, should be tagged as adjectives otherwise as nouns.

व /DMD फूल /NN **सुन्दर /JJ** है /VM । /PUNC
मोटकवा /JJ गदेल /NN गिर /VM परा /VAUX । /PUNC

4.6 Adverb

Only manner adverbs are to be tagged as Adverbs in this framework.

चुप्पेसे /RB निकर /VM जाओ/VAUX । /PUNC
तेजीसे /RB जा /VM । /PUNC

4.7 Postposition

Case relations are expressed by postpositions in Bundeli.

घर /NN **म /PSP** समान /NN धरा /VM है /VAUX । /PUNC व /PPR राम /NNP **क /PSP** कहे /PRQ मारिस /VM ही/VAUX । /PUNC

4.8 Conjunction (CC)

Conjunction is a major category in the tagset and has co-ordinator, subordinator and quotative as subtypes. We have to first enlist the conjunctions in these subcategories and then tag accordingly.

4.8.1 Co-ordinator (CCD)

The conjunctions that join two or more items of equal syntactic importance will be assigned CCD label. The list mainly includes

हम /PPR **अ /CCD** उँ /PPR एकै /QTC परवार /NN क्यार/PSP आहीं /VM । /PUNC

4.8.2 Subordinator (CCS)

The conjunctions that introduce a dependent clause are subordinators.The conjunctions यत्, य ेन , यदि etc. will be labelled as CCS.

व /PRP कहिस /VF कि/**CCS** तुम्हा/PRP काम /NN होइगा /VM । /PUNC

4.9 Particles (RP)

Particles have many a role to play in the language. In the tagset, there are default, classifier, interjection, intensifier and negation subtypes of the Particle category.

4.9.1 Default (RPD)

Words that express emotion are interjections, and also the particles which we use for getting the attention of people.

व /PRP तो/**RPD** चला /VM गा/VAUX । /PUNC
भला **/RPD** एइस े /PRP अचछा /JJ का/PRQ होइ /VM । /PUNC

4.9.2 Classifier (CL)

Unlike Hindi, Bundeli has a couple of classifiers.

भाइ /NN !/PUNC **एक** /QTC **ठो /CL बाल्टी** /NN **दे** /VM । /PUNC
कौं/DMQ गदेल /NN **आय /CL** भूँखा /JJ है /VM ?/PUNC

4.9.3 **Interjection (INJ)**

Words that express emotion are interjections, and also the particles which we use for getting the attention of people.

अरे /INJ !/PUNC **येही** /PRP **का** /PRQ **होइगा** /VM । /PUNC

4.9.4 **Intensifier (INTF)**

Adverbial elements with an intensifying role are intensifiers. They could be both, either positive or negative. *jyadasejyada, adhikaseadhik, kamsekam, Tapar* etc. will fall in this category.

व /PRP टपर **/INTF** के /PSP खाना/NN खाइस /VM ही/VAUX । /PUNC

4.9.5 Negation (NEG)

The indeclinables which are used for negative meaning are treated under this category.

व /PRP खाना/NN न **/NEG** खई /VM । /PUNC

4.10 Quantifiers (QT)

A quantifier is a word which quantifies the noun, i.e., it expresses the noun's definite or indefinite number or amount. The Quantifier category includes general, cardinal, and ordinal subtypes. These terms are equally applicable to both types of quantifiers: written in words (like five, fifth etc.) and in digits (like 5, 5th etc.).

4.10.1 General quantifier (QTF)

This tag is for general kind of quantifiers.

बहुत /QTF गद्याल /NN भूँखे /JJ हैं /VM । /PUNC

4.8.2 Cardinal quantifier (QTC)

The numbers which quantify objects are cardinal quantifiers.

पाँच /QTC गद्याल /NN भूँखे /JJ हैं /VM । /PUNC

4.8.3 Ordinal quantifier (QTO)

Quantifiers that specify the order in which a particular object is placed in a given world are ordinal quantifiers.

पँचवाँ /QTO गदेल /NN भूँखा /JJ है /VM । /PUNC

4.11 Residuals (RD)

Residual as a major category in this tagset has five subtypes; foreign word, symbol, punctuation, unknown and echo words as subtypes.

4.11.1 Foreign (RDF)

In this framework a word is considered a foreign one if it is written in a script other than Devanagari script.

हम /PPR अ /CCD उँ /PPR एकै /QTC **family/RDF** क्यार/PSP आहीं /VM । /PUNC

4.11.2 Symbol (SYM)

The symbol subtype is for symbols like $, %, # etc.

व /DMD फूल /NN १०० /QTC **%/SYM** सुन्दर/JJ है /VM । /PUNC

4.11.3 Punctuation (PUNC)

Only for punctuations like ?, ;, " , । , etc., so other symbols than punctuations will be tagged as Symbol.

सुषमा /NNP मदरसा /NNP जाथि/VM । **/PUNC**

4.11.4 Echo words (ECH)

Echo words are two words that occur together and the second one has no meaning on its own and it cannot occur on its own. It enhances the meaning of the word with which it occurs.

व /PPR घरै /NN **वरै /ECH** न /NEG जई /VM । /PUNC

5. Conclusion

In this paper we have presented a scheme for developing a standard annotated corpus of Bundeli. At experiment level around 3000 words have been manually tagged and we have updated our guidelines as we have explored more data. This Bundeli tagset along with the annotation guidelines and tagged corpus will be available in our

website: http://sanskrit.jnu.ac.in in the near future.

This scheme captures appropriate linguistic information, and also ensures the sharing, interchangeability and reusability of linguistic resources being a common scheme for all the Indian languages. The initiative for tagging Indian languages with the present standard tagset is a promising effort in this direction with the hope that all Indian language corpora annotation programmes will follow these linguistic standards for enriching their linguistic resources. The uniformity in tagging all Indian languages will help in identifying linguistic differences and similarities among Indian languages, and thus facilitate other NLP/linguistic researches.

6. Acknowledgements

I would like to express my sincerest gratefulness to Anoop Kumar and Ruchi Jain for assisting me in getting the Bundeli data. I am also thankful to Hindi annotators Esha, Rachita and Pinkey for explaining me the same scheme for Hindi.

7. References

Alok, Deepak. (2010). *Magahi Noun Particles.* Paper presented in 4th International Students' Conferenceof Linguistics in India (SCONLI-4), Mumbai, India.

Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Choudhury, M., Jha, Girish Nath, Rajendran, S., Saravanan, K., Sobha L., Subbarao, K.V. (2008). A Common Parts-of-Speech Tagset Framework for Indian Languages. In *Proceedings of LREC*, Marrakesh, Morocco, pp. 1331--1337.

Choudhary, Narayan and Jha, Girish N. (2011). Creating multilingual parallel corpora in Indian Languages. In Vetulani, Z. (ed.) *Proceedings of the 5th Language and Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics,* pp. 85-89.

Dash, Niladri sekhar (2011). Principles of Part-of-Speech (POS) Tagging of Indian Language Corpora. In Vetulani, Z. (ed.) *Proceedings of the 5th Language and Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics*, pp.101—105.

Gopal, Madhav, Mishra, Diwakar and Singh, Devi Priyanka (2010). Evaluating Tagsets for Sanskrit. In: Jha, Girish Nath (ed.) *Proceedings of the Fourth International Sanskrit Computational Linguistics Symposium*, LNCS, vol. 6465, pp. 150--161. Springer, Heidelberg.

Gopal, Madhav and Jha, Girish N. (2011). Tagging Sanskrit Corpus Using BIS POS Tagset. In Singh, C., Lehal, G.S., Sengupta, J., Sharma, D.V., and Goyal, V. (eds.) *Proceedings of the International Conference, ICISIL 2011*, Patiala, India, CCIS 139, pp. 191-194, Heidelberg: Springer.

Gupta, N. P. (2001). *Bundeli Lok Sahitya, Parampara aur Itihas,* Madhya Pradesh Adivasi Lok Kala Parishad, Bhopal.

Hardie, Andrew (2003). *Developing a tagset for automated part-of-speech tagging in Urdu.* Presentation at the CL2003 conference, Lancaster University

IIIT-Tagset. *A Parts-of-Speech tagset for Indian Languages*. http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

Jaiswal, M.P. (1962). *A Linguistic Study of Bundeli (A Dialect of Madhyadēśa)*. Leiden.

Jha, Girish N., Gopal, M. and Mishra, D. (2009). Annotating Sanskrit Corpus: adapting IL-POSTS. In: Vetulani, Z. (ed.) *Proceedings of the 4th Language and Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics*, pp. 467-471.

Census of India website – Census 2001, Statement 1, http://censusindia.gov.in/Census_Data_2001/Census_Data_Online/Language/Statement1.htm.

Grierson, G.A. 1908 [Reprint 1968] *Linguistic Survey of India,* Volume- IX Part-I Motilal Banarsi Dass, Delhi.

Developing Sanskrit Corpora based on national standard: Issues and Challenges

Madhav Gopal[1], Girish Nath Jha[2]
[1]Centre for Linguistics, SLL & CS,
J.N.U., New Delhi
[2]Special Centre for Sanskrit Studies,
J.N.U., New Delhi
{mgopalt@gmail.com, girishjha}@gmail.com

Abstract

This paper addresses the issue of the development of annotated corpus of Sanskrit using the Bureau of Indian Standards (BIS) Part of Speech (POS) tagset, a standard tagset designed for tagging Indian languages. The BIS POS tagset is a hierarchical tagset developed under the Bureau of Indian Standards POS committee. The development of Sanskrit corpus is going on by various research groups but standardised tagged data are not easily available. The planned corpora is intended to be put in the public domain for use by the research community . Every effort is being made to make the data more and more useful from computational perspective. We will also discuss the issues and challenges that emerge during the POS tagging.

1. Introduction

Sanskrit has a status in India and Southeast Asia similar to that of Latin and Greek in Europe, and is a central part of Hindu and other indigenous traditions. It is one of the 22 scheduled languages of India and the official language of Uttarakhand – a Himalayan state of India. Sanskrit is one of the well-studied languages of the world, having a sophisticated vocabulary, morphology, literature, research, scholarship and most importantly a rich grammatical tradition. The ancient knowledge of Indian subcontinent is stored in the Sanskrit texts and it needs to be explored today to benefit humanity. Plenty of literature is available on the highly philosophical and ethical subjects, various sciences, linguistic investigations and other academic disciplines. The linguistic investigation of this language was focused on the morphology and phonology, describing it variously. However, its syntax has received least attention by linguists. Mainly, Speijer (1886), Delbruck (1888), and Hock (1991) could be counted among those who have been interested in syntactic structure of this language.

The quality of the POS annotation in a corpus is crucial for the development of POS Tagger (POST). Unfortunately, so for, there have been no such annotated corpora available for Sanskrit NLP tasks. Natural languages are intrinsically very complicated and Sanskrit is not an exception to this. Sanskrit is morphologically and lexically very rich language. It has a variety of words, lexemes, morphemes, and a rich productive mechanism of forming new words. Due to its inflective nature, most of its words are ambiguous and their disambiguation for NLP tasks is a must. A tagged Sanskrit corpus could be used for a wide variety of research like developing POS Taggers, chunkers, parser, Word Sense Disambiguation (WSD) etc.

A tagged text corpus is useful in many ways. More abstract levels of analysis benefit from reliable low-level information, e.g. parts of speech, so a good tagger can serve as a preprocessor. It is also useful for linguistic research for example to help find instances or frequencies of particular constructions in large corpora. Apart from this it is good for stemming in information retrieval (IR), since knowing a word's part of speech can help tell us which morphological affixes it can host. Automatic POS taggers can help in building automatic word-sense disambiguating algorithms; POS taggers are also used in advance ASR language models such as class-based N-grams. POS tagging is also useful for text to speech synthesis and alignment of parallel corpora.

The paper is structured as follows - 1. Digitization of corpus, 2. Validation of Corpus, 3. Designing a tagset for tagging Corpus, 4. Complexity of Sanskrit Texts, 5. Using BIS for POS tagging, 8. Conclusion.

2. POS Tagging and Its Purpose

POS tagging (or morpho-syntactic tagging) is the process of assigning to each word in a running text a label which indicates the status of that word within some system of categorizing the words of that language according to their morphological and/or syntactic properties (Hardie, 2003). POS tagging is the process of assigning to each word the correct tag (part of speech) in the context of the sentence. Putting it another way we can say that POS tagging is the process of identifying lexical category of a word according to its function in the sentence. While POS tagging is not a new research topic, it is, indeed, a new field as far as Sanskrit is concerned. For natural language processing tasks, annotated corpus of a language has a great importance. Annotated corpora serve as an important tool for such well-known NLP tasks as POST, Phrase Chunker, Parser, Structural Transfer, WSD etc. Unfortunately, so far there have been no such annotated corpora available for Sanskrit NLP tasks.

3. The BIS POS tagset

This is a national standard for Indian languages POS that has been recently designed under the banner of BIS by the Indian Languages Corpora Initiative (ILCI) consortium led by Jawaharlal Nehru University, New Delhi (JNU).

This scheme has 11 categories at the top level. The categories at the top level have further subtype level 1 and subtype level 2. The standard which has been followed in this tagset takes care of the linguistic richness of Indian languages. This is a hierarchical scheme and allows annotation of major categories along with their types and subtypes. In this framework, the granularity of the POS has been kept at a coarser level. Thus, the hierarchy for most POS categories is only of two levels. The maximum depth for the POS tags is three levels so far. Most of the categories of this scheme seem to have been adapted either from the Microsoft Research India (MSRI) sponsored Indian Languages POS Tag Set (IL-POSTS) and the Indian Languages Machine Translation (ILMT) tagset. For morphological analysis it will take help from Morphological Analyzer, so the morpho-syntactic features are not included in this scheme. The BIS scheme is comprehensive and extensible and can spawn tagsets for Indian languages based on individual applications. It captures appropriate linguistic information, and also ensures the sharing, interchangeability and reusability of linguistic resources.

The Sanskrit specific tagsets available so far (barring IL-POSTS) are not compatible with other Indian languages and with the exception of the IL-POSTS, all other tagsets are flat and brittle and do not capture the various linguistic information. The IL-POSTS, an appreciable framework, captures various linguistic information in one go and this, according to the designers of the BIS tagset, makes the annotation task complex. And from machine learning perspective also it is not so good. So, the BIS tagset, as a middle path, is suitable for tagging all Indian languages (Gopal and Jha, 2011).

3.1 Utility of the Annotated Corpora

There is a big initiative going on for the development of parallel tagged corpora for 12 major Indian languages including English funded by the government of India (Nainwani et. al, 2011). For the annotation of these corpora, the BIS POS tagset is being used. The Sanskrit corpus annotated with this standard tagset will be very useful, as it will be compatible with the tagged corpora of languages in the vicinity.

4. Complexity of Sanskrit Texts

In this section we discuss the characteristics of the Sanskrit language which are relevant for our purpose. There is a range of complexities in the language, but we selectively discuss those only which affect the design of the system.

4.1 *Sandhi* Phenomenon and the Writing Convention

Sandhi phenomenon is a prime feature of Sanskrit language. It is highly synthetic language and the word boundaries in spoken as well as in written forms are often blurred due to intense concatenation. To identify word boundaries, especially in written form, is not an easy task. There are, actually, two kinds of phenomena involved in *sandhi* (euphonic combinations). They are commonly described as external or *anitya sandhi* and internal or *nitya sandhi*. When we split the external *sandhi*, the components remain usable in the sentence; they do not require inflectional suffixes as they are already endowed with before and after *sandhi*-ing, whereas this is not the case with internal *sandhi*. To identify *nitya* and *anitya sandhi* in Sanskrit there is a famous and well established rule composed in the following *kārikā* (doctrine stated in a verse):

kārikā (1)

saṃhitaikapade nityā nityā dhātūpasargayoḥ ।

nityā samāse vākye tu sā vivakṣāmapekṣate ॥

'In a *pada* (roughly a word) a *sandhi* is mandatory and also in combination of prefixes and *dhātus* (verb roots). It is mandatory in compound constructions also, but in a sentence it requires the intention of the speaker.'

Thus, barring the mandatory or *nitya sandhi* cases, the rest instances of *sandhi* are completely dependent on the speaker or writer whether they wish to combine two or more *padas* or not. And this kind of *sandhi*, viz. external *sandhi* clearly involves combination of two or more *padas*. The idea in the above *kārikā* beautifully works in identifying internal and external *sandhis* in *sandhi*-splitting tasks.

Sandhis, especially the external ones, are serious obstacles to an easy tokenization of Sanskrit texts. The un-preprocessed text has a lot of problems in identifying word boundaries. The external *sandhis* have to be resolved first for smoothly POS tagging and for anaphora resolution also. To unglue each *pada* from euphonic combinations is itself a complex process which requires the identification of external (*anitya*) *sandhis* and internal ones in the text and the *sandhi*-splitting rules available in the grammar. For *sandhi*-splitting a system has been designed at Hyderabad Central University (HCU), JNU and by Gerard Huet at INRIA, Paris. For POS tagging these cases of *anitya sandhi* must be resolved first. To tag a sequence of words with *anitya sandhis* is impossible. After splitting *anitya sandhis* the *kārikā* (1) will look like the following:

saṃhitā ekapade nityā nityā dhatūpasargayoḥ ।
nityā samāse vākye tu sā vivakṣām apekṣate ॥

In this condition of the verse, each *pada* is standing alone, and now can be tagged easily. In our tagging scheme, each *pada* is tagged separately (Jha et al. 2009, and Gopal et al. 2010).

The orthographic system of Sanskrit language is rather complex and this is not only obstacle for anaphora

resolution but also for many other NLP tasks. Due to this complexity of the language sometimes two or more *padas* are concatenated and they seem to be one word, but actually they are not. And to separate a *pada* (a usable unit in a sentence) from other adjacent *padas* is sometimes not an easy task. Some systems have been developed for such tasks. This generally happens with consonant ending words followed by words having vowel in their initial position. These cases as per orthographic rule - *ajjhīnam pareṇa saṃyojyam* – are concatenated, that is, the ending consonant of preceding word hosts the starting vowel of the following word. For POS tagging of the text this concatenation has to be broken up.

4.2 Irregularity of Punctuation Marks

The punctuation marking in Sanskrit texts is bizarre; they do not use any kind of reliable punctuation. Originally, Sanskrit had no punctuation. In the 17th century, Sanskrit and Marathi, both written in the Devanagari script, started using the vertical bar "।" (single *daṇḍa*, also called *virāma* in Hindi)[1] to end a line of prose and double vertical bars "॥" (double *daṇḍa)* in verse (Wikipedia). In unpunctuated texts, the grammatical structure of sentences in classical writing is inferred from the context. Sanskrit by itself contains only "।" to indicate an end of a sentence or half of a verse and "॥" to indicate the end of a complete verse. However, with the advent of the printed books, most punctuation marks used in English are also being used in printed Sanskrit texts. Punctuation sometimes plays a significant role in understanding the text and misplacement of a punctuation mark can reverse the intended meaning. Thus punctuations are important features of a text and they certainly help in following the text.

The implementation of punctuation marks in Sanskrit texts has been rather irregular and complex. These kinds of irregularities are visible in *Panchatantra* too. A sample is being given here from "mitrasaṃprāptikam" (the second section of *Panchatantra*) from Shrishyamacharan Pandey's edition (2006:237):

तत्र च लघुपतनको नाम वायसः प्रतिवसति स्म । सः कदाचित्प्राणयात्रार्थं पुरमुद्दिश्य प्रचलितो यावत्पश्यति, तावत् जालहस्तोऽतिकृष्णतनुः, स्फुटितचरणः, ऊर्ध्वकेशो यमकिङ्करकाकारो नरः संमुखो बभूव ।

'A crow named Laghupatanaka was living there. One day when he was going to the city in quest of food, he saw a man passing before him who was with a net in his hands, dark colour, splay-footed, hair raised up, and looking like the servant of Yama (the god of death).'

In sandhi places it is not necessary that the two words would be concatenated in writing too; they might be written separately too, depending on the nature of the *sandhi*. In the given sample due to *sandhi*, commas have not been put in the two requiring places: जालहस्तोऽतिकृष्णतनुः and ऊर्ध्वकेशो यमकिङ्करकाकारो; because, then, it would have invited the *sandhi*-splitting which the editors of texts do not do. But the computational linguists have to split them up in order to process the language. The *sandhi* free version of the above text would look like the one given below. Herein, the above mentioned two places have been marked with commas. This was possible only when the optional or external *sandhis* were split and the internal *sandhis* kept intact. The concerned places are underlined.

तत्र च लघुपतनकः नाम वायसः प्रतिवसति स्म । सः कदाचित् प्राणयात्रार्थम् पुरम् उद्दिश्य प्रचलितः यावत् पश्यति, तावत् जालहस्तः, अतिकृष्णतनुः, स्फुटितचरणः, ऊर्ध्वकेशः, यमकिङ्करकाकारः नरः संमुखः बभूव ।

This punctuation disorder creates problems in identifying sentence boundaries and clause boundaries which are very crucial for anaphora resolution system design. Hellwig (2007:38) observes:

daṇḍas may be helpful in generating hypotheses about the syntactic structure of a text, but cannot be considered as punctuation marks in a strict sense. This lack has a far reaching effect on any tagging or parsing process applied to a Sanskrit text, because it cannot be guaranteed that all words necessary for a complete analysis are really contained in the text delimited by these marks.

The text Pañcatantra is full of verses. The completion of verse, as stated earlier, is marked by double *daṇḍas* and this marking has been a great problem in tokenizing the text. The double *daṇḍas* are used in headings also and are typical style of Sanskrit text writing. In tokenization the double *daṇḍas* were first replaced by single *daṇḍas* and then the text was tokenized basing the delimitation on the single *daṇḍa.*

4.3 Compounding

Compounding is a very prevalent feature of Sanskrit word formation. Independent words are often compounded with one another, thus producing long strings of linguistic expressions. Even the category of pronouns is sometimes compounded with nouns and participles. In such a compounding process, roots of pronouns, possessive pronouns of all persons, and roots of lexical anaphors (reflexives and reciprocals) undergo compounding. In the process of compounding only the root of the pronoun is left and the case, number, and gender features are removed. These kinds of cases demand different treatment, like the compound processor and then POS tagger which are complex things to be done. For example in (96) *sva-gṛham* is such a compound; it is as a whole a *pada* and in the present situation this would be tagged as common noun with grammatical features. Now, without splitting this compound there is no way to recognize the reflexive *sva*. To enable the system to recognize it, one has to get this compound splitted and then transform the components into padas and then get them POS tagged

[1] There is a common practice in linguistic literature of Sanskrit written in English to refer the "।" sign as *daṇḍa* (see Huet 2009 and Hellwig 2007). In common parlance, however, it is called *virāma* also.

separately. This extra work has to be done manually or automatically to get the *sva* out from the compound. After the POS tagging the system would be able to find its referent.

4.4 Homophony and Syncretism

There are many cases of homophony and syncretism in the language. For example, some third person pronoun, relative and demonstrative forms are homophonous with some conjunctions in the language. These forms include ***tasmāt, tat, yat,*** and ***yena*** which have been used in the text in question. These words serve as linkers and they join the preceding sentence/clause to the following sentence. Their position in a sentence is also fixed as they invariably occur in between two sentences. Their syntactic position cannot be changed. For the systems mainly dependent on the POS tagging of the texts, the POS tagging needs to be done very carefully, as the entire burden is on POS tags. The system be it automatic POS tagger or anaphora resolution system or any other system, must be able to identify the linkers and the pronominals.

5. Corpus Creation

Corpus creation of any language is a challenging task. But for Sanskrit it is not so challenging at it has a huge number of texts of a variety of genres and a lot of them are available in digital form. However, there are many old manuscripts available which are yet to be published. The corpus development of Sanskrit was a joint work of many Indian universities which were part of our consortium. The texts of various genres were collected and digitized in UTF-8 format. Some of these texts are Nalacaritam, AbhiShekanatakam, Urubhangam, Hitopadesha, Pancatantram, Dutghatotkacam, Swapnavasavdatta, Pratijnyayaugandharayan, Avimarakam, Balacaritam, Kumarvijayam, Charudattam, Balaramayan, Agnipurana, Pancharaatra, Sankshipta Ramayan, Meghaduta, Raghuvansha, Abhijnana Shakuntalam, vetalakatha, Sanskritakathakunja and many more. The size of the corpus is approximately 700000 words.

6. Using BIS for POS tagging

Within a limited tagset we have to describe the language unambiguously and consequently we may have to compromise in certain areas. In the following we attempt to apply the tagset to the Sanskrit language.

6.1 Noun (N)

In the BIS scheme, the top level category of noun has four subtypes at level 1: common, proper, verbal and noun location.

6.1.1 Common noun (NN)

Common nouns in this framework are the words that belong to the types of common noun (person, place or a thing), abstract noun (emotions, ideas etc), collective noun (group of things, animals, or persons), countable and non-countable nouns. In the examples below the concerned tokens have been put in bold face.

गौः/NN ग्रामम्/NN स्वयम्/PRF याति/VF ।/PUNC
"The cow is going to the village herself."

6.1.2 Proper noun (NNP)

When the word denotes a specific name of a person, place, shop, institution, date, day, month, species, etc., or whatever is considered to be a name would be marked as proper noun. If the word is of some other category, but is used as a proper noun in a context; should be marked as proper noun.

कल्पना/NNP गोरखपुरम्/NNP गच्छति/VF ।/PUNC
"Kalpana is going to Gorakhpur."

6.1.3 Verbal noun (NNV)

The verbal noun in this framework is for languages such as Tamil and Malayalam. The *kṛdantas* like *āgamanam* and *hasanam* which could have been tagged as verbal noun will be treated as verb non-finite in this framework.

6.1.4 Nloc (space and time) (NST)

The fourth subtype under the category noun is Nloc. This category has been included to register the distinctive nature of some of the locational nouns which also function as part of complex postpositions. In the Sanskrit indeclinables like *agré* and *pūrvam* could be labeled as noun location.

अग्रे/NST साधुपुरुषाः/NN मिलिष्यन्ति/VF । /PUNC तेषाम्/PRP **पूर्वम्/NST** दुष्टाः/NN अपि/RPD मिलितुम्/VINF शक्नुवन्ति/VF ।/PUNC

6.2 Pronoun (PR)

The pronoun category is divided in 5 subtypes: personal, reflexive, relative, reciprocal, and wh-word.

6.2.1 Personal pronouns (PRP)

Personal pronouns are those which encode person feature in themselves. Pronouns like अहम्, त्वम्, भवान्, सः; inclusive pronouns like सर्वम्, उभयम्; indefinite pronouns like कश्चित्, किंस्वित् will fall in this category.

सः/PRP स्यूतम्/NN आदाय/VNG पाठशालाम्/NN गच्छति/VF ।/PUNC
"Taking the bag he is going to the school."

6.2.2 Reflexive pronouns (PRF)

A reflexive pronoun is a pronoun that is preceded by the noun or pronoun to which it refers (its antecedent). In Sanskrit, reflexivity is expressed by the words like आत्मन्, स्वयम्, स्व, स्वकीय, निज, आत्मीय etc. and they are tagged as reflexives.

गौः/NN ग्रामम्/NN **स्वयम्/PRF** याति/VF ।/PUNC
"The cow is going to the village herself."

6.2.3 Relative pronoun (PRL)

A relative pronoun is a pronoun that links two clauses into a single complex clause. In Sanskrit pronoun यत् and its grammatical variants fall in this category.

यस्य/PRL न /NEG अस्ति/VF स्वयम्/PRF प्रज्ञा/NN शास्त्रम्/NN तस्य/PRP करोति/VF किम्/PRQ ?/PUNC
"The one who does not have his own intellect; the scripture does nothing to him."

6.2.4 Reciprocal pronoun (PRC)
Reciprocity is expressed by the words like अन्योन्य, इतरेतर, मिथः and परस्पर. These are generally used in the singular.

बालकाः/NN **परस्परम्/PRC** क्रीडन्ति/VF ।/PUNC
"The boys are playing with each other.

6.2.5 Wh word (PRQ)
Wh- Pronouns like कः, किम्, etc. fall in this category.

तेन/PRP सह /PSP उद्यानम्/NN कः/PRQ गच्छति/VF ?/PUNC
"Who is going to the park with him."

6.3 Demonstrative (DM)
The next top level category is of demonstrative. Demonstratives have the same form as the pronouns, but distributionally they are different from the pronouns as they are always followed by a noun, adjective or another pronoun. In this category only deictic, relative and wh-word subtypes fall.

6.3.1 Deictic demonstrative (DMD)
Deictics are mainly personal pronouns. Sanskrit does not differentiate between demonstrative pronouns and third person pronouns.

तत्/DMD पुस्तकम्/NN लिसायाः/NNP अस्ति/VF ,/PUNC इदम्/DMD पुस्तकम्/NN च /CCD ललितस्य/NNP । /PUNC
"That book is Lisa's, this book is Lalit's."

6.3.2 Relative demonstrative (DMR)
Relative demonstratives are non-distinguishable from relative pronouns, except for that a demonstrative is followed by a noun, pronoun or adjective.

या/DMR बाला/NN तत्र/RB क्रीडति/VF सा/PRP नृत्याङ्गना/NN अपि/RPD अस्ति/VF । /PUNC
"The girl who is playing there is a dancer also."

6.3.3 Wh-word demonstrative (DMQ)
Wh demonstratives are non-distinguishable from wh pronouns, except for that a demonstrative is followed by a noun, pronoun or adjective. The change in the morphological form is not found. E.g.,

कः/DMQ माणवकः/NN वनम्/NN गन्तुम्/VINF इच्छति/VF ?/PUNC
"Which boy wants to go forest?"

6.4 Verbs
Sanskrit verbs are generally classified in three categories: *parasmaipada*, *ātmanepada* and *ubhayapada*. The difference between *parasmaipada* and *ātmanepada* is "for the greater part only a formal one…. Many verbs are used in the *parasmaipada,* but not in the *ātmanepada*, and inversely" (Speijer, 1886). A verb having both kind of forms said to be *ubhayapadi*. In such a kind of verb the *parasmaipada* form denotes that the fruit of the action goes to someone different other than the agent whereas the *ātmanepada* form denotes the fruit of the action goes to the agent herself. There is a further classification into *sakarmaka* (transitive) and *akarmaka* (intransitive) categories. Their usage can also be categorized into three categories: *kartṛvācya*, *karmavācya* and *bhavavācya*. They can again be classified into primary and derivative verbs depending on the type of verbal root. However, these classifications are of no use in the BIS paradigm. One has to understand things according to the framework. One has to apply the tag available in the tagset. It has main and auxiliary divisions under subtype level 1 and finite, non-finite, infinitive and gerund divisions under subtype level 2.

6.4.1 Main verb (VM)
At level 1verb main does not seem to be an appropriate tag in the case of Sanskrit language. However, if anybody wants to use it, it can be utilized in tagging the verbs of present tense when followed by a *sma* and also the *kta* and *ktavat pratyayāntas* when followed by an auxiliary, and in doing so the auxiliary verbs and *sma* have to retain their Auxiliary tags. However, we have not used this tag in tagging the Sanskrit corpus.

6.4.2 Finite (VF)
All the conjugations of the *dhātus* are finite verbs (VF). However, when some of these forms will be used to express the aspectual meaning of the preceding *kṛdanta* will be tagged as auxiliary, as is stated above. In addition, *kta* and *ktavat pratyayāntas* will also be tagged as VF when they are not followed by an auxiliary. As we do not have a separate tag for gerundives (like *kāryam*, *karaṇīyam*, *kartavyam*), VF tag could be applied for them as well.

मोहनः/NNP हैदराबादम्/NNP **गतवान्/VF** । /PUNC सः/PRP मम /PRP भ्राता/NN अस्ति/VF । /PUNC कल्पना/NNP विशाखापत्तनम्/NNP **गच्छति/VF** ।/PUNC

6.4.3 Non-finite (VNF)
kta and *ktavat pratyayāntas* (these are generally described as participles in literature) will be tagged as verb non-finite (VNF) when followed by an auxiliary and other *kṛidantas* like *śatṛ*, *śānac* and *kānac* will also get the same tag.

कल्पना/NNP प्रयागम्/NNP **गच्छन्ती/VNF** तेन/PRP सह /PSP वार्ताम्/NN करिष्यति/VF । //PUNC अधुना/RB सा/PRP सिंगापुरम्/NNP गता/VNF अस्ति/VAUX ।/PUNC

6.4.4 Infinite (VINF)

Sanskrit infinitives are different from other Indian languages and English. They correspond to the infinitive of purpose in English. They are formed by adding *tumun* suffix in the verb root. Only *tumun pratyayāntas* will be tagged as VINF.

सा/PRP जयपुरम्/NNP अपि/RPD **गन्तुम्/VINF** इच्छति/VF ।/PUNC

6.4.5 Gerund (VNG)

In the literature *ktvānta* and *lyabanta* forms are described as gerund. So, these kinds of constructions will be labeled with the gerund (VNG) tag.

कल्पना/NNP गोरखपुरम्/NNP **गत्वा/VNG** प्रयागम्/NNP गमिष्यति/VF । /PUNC तत्र/RB च /CCD स्वकीयाम्/PRF मातरम्/NN **आदाय/VNG** गङ्गास्नानम्/NN करिष्यति/VF ।/PUNC

6.4.6 Auxiliary (VAUX)

In the language some *tiṅantas* (like verbal inflections of *as, ās, sthā, kṛ*, and *bhū* only) that follow a *kṛdanta* to express its (*kṛdanta's*) aspectual meaning, will be tagged with Auxiliary label and the indeclinable *sma* will also get the same tag when follows a verb in present tense and modifies the meaning of the associated verb.

ततः/NST च /CCD पिङ्गलकः/NNP सञ्जीवकेन/NNP सह /PSP सुभाषितगोष्ठीसुखम्/NN अनुभवन्/VNF **आस्ते/VAUX** । /PUNC तस्मिन्/DMD वने/NN भासुरकः/NNP नाम/JJ सिंहः/NN प्रतिवसति/VF **स्म/VAUX** । /PUNC सः/PRP अधुना/RB सिंगापुरम्/NNP गतः/VNF **अस्ति/VAUX** ।/PUNC

6.5 Adjective (JJ)

An adjective modifies a noun. Though adjectives are not always followed by nouns, it can be used as a predicate too. An adjective can function as a noun if not followed by a modified noun; in that case it is called an absolute adjective. Adjectives in Sanskrit are rarely realized as modifiers. Often they occur as substantives. However, there is no dearth of pure adjective usages in the language. When they are used with their modified item, should be tagged as adjectives otherwise as nouns.

धीरोदात्तः/JJ नायकः/NN कलहम्/NN ,/PUNC ईर्ष्याम्/NN च/CCD न/NEG करोति/VF ।/PUNC

6.6 Adverb (RB)

Only manner adverbs are to be tagged as Adverbs in this framework; thus *uccaiḥ* (loudly), *sukhaṃ* (happily) etc. will get the adverb tag.

यानम्/NN **वेगेन/RB** गच्छति/VF ।/PUNC

6.7 Postposition (PSP)

There is a top level category for Postpositions. Sanskrit does not have postposition as such. But we can tag the *upapada* indeclinables as postpositions as they are indeed ambipositions and cause the assignment of a particular *vibhakti* in the concerned nominal.

दुर्गम्/NN **अभितः/PSP** परिखा/NN अस्ति/VF ।/PUNC

6.8 Conjunction (CC)

Conjunction is a major category in the tagset and has co-ordinator, subordinator and quotative as subtypes. We have to first enlist the conjunctions in these subcategories and then tag accordingly.

6.8.1 Co-ordinator (CCD)

The conjunctions that join two or more items of equal syntactic importance, will be assigned CCD label. The list mainly includes च, अपि च, तथा च, तथा.

नायकः/NN खलनायकः/NN **च /CCD** सहरूपेण/UNK गच्छन्ति/VF ।/PUNC

6.8.2 Subordinator (CCS)

The conjunctions that introduce a dependent clause are subordinators. The conjunctions यत्, येन, यदि etc. will be labelled as CCS.

रामः/NNP अकथयत्/VF **यत्/CCS** सः/PRP आपणम्/NN गमिष्यति/VF ।/PUNC

6.8.2.1 Quotative (UT)

The subordinators have a further sub type of 'quotatives'. Quotatives occur in many languages and have the role of conjoining a subordinate clause to the main clause. Therefore, it has been included at the third level of hierarchy within Conjuncts, however, it is left optional to the languages to go to this level of granularity or remain at the higher level keeping only two level hierarchy for Conjuncts.

"/PUNC सर्वे/PRP भवन्तु/VF सुखिनः/NN "/PUNC इति/UT केन/PRQ उक्तम्/VF ?/PUNC

6.9 Particle

Particle is a very important category for Sanskrit, as they play many kinds of role and are of many kinds and used for a number of purposes. Some of the indeclinables described as *avyayas* in the tradition fall in this category. The Sanskrit conjunctions are also described as *avyyas* in tradition, so I put these two categories together here to understand them clearly. In the tagset, there are default, classifier, interjection, intensifier and negation subtypes of the Particle category whereas conjunction has co-ordinator and subordinator subtypes level 1 and quotative subtype level 2.

6.9.1 Default Particle (RPD)

In the current system this would be applied for all avyayas which don't have specific tag in this framework. This will include the *avyaya* types सादृश्यादि, अवधारणम्, and प्रश्नार्थक.

अथ /RPD किम्/PRQ करणीयम्/VF ?/PUNC सुकुमारा/JJ

खलु/RPD इयम्/PRP ?/PUNC अपि/RPD गच्छति/VF सः/PRP ?/PUNC आम्/INJ ,/PUNC सः/PRP एव /RPD गन्तुम्/VINF शक्नोति/VF ।/PUNC

6.9.2 Classifier Particle (CL)
The classifier tag is not applicable for Sanskrit. It can be removed.

6.9.3 Interjection (INJ)
Words that express emotion are interjections, and also the particles which we use for getting the attention of people, e.g., बत, अहो, हा, धिक्, स्वधा, हे, भो etc.

भो/INJ बालकाः/NN !/PUNC यूयम्/PRP किम्/PRQ कुरुथ/VF ?/PUNC

6.9.4 Intensifier (INTF)
Adverbial elements with an intensifying role are intensifiers. They could be both, either positive or negative. भृशम्, पूर्णतया, न्यूनतया, न्यूनातिन्यूनम् etc. will fall in this category.

तम्/PRP अवेक्ष्य/VNG रुरोद/VF सा/PRP **भृशम्/INTF** । /PUNC

6.9.5 Negation (NEG)
The indeclinables which are used for negative meaning are treated under this category.

चिन्ता/NN **मा/NEG** करोतु/VF ।/PUNC सः/PRP भवन्तम्/PRP न /NEG ताडयिष्यति/VF ।/PUNC

6.10 Quantifier (QT)

A quantifier is a word which quantifies the noun, i.e., it expresses the noun's definite or indefinite number or amount e.g., दशम्, तृतीयः, कतिपय, सर्वे. The Quantifier category includes general, cardinal, and ordinal subtypes. These terms are equally applicable to both types of quantifiers: written in words (like five, fifth etc.) and in digits (like 5, 5th etc.).

6.10.1 General quantifier (QTF)
This tag will be used for non numeral quantifiers (perhaps?).

कतिपय/QTF बालकाः/NN श्लोकम्/NN रटन्ति/VF ।/PUNC

6.10.2 Cardinal quantifier (QTC)
The numbers which quantify objects are cardinal quantifiers.

पञ्च/QTC बालकाः/NN POS/RDF tagging/RDF इति/UT कुर्वन्ति/VF ।/PUNC

6.10.3 Ordinal quantifier (QTO)
Quantifiers that specify the order in which a particular object is placed in a given world are ordinal quantifiers.

द्वितीयः/QTO बालकः/NN मेधावी/NN अस्ति/VF ।/PUNC

6.11 Residual (RD)
Residual as a major category in this tagset has five subtypes; foreign word, symbol, punctuation, unknown and echo words as subtypes.

6.11.1 Foreign word (RDF)
In this framework a word is considered a foreign one if it is written in a script other than Devanagari.

पञ्च/QTC मेधाविनः/JJ बालकाः/NN **POS/RDF tagging/RDF** इति/UT कुर्वन्ति/VF ।/PUNC

6.11.2 Symbol (SYM)
The symbol subtype is for symbols like $, %, # etc.

मम /PRP पारगमनपत्रस्य /NN मूल्यम्/NN **$/SYM** 500/QTC अस्ति/VF ।/PUNC

6.11.3 Punctuation (PUNC)
Only for punctuations, so other symbols than punctuations will be tagged as Symbol.

रामः/NNP **,/PUNC** लक्ष्मणः/NNP **,/PUNC** सीता/NNP च /CCD चित्रकूटम्/NNP गच्छन्ति/VF **।/PUNC**

6.11.4 Unknown
If a word does not fit in any of these categories, will be tagged unknown.
नायकः/NN खलनायकः/NN च /CCD **सहरूपेण/UNK** गच्छन्ति/VF ।/PUNC

6.11.5 Echo Words (ECH)
Echo words are two words that occur together and the second one has no meaning on its own and it cannot occur on its own. It enhances the meaning of the word with which it occurs. Such constructions are very rarely used in Sanskrit. And they are written together, so they will collectively get one lexical tag.

7. Conclusion and Future Work

In this paper we have presented a scheme for developing a standard corpus of Sanskrit language. This Sanskrit tagset along with the annotation guidelines (that we have designed for tagging Sanskrit text) and tagged corpus will be available in our website: http://sanskrit.jnu.ac.in in the near future. This initiative, we hope, will enrich Indian NLP and will eliminate the language barriers between different linguistic communities not only in India but across the world. The uniformity in tagging all Indian languages will help in identifying linguistic differences and similarities among Indian languages, and thus facilitate other NLP/linguistic researches.

Moreover, the corpus annotated with this tagset would be more useful as it is tagged by a standard tagset or paradigm. This will ensure the maximal use and sharing of the tagged data. The initiative for tagging Indian languages with the present standard tagset is a promising effort in this direction with the hope that all Indian

language corpora annotation programmes will follow these linguistic standards for enriching their linguistic resources. Thus, Indian NLP may grow faster!

8. Acknowledgements

This work is a part of the Consortium project entitled 'Development of Sanskrit computational tools and Sanskrit-Hindi Machine Translation system' sponsored by the DIT, Government of India.

9. References

Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharyya, P., Choudhury, M., Jha, Girish Nath, Rajendran, S., Saravanan, K., Sobha L., Subbarao, K.V. (2008), A *Common Parts-of-Speech Tagset Framework for Indian Languages*. LREC, Marrakesh, Morocco (2008)

Chandrashekar, R (2007), *Parts-of-Speech Tagging For Sanskrit*. Ph.D. thesis submitted to JNU, New Delhi

Dash, Niladri sekhar (2011) *Principles of Part-of-Speech (POS) Tagging of Indian Language Corpora*, proceedings of 5th LTC, Poznan, Poland Nov25-27, 2011

Gopal, Madhav (2011), *Computational Methods for Anaphora and Cataphora Resolution in the Sanskrit Text Panchatantra*, M. Phil. dissertation submitted to JNU, New Delhi.

Gopal, Madhav, Mishra, Diwakar and Singh, Devi Priyanka (2010), *Evaluating Tagsets for Sanskrit*. In: Jha, Girish Nath (ed.) *Proceedings of the Fourth International Sanskrit Computational Linguistics Symposium*, Springer Verlag, Germany

Gupta, N. P. (2001), *Bundeli Lok Sahitya, Parampara Aur Itihas,* Madhya Pradesh Adivasi Lok Kala Parishad, Bhopal

Hardie, A. (2003). *The Computational Analysis of Morphosyntactic Categories in Urdu*. Ph.D. Thesis submitted to Lancaster University.

Hellwig, O. (2007), *SANSKRITTAGGER, A Stochastic Lexical and POS Tagger for Sanskrit*. In: Huet, G. & Kulkarni A. (eds.) *Proceedings of the First International Symposium on Sanskrit Computational Linguistics*, Springer Verlag, Germany

IIIT-Tagset. *A Parts-of-Speech tagset for Indian Languages.* http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

Jha, Girish Nath, Gopal, Madhav, Mishra, Diwakar (2009), *Annotating Sanskrit Corpus: adapting IL-POSTS*. In: Z. Vetulani (ed.) *Proceedings of the 4th Language and Technology Conference: Human Language Technologies as a challenge for Computer Science and Linguistics*, pp. 467-471

Kale, M.R. (1995), *A Higher Sanskrit Grammar*. MLBD Publishers, New Delhi

Nainwani et. Al. (2011), *Issues in annotating less recourced languages - the case of Hindi from Indian Languages Corpora Initiative*, proceedings of 5th LTC, Poznan, Poland Nov25-27, 2011

Ramkrishnamacharyulu, K.V. (2009), *Annotating Sanskrit Texts Based on Sabdabodha Systems*. In: Kulkarni, A. and Huet G. (eds) *Proceedings of the Third International Symposium on Sanskrit Computational Linguistics*, pp. 26-39, Springer Verlag, Germany

Speijer, J.S. (1886, repr. 2006), *Sanskrit Syntax*. Motilal Banarsidass Pvt. Ltd., New Delhi

Practical Approach For Developing Hindi-Punjabi Parallel Corpus

Ajit Kumar*, Vishal Goyal**
*Multani Mal Modi College, Patiala,
**Department of Computer Science, Punjabi University, Patiala
Email: *ajit8671@gmail.com, **vishal.pup@gmail.com**

Abstract

The backbone of statistical analysis of any languages is the availability of very large corpus. We are working on Statistical Machine Translation System and require very large sentence-aligned parallel corpus. A number of parallel corpora do exist but due to copy right or other legal issues they are not shared by their developers. So we are developing our own Hindi-Punjabi sentence aligned parallel corpus. In this paper we are discussing the various approaches used by different researchers to develop monolingual and parallel corpora with their advantages and limitations. We are also discussing tools and techniques used by us in corpus development. We have automated some part of corpus development and rest of the work is being done manually. We have taken typed text from various sources and aligned it, where ever parallel documents are not available Hindi text is being translated into Punjabi text by using existing machine translation system. In this paper we discussed the dual approach applied by us in the development of Hindi-Punjabi sentence-aligned parallel corpus.

Keywords: corpus, monolingual corpus, parallel corpus, sentence aligner, spell checker, translator.

1. Introduction

A sentence-aligned parallel corpus consists of pair of sentences in two languages, which are exact translation of each other. A parallel corpus consists of documents pair which are more or less exact translation of each other whereas a comparable corpus consists of documents having similar vocabulary.

If X and Y are two text documents in two different languages, consisting of $x_1,x_2,x_3,...,x_n$ and $y_1,y_2,y_3,...,y_n$ sentences respectively then X and Y are said to be sentence-aligned if y_i is the exact translation of x_i for each i.

A lot of work has been done for developing monolingual, bilingual and multilingual corpora. A large number of corpora are available in various languages. [1]Some of the available corpora are **English** (British National Corpus, Brown Corpus, International Corpus of English (ICE), SUSANNE corpus, CHRISTINE corpus, Michigan Corpus of Academic Spoken English (MICASE), Penn-Helsinki Parsed Corpus of Middle English, Corpus of Professional, Spoken American-English (CPSA) , Lancaster Parsed Corpus, Dialogue Diversity Corpus, American National Corpus) , **Chinese** (The Lancaster Corpus of Mandarin Chinese (LCMC)), **Multilingual** (JRC-Acquis, EMILLE/CIIL, OPUS, World Health Organization Computer Assisted Translation page, Searchable Canadian Hansard French-English parallel texts (1986-1993), European Union web server , TELRI CD-ROMs), **Bosnian** (The Oslo Corpus of Bosnian Texts), **Czech** (Parallel Czech-English, Czech National Corpus project: SYN2000), **French**(Association des Bibliophiles Universals, American and French Research on the Treasury of the French Language (ARTFL) database), **German** (COSMAS Corpus, NEGRA Corpus), **Russian** (Russian National Corpus, Library of Russian Internet Libraries), **Slovene** (Slovene-English parallel corpus), **Croatian** (Croatian National Corpus), **Spanish and Portuguese** (TychoBrahe Parsed Corpus of Historical Portuguese, Information about Mark Davies' collection of (mainly historical Spanish and Portuguese, The CUMBRE corpus, The CRATER Spanish corpus, Corpus resources for Portuguese, Folha de S. Paulo newspaper, COMPARA), **Swedish** (Spraakdata).

In Indian languages we have Gyan-Nidhi and EMILLE/CIIL corpus. [2] Gyan-Nidhi corpus consists of text in English and 11 Indian languages (Hindi, Punjabi, Marathi, Bengali, Oriya, Gujarati, Telugu, Tamil, Kannada, Malayalam, and Assamese). It contains digitized 1 million pages containing at least 50,000 pages in each Indian language and English. [3]The EMILLE corpus contains three components i.e. The EMILLE Spoken Corpus, The EMILLE-CIIL Monolingual Written Corpora, The EMILLE Parallel Corpus. The parallel corpus consists of 200,000 words of text in English and accompanying translations in Hindi, Bengali, Punjabi, Gujarati and Urdu.

2. Various approaches used in literature

A number of approaches used in literature for monolingual corpus creation include manual text typing, OCR for text extraction from scanned documents, picking text from web etc. These techniques have their own problems for example manual text typing is time consuming and involves a lot of labor, time and cost but the accuracy of such corpus is much more than other techniques. Use of OCR for text extraction from scanned documents suffers from the limitations of OCR also OCR is not available for many Indian languages. Even if OCR do exists for some languages the accuracy of text produced is not satisfactory and need to be manually checked. The non availability of spell checker aggravates the problem of correcting the corpus. Picking text from web is one of the option that seems to be practical at first sight but relatively small repository of text as compared to English and other European languages, various coding techniques, different fonts and font sizes, use of pdf documents and copy right of text poses a lot of problems.

Paul Baker et. al. (2004) [4] describe the development of EMILLE Corpus which consists of monolingual corpora in fourteen South Asian Languages. The EMILLE Corpus also includes an annotated component, namely, part-of-speech tagged Urdu data, together with twenty written Hindi corpus files annotated to show the nature of demonstrative use in Hindi.

The major approach used for corpus development is manual typing of the documents available in printed, GIF, JPEG or PDF format and use of OCR for extracting text from these documents. Documents available in TTF are converted to Unicode. For parallel corpus creation, documents available in English are translated into other languages. Where ever machine translation is not available, manual translators are employed and translated documents are typed and saved as Unicode formats.

Sunita Arora et. al. (2010) [5] presented an approach for automatic creation of Hindi-Punjabi parallel corpus from comparable Hindi-Punjabi corpus. According to them the comparable documents are processed to find sentence boundaries and sentences are tokenized at word level. Then sentences are aligned using POS tagger and weight assignment techniques.

Pardeep Kumar et. al. (2010) [6] used on-line Hindi-Punjabi machine translation tool available at h2p.learnpunjabi.org to develop Hindi-Punjabi parallel corpus. The authors assume that monolingual Hindi corpus is available and it can be translated to corresponding Punjabi corpus for developing Hindi-Punjabi parallel corpus. We applied this approach for the development of corpus and faced some difficulties such as: only around 100 sentences can be translated at one time, some Hindi words are transliterated rather than translated. Minor spelling mistakes and wrongly translated words appear in output so manual editing of output is required.

Alexandra Antonova et. al. (2011) [7] describe the techniques used in the development of parallel corpus involving Russian-English Language pair. The authors assume the web as source of parallel documents. In this approach comparable documents are collected from the web and processed for sentence wise alignment.

Aasim Ali et. al. (2010) [8] discussed the problems faced by them during the development of English-Urdu parallel corpus. The main problems faced by them include non availability of parallel text in English and Urdu, sentence alignments, punctuation marks and translations issues. They managed to develop a parallel corpus of 6000 lines by manually translating English sentences to Urdu and applied Moses to develop Statistical Machine Translation System with BLEU score of 9.035.

Masood Ghayoomi et. al. (2010) [9] Mentioned some of the common problems experimentally faced by them in developing a corpus for the Persian language from written text and described some rough solutions to fix them. According to them the source of problems could be the Persian script mixed with Arabic script; Persian orthography; the typing style of typists; the control characters' code pages in the operating systems and word processors; having various linguistic style and creativity in the language. They found that before processing the Persian corpus, it is required to preprocess the raw data automatically, manually, and a combination of both by spending energy and time.

3. Closely Related Languages

Hindi and Punjabi are closely related languages. These languages have their origin in Sanskrit [11] and have same sentence structure i.e. Subject verb object (*Karta, Karam, Kria).* In both languages, sentence is comprised of Subject and Predicate. Both languages have eight numbers of basic elements called *Kaaraka (Karta, Karam, Karan, Sampardaan, Apadaan, Sambandh, Adhikaran, Sambodhan)* which by combining with each other create a sentence.

The general sequence for transitive Sentence is *Karta*

Karam Kria and for intransitive sentence is *Karta Kriya*. In both languages the relation between *Kaarka's* are shown by postpositions. Total eight parts-of-speech are recognized in both Hindi and Punjabi. Beside this, both have same types of Nouns, Genders, Number, Persons, Tenses and Cases. The close relationship between Hindi and Punjabi is established by a study by Josan and Lehal [10] and Goyal V. et. al.[11]. The authors have also concluded that Hindi and Punjabi are closely related languages from the machine translation point of view also.These languages are written from left to right.

4. Challenges

Development of bilingual sentence-aligned parallel corpus is a big challenge. Some of the techniques used in literature are manual typing and aligning, translating monolingual corpus into target language, Picking parallel documents and aligning, taking comparable text from web and aligning.

Manual creation of corpus is a challenging and laborious task. So the use of tools for the automatic creation of corpus is one approach that is being used in the literature. The algorithm used for automatic line alignment is based on the sentence length base such as Gale and Church. The practical application of this algorithm on Hindi-Punjabi language does not give satisfactory results.

The availability of parallel text in Hindi-Punjabi is very rare. Also where ever we found the parallel documents these are not sentence-wise translation of each other.

Limited vocabulary of existing translation software is another challenge. When we use the translation approach most of the Hindi words are transliterated rather than translated.

Fig. 1: Raw Hindi Document

Gyan-Nidhi corpus is not clean corpus. Text documents contain unrecognizable characters.

The text pages contain header and footer information which is different in Hindi and Punjabi documents.

Reference documents details are given as footer on some pages and their place is different in Hindi and Punjabi documents.

The text documents contain Sanskrit Shalokas and Poems which are not translated in their Punjabi versions.

Spelling mistakes in Hindi-Punjabi documents is another big challenge.

Gyan-Nidhi corpus contain comparable documents, which are not sentences-wise translation of Hindi to Punjabi rather it is theme translation where multiple sentences from Hindi is translated into one Punjabi sentence or one Hindi sentence is translated to multiple Punjabi sentences, which make the process of sentence alignment even more challenging.

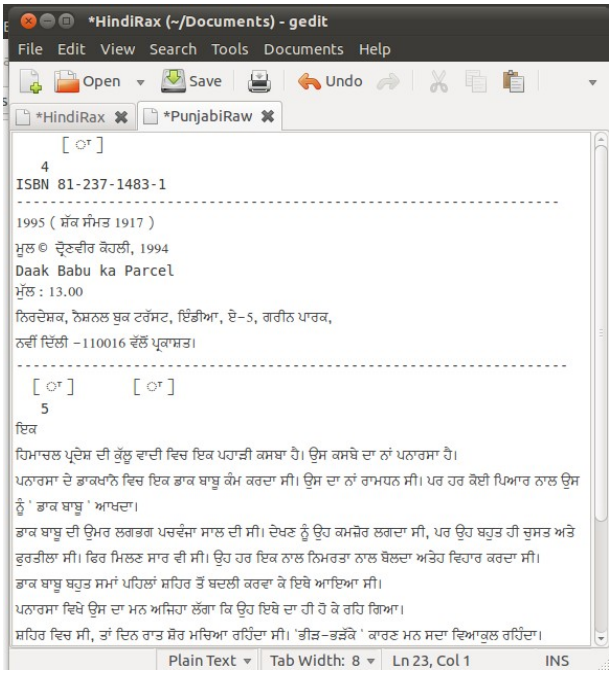


Fig. 2: Raw Punjabi Document

Hindi text available in on-line newspapers contain spelling mistakes, different spelling for same word.

Limited vocabulary of Hindi-Punjabi machine Translation System results in poor quality of translated text.

These challenges need to be addressed for successful development of sentence-aligned parallel corpus.

5. Approach used

We are using machine assisted approach to develop

Hindi-Punjabi sentence-aligned parallel corpus.

On the one side we are using parallel Hindi-Punjabi documents from Gyan-Nidhi corpus, on the other side we are using existing Hindi-Punjabi machine translation system to translate Hindi text taken from www.bbc.co.uk/hindi and other Hindi newspaper available on-line.

HINDI - PUNJABI PARALLEL CORPUS			
DOMAIN	**Number of Lines**	**Hindi words**	**Punjabi Words**
History, Literature, Religion	69894	930778	931068
Politics, Crime, Business & Sports news	33203	434708	437360
Total Size	**103097**	**1365486**	**1368428**

Table 1: Size of Corpus

Using these techniques we have successfully created a corpus of 1,03,097 lines each containing 13,65,486 words in Hindi and 13,68,428 words in Punjabi.

5.1 Aligning Documents from Gyan-Nidhi corpus

The source of text in Gyan-Nidhi corpus is National Book Trust India, Sahitya Akademi, Navjivan Publishing House, Publications Division and SABDA, Pondicherry.

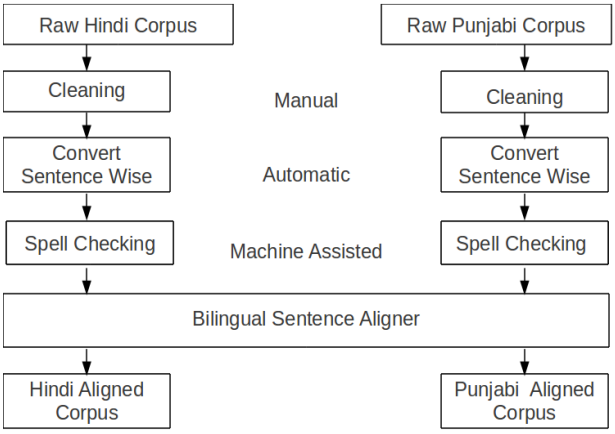


Fig. 3: Sentence Aligning Parallel Documents

The Hindi-Punjabi text taken from these sources contains a lot of noise in the form of unwanted characters, header, footer, poems, and Sanskrit shalokas etc. which are manually removed from the text documents. The cleaned documents are tokenized at sentence level to arrange document sentences-wise. Hindi spell checker which is available as add-on with Open-Office is used to correct the spelling of Hindi document. Punjabi spell checker which is available as a part of Akhar (Punjabi Typing software developed by Dr. G. S. Lehal) is used to correct the spelling of Punjabi Document.

After correcting spelling Bilingual Sentence Aligner [12] developed by Microsoft Corporation available at http://research.microsoft.com/en-us/downloads is used to align sentences from cleaned sentence-wise documents. The accuracy of this alignment tool is more that 98% on parallel documents as calculated by us during manual checking of sentence aligned parallel documents. The final aligned documents are manually corrected by removing discrepancies detected during manual checking.

5.2 Using Hindi-Punjabi Translation Software

The text from on-line Hindi news papers such as www.bhaskar.com, www.bbc.co.uk/hindi, and www.jagran.com, is saved to a document in Unicode format. The spell checker is used to correct the spelling mistakes and tokenized at sentence level to arrange the document sentences-wise. For spell checking we used the spell check facility of Open-Office on Ubuntu platform and for Tokenizing the document at sentence level we have written our own Python script to detect sentence boundaries and split the sentences by inserting new line character.

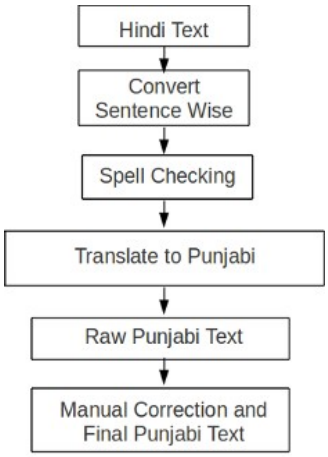


Fig. 4: Corpus Development Using Translation Software

The [13] Hindi-Punjabi Translation Software available at http://h2p.learnpunjabi.org developed by Dr. Vishal Goyal and Dr. G.S. Lehal is used to translate these sentences to Punjabi language. The said translation software is based on hybrid approach. It consists of combination of word for word translation approach and rule based approach. The translation accuracy of this software is 94%. The translated Punjabi text is aligned but contains some spelling mistakes and wrongly translated

words. So the produced output is manually corrected and finalized. Some part of sentence aligned Hindi-Punjabi parallel corpus is given:

1. जब अपनी उदासीनता के कारण उसने मेरी दशा बिगडते देखी तो अपना सारा शोक भूल गयी।

jab apnī udāsīntā kē kāraṇ usnē mērī dashā bigḍatē dēkhī tō apnā sārā shōk bhūl gayī.

1. ਜਦੋਂ ਆਪਣੀ ਉਦਾਸੀਨਤਾ ਦੇ ਕਾਰਨ ਉਸਨੇ ਮੇਰੀ ਹਾਲਤ ਵਿਗੜਦੇ ਵੇਖੀ ਤਾਂ ਆਪਣ ਸਾਰਾ ਸੋਗ ਭੁੱਲ ਗਈ ।

jadōṃ āpṇī udāsīntā dē kāran usnē mērī hālat vigaṛdē vēkhī tāṃ āpaṇ sārā sōg bhull gaī .

2. आज मैंने उसे अपने आभूषण पहनकर मुस्कराते हुंए देखा तो मेरी आत्मा पुलकित हो उठी ।

āj mainnē usē apnē ābhūshaṇ pahnakar muskrātē huṃē dēkhā tō mērī ātmā pulkit hō uṭhī .

2. ਅੱਜ ਮੈਂ ਉਸਨੂੰ ਆਪਣੇ ਗਹਿਣੇ ਪਾਕੇ ਮੁਸਕਰਾਉਂਦੇ ਹੋਏ ਵੇਖਿਆ ਤਾਂ ਮੇਰੀ ਆਤਮਾ ਖੁਸ਼ ਹੋ ਉੱਠੀ ।

ajj maiṃ usnūṃ āpaṇē gahiṇē pākē muskarāundē hōē vēkhiā tāṃ mērī ātmā khush hō uṭṭhī .

3. मुझे ऐसा मालूम हो रहा है कि वह स्वर्ग की देवी है

mujhē aisā mālūm hō rahā hai ki vah svarg kī dēvī hai

3. ਮੈਨੂੰ ਅਜਿਹਾ ਪਤਾ ਹੋ ਰਿਹਾ ਹੈ ਕਿ ਉਹ ਸਵਰਗ ਦੀ ਦੇਵੀ ਹੈ

mainūṃ ajihā patā hō rihā hai ki uh savarag dī dēvī hai

4. मुझे जैसे दुर्बल प्राणी की रक्षा करने भेजी गयी है।

mujhē jaisē durbal prāṇī kī rakshā karnē bhējī gayī hai.

4. ਮੇਰੇ ਵਰਗੇ ਕਮਜੋਰ ਪ੍ਰਾਣੀ ਦੀ ਰੱਖਿਆ ਕਰਣ ਭੇਜੀ ਗਈ ਹੈ ।

mērē vargē kamjōr prāṇī dī rakkhiā karaṇ bhējī gaī hai .

5. मैने उसे कठोर शब्द कहे,

mainē usē kaṭhōr shabd kahē,

5. ਮੈਂ ਉਸਨੂੰ ਕਠੋਰ ਸ਼ਬਦ ਕਹੇ ,

maiṃ usnūṃ kaṭhōr shabad kahē ,

6. Conclusion

We have observed that a lot of work has been done on corpus development in English language and most of the corpora are freely available for download on the web for researchers. But in Indian languages we are struggling at individual level to work from scratches and reinventing the wheel. The language translators spell checkers and OCRs are being language dependent, need to be enhanced for all languages to ease the work of corpus creation. The tools used by us such as Akhar, spell checker, Tokenizer and translation software are language specific and can't be used for other languages. If such tools do exist for other languages the same technique may be applied for corpus creation in other languages as well. We will make our corpus available on the web after completion.

7. References

[1] http://nlp.stanford.edu/links/statnlp.html#MT, as accessed on 16 Feb. 2012.

[2] http://www.cdacnoida.in/snlp/digital_library/gyan_nidhi.asp, as accessed on 16 Feb. 2012.

[3] EMILLE Corpus Documentation available on www.emille.lancs.ac.uk/manual.pdf, as accessed on 17 Feb. 2012.

[4] Paul Baker et al, (2004). *Corpus Linguistics and South Asian Languages: Corpus Creation and Tool Development*, Literary and Linguistic Computing, Vol. 19, No. 4 C ALLC, pp. 509-524

[5] Sunita Arora et al, (2010). *Creation of Parallel Corpus from Comparable Corpus*, Proceedings of ASCNT, CDAC, Noida, India, pp. 77 – 83

[6] Pardeep Kumar, Vishal Goyal, (2010). *Development of Hindi-Punjabi Parallel Corpus Using Existing Hindi-Punjabi Machine Translation System and Using Sentence Alignments*, International Journal of Computer Applications (0975 – 8887), Volume 5– No.9, pp.15-19

[7] Alexandra Antonova, Alexey Misyurev, (2011). *Building a Web-based parallel corpus and filtering out machine-translated text*, Proceedings of the 4th Workshop on Building and Using Comparable Corpora, Portland, Oregon, pp. 136–144,

[8] Aasim Ali et al, (2010). *Development of Parallel Corpus and English to Urdu Statistical Machine Translation*, International Journal of Engineering & Technology IJET-IJENS, Vol: 10 pp.30-33

[9] Masood Ghayoomi et al, (2010). *A Study of Corpus Development for Persian*, International Journal on Asian Language Processing, Vol. 20, pp.17-33

[10] G. S. Josan, G. S. Lehal, (2008), *A Punjabi to Hindi Machine Translation System*, Coling: Companion volume: Posters and Demonstrations, Manchester, UK, pp. 157-160

[11] Vishal Goyal, G. S. Lehal, (2008), *Comparative Study of Hindi and Punjabi Language Scripts*, Napalese Linguistics, Journal of the Linguistics Society of Nepal, Volume 23, pp 67-82

[12] http://research.microsoft.com/en-us/downloads/aafd5dcf-4dcc-49b2-8a22-f7055113e656/

[13] Vishal Goyal, G. S. Lehal, (2010), *Web Based Hindi to Punjabi Machine Translation System*, Journal of Emerging Technologies in Web Intelligence, Vol. 2 No. 2, pp. 148-151

Challenges in Developing Named Entity Recognition System for Sanskrit

Sachin Kumar[1], Girish Nath Jha[1], Sobha Lalitha Devi[2]

[1] Jawaharlal Nehru University, New Delhi (India)

[2] AU-KBC Research Centre, Anna University, Chennai (India)

sachinjnu@gmail.com, girishjha@gmail.com, sobhanair@yahoo.com

Abstract

In this paper, we discuss several challenges in developing Named Entity Recognition (NER) system for Sanskrit. The paper also presents a framework for a Name Entity Tagset for Sanskrit (NETS), suitability and process-flow of the hybrid approach for Sanskrit NER system. The paper mainly focuses on the issues related to developing NER system for Indian languages (ILs) especially Sanskrit. It also talks about intermediate results and its analysis based on a Machine Learning (ML) algorithm i.e. Conditional Random Field (CRF) applied on *Pañcatantra*.

Keywords: NER, Sanskrit NE, NETS, Hybrid Approach, Machine Learning, CRF, *Pañcatantra*

1. Introduction

Named Entity Recognition (NER) is a subtask of Information Extraction (IE) that seeks to locate the entities in Natural Language text and specify their types. It categorizes entities in a text into predefined categories such as the names of persons, designations, organizations, locations, abbreviations, expression of time, date, measure etc. NER mainly includes two tasks: identification and classification of named entities (NEs). In Identification, word or phrase is identified as named entity (NE) while in classification these identified NEs are assigned a category. For example in the sentence- *astyatra dharātale vardhamānaṁ nāma nagaram. Tatra dantilo nāma nānābhāṇḍapatiḥ sakalapuranāyakaḥ prativasati sma,* a Sanskrit NER system would identify *vardhamānaṁ* as the location name and *dantilo* as the person name. Bogers[1] (2005) says that NER is intuitively simple for humans as they determine that a word represents a name if it starts with an uppercase letter (for Roman based languages) or it is already observed or from contextual clues, although all these methods may have their limitations. But the question is how we teach this to computer.

2. Usefulness of the system

As a large amount of information in any text is simply the entities it refers, identifying and classifying them automatically can help in processing text. It is an important component in Natural Language Processing of Sanskrit for computational purposes like Machine Translation, developing Sanskrit search engine, automatic indexing, document classification and text summarization etc. It will also be helpful in many cross-linguistic applications as it will be relevant for other ILs.

3. NER for Indian Languages

Several attempts are being made to develop NER systems for various ILs using different techniques. The workshop on NERSSEAL, held in 2008 at IIIT Hyderabad, is a first major event in this direction. In the NERSSEAL-08 shared task, Saha *et al*. (2008) use a hybrid approach that includes Maximum Entropy Model (MaxEnt), language specific rules, gazetteers and context patterns to build NER systems for the five languages- Bengali, Hindi, Telugu, Oriya and Urdu. The system recognizes 12 classes of NEs and report 65.96%, 65.13%, 18.74%, 44.65% and 35.47% F-measure respectively for these languages. Gali *et al.* (2008) uses CRF model, followed by post-processing which involves some heuristics or rules for the same five languages and report F-measure of 40.63%, 50.06%, 40.94%, 39.04%, and 43.46% respectively. Ekbal *et. al* (2008) use CRF model and report 59.39%, 33.12%, 04.75%, 28.71% and 35.52% respectively. Praveen P *et al.* (2008) use the hybrid approach of Hidden Markov Model (HMM) and CRF. The lexical F-measure of HMM based system is 39.77%, 46.84%, 46.58%, 45.84%, 44.73% respectively while the performance of CRF system

[1] Bogers, T. (2005). *Named Entity Recognition*, [Online:Web] Accessed on 1 Feb. 2012 URL: http://ilk.uvt.nl/~antalb/textmining/tm5.pdf.

for these languages is 35.71%, 40.49%, 45.62%, 36.76%, 38.25% respectively.

For Sanskrit, this is a starting point as no NER system for Sanskrit is available till date. This work aims to take insights from the NER systems developed for ILs. The work also takes insights from the theoretical and philosophical discussion related to word and name in Sanskrit knowledge tradition. The major texts related to this are *Nirukta* of Yāska, *Amarkośa* of Amarsiṁha, *Māhābhāṣya* of Patañjali, *Vākyapadīya* of Bhartṛhari, *Vaiyākaraṇbhūṣaṇasāraḥ* of Kauṇḍabhaṭṭa, *Paramalaghumañjūṣā* of Nāgeśa Bhaṭṭa, *Arthasaṅgrahaḥ* of Laugākṣibhāskara, *Nyāyasiddhāntmuktāvalī* of Viśvanāthapañcānana. The work also aims to compare this discussion with the work in the western analytical philosophy related to proper names and definite description.

4. Definition of Sanskrit Named Entity

The task of defining the NE involves the practical considerations. The application for which the task is being done influences the definition of NEs. These applications may be translation, Question-answering, Information Extraction etc. The latter is a broader area among these applications which include NEs related to 'who', 'what', 'where', and 'when'. The present paper mainly focuses on deciding NE for Sanskrit, keeping in mind the translation purpose.

The main criterion that is generally followed to determine NEs is that an NE should be a rigid designator. Rigid designation[2], as defined by the western analytical philosopher Kripke, refers to the named object which denotes the same thing in all the possible world. For example 'Atal Bihari Vajpayee' refers to the same person in every possible world in which 'Atal Bihari Vajpayee' exists. This criterion for determining the NEs is also relaxed for practical reasons. This relaxed criterion includes NEs which denotes definite description. Definite description[3] refers to the named object which may not mean the same thing in all the possible worlds. Here context plays a big role in determining the referent of an NE. For example, Atal Bihari Vajpayee is a rigid designator and hence an NE. But 'the person who was Prime Minister of India in 1996' is not a rigid designator and it may refer to Atal Bihari Vajpayee, H.D. Deve Gowda and P.V. Narasimha Rao. But context of the text can help us to determine its referent. So this is also considered as an NE.

In the context of defining Sanskrit NE and developing tag set for Sanskrit NER, both of these kinds of expressions are taken as NEs. Besides the conventional names in Sanskrit text, lots of derivatively conventional names are also found. In this context the classification of words into 4 ways[4] in *Nyāya* (logic) text *Nyāyasiddhāntmuktāvalī* written by *Vishvanāpañcānan* can be described. In this text words are divided into four categories, viz some are derivatives (*yaugika*), some are conventional (*rūḍha*), some are derivatively conventional (*yogarūḍha*) and some are derivative and conventional (*yaugikarūḍha*).

These four types of words can be compared to the above mentioned two criteria which are followed in deciding NE for several languages.The following are some examples from Sanskrit text to illustrate the above criteria:

arjunaḥ: rigid designator (conventional (*ruḍha*)), single individual, not translatable

pāṇḍavaḥ: rigid designator (conventional (*ruḍha*)), group, not translatable

kuruvaṁśa: rigid designator (conventional (*ruḍha*)), family name, partly translatable

saubhdraḥ: definite description (derivatively conventional (*yogaruḍha*)) , singular, translatable/not translatable

draupadeyāḥ: definite description derivatively conventional (*yogaruḍha*), group, translatable/not translatable

4.1 NER Tag set

NEs have been classified in different ways in the NER tasks. For example, MUC-6 (1995)[5], in which the task of NE was defined for the first time, has a total of seven NE tags classified into three kinds: ENAMEX (person, location and organization), TIMEX (date, time) and NUMEX (money, percentage). In the CoNLL 2003 shared task[6], only four types of NEs are identified: person, location, organization and miscellaneous. NERSSEAL-08 shared task[7] performed

[2] http://en.wikipedia.org/wiki/Rigid_designator

[3] http://en.wikipedia.org/wiki/Definite_description

[4] śaktaṁ padam. Taccaturvidham. Kvacid yaugikaṁ, kvadidrūḍhaṁ, kvacid yogarūḍhaṁ, kvacid yaugikarūḍham. In Viśvanāthapañcānana (2011). *Nyāyasiddhāntmuktāvalī* Caukhambā Surabhāratī Prakāśana, Vārāṇasī. (śabdakhaṇḍa p. 100)

[5] http://www.cnts.ua.ac.be/conll2003/ner/

[6] http://cs.nyu.edu/cs/faculty/grishman/muc6.html

[7] http://ltrc.iiit.ac.in/ner-ssea-08/index.cgi?topic=3

NE recognition on 12 types of NEs – person, designation, organization, abbreviation, brand, title-person, title-object, location, time, number, measure and terms. Another tagset developed at AUKBC, Chennai (Vijayakrishna R *et al.* 2008), has a hierarchical scheme consisting of 106 tags for tourism domain and is divided into the same three main categories

As no NER tagset for Sanskrit is available, the main focus of this paper is to explore and report the Named Entity Tagset for Sanskrit (NETS). The paper takes insights from the above tagsets and looks to explore the characteristics of Sanskrit NEs. For this task, various Sanskrit texts (covering around 5 lakh tokens) of different genre, time and style of Sanskrit writing has been observed and the following types of NEs are found. These NEs have been classified into the same three main categories of ENAMEX, TIMEX and NUMEX. This NETS does not aim to be either coarse-grained or fine-grained at this juncture as this division of two types of tagset is influenced by the purpose for which the task is being done. But the NETS described in this paper just aims to explore the types of NEs found in the Sanskrit texts. In the future, with an increased coverage of Sanskrit texts, extended NETS is possible. The details of the proposed NETS are as follows:

1 ENAMEX
- 1.1 Person
  - 1.1.1 Individual
    - 1.1.1.1 Family_Name
    - 1.1.1.2 Title
    - 1.1.1.3 Designation
  - 1.1.2 Group
- 1.2 Organization
- 1.3 Location
  - 1.3.1 Place
    - 1.3.1.1 Village
    - 1.3.1.2 City
    - 1.3.1.3 District
    - 1.3.1.4 State
    - 1.3.1.5 Nation
  - 1.3.2 Landscape
  - 1.3.3 Water_Bodies
- 1.4 Character
  - 1.4.1 Bird_Character
  - 1.4.2 Animal_Character
- 1.5 Literary_Work
  - 1.5.1 Book_Name
  - 1.5.2 Part_of_the_book
  - 1.5.3 Story_in_the _text
- 1.6 Entertainment
  - 1.6.1 Music
  - 1.6.2 Sports
- 1.7 Direction
- 1.8 Fort
- 1.9 Celestial_body
- 1.10 Philosophical_School
- 1.11 Language
- 1.12 Temple
- 1.13 Technical_words
  - 1.13.1 Marriage_name
  - 1.13.2 Ritual_name
  - 1.13.3Word_For_Auspicious_Beginning
- 1.14 Abbreviation

2. TIMEX
- 2.1 Time
  - 2.1.1 Day
  - 2.1.2 Date
  - 2.1.3 Month
  - 2.1.4 Year
  - 2.1.5 Period
  - 2.1.6 Festive Day

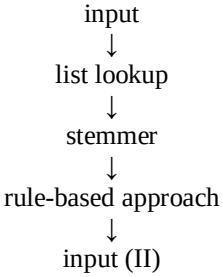
3. NUMEX
- 3.1 money

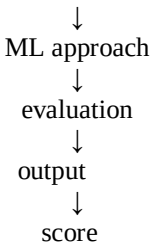
5. Hybrid approach for the system

The system proposes to adopt a hybrid approach of rule-base and ML, the process flow of which is described in the following section. However, as an intermediate result at present, the results based on ML approach only are described. For the rule-based approach, Sanskrit rules and heuristics are being studied and various lexica are being developed and adapted. The rule-based approach will include POS and morphological analysis, heuristics for locations, numerical and time expressions. The lexica will include lists of first name, middle name, last name, day name, month name, location name, verb, indeclinables, pronoun, designation and titles, currency etc. In the ML approach, the system learns from annotated corpus and applies it on unseen text.

This hybrid approach aims to get the advantages of both the approaches. As Sanskrit is a rule-based language, so the hybrid approach will try to capture the definiteness in identifying NEs through various ways of rule-based approach and at the same time since Sanskrit is not a fixed-order language, the probabilistic characteristic of the ML algorithm will be helpful in finding the patterns from the data and thus using various features to identify the NEs.

5.1 Process flow





5.1.1 Description of the process flow

The input for the system will be Sanskrit text file in *devanāgarī* Utf-8. The pre-processor will arrange the text into a row and column format as ML algorithm accepts the input in this way. The data will be arranged into three columns as tokens, POS tagging and chunking. After pre-processing, the system will apply list look up method to identify and mark NEs found as it is in the input. It will also identify and mark the verb, *avyaya* (indeclinable) and pronoun as these words need not to be considered in NEs identification. The system will fill this information as fourth column in the text. After this, the next step is to segment the words (still not marked) into base (*pratipadika)* and affix *(vibhakti)*. The next step of applying list look up method again is to identify the NEs appearing in base form (*pratipadika)* which could not be found earlier. This step will also apply some heuristics on the input. The result of this step will be in the fourth column. Now this input with result of rule-based approach becomes input for ML approach. This is referred as input (ii) in the process flow. The next step will be to apply CRF algorithm on this input. This algorithm will assign the NEs based on template file created by training file and features file. The result of this level will be put in the fifth column. At the next level of processing, the results of both the approaches will be combined and compared. To compare the result, there are certain parameters i.e. if an NE is marked same by both the approaches, then this is to be considered as final. If an NE is marked by one of the two approaches, then this is to be taken as final and if an NE is marked differently by both the approaches, then result of rule based approach is final. After this comparison, the final results of both the approaches to be put as the sixth column. After applying both the processes, the system will give output which will comprise the original input with three columns, fourth column of rule based approach, fifth of ML approach, and sixth column as final result. This output will be analyzed on two bases i.e. how many NEs identified are correct (precision) and how many NEs are totally identified (recall).

6. Challenges in developing NER for Indian Languages (ILs) in general and for Sanskrit in Particular

There are several issues related to develop NER system for ILs in general and Sanskrit in particular. These issues are listed as follows:

6.1 General issues in developing NER for ILs

- no graphical mark like capitalization
- low POS tagging accuracy for nouns
- lack of standardization and spelling variation
- NE with different case marker/postpositions
- lack of large gazetteers
- insufficiency of labeled data for ML
- role of context in deciding an NE
- different styles of writing abbreviation

6.2 Issues related to Sanskrit NER

All the above are the issues which make NER task difficult for ILs. But the case of Sanskrit is more complex and it is the primary focus of this paper. This complexity is due to the grammatical and contextual characteristics of Sanskrit. The issues faced in developing Sanskrit NER system can be divided into following two categories. Though some of the issues mentioned in these categories may also be common to other ILs, but their frequency and complexity appeared high in Sanskrit text.

(i) Various ways of referring an NE in a text:

- Appearance of synonymous expression of an NE
- descriptive expression describing an NE
- personified natural objects as NE
- frequent availability of the name of an individual as a son or daughter of someone
- name of an individual based on relation with someone or based on any characteristics or activity
- names of animal character and bird character
- limitation of certain clue words like *bho, nāma, deśa, pradeśa* etc. as these words are attached with non-NEs also
- large number of mythological names

(ii) Various ways of appearing an NE in a text

- longness of the string in which the NE is appearing
- different types of combination like *sandhi*, *samāsa*, or simple concatenation of NEs with other words in the text
- large string of Sanskrit words in which NE may appear at any place
- continuously lots of NEs and also a long gap between NEs
- relatively free word order so partial chunking possible

But the advantage with Sanskrit NER is that it will have many cross-linguistic applications as it will be relevant for other ILs.

7. Intermediate result and its analysis

As an intermediate result, a ML algorithm i.e. CRF was tested on *Pañcatantra* which is a blend of prose and poetry. The reasons to select *Pañcatantra* as a model text were that it has different types of names in a single text. This adds complexity to the NE identification as it has names for persons, animal characters and bird characters which make it difficult in selecting template features for ML. Due to its prose-poetry blended language, it proved to be a worthwhile text to test the computational complexity in assigning chunks in both kinds of text.

7.1 Training data

To train ML algorithm, a training file of 22900 tokens was created. This training file has 518 NEs classified according to 10 types of NEs mentioned above. To train the algorithm, a template file with 5 features was also created. The training file has three columns i.e. token, POS tagset and NE type respectively. The empty line marks the sentence boundary and 0 in the third column marks non_NE. Table 1 shows the Sample Training data:

अस्ति	KP	0
दाक्षिणात्ये	NV	0
जनपदे	NP	0
महिलारोप्यम्	NS	NE_Location_Place
नाम	NP	0
नगरम्	NP	0
।	PUNC	0
तत्र	A	0
सकलार्थिकल्पद्रुमः	NV	0
प्रवरमुकुटमणिमरीचिमञ्जरीचयचर्चितचरणयुगलः	NV	0
सकलकलापारंगतः	NV	0
अमरशक्तिः	NS	NE_Person_Individual
नाम	NP	0
राजा	NP	0
बभूव	KP	0
।	PUNC	0

Table1: Training data sample

7.2 Named Entity Tagset used

The system uses the following 10 types of NEs. These entities spawn from the broad framework of NETS.

- NE_Person_Individual
- NE_Person_Title
- NE_Animal_Character
- NE_Bird_Character
- NE_Book
- NE_Part_Of_the_Book
- NE_Story_in_the _text
- NE_Location_Place
- NE_WaterBody
- NE_Technical_Words

7.3 Testing data

To test the algorithm, a test file of 21067 tokens was created. This file has total 322 NEs out of which machine could identify 166. Total time taken to read the training file and test file was 40 sec. and 110 sec. respectively on a laptop with the processor of 2.00GHz and RAM of 2.00 GB. The testing file has three columns i.e. token, POS category and NE type respectively. The first two columns are given and the third is filled by machine. The empty line marks the sentence boundary and 0 in the third column marks non_NE.as given in table 2:

(	PUNC	0
वल्मीकोदरस्थसर्पकथा	NS	NE_Story_in_the _text
)	PUNC	0
अस्ति	KP	0
कस्मिंश्चित्	SNV	0
नगरे	NP	0
देवशक्तिः	NS	NE_Person_Individual
नाम	NP	0
राजा	NP	0
।	PUNC	0

Table 2: Testing data sample

7.4 Analysis of the result

- Total NEs 322
- Identified by machine 166
- Machine looked only those 10 types of NEs in the test file which were in training file
- Right 87 (the precision is 52%, recall is 27% and F-measure is 35.54)
- Out of 79 wrong results:
    - Most of 45 wrong results can be corrected by analyzing the patterns in the training file and embedding more features in template file
    - 22 mistakes in the words preceded by *nāma*,
    - 12 mistakes in the word followed by *bho*,
    - Most of these 34 (22+12) results are identified wrong as the preceding word of *nāma* and the following word of *bho* in

the data is appeared either as person name, bird character or animal character.

8. Conclusions

The paper has described the process flow of a Sanskrit NER system based on a hybrid approach. It also describes the definition of Sanskrit NE and a tagset for Sanskrit NER. An example of NER and the intermediate result based on ML algorithm are also presented. The paper has also listed the challenges in developing the system. NER for ILs has been one of the most challenging aspects of computational processing. Therefore, we hope that this work and the associated algorithm will be useful in Sanskrit NER and also helpful for NER of other ILs in many ways.

9. Acknowledgements

The authors duly acknowledge the DIT sponsored consortium project SHMT for allowing us to use the POS tagged corpora. We also thank the Computational Linguistics R&D at Jawaharlal Nehru University for allowing to use all of their tagged text corpora.

10. References

Ekbal, A., Haque, R., Das, A., Poka, V., and Bandyopadhyay, S. (2008). Language Independent Named Entity Recognition in Indian Languages. In *Proceedings of the workshop on NER for South and South East Asian Languages*. IIIT Hyderabad, pp. 33-40.

Gali, K., Surana, H., Vaidya, A., Shishtla, P., and Sharma, D.M. (2008). Aggregating Machine Learning and Rule Based Heuristics for Named Entity Recognition. In *Proceedings of the workshop on NER for South and South East Asian Languages*. IIIT Hyderabad, pp. 25-32.

Praveen P., and Ravi Kiran V. (2008) Hybrid Named Entity Recognition System for South and South East Asian Languages. In *Proceedings of the workshop on NER for South and South East Asian Languages*. IIIT Hyderabad, pp. 83-88.

Saha, S.K., Chatterji, S., Dandapat, S., Sarkar, S., and Mitra, P. (2008). A Hybrid Named Entity Recognition System for South and South East Asian Languages. In *Proceedings of the workshop on NER for South and South East Asian Languages*. IIIT Hyderabad, pp. 17-24.

Vijayakrishna, R. and Sobha, L. (2008). Domain Focused Named Entity Recognizer for Tamil Using Conditional Random Fields. In *Proceedings of the workshop on NER for South and South East Asian Languages*. IIIT Hyderabad, pp. 59-67.

Viṣṇuśarma (2002). *Pañcatantram.* Motilal Banarasidasa, New Delhi.

Exploratory Analysis of Punjabi Tones in relation to orthographic characters: A Case Study

Swaran Lata, Swati Arora[1]
[1]Jawaharlal Nehru University, New Delhi 110067
W3C India Office, New Delhi 110003

Abstract

Punjabi is known tonal language of Indo-Aryan family with a very wide linguistic coverage across two countries. Bimodal one Male and one Female Data Repository and analysis for Indo-Aryan languages especially Punjabi is presently non-existent. Tone is the inherent feature of Punjabi due to the presence of 5 tonal characters. These Tonal Characters are represented by the corresponding aspirated or un-aspirated and voiced or unvoiced forms and also marked with high rising tone / ó / and low rising tone /ò/ on top of the accompanying vowel. Gender specific samples of recorded data from native speakers is being used for the analysis of Punjabi Tones in relation to Orthographic characters i.e. ਭ(bh) /p/ with a tone, ਧ (dh) /t/ with a tone, ਢ (dh)/ʈ/ with a tone, ਘ (gh)/k/ with a tone, and ਝ (Jh)/tʃ/ with a tone. These orthographic characters have lost their aspiration and have become tonal over a period of time. This analysis will help in the Speech Technology Research.

1. Introduction

Punjabi is a member of Indo-Aryan Language family and it is mainly spoken by inhabitants of north western India and north eastern Pakistan. It is a descendant of the Shauraseni language, which was the chief language of medieval northern India. According to the ethnologies 2005 estimate, there 88 million native speakers of the Punjabi language, which makes it approximately the 10th most widely spoken language in the world and according to 2001 census of India, there are 29, 102, 477 Punjabi speakers in India.

Punjabi is one of the few Indo-Aryan languages which has developed tonal contrast. Unlike mandarin, Punjabi doesn't have contour tones. Punjabi has three phonemically distinct tones i.e. high-tone /Ó/, low-tone /Ò/ and mid-tone /ō/ and there are five tonal characters i.e. ਭ(bh) /p/ with a tone, ਧ (dh) /t/ with a tone, ਢ (dh) /ʈ/ with a tone, ਘ (gh) /k/ with a tone, and ਝ (Jh) /tʃ/ with a tone. Another salient feature of Punjabi is the occurrence of double consonants, i.e. geminates which results in phonemic stress which can occur on both initial or final syllable.

PLS is a standard of World-Wide Web Consortium (W3C) and its current version is PLS 1.0 (2008) produced by Voice Browser Working Group of W3C [1]. The PLS has been designed with a goal to have inter-operable specifications of pronunciation information which can be used for speech technology development. It provides a mapping between the words or short phrases, their written representations and their pronunciation especially for use by speech engines. The PLS data will be prepared in the XML format for specific language using the base line PLS specification of W3C. The phonetic nuances of specific languages need to be captured in the PLS which requires detailed study of phonological features. Tonal study of Punjabi is crucial for proper phonological representation in PLS.

The present study attempts to characterize the tonal parameters in terms of intensity analysis, minimum and maximum pitch of falling and rising tones present in Punjabi language. PLS development is the foundation of future voice interface of web browsers as represented below in figure – 1.

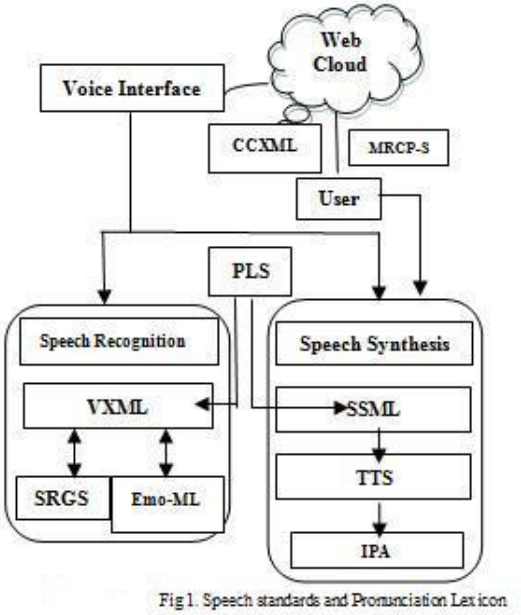


Fig 1. Speech standards and Pronunciation Lexicon

The tones in Punjabi arise as a reinterpretation of different consonant series in terms of pitch. Current

literature survey reveals that, the analytical study of the tonal features of Punjabi Language has not been

investigated, however some initial attempts of linguistic characteristics analysis of Punjabi language has been made by Baart [2] and Karamat [3]et.al. The findings of tonal characteristics would be helpful for research in contextual tonal variations and their contributions to global F_o contour as well as linguistic requirements. The paper is organized as follows. Section – 2 describes the methodology adopted for preparation and recording of data. Section – 3 covers the annotation of the recorded data and its spectrographic analysis for identifying the nature of tones. Section – 4 presents the experimental analysis of Punjabi tones associated with these 5 tonal characters while these occur in initial medium & final position in the words. Finally the conclusion is drawn at section 5.

2. Preparation & Recording of Data

There are 5 Tonal consonants in Punjabi [4], namely ਭ(bh) /p/, ਧ (dh) /t/ ਢ (dh)/ʈ/, ਘ (gh)/k/, and ਝ (Jh)/tʃ/; four of these are stops and the last one is affricate. These are represented phonetically (IPA) by corresponding aspirated/ un-aspirated and voiced/unvoiced forms and also marked with high rising tone / ◌́ / and low rising tone /◌̀/ on top of the accompanying vowel.For the present study, a word list of these characters has been compiled.

2.1 Word List

Five words each for these 5 Tonal Characters with their occurrence in initial, medial and final position have been selected as described in annexure.

Data Recording Specifications

For the recoding of the Punjabi speech data, standardized procedure for speech corpora development based on the ITU recommendations has been adopted. The recording of the annexed word list has been done in standard recording environment having SNR>=45dB. The recording format is 16 bit, PCM, Mono and sampling rat is 48 KHz and the speech rate is medium with neutral emotion.

2.3 Recording of Data

The numbers of informants used for the speech data of the present analysis are 1 male 1 female between 25-35 age group.The orthographic representations of words involving tonal consonants in initial, medial & final position are recorded by these informants.

3. Annotation and spectrographic analysis of Data

3.1 Annotation

The annotation of the recorded speech data has been carried out using the PRAAT software package since it is a very flexible tool to do speech analysis**.** The spectrographic

analysis of all the male & female samples was carried out and phoneme level annotation was done. Punjabi tone is normally realized over two syllables. Of these, first is the most important and is called onset syllable. During transcription also, tone is represented on this syllable. [Gill,1986]. This syllable was identified from the annotated data of words with the tonal character occurring in the initial, medial and final position.

The PRAAT tool has also been used for analysis of the F_0 contour and the slope of the contour over the pitch area of the vowel accompanying that syllable. In the F_0 analysis the ESPS signal processing software has been used. The ESPS *epochs* program was used to mark every vocal cycle in the sentences.

3.2 Effect of tone vs F_0

F_0 contour is one of the major acoustical manifestations of supra-segmental features such as tone, pitch accent and intonation. These features are critical to perceptual naturalness of human speech. In tonal languages, different tones are associated with the same syllable pronunciation to express different lexical meanings, which needs to be captured in the PLS.

The principal phonetic feature of tone is found in the domain of pitch. Its primary acoustic correlate is fundamental frequency F_0. It is very difficult to use pure tone as a stimuli, as the ear cannot perceptualize tone distinction while processing the acoustic signal hence the fundament frequency F_0 analysis can help in the study of Tones.

4. Experimental Analysis of Punjabi Tones

4.1 Intensity Graphs:

For the systematic analysis of Punjabi Tones, Intensity Graphs have been first studied using the PRAAT tool. The intensity graph for the word ਸਾਂਝ /sa̋d͡ʒ/ (partnership) is depicted in Fig 2 below.

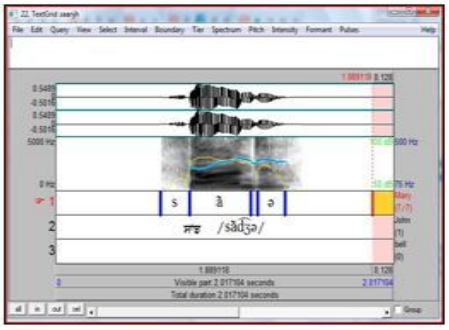


Fig 2. Intensity Graph for ਸਾਂਝ /sə̃dʒ/ (partnership)

Similar analysis for entire word list has been carried out. The intensity of the tone bearing vowels has been measured for 2 words in each category of 5 tonal characters and separately for male and female voices. The details are tabulated below in the tables Table 1 & Table 2.

4.2 Tone Analysis

Tone patterns have been observed from the annotated spectrographs for the entire word-list tom study the pattern of tones for these characters while these occur in the initial, medial and final positions. Falling tone was observed in initial and medial position and rising tone was observed in the final position. One female sample for one word (Initial-Red, Medial-Green, Final-Blue) in each category is given below corroborating the above findings:

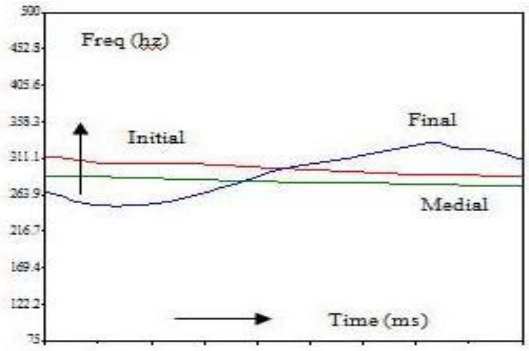


Fig 3. ਭ(bh) /p/ ਭਸਮ(initial)(ash)(/pəsəm/) ;
ਗਭੀਰ(medial)(serious)(/gəbir/); ਜੀਭ(final)(tongue)(/ /dʒib //)

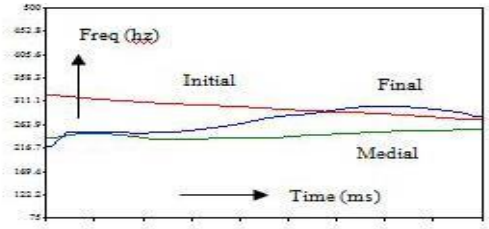


Fig. 4 ਧ (dh) /t/ ਧਰਮ(Initial)(religion)(/tərəm/) ;
ਗਧਲਾ(medial)(muddy)(/gəd̪la/);ਪ੍ਰਬੰਧ(final) (arrangement) (prbə̃d)

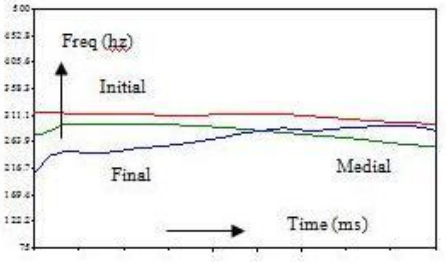


Fig 5. ਢ (dh)/ʈ/ : ਢਹਿਣਾ(initial)(to fall)(/ʈəhɳa/) ;
ਵਢਾਈ(medial) Harvest reaping(/wəd̪ai/); ਸੀਂਢ (final)(Nosy); /sĩd/

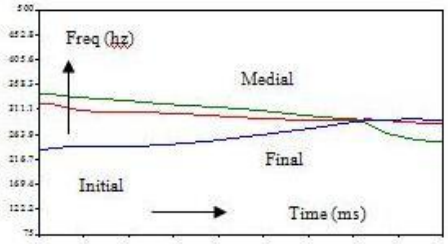


Fig 6. ਘ (gh)/k/ : [ਘੜੀ(Initial)[Watch] /kəɾi/ ; ਜਮਘਟ
(Medial)[Crowd] /jmgət/; ਮਾਘ (Final)[Name of Month] /məg/

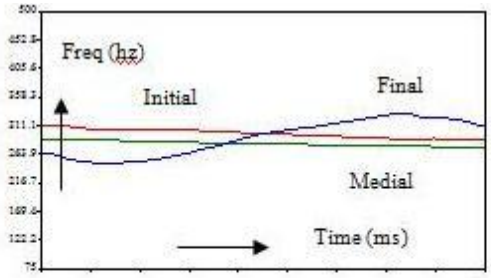


Fig 7. ਝ (Jh)/ʧ/ : ਝਗੜਾ(initial)(Quarrel)(/ʧəgɾa/);
ਅਝੱਕ(medial) / Bold ; ਸਾਂਝ (final)(/sə̃dʒ/)

4.3 F_0 Analysis

The F0 analysis of five tonal characters existing in Punjabi language namely ਭ, ਧ, ਢ, ਝ, ਘ in their initial, medial and final positions have been carried out as per the methodology mentioned above. The F0 contour plots for one female sample are depicted below .The maxima and minima for the five Punjabi tonal characters for male and female voice has also been investigated. They are enumerated in the Tables I and Table 2 below.

5. Summary and Conclusion

In the present analysis, we have observed the following with respect to tonal characters of Punjabi language:

(1) The tonal characters in the initial and medial position results in a falling tone. However,

if the medial consonant is nasalized, the pitch pattern has a rising –falling pattern as observed in the Fig 6.

(2) The tonal characters in the final position indicate a rising tone.

(3) The intensity figures from the tables for male and female voices corroborate the higher female pitch.

(4) The medial character if followed by ਅ /a/ results in a decrease in slope of the falling tone

(5) The germination in the final position results in rising-falling tone and in the medial position gives rise to a slow decrease in falling tone.

(6) The nasalization in character prior to the final tonal character results in rising tone followed by sudden fall in the tone.

(7) All the tonal characters in the final position indicate a release-vowel.

(8) The study can be considered as an empirical proof of the linguistic rules on Punjabi tones illustrated by Gill, H.S. (1986),[Ref no:5] Chander Shekhar (2001) [Ref no:8] and the author's earlier work. [Ref no:9]

6. References

[1] W3C Recommendation (2008), Pronunciation Lexicon Specification Ver 1.0.

[2] Baart Joan L.G, Tonal features in languages of Northern Pakistan, Pakistani languages and society: problems and prospects

[3] Karamat Nayyara (2010), Phonetic Inventory of Punjabi: Pakistan, Center for Research in Urdu Language Processing

[4] Singh,Harkirat (1991), Prominent features of Punjabi language. Patiala: Publication Bureau, Punjabi.

[5] Gill, H.S. (1986) , A Reference Grammar of Punjabi

[6] Joshi S.S. (1973), Pitch and related phenomena in Punjabi , . Patiala:Pakha Sanjam.

[7] Haudricourt, A.G. (1971), Tones in Punjabi. Paris: C.N.R.S.

[8] Singh Chander Shekhar (2001), Punjabi Prosody: The old Tradition & The new Paradigm

[9] Lata Swaran (2011), Challenges for Design of Pronunciation Lexicon Specification (PLS) for Punjabi Language.

[10] Yi Xu (2004), Understandin Tone From The Perspective Of Production and Perception.

[11] Yi Xu, Sources Of Tonal Variations In Connected Speech, Journal Of Chinese Linguistics.

[12] Ching X.Xu, Yi Xu, Li-Shi Luo, A Pitch Target Approximation Model For F0 Contours in Mandarin.

	Initial				Medial				Final			
	Tonal Words	Pitch (Hz)		Intensity (dB)	Tonal Words	Pitch (Hz)		Intensity (dB)	Tonal Words	Pitch (Hz)		Intensity (dB)
		Min	Max			Min	Max			Min	Max	
ਭ	ਭਬਕ /pə̀bə̀k/	116.9	164.5	79.6	ਅਭਿਆਸ /əbìas/	132.7	140.4	79.8	ਗੁਭ /gÚb/	121.7	157.0	75.5
	ਭਸਮ /pə̀sə̀m/	122.5	166.5	79.2	ਗੰਭੀਰ /gə̀̃bir/	117.8	124.8	75.3	ਜੀਭ / d͡ʒíb /	114.6	179.1	73.7
ਧ	ਧੜ /tə̀ɽ/	115.7	173.3	81.1	ਦੁਧੀਆ /dUdìa/	140.1	154.3	75.7	ਪ੍ਰਬੰਧ /prbə̃́d/	105.7	155.7	73.5
	ਧਰਮ /tə̀rə̀m/	127.5	162.7	82.7	ਗੰਧਲਾ /gə̀̃dla/	114.4	125.9	73.5	ਕੰਧ /kə̃́d/	118.6	159.6	74.6
ਢ	ਢਹਿਣਾ /ʈìɳa/	117.2	136.1	74.7	ਬੀਂਢਲ / bǐ̃ɖə̀l /	119.0	133.4	74.8	ਗੰਢ /gə̃́ɖ/	113.0	157.8	74.6
	ਢਕੋਸਲਾ /ʈə̀kosla/	120.3	136.1	68.0	ਵਢਾਈ /waɖə̀ii/	114.3	134.9	80.4	ਸੀਂਢ /sǐ̃ɖ/	129.3	164.5	74
ਘ	ਘਰ /kə̀r/	133.4	199.1	81	ਅਨਘੜ /əngè̞ɽ/	131.5	148.7	82.0	ਜੰਘ /jə̃́g/	112.2	188.8	78.8
	ਘੜੀ /kə̀ɽi/	128.8	171.5	79.6	ਜਮਘਟ /jmgə̀t/	123.1	154.9	80.6	ਮਾਘ /mág/	115.6	162.0	80.3
ਝ	ਝੱਗ /t͡ʃə̀gg/	113.9	173.5	79.1	ਅਝੱਕ /ə̀d͡ʒkk/	115.3	166.9	78.5	ਜੰਝ /d͡ʒə̃́d͡ʒ/	116.1	174.8	78.2
	ਝਗੜਾ /t͡ʃə̀gɽa/	117.2	172.9	79.5	ਸੁਝਾਈ /sUd͡ʒə̀i/	117.5	147.1	79.5	ਸਾਂਝ /sə̃́d͡ʒ/	249.3	332.5	79.1

Table 1: Male Pitch and Intensity

	Initial				Medial				Final			
	Tonal Words	Pitch (Hz)		Intensity (dB)	Tonal Words	Pitch (Hz)		Intensity (dB)	Tonal Words	Pitch (Hz)		Intensity (dB)
		Min	Max			Min	Max			Min	Max	
ਭ	ਭਬਕ /pə̀bə̀k/	256.8	308.0	80.8	ਅਭਿਆਸ /əbìas/	270.8	298.6	78.3	ਗੁਭ /gÚb/	237.1	283.7	75.3
	ਭਸਮ /pə̀sə̀m/	253.4	294.9	77.1	ਗੰਭੀਰ /gə̀̃bir/	212.3	277.8	79.1	ਜੀਭ / d͡ʒíb /	195.8	331.4	74.8
ਧ	ਧੜ /tə̀ɽ/	253.0	321.0	79.5	ਦੁਧੀਆ /dUdìa/	253.3	324.0	77.5	ਪ੍ਰਬੰਧ /prbə̃́d/	217.9	301.0	76.8
	ਧਰਮ /tə̀rə̀m/	269.4	326.3	81.7	ਗੰਧਲਾ /gə̀̃dla/	226.3	253.5	77.3	ਕੰਧ /kə̃́d/	260.9	304.6	78.3
ਢ	ਢਹਿਣਾ /ʈìɳa/	296.3	319.5	81.6	ਬੀਂਢਲ / bǐ̃ɖə̀l /	243.7	288.2	77.8	ਗੰਢ /gə̃́ɖ/	217.9	292.6	78.3
	ਢਕੋਸਲਾ /ʈə̀kosla/	255.5	292.1	79.9	ਵਢਾਈ /waɖə̀ii/	255.1	296.3	82.3	ਸੀਂਢ /sǐ̃ɖ/	268.4	324.3	75.8
ਘ	ਘਰ /kə̀r/	253.1	316.4	80.6	ਅਨਘੜ /əngè̞ɽ/	287.7	348.2	79.3	ਜੰਘ /jə̃́g/	205.5	317.7	77.1
	ਘੜੀ /kə̀ɽi/	274.3	323.2	78.4	ਜਮਘਟ /jmgə̀t/	248.0	353.8	79.5	ਮਾਘ /mág/	234.7	292.7	79.1
ਝ	ਝੱਗ /t͡ʃə̀gg/	234.1	316.1	79.7	ਅਝੱਕ /ə̀d͡ʒkk/	274.0	287.1	78.1	ਜੰਝ /d͡ʒə̃́d͡ʒ/	243.8	307.6	78.0
	ਝਗੜਾ /t͡ʃə̀gɽa/	278.1	329.8	79.4	ਸੁਝਾਈ /sUd͡ʒə̀i/	263.4	306.8	79.5	ਸਾਂਝ /sə̃́d͡ʒ/	249.3	331.8	78.3

Table 2: Female Pitch and Intensity

Grapheme-to-Phoneme Converter for Sanskrit Speech Synthesis

Diwakar Mishra, Girish Nath Jha, Kalika Bali
{1,2} Jawaharlal Nehru University, New Delhi
{3} Microsoft Research Lab India, Bangalore
E-mail: diwakarmishra@gmail.com, girishjha@gmail.com, kalikab@microsoft.com

Abstract

The paper presents a Grapheme-to-Phoneme (G2P) converter as a module for Sanskrit speech synthesis. The G2P and the speech synthesis this is going to be a part of, are for post Vedic or classical Sanskrit prose only. This does not apply to Vedic because that has different writing system with accent marks which is not with classical Sanskrit. It also does not apply with meters because meter does not only decide pause boundary but phones also. While Spoken Sanskrit is used in a limited specific context, and for general purpose by a fewer number of people (according to census of India data), the socio-cultural value of the language retains its significance in the modern Indian milieu. The accessibility to Sanskrit resources is of utmost importance in India, and also in the world for the knowledge discourse of Sanskrit. This paper, presents the development of a standalone G2P converter for Sanskrit based on the model developed by HP Labs India (and released through Local Language Speech technology Initiative). The converter system takes as input the Sanskrit Unicode text in UTF-8 format and returns the sequence of phones with word and sentence boundaries written in the output file. The input text for this system is supposed to be in normal word form, i.e., if there are numbers or abbreviations, those should be expanded into words. The system maps the characters applies the specific rules that are necessary for a conversion of orthographic representation to a phonetic representation of Sanskrit. The system converts into phoneme word by word, thus cross-word modifications are not dealt with. The authors will also demonstrate the system with the presentation. New abstract starts from here.

Keywords: Speech synthesis, Text-to-Speech, TTS, Grapheme-to-Phoneme, G2P, Sanskrit, Phonetics, Phonology

1. Introduction

Sanskrit as a language is unique in nature, as it represents a continuity of tradition since the Vedic period, has available one of the most exhaustive grammar, and plays an important role in the socio-cultural psyche of the people of India. It can truly be said of Sanskrit that "Languages are the repository of thousands of years of people's science and art" (Harrison, 2007). In the course of time, Sanskrit has lost much of its heritage and its scope and usage may have become restricted. Even then, there still exist numerous resources, mainly text, on a variety of subjects and from various historical eras.

An easy natural means of exploring, preserving and accessing these resources will help preserve a significant part of the cultural heritage of India as well as provide an insight into the historical past of the country. The advancement in digital storage and retrieval techniques has led to a cost effective means for easy dissemination of these resources. And while Indian languages, including Sanskrit, remain relatively resource poor, the present focus by the government, academic institutions as well as industry on language resources and technology provides much to be hopeful about. In fact, e-corpora for Sanskrit is already available which Babeu (2011) summarizes precisely. The next logical step from a digital repository of Sanskrit resources is to provide access in various modes – text, audio, video or other multimedia. Till date, there has been little effort to develop a TTS for Sanskrit except for a prototype Text-to-Speech system at International Institute of Information Technology (IIIT), Hyderabad (Mahananda et al, 2010). Acharya, multilingual computing website of IIT Madras, has link of online demo of Sanskrit speech synthesis. The synthesizer is developed on MBROLA speech synthesis engine and does syllable level synthesis (lacking intonation). In the website, it is embedded as a Java applet which could not be tested online.

We believe that a Text-to-Speech (TTS) for Sanskrit is necessary to provide a natural auditory means of access to these resources. A TTS is a computer system that allows access to text data in an audio form. Its usefulness lies in making Sanskrit accessible to a larger audience through a number of devices like PCs and phones. Such a system would also greatly benefit those with visual impairment. Another point to note is that while Sanskrit is written in Devanagari script it is used for traditional purposes across the country by people who may not necessarily be conversant in reading Devanagari text. Sanskrit TTS will help lifting barrier of Devanagari script for such people. Besides, Sanskrit has always had a great oral tradition where stress has always been on how and what is recited aloud than a textual reading. Further, use of multimedia in teaching and learning of languages has also gained popularity and credibility. Hence, the value of a TTS as a part of an e-learning program for Sanskrit is immense and undeniable. The paper in coming sections will describe a Grapheme-to-Phoneme (G2P) converter for Sanskrit, which applies on classical Sanskrit prose only. This does not apply on Vedic Sanskrit because that uses many more symbols which make its writing system different. This also does not apply to the meter or poetry because meter not only affects pause boundary but also the phones. It does not mean that proposed system will skip the metrical text, but it will deal with it in the same manner as the prose.

2. Grapheme-to-Phoneme Conversion

In a TTS system, the Grapheme-to-Phoneme module converts the normalized orthographic text input into the underlying linguistic and phonetic representation (Bali et al, 2004). Therefore G2P conversion is the most basic step in a TTS system.

Most Indian languages' scripts are syllabic and largely phonetic in nature. Therefore, in most cases, there exists a one-to-one mapping between the phone and its orthographic representation. However, in many languages like Hindi, Sanskrit, and Bangla etc. certain phonological phenomena like consonant clusters, long consonants and schwa-headed syllables, require special rules. The abovementioned three major exceptions are very regular in nature and can be handled with simple rules. Only few more rules are required to handle other kinds of exceptions like, schwa deletion in Hindi, consonant lengthening before semivowel in Sanskrit etc.

The model adapted for this G2P converter is of the Hindi G2P converter developed as a part of the Hindi TTS at HP Labs India and released by Local Languages Speech Technology Initiative (LLSTI) (Bali et al, 2005; Krishna, 2004; Bali et al, 2007; Talukdar, LLSTI). There are other TTS systems for Hindi and other Indian languages which are not mentioned here as this paper takes this particular system as a model. For this implementation, the data structure is similar to the abovementioned however; there are certain differences that are described in the coming sections.

3. Sanskrit G2P Converter

Sanskrit Grapheme-to-Phoneme converter is supposed to run on the normalized text and therefore assumes that the input text is normalized, it means is plain Devanagari Sanskrit text with no symbols and abbreviations, numbers etc. expanded into words. The following subsections of this section describe the phoneset, data-structure, program architecture and its implementation.

The Sanskrit G2P is being developed as a part of a project to develop a Sanskrit TTS using the Festival TTS framework (Taylor et al, 1998; Richmond et al, 2007) which has already been used extensively by research community in speech synthesis. However, it has been felt that the language processing modules in Festival are not adequate for certain languages and hence, the need for a different G2P and other modules to be plugged into Festival (Bali et al, 2004). Further, a standalone G2P can also be used for other related applications; for example, for conversion of text for an Optimal Text Selection (Bali et al 2004), as one of the components for a Machine Transliteration system etc.

3.1 Sanskrit Phoneset

The first step towards the development of a G2P converter is to define a phoneset for the language. Astadhyayi's phoneme inventory as explained in Siddhanta kaumudi (Panashikar, 1994) has 44 phonemes (9 vowels, 33 consonants and anusvara, visarga); and the same explains the total number of phones excluding accents as 44 vowels' variations (with accents, 132), 56 consonants' variations (including 20 yama and 3 nasal semivowels), 3 visarga variations and 1 anusvara, thus a total of 104 or 192 with accented vowels. The Sanskrit phoneset under presented G2P converter system has different enumeration as the description follows. It contains 93 phones including 21 vowels, *anusvara*, *visarga*[1] and 69 consonants. Of these, no examples were found in the text corpora of 3 viz., protracted variation of |i|, |u| and nasal palatal fricative long consonant. Excluding these, there are 90 phones. Among these, allophones of *visarga* and nasal variations of vowels are also excluded. The allophones of *visarga* don't have representative characters in Devanagari Unicode scheme and are written by *visarga* character only. Also, the observation of the spoken Sanskrit data shows no difference between common *visarga* and its allophones before velar and labial unvoiced stops. The reason for excluding the nasal variation of vowels is that they are not found in Sanskrit speech as well as writing except some technical texts (Sastri, 1983). The included phones, which are not famous in the traditional Sanskrit grammar, come from splitting of protracted vowels into two for both tense and lax vowel. [2] The other included phones are long consonants which are recognized in linguistics as phones but as consonant clusters in traditional Sanskrit grammar. Among those, the nasal palatal affricate does not find any example. Thus, a total of 89 phones are considered for the speech synthesis. The descriptive phoneset in the form of a table with six columns, namely | phone/phoneme | Grapheme, Devanagari | phone description | context | example roman | example Devanagari | is given in the appendix.

3.2 Data-structure and Statistics

As mentioned earlier, this module follows the methodology followed by Hindi G2P converter released at LLSTI website (llsti.org). This module has three data files – Sanskrit mapping, Sanskrit rules, and Sanskrit pronunciation lexicon.[3] Expecting that in modern Sanskrit,

[1] The status of these two sounds lies between vowel and consonants but are not semivowels. We can say them semiconsonants, though this word is not used for them. These can occur only in coda of a syllable immediately preceded by a vowel. *Anusvara* is nasal and *visarga* is glottal affricate.

[2] Traditionally, a Sanskrit vowel has three variations – short, long and protracted. Here short and tense vowels are taken as different vowels – lax and tense. And both have their protracted (longer) variation. The reason for that is that the protracted vowel is used in the place of its allophonic vowel in special cases. The lengthened variation of both the lax and tense vowel is not the same. So the protracted allophones of both are different.

[3] Please refer to Bali et al (2004) and documentation of Hindi G2P at LLSTI (Talukdar, LLSTI) for detailed description. The structure followed is very similar to that used in that Hindi G2P.

imported words can import non-Sanskrit Devanagari characters, the characters representing the foreign sounds are also included in the mapping file and mapped to the corresponding Sanskrit sound character, e.g., variations of ‘k’, ‘kh’, ‘g’ in the sample of mapping file. Sanskrit mapping file is a four column table with a line as a row and white space as the column separator. The four columns are:

Character Type Class Phoneme

Following are some examples from the Sanskrit mapping file:

```
ॅ A CBD MM
ं A ANSW M
ः A SG H
अ V VWL ah
आ V VWL aa
ऑ V VWL aa
इ V VWL ih
ई V VWL ii
उ V VWL uh
ऊ V VWL uu
ऋ V VWL rhi
ॠ V VWL rri
ऌ V VWL li
ए V VWL e
ऐ V VWL ai
ओ V VWL o
औ V VWL au
क C KS1 k
क़ C KS1 k
ख C KS2 kh
ख़ C KS2 kh
ग C KS3 g
ग़ C KS3 g
घ C KS4 gh
ङ C KS5 ng
च C CS1 c
…
ॄ V VWL li
े V VWL e
ै V VWL ai
ो V VWL o
ौ V VWL au
़ S NUK NU
, S COM #
। S VRM ##
॥ S VRM ##
? S VRM ?
```

The rule file is in a particular format with the following structure (for explanation, refer to Bali et al, 2004):

$COND_1\ COND_2\ \ldots\ COND_m\ \{\ ACT_1\ ACT_2\ \ldots\ ACT_n\ \}$

Where the LHS (before “{”) is the context sequence of characters and the RHS is sequence of the actions to be taken on this context. Following are some samples from the rule file:

```
KS1 NUK { R:1:k }
KS2 NUK { R:1:kh }
KS3 NUK { R:1:g }
CS3 NUK { R:1:j }
… …
KS1 HAL KS1 { R:1:kk }
KS2 HAL KS2 { R:1:kkh }
KS3 HAL KS3 { R:1:gg }
KS4 HAL KS4 { R:1:ggh }
KS5 HAL KS5 { R:1:nng }
CS1 HAL CS1 { R:1:cc }
… …
VWL KS1 HAL YA { K:1:X R:2:kk K:4:X }
VWL KS1 HAL RA { K:1:X R:2:kk K:4:X }
VWL KS1 HAL LA { K:1:X R:2:kk K:4:X }
VWL KS1 HAL VA { K:1:X R:2:kk K:4:X }
VWL KS2 HAL YA { K:1:X R:2:kkh K:4:X }
VWL KS2 HAL RA { K:1:X R:2:kkh K:4:X }
… …
ANSW PS3 { R:1:m K:2:X }
ANSW PS4 { R:1:m K:2:X }
ANSW PS5 { R:1:m K:2:X }
ANSW YA { R:1:~y K:2:X }
ANSW RA { R:1:nx K:2:X }
ANSW LA { R:1:~l K:2:X }
ANSW VA { R:1:~v K:2:X }
… …
ANSW HA HAL YA { R:1:~y K:2:X K:4:X }
ANSW HA LA { R:1:~l K:2:X K:3:X }
ANSW HA VA { R:1:~v K:2:X K:3:X }
```

The third data file for this module is Sanskrit pronunciation lexicon. Though most of the Sanskrit words are covered by general rules and Sanskrit does not need a lexicon like for example, English, it is kept for words which might be exceptions to the rules. This is in the form:

Unicode_skt_word phonetic_representation

The three data files are accessed by the G2P converter program for the conversion. The program reads the input file with Sanskrit text in Devanagari Unicode UTF-8 encoding and writes the output in a different text file. The process of conversion is described in the next section. The Sanskrit rule file contains 222 rules. The actual number of the rules is very less but a different class label had to be assigned to each consonant as they needed distinction for some different rules. First 8 rules are to convert special characters from imported words to the standard Sanskrit characters. 43 rules convert the double consonants into long consonant (one for each phoneme). 128 rules deal with consonant lengthening between the preceding vowel and the following semi-vowel. This simple rule took too many lines due to the abovementioned necessity of making distinction. The next 36 rules are for assigning

appropriate allophones to the *anusvaara*. One rule adds length to the vowel following digit 3 or *avagraha* to make it *pluta*. The last 6 rules convert the three characters of nasal semi-vowel into single phone. The Sanskrit character mapping file has 79 characters including four punctuations – comma, question mark, *viraama*, *purna viraama* and digit 3. Only a few examples are there in the pronunciation lexicon file. *Pluta* vowel and consonant lengthening between vowel and semivowel is Sanskrit's distinct feature from other Indian languages.

3.3 Process of the Conversion

This module accepts the text in Unicode UTF-8 format. The input file is read in the same format and it is considered token by token separated by white space. First the token is checked in the pronunciation lexicon file. If it is found there, its corresponding pronunciation or phone-sequence is returned to the output String buffer. And if not found there, then only it is sent to apply the phoneme conversion rules.

The converter method converts the Devanagari Unicode string in following steps:

- First it adds schwa after each consonant not followed by vowel sign or consonant marker (*halanta*).
- Then it starts the rule matching and applies the actions on the schwa added string.
- After this, the role of *halanta* is done with their role, and the nasal vowels are not phonemic (also not found in general Sanskrit writings) therefore both the *halanta* and the nasal marker are deleted from the phonetic string.
- *Chandra-bindu*, the nasal marker, is not deleted prior to rule application because it has role in the nasalization of semivowels, and it is used in the rules.

Thus the string of phones is returned for each word token and it is added to the output string buffer. In this independent module, the output is written in the output file. The same can be returned to the next module in the integrated step.

3.4 Technological Platform

The Hindi G2P converter taken for the present as model is developed in C++ and LISP. But the system described here is developed in Java. The program is stand alone and runs through command line. The data files are the text files (.txt) with Unicode compatibility.

3.5 Sample of Text Conversion

The following is given the sample input and its output generated by the above described system.

Input: लघुरयमाह न लोकः कामं गर्जन्तमपि पतिं पयसाम् ।

Output: `# l ah gh uh r ah y ah m aa h ah # n ah # l o k ah H # k aa m ah M # g ah r j ah n t ah m ah p ih # p ah t ih M # p ah y ah s aa m # %`

In the output, '#' is the word boundary and '%' is the sentence boundary. The system was internally evaluated and the found bugs were fixed. It does not have and external evaluation. It converts normal text with good accuracy but gives wrong result when the word is attached with some punctuation or a symbol comes within a word, which is out of assumed condition of the input. A different module- Sanskrit text normalizer is supposed to provide the input to G2P system in the form of words separated with space and detached from symbols.

4. Conclusion

In this paper, we have presented a Grapheme-to-Phoneme converter module for Sanskrit to be used in Sanskrit speech synthesis system. The developed module is currently being evaluated. The system is being developed as a part of PhD research of the first author. The authors will also demonstrate the system at the time of the presentation.

5. Acknowledgements

The authors acknowledge the Local Language Speech Technology Initiative for making the speech synthesis modules available for free. At the same time authors acknowledge the developer and copyright holder of the G2P module, Partha Pratim Talukdar for allowing the researchers to access and reuse his development.

6. References

Acharya: Multilingual Computing for Literacy and Education, IIT Madras website, online demo page, http://acharya.iitm.ac.in/demos.php

Babeu, Alison (2011). *"Rome Wasn't Digitized in a Day": Building a Cyberinfrastructure for Digital Classics*. Council on Library and Information Resources, Washington, DC, USA, pp. 20-23.

Bali, Kalika, Krishna, N. Sridhar, Badasker, Sameer, Anjaneyulu, K.S.R. (2007). Enabling IT Usage through the Creation of a High Quality Hindi Text-to-Speech System, *HP Laboratories India Technical Report*.

Bali, Kalika, Talukdar, Partha Pratim, Krishna, N. Sridhar, Ramakrishnan, A.G. (2004). Tools for the Development of a Hindi Speech Synthesis System, In *Proc. 5th ISCA Speech Synthesis Workshop (ISCA)*, Carnegie Mellon University, Pittsburgh, pp. 109-114.

Harrison, K. David, (2007). *When Languages Die: The Extinction of the World's Languages and The Erosion of Human Knowledge*, Oxford University Press

Local Language Speech Technology Initiative website http://llsti.org/

Mahananda, Baiju, Raju, C. M. S., Patil, Ramalinga Reddy, Jha, Narayan, Varakhedi, Srinivasa and Kishore, Prahallada (2010). Building a Prototype Text to Speech for Sanskrit. In *Lecture Notes in Computer Science (LNCS-6465), (In Proc. 4i-SCLS)*, Springer, Germany.

Panashikar, Vasudeva Lakshman Shastri (ed.) (1994). *Siddhāntakaumudī*, Choukhamba Sanskrit Pratishthan, Delhi.

Richmond, K., Strom, V., Clark, R., Yamagishi, J. and Fitt, S. (2007). Festival Multisyn Voices, In *Proc. Blizzard Challenge Workshop (in Proc. SSW6)*, Bonn, Germany

Sastri, Bhimsen (1983). *Laghusiddhantakaumudi Bhaimi Vyakhya*. Bhaimi Prakashan, Delhi

Talukdar, Partha Pratim. Grapheme to Phoneme Conversion System v 1.0. In documentations in Local Language Speech Technology Initiative (LLSTI).

Taylor, Paul A., Black, Alan, and Caley, Richard. (1998). The Architecture of the Festival Speech Synthesis System. In The Third ESCA Workshop in Speech Synthesis, Jenolan Caves, Australia, pp. 147-151.

Appendix: Sanskrit Phoneset

Following is the Sanskrit phoneset prepared for the Sanskrit speech synthesis system to be developed under the research of the authors.

Phone / phoneme	**Grapheme Devanagari**	**Sound description**	**Context**	**Example Roman**	**Example Devanagari**
		vowels			
ah	अ	[+vocalic –high –front –length –tense –round]	Any	k**A**m**A**l**A**m	कमलम्
aa	आ	[+vocalic –high –front –length +tense –round]	Any	r**A**j**A**	राजा
ah3	अ३	[+vocalic –high –front +length –tense –round]	In specific places, only in vocative case	bho bak**A**	भो बक
aa3	आ३	[+vocalic –high –front +length +tense –round]	In specific places, only in vocative case	bho dev**A**	भो देवा
ih	इ	[+vocalic +high +front –length –tense +spread]	Any	**It****I**	इति
ii	ई	[+vocalic +high +front –length +tense +spread]	Any	sh**I**rsham	शीर्षम्
ih3	इ३	[+vocalic +high +front +length –tense +spread]	In specific places, only in vocative case	ayi janan**I**	अयि जननि
ii3	ई३	[+vocalic +high +front +length +tense +spread]	In specific places, only in vocative case	Not found	
uh	उ	[+vocalic +high –front –length –tense +round]	Any	**U**payah	उपायः
uu	ऊ	[+vocalic +high –front –length +tense +round]	Any	bh**U**mih	भूमिः
uh3	उ३	[+vocalic +high –front +length –tense +round]	In specific places, only in vocative case	he vadh**U**	हे वधु
uu3	ऊ३	[+vocalic +high –front +length +tense +round]	In specific places, only in vocative case	Not found	
rri	ऋ	[+vocalic +high +front –length –tense –round]	Any	v**RI**kshah	वृक्षः
rhi	ॠ	[+vocalic +high +front +length +tense –round]	Any	pit**RI**n	पितॄन्
lli	ऌ	[+vocalic +high +front +length +tense –round]	Any	k**LRI**ptam	कॢप्तम्
e	ए	[+vocalic +high +front +length +tense +spread]	Any	d**E**shah	देशः
e3	ए३	[+vocalic +high +front +length +tense +spread]	In specific places, only in vocative case	mun**E**	मुने
ai	ऐ	[+vocalic –high +front +length +tense +spread +diphthong]	Any	d**AI**shikah	दैशिकः
o	ओ	[+vocalic +high –front +length +tense +round]	Any	l**O**kah	लोकः
o3	ओ३	[+vocalic +high –front +length +tense +round]	In specific places, only in vocative case	bhan**O**	भानो
au	औ	[+vocalic –high –front +length +tense +round +diphthong]	Any	l**AU**kikah	लौकिकः
M	ं	[–vocalic	After	sa**M**skrita	संस्कृत

(anusvaara)		+nasal +voice]	vocalic, never before vocalic, anusvaara, visarga	m	म्
H (visarga)	ः	[–vocalic +glottal –voice]	After vocalic, never before anusvaara, visarga	anta**H**karanam	अन्तःकरणम्
		consonants			
k	क्	[velar stop –aspiration - –voice]	Any except between vowel _ semivowel	**K**amalam	कमलम्
kk	क्क्	[velar stop –aspiration - –voice]	Between two vowels, between vowel _ semivowel	hi**KK**a sha**K**yah shu**K**rah shu**K**lah pa**K**vah	हिक्का शक्यः शुक्रः शुक्लः पक्वः
kh	ख्	[velar stop +aspiration - –voice]	Any except between vowel _ semivowel	**KH**adati	खादति
kkh	क्ख्	[velar stop +aspiration - –voice]	Between two vowels, between vowel _ semivowel	vi**KH**yata h	विख्यातः
g	ग्	[velar stop –aspiration +voice]	Any except between vowel _ semivowel	**G**acchati	गच्छति
gg	ग्ग्	[velar stop –aspiration +voice]	Between two vowels, between vowel _ semivowel	samya**G** **G**anam	सम्यग्गानम्
gh	घ्	[velar stop +aspiration +voice]	Any except between vowel _ semivowel	**GH**orah	घोरः
ggh	ग्घ्	[velar stop +aspiration +voice]	Between two vowels, between vowel _ semivowel	ji**GH**rati	जिघ्रति
ng	ङ्	[velar stop –aspiration +voice +nasal]	Any	ra**N**gah	रङ्गः
nng	ङ्ङ्	[velar stop –aspiration +voice +nasal]	Between two vowels	pratya**NN** atma	प्रत्यङ्ङात्मा
c	च्	[palatal affricate –aspiration - –voice]	Any except between vowel _ semivowel	**C**amasah	चमसः
cc	च्च्	[palatal affricate –aspiration - –voice]	Between two vowels, between vowel _ semivowel	u**CHCH**a ranam	उच्चारणम्
ch	छ्	[palatal affricate	Any except between	**CH**atrah	छात्रः
		+aspiration - –voice]	vowel _ semivowel		
cch	च्छ्	[palatal affricate +aspiration - –voice]	Between two vowels, between vowel _ semivowel	i**CHCHH** a	इच्छा
j	ज्	[palatal affricate –aspiration +voice]	Any except between vowel _ semivowel	**J**agat	जगत्
jj	ज्ज्	[palatal affricate –aspiration - +voice]	Between two vowels, between vowel _ semivowel, between vowel _ ny	sa**JJ**a vi**J**nanam	सज्जा विज्ञानम्
jh	झ्	[palatal affricate +aspiration +voice]	Any except between vowel _ semivowel	**JH**ankara h	झङ्कारः
jjh	ज्झ्	[palatal affricate +aspiration +voice]	Between two vowels, between vowel _ semivowel	u**JJH**itah	उज्झितः
ny	ञ्	[palatal affricate –aspiration +voice +nasal]	Any	ca**N**calah	चञ्चलः
nny	ञ्ञ्	[palatal affricate –aspiration +voice +nasal]		No example found	
tx	ट्	[retroflex stop –aspiration - –voice]	Any except between vowel _ semivowel	**T**iTTibha h	टिट्टिभः
ttx	ट्ट्	[retroflex stop –aspiration - –voice]	Between two vowels, between vowel _ semivowel	Ti**TT**ibha h	टिट्टिभः
thx	ठ्	[retroflex stop +aspiration - –voice]	Any except between vowel _ semivowel	nish**TH**a	निष्ठा
txx	ट्ठ्	[retroflex stop +aspiration - –voice]	Between two vowels, between vowel _ semivowel	prapa**TH** ya	प्रपठ्य
dx	ड्	[retroflex stop –aspiration +voice]	Any except between vowel _ semivowel	**D**amaruh	डमरुः
ddx	ड्ड्	[retroflex stop –aspiration +voice]	Between two vowels, between vowel _ semivowel	u**DD**ayati	उड्डयति
dhx	ढ्	[retroflex stop +aspiration +voice]	Any except between vowel _ semivowel	ru**DH**ih	रूढिः

dxx	ड्ढ्	[retroflex stop +aspiration +voice]	Between two vowels, between vowel _ semivowel	aru**DH**ya	आरुढ्य
nx	ण्	[retroflex stop –aspiration +voice +nasal]	Any except between vowel _ semivowel	rame**N**a	रामेण
nnx	ण्ण्	[retroflex stop –aspiration +voice +nasal]	Between two vowels, between vowel _ semivowel	sha**NN**am	षण्णाम्
t	त्	[dental stop –aspiration - –voice]	Any except between vowel _ semivowel	**T**amalah	तमालः
tt	त्त्	[dental stop –aspiration - –voice]	Between two vowels, between vowel _ semivowel	sa**TT**a	सत्ता
th	थ्	[dental stop +aspiration - –voice]	Any except between vowel _ semivowel	s**TH**irah	स्थिरः
tth	त्थ्	[dental stop +aspiration - –voice]	Between two vowels, between vowel _ semivowel	u**TTH**anam	उत्थानम्
d	द्	[dental stop –aspiration +voice]	Any except between vowel _ semivowel	**D**evah	देवः
dd	द्द्	[dental stop –aspiration +voice]	Between two vowels, between vowel _ semivowel	sa**DD**evah	सद्देवः
dh	ध्	[dental stop +aspiration +voice]	Any except between vowel _ semivowel	ra**DH**a	राधा
ddh	द्ध्	[dental stop +aspiration +voice]	Between two vowels, between vowel _ semivowel	uDDHarah	उद्धारः
n	न्	[dental stop –aspiration +voice +nasal]	Any except between vowel _ semivowel	**N**amah	नमः
nn	न्न्	[dental stop –aspiration +voice +nasal]	Between two vowels, between vowel _ semivowel	jana**NN**api	जानन्नपि
p	प्	[labial stop –aspiration - –voice]	Any except between vowel _ semivowel	**P**atni	पत्नी
pp	प्प्	[labial stop –aspiration - –voice]	Between two vowels, between vowel _ semivowel	a**PP**ayah	अप्पयः
ph	फ्	[labial stop +aspiration - –voice]	Any except between vowel _ semivowel	**PH**alam	फलम्
pph	प्फ्	[labial stop +aspiration - –voice]	Between two vowels, between vowel _ semivowel	anustu**PPH**alam	अनुष्टुप्फलम्
b	ब्	[labial stop –aspiration +voice]	Any except between vowel _ semivowel	**B**alam	बलम्
bb	ब्ब्	[labial stop –aspiration +voice]	Between two vowels, between vowel _ semivowel	ba**BB**anah	बब्बनः
bh	भ्	[labial stop +aspiration +voice]	Any except between vowel _ semivowel	**BH**aratam	भारतम्
bbh	ब्भ्	[labial stop +aspiration +voice]	Between two vowels, between vowel _ semivowel	sa**BH**yah	सभ्यः
m	म्	[labial stop –aspiration +voice +nasal]	Any except between vowel _ semivowel	**M**ata	माता
mm	म्म्	[labial stop –aspiration +voice +nasal]	Between two vowels, between vowel _ semivowel	sa**MM**ukhe	सम्मुखे
y	य्	[palatal fricative –aspiration +voice]	Any	**Y**ajnah	यज्ञः
yy	य्य्	[palatal fricative –aspiration +voice]	Between two vowels	a**YY**arah	अय्यरः
~y	यँ्	[palatal fricative –aspiration +voice +nasal]	Preceded by vowel, followed by y or hy	sa**M**yanta	सयँ्यन्ता
r	र्	[retroflex fricative –aspiration +voice]	Any except between vowel _ semivowel	**R**ama	रमा
rr	ऱ्	[retroflex fricative –aspiration +voice]	Between two vowels, between vowel _ semivowel	ca**R**ya	चर्या
l	ल्	[palatal liquid –aspiration +voice]	Any except between vowel _ semivowel	**L**a**L**ita	ललिता
ll	ल्ल्	[palatal liquid –aspiration +voice]	Between two vowels, between vowel _ semivowel	u**LL**asah	उल्लासः
~l	लँ्	[palatal liquid	Preceded by	sa**M**lapah	सलँ्लाप

		–aspiration +voice +nasal]	vowel, followed by l or hl		पः
v	व्	[labiodental fricative –aspiration +voice]	Any except between vowel _ semivowel	di**V**a	दिवा
vv	व्व्	[labiodental fricative –aspiration +voice]	Between two vowels, between vowel _ semivowel	u**VV**atah	उव्वटः
~v	वँ्	[labiodental fricative –aspiration +voice +nasal]	Preceded by vowel, followed by v or hv	sa**M**vatsarah	सवँ्वत्सरः
sh	श्	[palatal fricative +aspiration –voice]	Any except between vowel _ semivowel	**SH**akrah	शक्रः
ssh	श्श्	[palatal fricative +aspiration –voice]	Between two vowels, between vowel _ semivowel	ni**SHSH**okah	निश्शोकः
sx	ष्	[retroflex fricative +aspiration –voice]	Any	bha**SH**a	भाषा
ssx	ष्ष्	[retroflex fricative +aspiration –voice]	Between vowel _ semivowel	bha**SH**yam	भाष्यम्
s	स्	[dental fricative +aspiration –voice]	Any except between vowel _ semivowel	bha**S**ate	भासते
ss	स्स्	[dental fricative +aspiration –voice]	Between two vowels, between vowel _ semivowel	ni**SS**ritah	निस्सृतः
h	ह्	[glottal fricative +aspiration +voice]	Any	su**H**asini	सुहासिनी
hh	ह्ह्	[glottal fricative +aspiration +voice]	Between vowel _ semivowel	sa**H**yadri	सह्याद्रि
#		short silence/ word boundary			

Phonetic Dictionary for Indian English

[1]Aparna Mukherjee, [2]Alok Dadhekar

[1]Centre for Linguistics, Jawaharlal Nehru University, New Delhi-110 067, India.
aparna.jnu27@gmail.com
[1]Aurionpro Solutions HK Ltd, 33 Hysan Avenue, Causeway Bay, Hong Kong.
alok@aurionpro.com

Abstract

The paper presents the task of building an electronic phonetic dictionary for Indian English, with an aim to have a common source of English words, as they are pronounced in Indian English. The dictionary is a customized version of a pre-existing dictionary – the Carnegie Mellon University (CMU) Pronouncing Dictionary. The differences between pronunciation given in the North American English based CMU Dictionary and Indian English pronunciation, were identified and categorized. The identified categories and patterns were searched and replaced with the desired phonemes with the help of advanced regular expressions with back references. The data was edited manually as well, to avoid any error, since generalization of pattern in natural language is difficult. The phonetic dictionary was then used to generate English words in Devanagari script by following a mapping algorithm. The phonetic dictionary thus built for Indian English pronunciation can be further used to generate words in any Indian script.

1. Introduction

With proliferation of Information and Communication Technology (ICT), information resources are now easily made available to most of us, say within a mouse click on the Internet. But there still remains the issue of language divide and hence inaccessibility hindering a steady growth and development of masses. An individual who does not know the language this information is made available in, still remains unreached and untouched by the technological advances. As a result, Human Language Technology (HLT), which attempts at addressing these issues, continues to play a crucial role. The ongoing research in the field of HLT aims to bridge this linguistic gap. This also calls for a huge requirement of diverse linguistic resources, especially for a country like India with 22 national languages and over 1600 other languages spoken in various regions. There are more than 18 scripts in India, with Devanagari being used by more than 6 languages. All these languages and scripts need to be supported by technology for the benefit of the end-user and to achieve our ultimate aim. This can be best done by attempting to support technology with a sizeable collection of appropriate resources that are customizable and dynamically tunable as per requirement. Only then can we expect technology to work well and bear fruits.

In this paper we make one such attempt. We present the task of building a machine-readable phonetic dictionary for Indian English, (henceforth IE), from an existing dictionary – the Carnegie Mellon University Pronouncing Dictionary[1]. This is a machine-readable pronunciation dictionary for North American English that contains more than 125,000 words and their transcriptions. Each word is phonetically transcribed; the pronunciation is mapped using a phoneme or phone set that contains 39 phonemes. This phoneme set is based on the ARPAbet symbol set, a phonetic transcription code developed by the Advanced Research Projects Agency (ARPA)(ARP, 1993; ARP, 1996) as a part of their Speech Understanding Project (1971–1976). It represents each phoneme of General American English with a distinct sequence of ASCII characters. In ARPAbet, every phoneme is represented by one or two capital letters. Digits are used as stress indicators and are placed at the end of the stressed syllabic vowel.

In the following sections we will discuss how the task of creating an electronic phonetic dictionary for IE was carried out and how we have further used this dictionary for generating English words in Devanagari script. The dictionary can serve as a common source for generating English words in any Indian language script.

2. Utility of the Dictionary

A phonetic dictionary can be directly used in the development of speech synthesis and speech understanding tools. It can also serve as a common source to generate English words in any Indian Language script. The dictionary of English words in an Indian script is required in building various natural language tools, as in:

- machine transliteration systems where it can be used as direct input or the parallel data can be used to train the system.
- machine translation systems where it helps to render the abbreviations and accronyms such as UNESCO, UNICEF and proper names that need not be translated but transliterated into the desired Indian Language.

3. Building an Indian English Phonetic Dictionary

As discussed in section 1, the said Indian English phonetic dictionary is built by customizing the CMU dictionary for IE pronunciation. The CMU dictionary is based on North American (NA) pronunciation, which is quite different from IE. The transcription of the English words in the existing dictionary have been optimized so as to bring the pronunciation as close to the English spoken in most part of India (Gargesh R., 2004; Maxwell and Fletcher, 2010). The

[1]The CMU Phonetic Dictionary is available at https://cmusphinx.svn.sourceforge.net/svnroot/cmusphinx/trunk/cmudict/

pronunciation given in the printed dictionaries, which may not be machine readable, has been closely studied while optimizing the NA pronunciation in building up this dictionary (Jones D., 2003; Jones D., 2004). The phonetic dictionary thus built for the pronunciation of words in IE, has been used to generate English words in Devanagari script.

3.1. Methodology

Here we have used this electronic dictionary in producing English words in Devanagari script. The ARPAbet phonemes were mapped into the corresponding grapheme of the desired script.There may be some English sounds that may not correspond to any character in a particular Indian script like JH (of ARPAbet) in 'pleasure' cannot be represented by any alphabet in Devanagari. To meet this gap, an alphabet in Indian script that closely correspond to a particular character has been chosen, like pleasure is written as प्लेजर in Devanagari. The mapping of the phoneme is given in Table 1.

AH → अ	AA → आ	IH → इ	IY → ई
UH → उ	UW → ऊ	EH → ए	EY → ए
AE → ऐ	OW → ओ	OY → ऑय	AO → ऑ
AW → आउ	AY → आइ	NG → ं	BH → भ
B → ब	CH → च	DD → ध	DHH → ढ
DH → द	D → ड	F → फ	GH → घ
G → ग	NN → ण	HH → ह	JHH → झ
JH → ज	KH → ख	K → क	L → ल
M → म	N → न	P → प	R → र
SHH → ष	SH → श	S → स	TTH → ठ
TH → थ	TT → त	T → ट	V → व
W → व	Y → य	ZH → ज	Z → ज़

Table 1: ARPAbet to Devanagari script

The first step in building the phonetic dictionary for IE involved identifying the major differences between the NA pronunciation, as given in the CMU Pronunciation Dictionary, and the IE pronunciation; and categorizing the phonemes of the words transcribed in the CMU dictionary. Once identified, the phonemes were easily replaced with the appropriate phoneme for IE pronunciation.

For instance, -ed in 'aborted' in IE is pronounced as ED /ed/ unlike the NA pronunciation ID /ɪd/; 'odd' is pronounced as AOD /ɔd/, while it is AAD /ɑd/ in NA English. The R–coloured vowel in 'her' /hɝ/ is pronounced as /hər/ in IE. So the phoneme ID, AA and ER need to be mapped as ED, AO and AH R respectively. Based on the phonemic transcription, the words were categorized and their corresponding pronunciation in IE were mapped in ARPAbet symbols. This list of phonemes and the corresponding mapping helped in identifying the patterns that needed manipulation and further helped in generalizing patterns that required replacement. An exemplar list of phonemes and their pronunciation in IE is given in Table 2.

The phonemes in particular environment in each word that were identified to differ from the IE pronunciation were replaced with the desired phonemes using regular expressions (Goyvaerts J., 2009) with back references. Some regular expressions used to replace the phonemes are shown in Table 3. The first column of the table lists the environments where a specific phoneme for each case needs to be replaced. The replacements were done only semi-automatically using back referencing technique in the 'regex' tool. Everything could not be automatized since the pattern of variation cannot be generalized for all cases.

Phoneme	Phoneme mapping for IE	Example
IZ/IS/AS/AZ	ES/EZ	Abolishes
AD/ID	ED	Aborted
AA	AO	Accomplish
JH	D JH	Adjective
M	MB	Comb
NG	NG G	King
ER	AH R	Accelerator

Table 2: North American pronunciation to Indian English pronunciation mapping

Search for	Replace with	No. of words edited (approx.)
(.∗E(S |SS).∗)(AH |IH |IY)(S|Z)$	\1EH\4	8403
(.*ED.*)(AH|IH|IY)(D)	\1EH\3	9500
(.*O.*)AA(.*)	\1AO\2	10123
(.∗DJ.∗[^D])JH(.∗)	\1D JH\2	58
(.∗MB.∗M)([^B].∗)	\1 B\2	192
(.*NG.*NG)$	\1 G	5843
ER ER$	AH R AH R	18557 10017

Table 3: Indicative list of regular expressions with back references

4. Results : What We Achieved

We could successfully complete the task of customizing the CMU Phonetic Dictionary by characterizing the major differences, categorizing patterns, and then employing complex regular expressions. We thus accomplished the tedious task of having 125,000 words with IE pronunciations. A collection of the identified patterns and the phonemic mappings for IE are given in Tables 4 and 5 [2].

To test and validate the proposed phoneme set, we compared the pronunciation of the words that are commonly used, based on the pronunciation of Indian speakers – in and around Mumbai and Delhi. A detailed and exhaustive evaluation of this dictionary would be a part of the future work. The modified dictionary was still roughly evaluated based on the available test cases and a few errors were noted for rectification. Such rectifications required in the dictionary have been exemplified by the following paradigm of the same word as noted below:

(a) Cottage

[2]The | sign very typically represents OR in Tables 4 and 5

Phoneme [for NA]	Example Words	Pattern → Phoneme [for IE]
AA	father, Yugoslavia, aamodt	A |aa → AA
AA	Odd, bought	o |ough → AO
AE	at, fact	a → AE
AH	hut, synonymous, vision, chryseis	u |ou |o |ei → AH
G	green, aggressive, cage	g |gg |ge → G
HH	he, Achmed	h |ch → HH
IH	it, select	i |e → IH
IY	eat, diesel, crazier, crazy, create, creep, capemaum, ceiling	ea |ie |i |y |e |ee |a |ei → IY
JH	singe, ginger, bridge, silajdzic, zawadzki, zhao	ge |g |dge |dg |j |jdz |dz |zh → JH
JH	adjective	dj → D JH (exception: djakarta, guandjo)
K	key, accident, Christian, brake	k |c |ch |ke → K
L	lee, annabelle, tall, fable	l |lle |ll |le → L

Table 4: Mapping of North American pronunciation to Indian English pronunciation - I

CMU Pronunciation → K AA T AH JH
Optimal IE Pronunciation → K AO T AH JH
Desired IE Pronunciation → K AO T EH JH
(b) Cottages
CMU Pronunciation → K AA T IH JH IH Z
Optimal IE Pronunciation → K AO T IH JH IH Z
Desired IE Pronunciation → K AO T EH JH EH Z

With the phoneme grapheme mapping the phonetic dictionary has been used to generate English words in Devanagari script programatically. Some examples of Devanagari output are as in Table 6:
The following input files are required for the generation of the Indian Language Dictionary:

(a) the Indian English phonetic dictionary;

(b) phonemes and Devanagari grapheme mapping as given in Table 5;

(c) vowel phoneme set used in CMU dictionary, $eg.$ AA, AE;

(d) vowel phoneme set to Devanagari half vowel mapping, $eg.$ AA → ा, IH → ि

The input is kept in separate modules so that they can be modified as and when required according to the requirement. The IE Phonetic Dictionary can be optimized based on specific regional variation in pronunciation. Words in different script can be generated by just replacing the three files for phoneme grapheme, as mentioned above.

Phoneme [for NA]	Example Words	Pattern → Phoneme [for IE]
M	me, come, committee	m |me |mm → M
M	comb	mb → MB
N	Net, knee, cannibal, nine	n |kn |nn |ne → N
NG	ping	ng |ngh → NG G
OW	Oat, abaco, kingsborough, borrow, bureau, tableaux, seoul	oa |o |ough |ow |eau |eaux |eou → OW
OY	toy, oil, buoyant, greubel, bouyer	oy |oi |uoy |eu |ou → OY
P	pen, apple, ape	p |pp |pe → P
R	read, ohrt, oherron	r |hr |rr → R
S	sea, cetacean, scissors, fitness, szilard	s |c |sc |ss |sz → S
SH	she, chanel, nation, kollasch, cretien	sh |ch|tio |sch |t |s |sz |szcz → SH
T	tea, attic, nute, capizzi, aamodt	t |tt |te |z |dt → T
TH	theta, matthew	th|tth → TH
UH	hood, should, joora	oo |ou |u → UH
UW	Two, caribou, loose, lose, jude	wo |ou |oo |o |u → UW
UW	New, obtuse, boulet, greuel	ew|u|ou|eu → UW
V	vee, proactive	v|ve → V
W	we, guam	w|u → W
Y	yield, europe, view, nadja	y|e|i|j → Y
Z	zee, feasible, scissors, szilard	z|ze|s|ss|sz→ Z
ZH	seizure, division, zsazsa, genre	z|si|zs|g → ZH
Y UW	nute	u → Y UW
Y UH	politburo	u → Y UH

Table 5: Mapping of North American pronunciation to Indian English pronunciation - II

aback → अबैक	abaco → ऐबको
abacus → ऐबकस	bridge → ब्रिज
cots → कॉट्स	display → डिस्प्ले
displease → डिस्प्रीज	pleasure → प्लेजर

Table 6: Examples of Devanagari output

5. Conclusion and Road ahead

An electronic phonetic dictionary in Indian English thus built by customizing a pre–existing dictionary, is a valuable language resource for Language Processing and Technology. In both speech and text processing this dictionary is of immense utility. As we have created a dictionary for English words in Devanagari script based on this IE phonetic dictionary, many more such dictionaries in various Indian scripts can be generated. This in turn can be used in various NLP tools. There also is a need to have a thorough procedure of evaluation and validation of such dictionaries.

6. References

Maxwell, O. and Fletcher, J. 2010. "The acoustic characteristics of diphthongs in Indian English". *World Englishes.* 29:27-44. doi: 10.1111/j.1467-971X.2009.01623.x.

Gargesh Ravinder. 2004. "Indian English: Phonology". In *A Handbook of Varieties of English, ed. E W Schneider, K Burridge, B Kortmann and R Mesthrie*. pg. 992-1002. Berlin: Mouton de Gruyter.

Jones, Daniel. 2004. "The Pronunciation of English". Cambridge University Press.

Jones, Daniel. 2003. "English Pronouncing Dictionary". Ed. Peter Roach, James Hartman & Jane Setter. Cambridge University Press.

CMU Dictionary. Available online: http://www.speech.cs.cmu.edu/cgi-bin/cmudict?

Advanced research projects agency. 1993. In *Proceedings of TIPSTER Text Program (Phase I)*. Morgan Kauffman.

Advanced research projects agency. 1996. In *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kauffman.

Levithan S., Goyvaerts J. 2009. *Regular Expressions Cookbook: Detailed Solutions in Eight Progamming Languages.* O'Reilly Media.

Development of an Online Repository of Bangla Literary Texts and its Ontological Representation for Advance Search Options

Sibansu Mukhopadhyay, Tirthankar Dasgupta[1,2], Anupam Basu[1]
[1]Society for Natural Language Technology Research, Kolkata
[2] Indian Institute of Technology Kharagpur, India
E-mail: sibansu@gmail.com, iamtirthankar@gmail.com, anupambas@gmail.com

Abstract

This paper presents the design and development of an online repository of Bangla literary texts written by eminent Bangla writers. This will allow large readers of Bangla language to access a huge storehouse for Bangla literary resources in Unicode to read them online. Moreover, such a large collection of Bangla literary texts will help numerous computational linguists to perform different interesting language related analysis and build linguistic applications. The paper also proposes the major role of ontological design to establish a knowledge sharing system within building a grand literary corpus of Bangla language. Finally, the paper presents the ontological representational schema for storing such a large collection of data that may help different search engines to efficiently retrieve the different

Keywords: Bangla Online Repository, Rabindra Rachanabali, Ontology

1. Introduction

Due to the advancement in the area of computer science, artificial intelligence and machine learning, numerous novel applications are being identified by researchers that can take the potential benefits of the use of an electronic corpus as a source of empirical language data for different types of analysis and development. One of the key issues regarding this is the collection of large data. Moreover, the data should be in such a format so that it can be easily read and processed by any standard data processing tools available. Thus, it is evident that collection of such a huge corpus is an extremely important task behind any resource building effort for developing any applications related to the field of natural language processing and data mining. In order to get some specific linguistic information, the corpus is needed to be well annotated. This annotation of corpus may be performed at different levels of granularity namely; phonological, syllabic, lexical (part of speech), phrasal (as far as chunking is concern), syntactic, semantic and also discursive. However, the level of granularity and annotation depends upon the underlying application. Moreover, there are certain applications that require different stylometric information about a given document like, name and gender of the author of a document, year and place of publication etc. Such type of document specific metadata is typically not present in the existing corpora, we are referring to. Hence, developing a sizable amount of corpus that can be used by different applications for varied purposes is an extremely important task.

It is well-known that Bangla belongs to the eastern Indo-Aryan languages. The language along with its different dialects as well as varieties is widely spoken in different parts of India that includes West Bengal, Tripura, and Assam. Further, Bangla is the national language of Bangladesh. It has been estimated that more than 230 million people speaks in Bangla. It is considered to be one of the most frequently spoken languages ranking fifth in the world[1]. In spite of such a huge population of Bangla writers and speakers, sadly the amount of electronic text document, in editable format, available in Bangla language is very limited as compared to other languages spoken by less number of people.

Based on the above mentioned issues the primary objective of this paper is to develop a large collection of online repository of literary texts of eminent Bangla litterateurs. Presently we have developed and uploaded the complete works of Rabindranath Thakur and Bankim Chandra Chattopadhyay. The purpose of this online resource is not only to allow common people to access these literary documents freely, but also these collected documents can be ready to be used as a potential source of Bangla language repository that can be used for various text analysis and text mining purpose. Moreover, as the corpus has been built with the writings of different Bangla litterateurs, we have also provided different metadata information about the author and the document as a whole for different stylometric analysis of the text documents.

The rest of the paper is organized as follows: In section 2 we have discussed about the technique and perspective of building the Bangla repository and this section discusses about the hierarchical orderings of the text documents. Section 3 concludes with remarks. Future possibility may also include various phonemic and part-of speech analysis of the input text corpora.

2. Repository for Bangla Literary Resources

Bankim Chandra Chattopadhyay (27 June 1838 – 8 April 1894) and Rabindranath Thakur (7 May 1861 – 7 August 1941) are the two representative icons in the Bangla literary tradition. Thus, the collected works of both Bankim Chandra and Rabindranath have the immense importance to all the Bangla literature readers throughout the world. In this paper we present the schema deployed in the online development and representation of the complete works of both stalwarts. This repository of their complete literary works have been created in a Unicode compliant way so that any other interested person can freely download and use it for further analysis and

[1] "Statistical Summaries". Ethnologue. 2005. Retrieved 2007-03-03.

processing works[2]. Developing such a huge repository will enable readers:

- To access all works of Bankim Chandra Chattopadhyay and Rabindranath Thakur on-line
- To use the necessary texts from their different works
- To search for different information about the writings
- To search for different lines and text segments across the texts
- To have access to the different information and commentaries about the works

The system is using the UTF-8 encoding for data storing. UTF-8 (8-bit Universal Character Set/ Unicode Transformation Format) is a variable-length character encoding of byte codes and character assignments for UTF-8 is backwards compatible with ASCII. The total work is being digitized manually by using Unicode compatible software.

The details of the Rabindra-rachanabali and Bankim-rachanabali (such coinage is used by the publishers of West Bengal or Bangladesh to recognize the collected works of any writers in a thumbnail) have been depicted in Table 1 and Table 2 respectively. From Table 1 we can observe that, Rabindranath wrote vast literature comprising of different types literary pieces such as novels, poems, short stories in Bangla. For example, he wrote 13 novels, more than 2000 poems, more than 150 short stories, near about 2000 lyrics, 70 dramas and 51 collections of essays along with letters, English writings and huge amount of composition for his lyrics. Bankim Chandra Chattopadhyay has also written 14 novels and so many essays, etc, but his writings are not as the varieties and quantities like Rabindranath.

The total work of Rabindranath is classified into eight principal categories, such as; novels (upanyAsa), stories(galapA), dramas (nAtaka), songs (gAna), poems (kabitA), essays (prabandha), letters(cithipatrA) and collections (achalita sangraha). We also have another option to incorporate the English writings (ingreji rachanA) of Rabindranath, in which there are also such typological classifications. On the other hand, for Bankim Chandra's writings, we have to deploy a simple sketch of classification, because his works are classified, according to the edition, into two major categories, i.e., Novel and Other Works, although there are many subcategories like essays (satirical or historical or religious), short articles, fictions, dramas, introductory notes for various Bangla books, letters and even poems are incorporated in the 'Other Works'.

	Rabindra Rachanabali
Novels	13
Stories	150
Play	70
Songs	2000
Poems	2000
Article &Letters	51
	Bankim Rachanabali
Novel	14
Other Works	269

Table 1: Details of the Rabindra-rachanabali and Bankim-rachanabali Text Corpora

	Rabindra Rachanabali	**Bankim Rachanabali**
Number of words	5800000	253000
Number of unique words	173000	101182
Number of content words	141000	91000
Number of sentences	63582	35379

Table 2: Analysis of the Two Text Corpora

We have followed a two-tier categorization for the literary works. One tier represents the names of the head categories and second level representation is for the members lying under one head. Every category has many members. For example, there are thirteen novels written by Rabindranath. Each novel is a member of the category 'novels (upanyAsa)'. In the similar manner, the category 'stories (galapa)' has its members; 'dramas (nAtaka)' has its own and so on. Each category of "Rabindra-rachana" (writing of Rabindranath) has been structured and designed under the different schemas. For every category we have created specific data-structure that shows inter-relations between the prime members of the category. Information of different aspects like member's classification, publication date, place where the script were written, further modification of the script, characters of novel or story, any other application of the script like conversion to cinema or a stage performance or other else, has been populated essentially into *a shared data-base* Gruber (1993).

For the sake of indexing of the rabindra-rachanabali website we have also followed different schemes according to the aspectuality as well as functionality of the literatures. For example, for novels (*upanyas)* we have considered to design the structure according to their name where as for stories (*galapa)* it is considered that after clicking "galpa" readers will see the name story collections rather than the list of stories because of their huge number. In case of *natak* (dramas) of Rabindranath, the classification has been created according to their 'form' and 'type' and the classification of drama enhanced. This is illustrated in the following subsections.

2.1 Classification of Drama, Songs and Poems

The category like song (*gAna*) and poems (*kabitA*) has extremely large number of members, so that the task of a 3rd level indexing of these categories is very difficult. For indexing songs, we have followed the content distribution of Gitabitan published by Visvabharati. According to Gitabitan, the content of the songs (commonly known as Rabindrasangeet) has been developed through two types of indexing simultaneously. One is according to its theme and the other is according to the alphabetical order of first line of songs. Each of those category-heads precedes the

[2] The entire repository is available at www.rabindra-rachanabali.nltr.org & www.bankim-rachanabali.nltr.org

names of the items. For thesong category, there are altogether 15different head classes namely “prema”, “prakriti”, “pujA”, “pujA-o-prakriti”etc. Each of the head class is further classified into several subclasses depending on the number of entries. Moreover, there are several entries of one class that also belongs to another class or subclass of the given hierarchy. This is illustrated in Figure1(a). From figure1(a) we can observe that, the “prema” class under the “gAna” category, has got some entries that also belongs to the “prema-o-prakriti” class. Further, there exist a number of cross-category linkages between items on two different classes or categories. For example, figure 1(a) shows the hierarchical representation of the song “gAna” category and figure 1(b) represents the hierarchical representation of the drama category. However, in the song category there is a class called “gitinAtya-o-nrityanAtya” and in the drama category we have two different classes “gitinAtya” and “nrityanAtya”. Both the two classes of drama maps to the same class “gitinAtya-o-nrityanAtya” of songs. This marking of intra and inter category cross linkage of different category classes are illustrated in figure 1(a) and figure 1 (b).

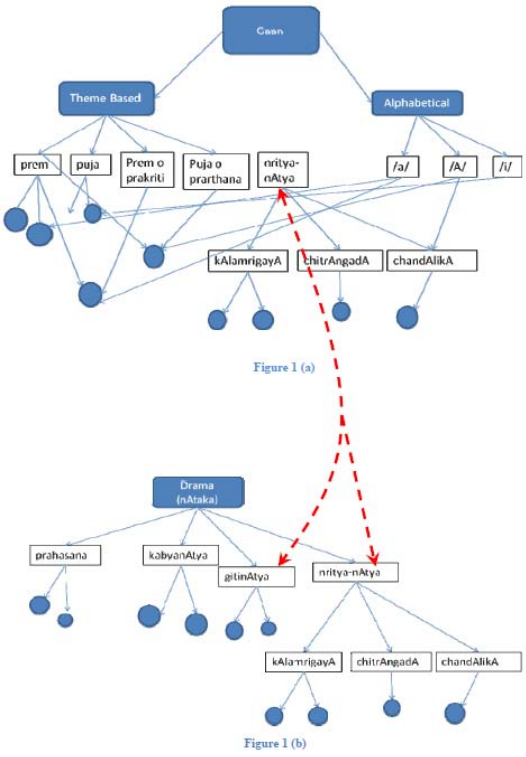


Figure 1(a) and 1(b): Illustrates the hierarchical representation of the two different categories of Rabindra-Rachanabali namely, Drama (nAtaka) and Songs (gAna). The figure also illustrates the cross category relationships between different classes of the two different categories.

On the other hand, in case of drama written by Rabindranath, we have added an extra level which first describes the conceptual nature of the dramas and then goes to the names of the items. For example, clicking the tab ‘natak’ (drama) one first get a classification that comprises ‘natak’ (traditional drama) ‘prahasan’ (farce) ‘hasyakoutuk’ (comedy of manners) ‘byangakoutuk’ (satire) ‘gitinatya’ (kind of ballad) ‘nrityanatya’ (dance-drama) ‘kavya-natya’ (poetic-drama) and ‘natya-kabita’ (dramatic poetry). And each of these classes provides the name of the items like the other categories.

Most considerable task in these repositories is to work out the designs based upon the conceptualization of the events. And this explicit specialization of texts instigates us to accept the challenge of the future work on inter-relationality between the genres.

2.2 Hierarchical (Taxonomy) ordering of Rabindranath's writing

The concept of populated schema for using classification of instances in respect to ontology mapping is best applied in Kalfoglou and Schorlemmer (2002). Populated schema for the complete work of Rabindranath is characterized by augmenting different events with classified relation, which describes the instances to the concepts. The following taxonomical figure is showing the framework for categorization of Rabindranath's writing developed in this ontological conceptualization. The hierarchical structure shows an ISA chain (ISA hierarchies are used to classify and represent domain concepts (Cheung, C. and Salagean, 2006)).

There are obviously more relations, we are just sampling examples for our purpose. We can establish the relationship (an example deployed in the Appendix-I) with the specific structure of data which is populated manually. We have done a two-folded task on structuring and categorization of the digital repository of the complete works of Rabindranath Thakur and Bankim Chandra Chattopadhyay. The first significant task in this respect is a schema design for an ontological structure of the Rabindra-rachanabali and Bankim-rachanabali, through which an explicit specialization of the writings or conceptualization of the events can be described and represented. This schema includes the relationship between the writing events of the litterateurs and the different information as well as commentaries about their works. The second task is, depending on this schema design we have classified and illustrated the total works and created WEB-based GUI of the websites for those collected works. As an illustration, consider the organization of the Rabindra-rachanabali. Here, we have created a head tag called the <*author*> under which the name of the author is present. Under the <*author*> tag, we have defined eight different tags based on the different categories of work like, <*story*>, <*drama*>, and <*songs*> as written by the author. Further, each of the categories is further divided into different subcategories depending upon the theme of the work. For example, the category <*song*> is further classified into 18 different sub classes. The entire organizational structure of the works of Rabindranath Thakur is depicted in figure 2. Here, the variables (x1, x2,…, xn) are used to denote leaf node, i.e., the texts and the caps are used for the immediate higher node of the leaf nodes.

In order to address the issue of sharing common understanding of the structure of information among the people from the literary website, we have followed the ISA chain concept.

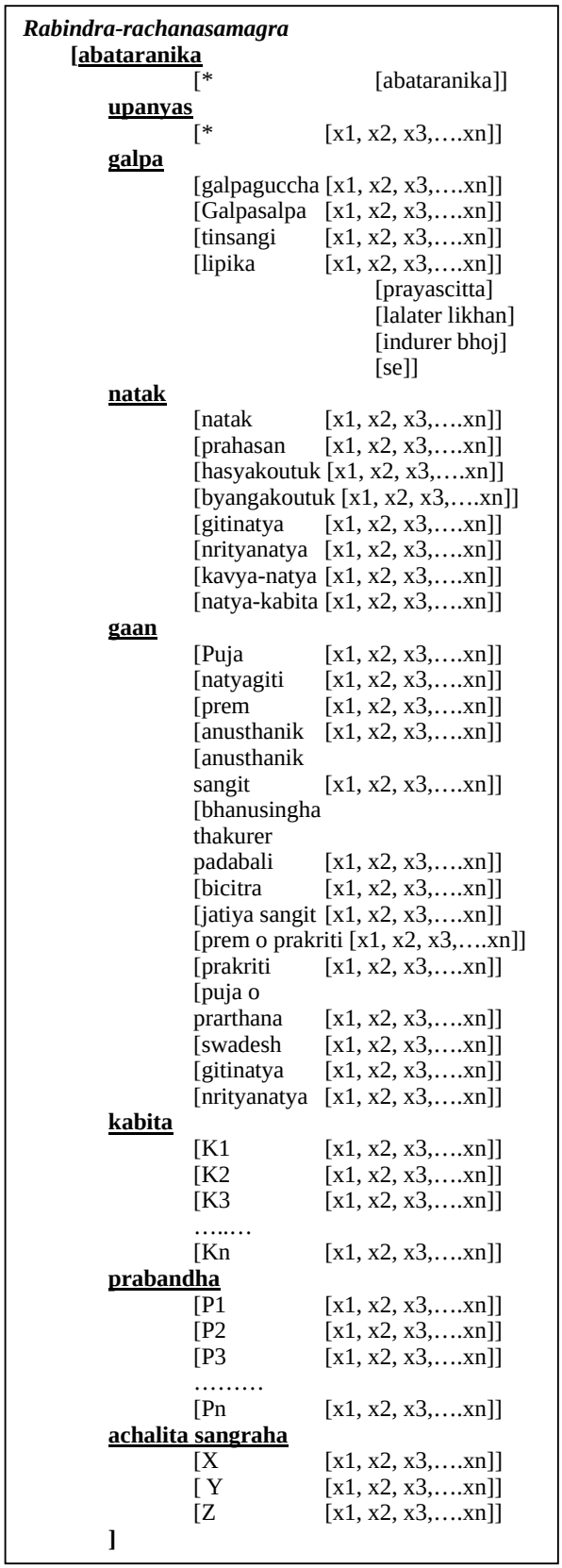


Figure 2: Hierarchical organization of the works of Rabindranath Thakur

Depending on the ISA-chain concept we can draw a directional map to show sample of ontology for understanding the structure of information of an author's writing. These are the fields like novel's name, published on, tribute to, first published on, first published in, characters, theme, either developed as a feature film; film_name, film_director, film_producer, role_players, songs used in the cinema, and appreciations or criticisms. Each author is associated with the directional map.

3. Conclusion

In this paper we have presented the development of an online repository of Bangla literary texts written by eminent Bangla writers. This will not only allow readers of Bangla language to access a huge storehouse for Bangla literary resources in Unicode to read them online, but also such a large collection of Bangla literary texts will help numerous computational linguists to perform different interesting language related analysis and build linguistic applications. Further, the paper also discuss about a representational schema for storing such a large collection of data that may help different search engines to retrieve the different metadata related to the author and the document efficiently.

References

Cheung, C., Salagean. (2006). A set theoretic view of the ISA hierarchy. In Proceedings of the *19th international conference on Advances in Applied Artificial Intelligence: industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 127-136.

Gruber, T. R. (1993). A translation approach to portable ontologies. *Knowledge Acquisition*, 5(2). pp.199-220

Kalfoglou, Y., Schorlemmer, M. (2002). Information-Flow-Based Ontology Mapping.In proceedings of On the Move to Meaningful Internet Systems 2002 : CoopIS/DOA/ODBASE2002. pp. 132-1151

Challenges in Sanskrit-Hindi Adjective Mapping

Kumar Nripendra Pathak
Special Centre for Sanskrit Studies,
Jawaharlal Nehru University, New Delhi -110067
nri.pathak@gmail.com

Abstract

In this paper author is describing the adjective identification and in handling process theoretically for Sanskrit-Hindi Machine Translation (SHMT). In the diverse linguistic scenario in India, MT is needed to transfer the knowledge from one language to another. The overwhelming literary superiority of Sanskrit has attracted intellectuals worldwide and attempts to translate desired Sanskrit texts into other languages have been made since 17th century. Both the languages differ at various levels. Transferring Sanskrit linguistic features to Hindi is similarly challenging given the structural nuances that Hindi has developed in the course of its evolution. In order to provide comprehensible translation, an MT system should be capable of identifying an adjective and map it correctly into Hindi.

1. Introduction

In Sanskrit-Hindi Machine Translation (SHMT), adjective mapping is required to be handled carefully. It needs the detailed discussion on Sanskrit adjectives and the resultant came from the discussion can help a lot in identifying the Sanskrit adjective to adjust it according to the target languages structure. Sanskrit language has relatively free word-order feature and thus it differs from the structure of Hindi language where adjective comes before the substantive. For ex- *bālakaḥ capalaḥ gacchati- capala bālaka jātāhai*. Here *capala* is an adjective of *bālaka.* In Sanskrit, itis kept after the substantive and in Hindi, it is kept before the substantive.Generally, adjective and substantive have the same number in Sanskrit but there are few exception as well where we find some mismatches as *vedāḥ pramānaṃ*, *śataṃbrāhmaṇāḥ*etc. The different types of adjectives can be identified to handle the adjective mapping issues to produce respectively good output in SHMT.

2. Related Work

Dash and Gillon (1995)[1]say that there are two ways to categorize words: on the basis of properties intrinsic to words and on the basis of relations which words bear to be other words, or even extra grammatical entities. First way leads to the categories of noun, adjective, verb, preposition etc.; while the second way leads to such categories as predicate and modifiers. Sanskrit, being an inflectional language, requires essentially morphological criteria for determining membership in lexical categories. While nouns and adjectives, both take declensional inflection, nouns have gender intrinsically, while adjectives do not.

One element modifies another when the two elements are co-constituents in a constituent of which the later element is the head. It turns out that, in general, nouns, adjectives and prepositions can be modifiers in compounds. In every compound in Sanskrit which is not a co-ordinate, the non-head element modifies the head element. Even in phrase structure both adjectives and nouns can be modifiers. Adjectives agree with the nouns they modify.. Structural conditions for agreement of adjectives with nouns are quite precise: an adjective agrees with a noun in case, number and gender when the adjective is the head of an adjective phrase which modifies the noun (where the relation of modification is taken in its unextended sense). Relation of predication is the relation which the verb phrase of a clause bears to the subject noun phrase of the same clause. In *rāmaḥ sītāṃ paṣyati*, *sītāṃ paṣyati* is the predicate of the subject *rāmaḥ*. But in *strīsundarī,* the adjective *sundarī* is a predicate of the noun *strī* and there is no verb. in this condition, assuming null copula in absence of VP, we can say that the adjective *sundarī*is a predicate of the noun *strī.* Dash and Gillon (1995) present empirical support to the assumption of a phonetically null copula and shows that the adjective agrees with a noun of which it is a predicate or modifier (in phrase structure).

[1] The Adyar Library Bulletin, pp. 285-294, 1995

Adjectives (modifiers) seem to be nouns in absence of modified nouns. In *dīnaṃ pālaya*,*dīnaṃ*is performing the role of noun as this sentence contains a phonetically null noun/pronoun. Null noun/pronoun serves as a head noun in a NP in which the bare adjective is modifier. Thus Dash and Gillon (1995) conclude that a noun is provided for the adjective to modify, thereby supplying a noun phrase to be the subject of the sentence. They further give a template for this type of construction saying that these may be considered as compound. In this template, [N {Adj.} {Nominal}], head of such compound is a noun, so the compound itself has the status of the head.

Dr S D Joshi did the classification of nouns into adjectives and substantivesin his paper titled *Adjectives and Substantives as a single class in the "Parts of Speech"*[2]. The structure of Sanskrit is such that it makes a clear distinction between verbs and nouns which can be defined semantically as well as morphologically. A verb denotes activities in process which consists of definite sequence of beginning middle and end. Nouns, on the other hand, do not denote a process but frozen actions in the form of a substance. Particles serve to reveal the relation existing between two different words, whereas the prepositions serve to specify the meaning of the verb. Grammatically *upasargas* are always connected to verbs while *nipātas* shows relation between the different words. The distinction between *upasarga* and *nipātas* is not structural or morphological but it is functional. This functional classification corresponds to the grammatical fact that prepositions are always connected with verbs in the sentence and bring out inherent signification of verbs whereas the *nipātas* are regarded as word-connectives or sentence connectives. this division of words is based on the combined aspect of form, meaning and function, and it is suitable to the structure of Sanskrit.

According to Pānini, prepositions, particles, adverbs, adjectives fall under the category *nāman* or *prātipadika*. In his descriptive categorization, Pānini includes *upsarga* and *nipāta* under the single category 'noun'. As mentioned earlier, noun presents the static notion and verb presents the process of happening, thus all adjectives, pronouns, conjunctions and indeclinables are noun. The adverbs are grouped under the category of nouns, because structurally they have similar forms with nouns. To Bhartṛhari, pronouns might be a subclass of nouns or adjective. It is either restrictive adjective or stands for the things in general. Thus they function either like adjectives or like noun, and when they restrict the sense of substantive, they are adjectives.

The adjectives and substantives have the same inflection in Sanskrit. The main difference one may note is that the former class varies in three genders, as it is shown by their agreement with substantives. The substantives have definite gender[3]. But there are some exceptions pointed out in *vyutpattivāda* where *vedāḥ pramānaṃ*, *śataṃbrāhmaṇāḥ*etc are shown as examples. Here we find the disagreement between adjectives and nouns in either their gender or number, or their gender and number.

Joshi (1996) states that Pānini does not define the terms *viśeṣaṇa* and viśeṣya semantically or structurally but used the term in the rule II.1.57 (which means: a case inflected word standing for a qualifier is compounded with a case-inflected word standing for a qualificand .For example - *nīlotpala (nīlā kamala), raktotpala (lāla kamala)*etc. Here *nīla* and *rakta* are used as modifier which differentiate the blue and red lotuses from the white lotus etc). He further says that Patañjali also finds it difficult to explain the term *viśeṣaṇa* and *viśeṣya.* The difference between *viśeṣaṇa* and *viśeṣya* lies in the point of view which we put forth. When the notion of qualifier and qualified is purely subjective with the reference to the wish of the speaker, adjectival notion can be easily turned into the substantive notion. Therefore Patañjali gives suitable terminology to point out the intrinsic difference between the adjectives and substantives. He calls adjectives *guṇavacanas*: 'denotative of qualities', that are found in substances and terms substantives (*viśeṣya*) *dravyavacana*: 'denotative of substances'. Pānini has also used the term *guṇavacana* in sense of qualifying attribute[4]. The grammarian's term *guṇa* stands for the attribute or qualities separable or inseparable from the substances. Here the term*guṇavacana* means any attributive word which serves to distinguish one from the other which is cleared by the earlier example of *nīlotpala*, *raktotpala* etc. but this concept of*guṇavacana*and *dravyavacana*is not useful for the *karmadhāraya* compound prescribed for the *viśeṣya-viśeṣaṇa* words because*karmadhāraya* compound is formed with the two *dravyavacana* as well as two

[2]Journal of the University of Poona, vol.25, pp.19-30, 1966.

[3]yalliñgaṃyadvacanaṃ yācavibhaktirviśeṣyasya, talliñgaṃtadvacanaṃ sācavibhaktirviśeṣaṇasyāpi. (samāsacakraṃ)

[4] Panini II.1.30;VI.2.115.

guṇavacana. For example, in *āmravṛkṣaḥ*, we find two *dravyaāmra* and *vṛkṣa* respectively. Similarly *śuklakṛṣṇaḥ* has two *guṇavacanaśukla* and *kṛṣṇa* respectively. The terms *guṇavacana* and *dravyavacana* have definite meaning and they cannot be normally interchangeable. When the notion of *viśeṣya-viśeṣaṇa-bhāva* is purely dependent upon *vivakṣā*, a classification of adjectival words into *guṇavacana* and *dravyavacana* can handle the non-compound construction in language processing. The grammatical fact of agreement or concord between the adjectives and substantives is well brought by Patañjali by classifying them logically as denotative of qualities and substances.

Joshi (1996) also presents the views ofBhartṛhari and Helārāja from Vākyapadīya. The *guṇa* cannot be conceived without *guṇī*. Here the *guṇa* is modifier and thus, Helārāja says: *bhedakatvaṃ cātramukhyaṃguṇalakṣaṇaṃ*. Bhartṛhari, while discussing the term *viśeṣya* and *viśeṣaṇa*, pointed out that *viśeṣya-viśeṣaṇa* is a syntactic category and not a morphological or grammatical one. The designatives*viśeṣya* and *viśeṣaṇa* refer to a word as a member of combination and not as an isolated individual. If *nīla* and *ghaṭa* are not put together in combination but used separately, they only denote the nominal notion. But when it is used in combination, nīla becomes the modifier of the *ghaṭa.*

It is also presented by Joshi (1996) that the modifier is used not only for differentiation but also for the identification. In *prameyoghaṭaḥ,* the universal attribute prameya, 'knowledge', cannot eliminate anything because there is no unknowable worldly object. In such case adjectives do not indicate the difference but simply identify the object with its qualifier. The qualifying word differentiates the individuals of the same class and not individuals of the other classes. In nīlo *ghaṭaḥ, ghaṭa* is representing the whole class of *ghaṭa* and nīla is differentiating it from the other possible *colours*.

So we conclude that 1) the main difference between adjectives and substantives is that the substantives have a fixed gender while the adjectives vary in gender and number following the substantive character. And 2) the adjective is subordinate while the substantives areprimary.

3. Challenges in adjective mapping

After a conceptual discussion, we can move towards few examples which are related to the identification of adjectival *pada*, the problem of case handling in adjectives and gender aspect affecting the adjective.

Ex- sundaraḥ bālakaḥ.

In this example, the padasundaraḥ and bālakaḥ have same case marker which shows two possibilities 1) sundaraḥ is the name of the said boy (bālakaḥ) and 2) the padasundaraḥ is an adjective of the pada bālakaḥ (boy). In this analysis, it is known to us that the pada bālakaḥ is substantive. But a machine gets confused in identifying the substantive and adjective padas in the given input text because both have same case markers.

Moreover, adjectives in Hindi, either drops the case marker or takes the oblique form in the output (translation) of the input/given text.To show the gender affect in the given text, we can say that sundarī bālikā is translated as sundara laḍakī in Hindi which shows that sundaraḥ and sundarī, both has the same translation sundara in Hindi.

These aspects of natural language are the challenges in machine translation which needs great effort to identify and discuss the different angles of a particular problem.

3.1 Identification of adjectival pada

In Sanskrit, except some compound nominals, the place of adjective is not fixed. In compounding, the adjective and substantive is observed by the commentators on the basis of *vācakatva*. If *jātivācaka/saṃjñāvācaka* word is kept with the *guṇavācaka* and *kriyāvācaka* then *guṇavācaka* and *kriyāvācaka* will be the adjectives in those compounds. In *nilotpalaṃ*, *nilaṃ* is qualitative and hence this is the adjective of *utpala* which is *jātivācaka.* In *pācakabrāhmaṇaḥ, kriyavācakapada pāchakaḥ* is the adjective of *brahmaṇa* (*jāti/saṃjñā*). (When*kriyāvācakapada* will be the adjective, then Hindi will take the *wālā* construction). But when two qualitative words or the *kriyāvācakapadas* come together, there is no certain position for adjective.

Three types of nominal can function as an adjective: *subanta, kṛdanta and taddhita*. If two *subantapadas* are coming together as *samānādhikaraṇa* then anyone may be adjective. Same as, if two *kṛdantapadas* are coming together then also the place of adjective *pada* is uncertain. If *kṛdanta* and *subantapada* comes together, then the *kṛdantapada* is adjective and *subanta*is substantive. If

kṛdanta/subanta and*taddhita* is coming together, then *taddhita* is the adjective and the *kṛdanta/subanta* is substantive. As *taddhita* never becomes the adjective of a pronoun, when *taddhita*is used with a pronoun, pronoun will be the adjective and *taddhita* will be the substantive.

3.2 Case Marking and Word order

There are two issues in adjective mapping: the case marking and the word order. The first difference between Sanskrit and Hindi adjective is that an adjective and a substantive in Sanskrit have the same case marker while adjective in Hindi does not show the case marker accept oblique forms in pronominal adjectives and the adjectives with ā-ending. As- *tena bālakenakathitaṃ= usa bālaka ne kahā.* In *madhuraṃphalaṃ dadāti*, adjective and substantive has the same case marker in Sanskrit, but in *mīṭhāphaladetāhai,* adjective *mīṭhā* has no case marker. But the plural of *madhuraṃphalaṃ, madhurānīphalānī*(*dadāti*), (*mīṭhe phaloṅkodetāhai*), has oblique form with the adjective in Hindi. In *sundaraḥ bālakaḥasti,* adjective and substantive has the same case marker in Sanskrit, but in *sundara bālakahai*, adjective *sundar* has no case maker. So here we see that the case marker is dropped. The NP *sushilā bālikā*can be translated as *sushilā bālikā* as well as *sushila laḍakī*. Here it can be noticed that when *bālikā*word is not changed with its synonyms, we can keep*sushilā* as an adjective of *bālikā*. But *sushilā laḍakī* is not used in Hindi, instead of *sushil laḍakī*. In Hindi, *acchī laḍakī* cannot be pluralized as *acchiyāṅ laḍakīyāṅ* instead of *acchī laḍakīyāṅ.* So the claim of same number with adjective and substantive cannot be true in Hindi while this is applied in Sanskrit. We can see another example: *teja laḍakā*and *teja laḍakī*, where the gender of adjective word is not affected by the gender of substantive. As it is mentioned earlier that the adjectives and substantives have the same inflection in Sanskrit, the chief difference one may note is that the former class varies in three genders, as it is shown by their agreement with substantives. The substantives have definite gender[5]. But there are some exceptions pointed out in *vyutpattivāda* where *vedāḥ pramānaṃ*, *śataṃbrāhmaṇāḥ*etc are shown as examples. Here we find the disagreement between adjectives and nouns in either their gender or number, or their gender and number. So these challenges need to be address to get the correct output.

The next issue in mapping adjective is the place of adjective. In Sanskrit, adjective has no fixed place. In Sanskrit, it can be kept either before or after the substantive. In Hindi, adjective is kept before the substantive (i.e. Adjective + Noun) but the adjectival clause has no fixed place. So leaving the adjectival clause aside, we should focus on the adjective (non-adjectival clause) where we have, first, to identify the adjective in SL which will be arranged according to the Hindi output. One more thing can be notices that *bālakadvaya*types of constructions have numeral adjective where adjectives are attached with the substantive and making a single word. Here we have to separate the adjective from the substantive to handle the output.

4. Conclusion

On the basis of above discussion, it can be said that:

1) The main difference between adjectives and substantives is that the substantives have a fixed gender while the adjectives vary in gender and number following the substantive character.
2) 2) The adjective is subordinate while the substantives are primary.
3) The adjectives can be classified on the basis of 1) three nominal categories: *subanta, kṛdanta and taddhita*. and 2) *jātivācaka/saṃjñāvācaka, guṇavācaka* and *kriyāvācaka.*
4) The judgment of the nominal category in the given sentence and an exhaustive database of*jātivācaka/saṃjñāvācaka, guṇavācaka* and *kriyāvācaka* can help us in handling adjectives in SHMT.

5. References

Aklujkar, Ashok (1996): *Some theoretical observations on word order in Sanskrit,* Festschrift Paul Thieme, Stll, (1996), S.1-25.

Dash, Siniruddha and Brenden Gillon, (1995): *Adjectives in Sanskrit,*TheAdyar Library Bulletin-1995.

Deshpande, Madhav M. (1987); *Paninian Syntax and the changing notion of sentence,* in Hock (1991), Studies in Sanskrit Syntax: A Volume in Honour of

[5]*yalliñgaṃyadvacanaṃ yācavibhaktirviśeṣyasya, talliñgaṃtadvacanaṃ sācavibhaktirviśeṣaṇasyāpi. (samāsacakraṃ)*

the centennial of Speijir's Sanskrit Syntax, MLBD, New Delhi.

Deshpande, Madhav M. (1990); *Semantics of Karakas in Panini: an exploration of philosophical and linguistic issues*, in Sanskrit and related studies: contemporary researches and reflections. Matilal (Et.al.) Sri Satguru Publications, 33-57.

Joshi., S.D. (1966): Adjectives and Substantives as a single class in the "Parts of Speach", Journal of the University of Pune, Vol. 25, pp.19-30.

Joshi., S.D. (2001): *Synyactic and semantic device in Aṣṭādhyāyī of Pāṇini*, in journal of Indian Philosophy 29: 155-167, @ 2001 Kluwer Academic Publishers, Printed in the Netherlands.

Pathak, K. N. and G. N. Jha (2011): *Challenges in NP Case-Mapping in Sanskrit-Hindi Machine Translation*, in C. Singh (et al.) (Eds.): ICISIL 2011, CCIS 139, pp-289-293, 2011, @ Springer-Verlag Berlin Heidelberg 2011.

Subash and Girish N. Jha (2005). *Morphological analysis of nominal inflections in Sanskrit*, Presented at Platinum Jubilee International Conference, L.S.I. at Hyderabad University, Hyderabad.

Hindi Web Page Collection tagged with Tourism Health and Miscellaneous

Pattisapu Nikhil Priyatam, Srikanth Reddy Vaddepally, Vasudeva Varma

International Institute of Information Technology
Hyderabad, Andhra Pradesh, India
nikhil.priyatam@research.iiit.ac.in , srikanthreddy.v@research.iiit.ac.in , vv@iiit.ac.in

Abstract

Web page classification has wide number of applications in the area of Information Retrieval. It is a crucial part in building domain specific search engines. Be it 'Google Scholar' to search for scholarly articles or 'Google news' to search for news articles, searching within a specific domain is a common practice. **Sandhan** is one such project which offers domain specific search for *Tourism* and *Health* domains across 10 different Indian Languages. Much of the accuracy of a web page classification algorithm depends on the data it gets trained on. The motivation behind this paper is to provide a proper set of guidelines to collect and store this data in an efficient and an error free way. The major contribution of this paper would be a *Hindi* web page collection manually classified into *Tourism*,*Health* and *Miscellaneous*.

1. Introduction

Web search in Indian languages is constantly gaining importance.With the fast growth of Indian language content on the web, many classic IR problems (Web page classification, focused crawling, Ranking etc) need to be addressed. Though most of these problems seem to be solved for high resource languages like English, the solutions cannot be applied to Indian languages because of dearth of resources. Indian languages have rich morphological features which can be used to solve the problem in a better way.Hence preparation of Indian language resources for IR tasks is very important.

Problems like WPC require huge amount of training data to ensure diversity and coverage. This requires huge amount of manual labor. Moreover for these algorithms to work accurately, the training data should be error free. A good training dataset should be an actual representation of the real data. In an ideal scenario the training and testing documents should come from the same distribution. But generally this does not happen since the distribution is unknown to us. One of the features offered by **Sandhan** is *domian* specific search. In this context we define *domain* as a category of web pages which satisfy a particular user need. Currently our search engine is expected to support only two domains namely *Tourism* and *Health*. The domain identification module is supposed to classify a web page from the crawl as *Tourism* or *Health* or *Misc* (neither *Tourism* nor *Health*).
The purpose of this work is to collect good quality training data for Domain Identification module.
Documents belonging to the tourism domain are supposed to contain the following information:

- Information about historical places
- Tourist attractions in India
- Travel Guide which includes cost of trip,best time to visit, transport facilities, nearby hotels, accommodation facilities, and nearby places of interest of a particular tourist spot.
- Food and other popular cuisines.
- On line services (if any) provided by the respective tourist spots.
- Latest news about a particular place which might include weather reports and any other news alerts.

Documents belonging to the health domain are supposed to contain the following information:

- Complete information about any disease: Symptoms, precautions, cure and other related information.
- Information about medicines, their ingredients, possible side effects, availability and the type of medicine (*ayurvedic,allopathy,homeopathy etc*) .
- Information about clinics or hospitals, doctors etc.
- Latest research news about some ailments.
- Information related to nutrition, diet, personal hygiene etc.

We used the proposed guidelines to collect, classify and store *Hindi* web pages. These web pages contain information about Indian tourist attractions, General health topics and *Miscellaneous* (neither *Tourism* nor *Health*).

2. Background

Several areas in Information Retrieval like Web page classification (WPC) and focused crawlers (FC) require Machine Learning algorithms for training their classifiers. Significant amount of research has gone into exploring different kinds of algorithms and right feature combinations which would work, but not enough work has gone into what qualifies to be a good training data and how to collect it. No algorithm can learn anything significant on junk.

2.1. Web Page classification

Web Page classification is the task of categorizing web pages into different classes. This is an extended problem of document classification. Document classification is a fundamental learning problem that is at the heart of many information management and retrieval tasks (Power et al.,

2010).

A document classification works on plain text as compared to the rich set of features that can be explored in web pages. Most of the approaches use machine learning algorithms to solve this problem. Irrespective of which algorithm one uses, accuracy of a classification algorithm depends on the feature sets. Web page classification uses wide number of features from the simplest like URLs (Baykan et al., 2009) to the complex ones like topic models (Sriurai et al., 2010). In addition to these features there are also many common features that these algorithms use. Some of them are:

- In links
- Out links
- Page size
- Content
- Title

and any valid combination of these features. Mainly 3 types of algorithms are used to solve this problem.

2.1.1. Supervised Algorithms

Supervised algorithms are one specific class of learning algorithms which infer a function from supervised (labeled) training data (Duda et al., 2001). These are often called as classifiers. Once the classifier learns on the labeled data, it predicts the label of the unlabeled data. The most common supervised algorithm is the Naive Bayes algorithm (Duda et al., 2001).

2.1.2. Semi-supervised Algorithms

Semi-supervised learning is a learning paradigm concerned with the study of how computers and natural systems such as humans learn in the presence of both labeled and unlabeled data (Zhu and Goldberg, 2009). There also exist methods which use large unlabeled samples to boost performance of a learning algorithm when only a small set of labeled examples is available (Blum and Mitchell, 1998).

2.1.3. Unsupervised Algorithms

Unsupervised learning algorithms do not require any training data. These algorithms work by calculating the similarity between different samples and clustering them under one group. Multilingual document clustering is one such area. (Steinbach et al., 2000) compares different document clustering algorithms.

Be it a single algorithm or a mix of different algorithms, the labeled web page data is crucial to solve this problem. Even for unsupervised methods labeled data will always help in determining quality of a cluster (Steinbach et al., 2000).

2.2. Focused Crawlers

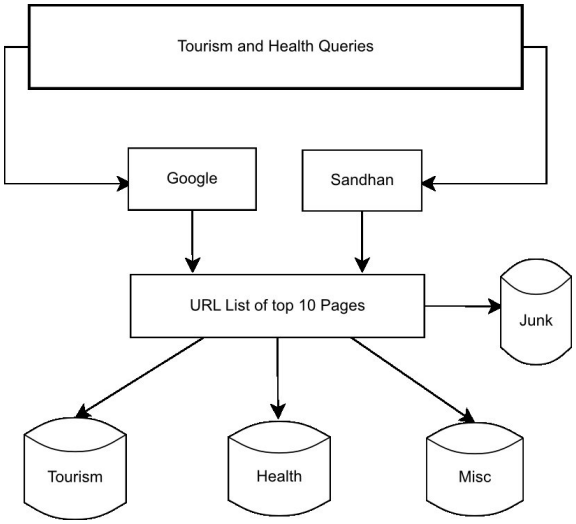
Focused crawlers are the ones which selectively seek out pages that are relevant to a pre-defined set of topics. Rather than collecting and indexing all accessible Web documents to be able to answer all possible ad-hoc queries, a focused crawler analyzes its crawl boundary to find the links that are likely to be most relevant for the crawl, and avoids irrelevant regions of the Web (Chakrabarti et al.,). To achieve this task focused crawlers need some training either on a predefined ontology as in (Chakrabarti et al.,) or tagged URLs.

3. Data Collection

For web page classification we require real web pages to be classified into respective domains. In our dataset we identify each web page with its URL classified into *Tourism*, *Health* and *Misc*. Since we cannot pick random pages from the web and tag them, a systematic approach needs to be followed. The Following section describes our approach.

3.1. Approach

Our data collection process can be logically broken down into 3 steps: Query collection, Firing them on a Search Engine and Classifying retrieved results. The data collection process can be represented in the form of a flowchart.



3.1.1. Query Collection

This is a very crucial part of our system. If queries are not proper and diverse enough we cannot expect our training set of URLs to be an actual representation of entire data. For query collection we have 30 people from different places in India speaking 10 different languages. Everyone of them is explained what qualifies to be a *Tourism* query and *Health* query and then each person is asked to prepare 3 set of queries:

- Regional Queries: These are tourism queries about specific places or locations where the person belongs.
- Non regional Queries: These are also tourism queries but are quite generic in nature (not related to the place where the person belongs).
- Health Queries: Any health related query.

Along with the queries the person is asked to provide English translation of the queries and the intent of the query.

First let us consider only *Tourism* queries, By collecting queries in this way we are ensure that the coverage is diverse enough. Note that one person's regional query can be others' non regional query, so we have both local as well as global coverage of queries. Even for *Health* queries, by collecting queries from different people we ensure sufficient diversity in the health queries too. Once we have the queries we translate all of them into ***Hindi*** for our experimental purpose. The translation is done manually so that no error creeps in this part. After translation we eliminate all duplicates (if any). We also make sure that all the queries are collected in isolation i.e. no one knows queries given by other people. The following table gives the split up of the queries.

Type of queries	No of queries
Regional	60
Non-Regional	60
Health	60

3.1.2. Firing Queries on Search Engines

Once we have the above set of queries we use a search engine to get the relevant web pages. Search engines like Google usually provide diverse results i.e. results from different hosts. For our work we actually fire the queries on Google as well as **Sandhan** to ensure wider coverage and pages for diverse websites.

3.1.3. Classifying Retrieved Results

The results retrieved for each query are manually looked at and classified into one of the 3 defined domains as per the set guidelines. The top 10 results for each of Google and **Sandhan** are considered for manual classification. More than 10 results were also considered for some queries whenever the top 10 results had many miscellaneous pages. The pages going into *Misc* category are the ones returned by the search engine in response to a *Tourism* or a *Health* query but belong to neither of the domains. Therefore we are not making any specific efforts to fetch Miscellaneous pages, because our main interest is to get *Tourism* and *Health* pages only. For our current work we do not do a further analysis on sub-domains of Tourism and Health. It has to be noted that a page is classified as *Tourism* or *Health* if it strictly falls within our definition. It need not be completely relevant to the fired query. Pages which have no or meagre text are discarded. The following table shows the statistics of the data collected.

Domain	No of Web Pages
Tourism	885
Health	978
Misc	1354

4. Data Storage

Storage of Indian language datasets needs to overcome various challenges. This problem is much more pronounced when it comes to storing web page collections. This is because Indian language content authors use very proprietary (Non standard) character based encoding formats alongside with the standard Unicode format. For proper distribution and machine readability Unicode is most preferable. We store the URLs (not the content) and its corresponding tag. This is to ensure that when we have the data ready to be used for Web Page Classification, we can crawl the collected set of URLs and extract any features that we wish (which may include features other than raw text like images,titles etc). Since all of these pages are not in the standard Unicode format, we provide Unicode converted pages along with the dataset.

We have chosen *SQLite3* [1] as the storage media. *SQLite3* is a portable, self-contained database. According to their website, it is the "most widely deployed database engine in the world". Also there are no licensing issues when using *SQLite3*, since it's in the public domain [2]. *SQLite3* has the following advantages over traditional storage mechanisms like flat files, Xml etc.

- *SQLite3* is a zero configuration fully featured RDBMS system. It fully supports the SQL 92 standard.
- This allows easy alteration like updates, inserts, deletes etc.
- All data is stored in a single file which makes it distributable. *SQLite3* also allows encryption of the database file if the data has to be distributed in a secure manner.
- The open storage standard of *SQLite3* allows it to be used independent of any operating system.
- *SQLite3* allows storage of Unicode data. It supports UTF-8, UTF-16BE and UTF-16LE [3].
- APIs are available in various programming languages which makes it easy to consume the data or convert it into any required format.

4.1. Table Structure

Our dataset is stored in a single *SQLite3* database with 2 tables namely URL_Tags and URL_Content. Table 1 shows the structure of URL_Tags. The URL_Content table provides the content of all the URLs after converting to Unicode. Table 2 shows the structure of this table.

One observation about our data is that even though our queries were purely *Tourism Queries* or *Health queries*, the number of general or miscellaneous documents outnumber the tourism and health documents. This behavior is consistent in both Google as well as **Sandhan**. This shows that most of the pages do contain tourism or health related keywords but do not belong to *Tourism* or *Health* domain, in specific. This might also be a consequence of our strict categorization policy. Miscellaneous pages are as important as tourism and health pages because in the task of web page classification they play the role of negative samples.

[1] http://www.sqlite.org/different.html

[2] http://www.switchonthecode.com/tutorials/php-tutorial-creating-and-modifying-sqlite-databases

[3] http://www.sqlite.org/datatype3.html

Table 1: URL Tag table structure

Column Name	Data type	Notes
Doc_id	INT	A numeric id for each web page
URL	TEXT	The URL of the web page.
Lang	TEXT	A 2 character ISO language code
Domain	TEXT	Domain tag of the web page

Table 2: URL Content Table structure

Column Name	Data Type	Notes
Doc_id	INT	A numeric id of a web page as given in table 1 (foreign key constraint)
Unicode Content	TEXT	UTF-8 HTML content obtained by converting proprietary encodings if any. The Text datatype can comfortably store large web pages.

4.2. SQLite3 API

SQLite3 is an open source database with API support in many languages. PHP 5 has in built support for *SQLite3*. This is the version that we have used for building our data collection tool. The following PHP code fragment shows how to issue a SELECT query to an *SQLite3* database.

```
<?php
$db_file = 'database_file_path';
$table_name = 'Some_table_Name';
$db = new SQLite3($db_file);
$query='SELECT * FROM $table_name';
$results = $db->query($query);
while($row = $results->fetchArray
(SQLITE3_ASSOC))
{
  \\ Process a single row.
}
$db_close();
?>
```

The following are some sample SQL statements that can be used to retrieve data from our dataset.

- Aggregate information: Selects all the information of a particular document by combining both the tables.

  *SELECT * FROM url_List,url_Content*

- Selecting Hindi tourism URLs.

  SELECT url FROM url_List WHERE lang='hi' AND domain='tourism'

Since *SQLite3* fully supports the SQL92 standard, any valid SQL statement can be used to retrieve data from the database.

5. Conclusion and Future Work

We conclude that this mechanism of data collection is simple and efficient. It minimizes manual effort and reduces errors. We also discuss an efficient storage mechanism which is easy to use and distributed. We also believe that the tagged *Hindi* web page collection prepared by us will find great use in further research on Web page Classification and Focused crawling.

We plan to collect data in a similar fashion for all other Indian languages and evaluate it. We also plan to further minimize manual effort by using these URLs as input to a semi supervised focused crawling algorithm. We plan to provide data quality evaluation statistics by using measures like cluster quality, classification accuracy etc for all our datasets. In future, we wish to extend this approach to do a much more fine grained analysis on the sub domains of Tourism, Health on the same data.

6. References

Eda Baykan, Monika Henzinger, Ludmila Marian, and Ingmar Weber. 2009. Purely url based topic classification. In *World wide web poster*, April.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co training. In *COLT' 98 Proceedings of the eleventh annual conference on Computational learning theory*.

Soumen Chakrabarti, Martin van den Berg, and Byron Dom. Focused crawling: a new approach to topic-specific web resource discovery. *Computer Networks*, 31.

Richard O. Duda, Peter E. Hart, and David G. Stork, editors. 2001. *Pattern Classification*. John Wiley and Sons. 2nd edition.

Russell Power, Jay Chen, Trishank Karthik, and Lakshminarayanan Subramanian. 2010. Document classification for focused topics. In *Association for the Advancement of Artificial Intelligence*.

Wongkot Sriurai, Phayung Meesad, and Choochart Haruechaiyasak. 2010. Hierarchical web page classification based on a topic model and neighboring pages integration. In *(IJCSIS) International Journal of Computer Science and Information Security*.

Michael Steinbach, George Karypis, and Vipin Kumar. 2000. A comparison of document clustering techniques. Technical Report 00-034.

Xiaojin Zhu and Andrew B. Goldberg. 2009. Introduction to semi-supervised learning. June.

Treatment of Tamil Deverbal Nouns in BIS Tagset

S.Arulmozi, G. Balasubramanian & S.Rajendran
Dravidian University, Kuppam Amrita Vishwavidyalayam, Coimbatore

Abstract

In Modern Tamil, the major Parts-Of-Speech (POS) categories are nouns, verbs, adjectives and adverbs. Of these, verbs take nominal suffixes such as *–al, -tal, -kai*, etc. to form "verbal nouns". Formation of a verbal noun is a regular and productive process in Modern Tamil. In this paper we attempt to examine the place of verbal nouns in Tamil language as well as in the Bureau of Indian Standards (BIS) guidelines for POS annotation with reference to Tamil language. We present a brief introduction to the ILCI Corpus and Tamil language followed by a detailed study of the POS Tagsets for Tamil in general and verbal nouns in particular. The peculiar behaviour of verbal nouns which made them to be subcategorized under verb is discussed elaborately.

1. Introduction

The Bureau of Indian Standards (BIS) has recommended the use of a unified Parts-Of-Speech (POS) standard in Indian languages (Draft Standard – Version1.0) for the annotation of corpus. The BIS tagset has a total of 38 annotation level tags which are common to all Dravidian languages. Out of the 38 tags, 11 tags are at the top level, 32 tags are at level 1 (subtype) and verbs have 4 tags at level 2 (subtype) and conjunction has 2 tags at level 2 (subtype).

2. ILCI Corpus

The Indian Languages Corpora Initiative (ILCI) is a research project for the Technology Development in Indian Languages (TDIL). The main objective of the project is to build annotated corpora from Hindi into 11 Indian languages along with English with standards for 12 major Indian languages in the domain of health and tourism. The major aims of the project are to evolve draft standards, build parallel corpora and also manually annotate parallel corpora.

3. The Tamil Language

The Tamil language is the oldest, richest and one of the classical languages of the Dravidian family of languages which are agglutinative in nature. The major Parts-Of-Speech categories in Modern Tamil are Nouns, Verbs, Adjectives and Adverbs. Nouns in Tamil inflect for number and case marking and verbs inflect for tense and person-number-gender suffixes. In Tamil, verb forms can be distinguished into finite, non-finite and nominalized verb forms. Finite verb forms can be inflected for tense (past, present and future), mood (imperative, optative and indicative), aspect (perfect and continuative) and subject agreement (marking person, number and gender). Non-finite and nominalized verb forms are inflected for tense – in some cases – and for the relational category which indicates the subordinate or nominal status of the verb.

4. Tagset for Tamil

1. Noun (N)

The Noun category has three tags at the subtype level1, viz. Common Noun, Proper Noun and Nloc in Tamil.

Sub-level	Tag	Example
Common	N_NN	puttakam, kaNNaaTi
Proper	N_NNP	aruL, ravi, maalati
Nloc	N_NST	meel,kiiz,mun,pin

2. Pronoun (PR)

Pronouns are divided into five sub-categories at the sub-level, viz. personal pronouns, reflexive pronouns, relative pronouns, reciprocal pronouns and wh-word pronouns.

Sub-level	Tag	Example
Personal	PR_PRP	naan, nii, avan
Reflexive	PR_PRF	taan
Relative	PR_PRL	yaar, etu, eppootu
Reciprocal	PR_PRC	avanavan, oruvarukkoruvar
Wh-word	PR_PRQ	yaarum, etuvum

3. Demonstrative (DM)

The Demonstratives have three tags at the sublevel, viz. deictic, relative and wh-word.

Sub-level	Tag	Example
Deictic	DM_DMD	anta, inta
Relative	DM_DMR	enta
Wh-word	DM__DMQ	enta, yaar

4. Verb (V)

The verb category has three tags at the sublevel, viz. Main (which has finite, non-finite, and infinitive and gerund as subtypes - level2), verbal noun, and auxiliary (which in turn has subtypes, non-finite and infinite at level2).

Sub-level	Tag	Example
Main - Finite	V_VM_VF	vizhutaan, pooneen, cirittaaL
Main – Nonfinite	V_VM_VNF	vizhunta, poonaal
Main – Infinitive	V_VM_VINF	vizha, pooka, cirikka
Main-Gerund	V_VM_VNG	vizhutal, cirittal,
Verbal Noun	V_NNV	paTittal, varukai
Auxiliary	V__VAUX	aakum, veeNTum

5. Adjective (JJ)

Adjectives have no sub-category and the tag is given below:

Sub-level	Tag	Example
Adjective	JJ	niRaiya, periya, azhakaana

6. Adverb (RB)

Adverbs also have no sub-category and this tag mainly covers manner adverbs.

Sub-level	Tag	Example
Adverb	RB	veekamaaka, viraivaaka

7. Postposition (PSP)

Postposition has only one tag and the detail is given below:

Sub-level	Tag	Example
Postposition	PSP	paRRi, kuRittu

8. Conjunction (CC)

Conjunctions are sub-divided into two namely, coordinator and subordinator (this has a sub-type under level2, i.e. quotative).

Sub-level	Tag	Example
Coordinator	CC_CCD	um, maRRum, aanaal, allatu
Subordinator	CC_CCS	enRu, ena, enpatu
Subordinator - Quotative	CC_CCS_UT	ena, enRu

9. Particles (RP)

Particles have five tags are the sub-type level1 viz. Default, Classifier, Interjection, Intensifier and Negation.

Sub-level	Tag	Example
Default	RP_RPD	maTTum, kuuTa
Classifier	RP_CL	not required
Interjection	RP_INJ	aayyoo, teey
Intensifier	RP_INTF	ati, veku, mika
Negation	RP_NEG	illai

10. Quantifiers (QT)

Quantifiers are sub-divided into three at the sub-type level1 viz. General, Cardinals and Ordinals.

Sub-level	Tag	Example
General	QT_QTF	konjam, niRaiya,
Cardinals	QT_QTC	onRu, iraNTu
Ordinal	QT_QTO	mutal, iraNTaam

11. Residuals (RD)

Residuals have five tags at the sub-type level1

and they are: Foreign word, Symbol, Punctuation, Unknown and Echowords.

Sub-level	Tag	Example
Foreign word	RD_RDF	
Symbol	RD_SYM	
Punctuation	RD_PUNC	
Unknown	UNK	
Echowords	RD_ECH	vaNTi kiNTi, paal kiil

5. Verbal Noun

The verbal nouns mentioned here are a special type of nouns derived from verbs (e.g. *paTittal* 'the act of reading', *vaaztal* 'act of living'). These deverbal nouns are differentiated from the other types of deverbal nouns such as *paTippu* 'education' (derived from the verb *paTi* 'to study' by suffixing *-ppu*), *vaazkkai* 'life' (derived from the verb *vaaz* 'to live' by suffixing *-kai*). The first types of deverbal nouns have both nominal and verbal characters. But the second type of deverbal nouns has only the nominal character; they do not retain the verbal character. Here verbal character means the character of these nouns to retain the argument structure of the verbs; and the nominal character means the capability of getting inflected for case. The following example will illustrate this point.

1a. *avan puttakam paTikkiRaan*
'he book read_PRE_he'
'He is reading a book'

1b. *avan puttakam paTittalai virumpukiRaan*
'he book reading like_PRE_he'
'He likes reading books.

1.c. *avan paTippai viTTuviTTaan*
'he education leave_PAS_he'
'He has left the education'

1d. **avan puttakam paTippai viTTuviTTaan*

From the example 1b it is clear that *paTittal* retains the argument structure of the verb *paTi*. At the same time examples 1c and 1d show that *paTippu* does not retain the argument structure of the verb. The first type of deverbal nouns is productive in their formation and the meaning of them can be predicted. But in the formation of the second type of deverbal nouns the formation is not productive and the meaning of the nominal forms cannot be predicted; they involve a large number of suffixes. The suffixes that are involved in the formation of these two types of deverbal nouns are listed into two sets.

First set of nominal suffixes
-tal ~ -ttal -al ~ -kal ~ -kkal -kai~ -kkai
Second set of nominal suffixes
-am, al, -i, --ai, cal, -ccal, -ci, -cci, paan, -pu, -ppu, -mai, -vi, -vu, -vai, etc

The deverbal nouns in Tamil have been extensively studied by various authors including Arden (1949), Andronov (1969), Paramasivam (1971), Kamaleswaran (1974), Kothandaraman (1997), Rajendran (2001). Here in this paper we are making use of the term verbal nouns in a narrow sense i.e. deverbal nouns having both nominal and verbal characteristics. The second type of deverbal nouns can be modified by adjectives whereas the first one cannot be modified by adjectives. The following example will illustrate this point.

2a. avan nalla vaazvai virumpukiRaan
'he good life like_PRE_he'
'He likes good life'

2b. *avan nalla vaaztalai virumpukiRaan

In 2a *vaazvu* 'life' is modified by an adjective whereas, in 2b *vaaztal* does not permit the modification by an adjective.

Both the types can be inflected for case suffixes, but the first one with certain restriction which is discussed later. Here for the sake of POS tagging we consider only the first types of deverbal nouns as verbal nouns. The nouns from verbs having only nominal characters are considered as simply nouns in BIS categorization. The verbal nouns considered here include the following types:

(1) *-tal* suffixed verbal nouns.
E.g. *vaaztal* 'living'
paTittal 'reading'

(2) *-al* suffixed verbal nouns
E.g. *vaazal* 'living'
paTikkal 'reading'

(3) *-kai* suffixed verbal nouns
E.g. *vaazkai* 'living',
paTikkai 'studying'

Though these verbal nouns denote the same meaning, they are distributionally different: *-al* suffixed verbal nouns occurs before the auxiliary verb *aam* 'is'; *-kai* suffixed verbal nouns occur when inflected by the case markers *–ai, -aal, -ku,* etc. to denote the meaning 'while'; *-tal* suffixed verbal nouns occurs elsewhere but do not take case markers.

3a. *avan inRu inku varal-aam*
'he today here coming_is'
'He may come here today'

3b. *avan varukaiyil naan paartteen*
'he coming_LOC I see_PAS_I
'I saw him while coming.

3c. *avan kooyilukku varutal illai*
'he temple_LOC coming not'
'He does not come to temple'

These types of verbal nouns need to be differentiated from the non-productive deverbal nouns with care. For example *vaazkai* 'living' which belongs to the first group is different form *vaazkkai* 'life which belongs to the second group.

5.1 Verbal nouns vs. Gerundival nouns

Verbal nouns need to be differentiated from the gerundival nouns (which are usually referred to as participial nouns in Tamil grammatical tradition). Gerundival nouns are formed by suffixing *–atu* the tensed suffixed stems or by suffixing *mai* to the negative suffixed verbal stems or past/present tense suffixed relativized verbal stems. The following example will substantiate the point.

E.g.
paTittatu 'read_PAS_NOM
paTikkiRatu 'read_PAS_NOM'
paTippatu 'read_FUT_NOM
paTikaamai 'read_not_NOM
paTittamai 'read_PAS_REL_NOM
paTikkiRamai 'read_PAS_REL_NOM

All these nominal forms denote the meaning 'reading' with the addition of tense and negation. Like the first type of deverbal nouns these gerundival verbal nouns too have both verbal and nominal character; they are capable of retaining the argument structure of the concerned verbs; they are productive in their formation and the nominal meaning is easily predictable.

5.2. Verbal nouns vs. pronominalized verbal nouns

The first type of deverbal nouns needs to be distinguished from the pronominalized verbal nouns. The prominalized verbal nouns are formed by adding the third person-number-gender suffixes (portmanteau morphs) to the tense suffixed or negative suffixed verbal nouns.

Eg.
paTittavan 'read_PAS_he'
'he who read'
paTikkiRavan 'read_PRE_he'
'he who reads'
paTippavan 'read_FUT_he'
'he who will read'
paTikkaatavan 'read_not_he'
'he who did not read'

These pronominalized verbal nouns too have both nominal and verbal character. They too are capable of retaining the argument structure of the concerned verbs; they are productive in formation and the nominal meaning is predictable.

6. Summary and Conclusion

It is clear from the above discussion that the formation of nouns from verbs is of different types. They have to be separated mainly into two types: non-productive deverbal nouns and productive deverbal nouns. The productive devebal nouns are divided into at least three types: non-tensed non-negativized verbal nouns, tensed or negativized gerundival nouns and tensed or negativized pronominal verbal nouns. The first type of verbal nouns which are differentiated by the use of different suffixes (*-tal*, *-al*, *-kai*) are different distributionally too.

Because of the above mentioned reasons the BIS tagset treat the non-productive deverbal nouns from productive deverbal nouns. On the similar score verbal nouns, gerundival nouns and prominalized verbal nouns are differentiated from one another.

7. References

Arden, A.H. 1937. (4th edition). A Progressive Grammar of the Telugu Language. Madras: CLS Publication.

Andronov, M. 1969. A Standard Grammar of Modern and Classical Tamil. Madras: New Century Book House.

Kamaleswaran, S. 1974. Nouns in Tamil. Annamalai University: Ph.D.Thesis.

Kothandaraman, Pon. 1997. A Grammar of Contemporary Literary Tamil. Chennai: International Institute of Tamil Studies

Lehmann, T. 1989. A Grammar of Modern Tamil. Pondicherry: Pondicherry Institute of Linguistics and Culture.

Manning, Christopher D. Analysing the Verbal Noun: Internal and External Constraints.www.essex.ac.uk/linguistics/external/clmt/.../manning/verbnoun.ps accessed on 25.2.2012

Paramasivam, K. 1971. Verbal Nouns in Literary Tamil. In: V.I.Subramoniam (ed.). Proceedings of the First Conference of Dravidian Linguists. Trivandrum: DLA.

Rajendran, S. 2001. Typology of Nominalization in Tamil. http://www.languageinindia.com/nov2001/rajendran1.html accessed on 05.02.2012

Schiffman, H F. 1989. `Deverbal Nominal Derivation in Tamil', Paper presented to the American Oriental Society, New Orleans (Downloaded from web).

TDIL, Unified Parts of Speech (POS) Standard in Indian Languages, Draft Standard, Version 1.0.

Zvelebil, K. 1957. Verbal Nouns in early Old Tamil. Tamil Culture. 6:2:87-91.

TschwaneLex Suite (5.0.0.414) Software to Create Italian-Hindi and Hindi-Italian Terminological Databases on Food, Nutrition, Biotechnology and Safety on Nutrition : a Case Study

Silvia Staurengo
Foreign Languages Department, Hindi Language and Indian Studies
University of Studies of Milan, Milan, Italy.
E-mail: silvia.stau@gmail.com

Abstract

This paper describes the working of Tlex Suite to create an Italian-Hindi terminological database. This research aims to develop communication without the English medium "forced step" – where it is possible - between workers on the field, in the domain of food and nutrition. This tool can capture socio-cultural implications and information as well, publicize unknown aspects of both countries to non-technicians. In few years we would like to launch on the market wide range of merchandises from papers to multimedia Lexicon; Machine-readable dictionary; pc, tablets, smart phone's applications etc. Further, we would like to create Speech corpora, too. Right now, we are submitting our work on progress paper as we have just started the research. Here we have described some of the problems faced during the development of terminological database as a case Study.

Keywords: Hindi Database, Hindi-Italian Database, Food, TLex, TlTerm

1. Introduction

With the Development of modern technological resources, it is high time to start thinking about the creation of a functional database to support interdisciplinary channel of communication between Hindi and Italian language and vice versa. Moreover considering contemporary European situation; Italian-Indian Bilateral Dialogue has become more important for mutual cooperation, economical relations, dynamics, prospects and trends between the two countries. Basically, believing in the rich linguistic heritage of both Hindi and Italian languages, through this work, we bridge the gap the exists in intellectual exchanges without using English as an intermediary language. In order to broaden the future prospects along with retaining the value of project, we need to keep doors open for inter-communicability. Therefore, we are working on this specific field as part of a group of researchers of several languages in order to put together a multilingual database. Database's focus is on subjects like Food, Nutrition, Biotechnology and Safety on Nutrition. We would like to present our Hindi language study in a Bilingual/Multilingual Database, Text Corpora as well as Lexicons, Machine-readable Dictionary comprehending not less than 500 lemmas. Our goal is to make multimedia's merchandises like DVDs, CD-ROMs, pen drives, personal computers, tablets and smart phones applications to be ready and launch on the market by 2014. In future if we collaborate with Indian Institutes, Italian researchers of Hindi would also like to develop Speech Corpora on these fields.

With these assumptions we would like to come closer to Customers interests, we aim to cover all ranges of linguistics users – from technical to non technical – with a diversify spread proposal. TschwaneLex Suite (TLex) is one of the best software applications which offers various features to create glossaries and terminology lists. It contains the TLex, tlTerm, tlCorpus and tlReader. We have already released the 2010 Suite (5.0.0.414). In addition to TLex we have been suggested to start working with WordSmithTool 3 and AntConc for terminological citation from selected corpus on Language-2 to Language-1 (L2-L1) in comparable texts. Since our research is at a starting step in this direction, our team has chosen Italian as pivot–language, setting it as L1. After group consultation, Italian linguistic experts have selected, scanned and loaded an essay (code is E1L1 followed by initials of the name and the surname of the loader etc.) on this specific topic in .txt file on tlCorpus with full details about it, like Name, File Name, Language, Encoding, Size etc.

Linguistic researchers know that .txt format can be used perfectly with WordSmithTool, AntConc as well as TLex Suite. Even if researchers do not reject the idea to try and compare AntConc which can be freely downloaded, on the basis of E1L1 they have moved on terminological extraction with WordSmithTool 3 as it has all necessary applications. This terminological extraction from text (E1L1) is semi-automatic. In other words software choose significant lemmas/clusters – verbs, nouns and adjectives – both from grammatical criteria and frequent appearances in the corpus. Biggest advantage of utility of this application is that it is a time saver. Software extracts from terminological unit (TU) the related concepts whereas researchers create conceptual-trees or conceptual-constellations in order to delete repeated, or synonymous lemmas and add others which software cannot extract from the corpus. Since E1L1 has been loaded on tlTerm software, team had got a basic L1 Word List to work on including its Italian terminological partially filled papers given us as example. But at the same time, we are looking for essays, texts, magazines, brochures, publications and web sites are considered in our textual corpus.

Working on tlTerm terminological Hindi papers is facing difficulties in format it in .txt and load it on tlCorpus as well as alphabet writing in Devanagari script with font provided by the software. Therefore, we are trying to solve above mentioned problems and we are looking for proper OCR which is good for Devanagari. We searched on the web for the best one. Biggest names on this area

seem to guarantee good outcomes; however as there were more than 100 languages supported it was impossible to find out specific mention of Hindi or, at least, to evaluate and test. Now, we are going to buy an Indian OCR named *Chitrankan* that is specialized for Hindi and Marathi alphabets.

2. tlTerm: terminological database "builder".

As we have already stated that our research on the field is in the initial stage, we could not rush forward even with "project-timetable and formulas". Maybe at a later stage, we will be able to give online and stand alone demonstrations. In any case, right now, our presentation is based on step-by-step approach. We took the assumption that meta project develops from L1 pivot language in order to have enough lemmas for a basic shared L1 word list. TschwaneLex is software that consents to create on line monolingual, bi or multilingual Lexicon. In a multilingual database, as our, users are previously requested to work on the highlighted macrostructure of the project; nevertheless TlTerm is a flexible software. This allows authors to join a research in progress and gives all the necessary autonomy to shape the step-by-step – features. They could add, delete or change information either general – involving the meta structure of the database – or specific to their own area of study. Since it is an online work, all could be able to check changes on the research at the same time. Moreover, most important feature of tlTerm is that is fully reversible. In other words this software permits authors to work sharing and comparing outcomes simultaneously. If someone of the team would like to delete an entry from the database related entries will be deleted. This is useful in order to avoid mismatched lemmas in each language. In addition to this tlTerm consent to have statistical information on the work done in order to advise on deficiencies in parts of the Lexicon (Figure 1 and 2).

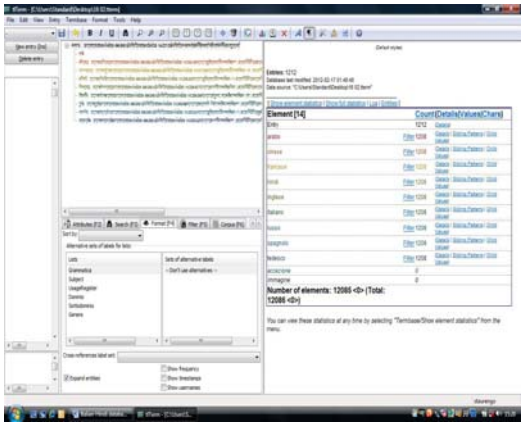


Figure 1: Termbase show element statistics

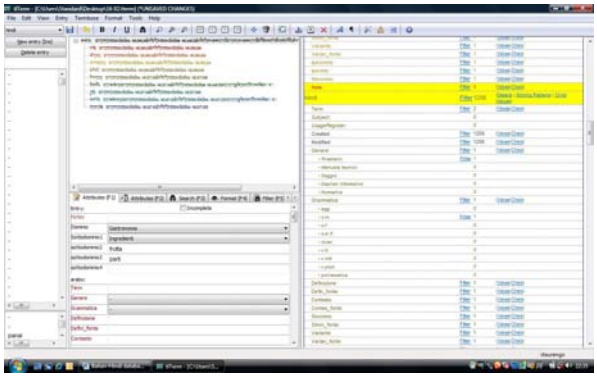


Figure 2: Termbase show element statistics (b)

On the other hand, tlTerm allowed us, in our specific area of research, to select and test several font from the tool bar format and tools (Figure 3 and 4).

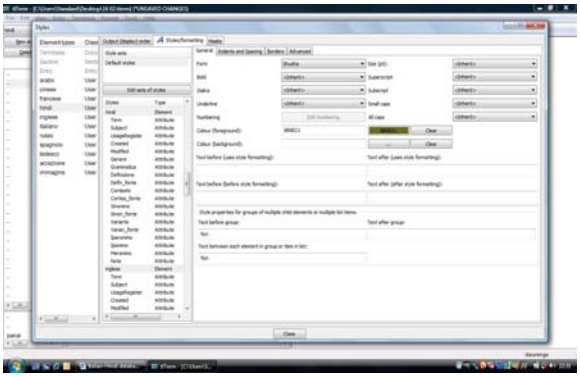


Figure 3: Format styles

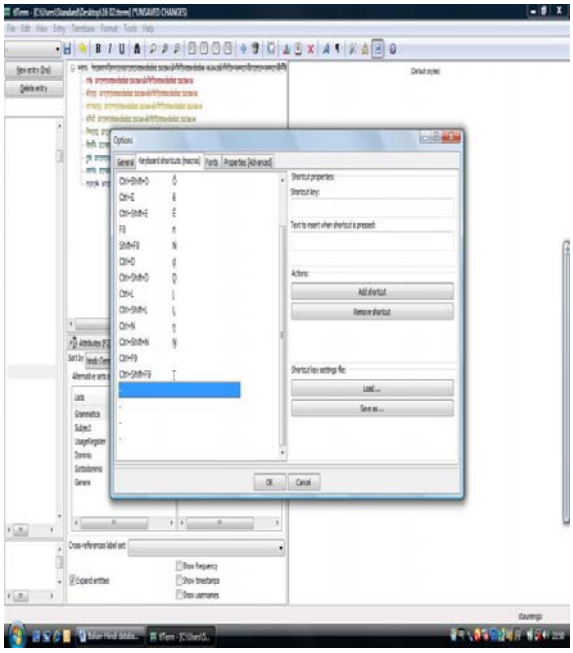


Figure 4: Tools Options

Since we decided to write on the screenshot by Devanagari font but faced problems we downloaded Shuhsa and started using it on tlTerm (we will talk more about the problems later). Once selected the entry and the language – Hindi – we started filling up TU or DTD (Document Type Definition) from L1 basic word list. We were able to see L2 and L1 work done on this entry simultaneously (Figure 5)

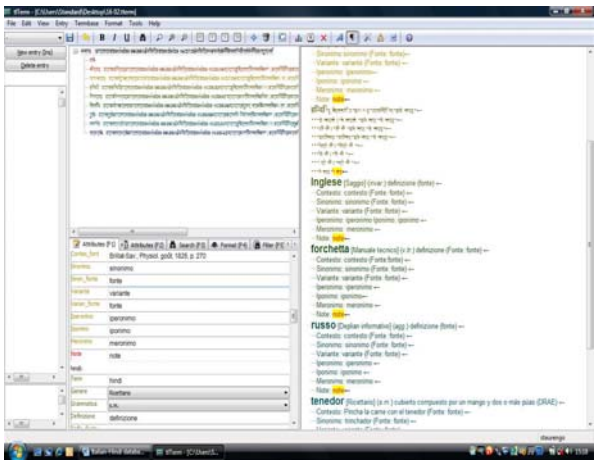


Figure 5: Output after manual entering

Right now, we would like to show a new entry in order to illustrate how Hindi database looks like.
We choose as an example the lemma ('acqua' in Italian, 'water' in English) as given in Figure 6.

पानी [Ricettario] (s.*m.*)
acqua [Ricettario] (*s.f.*)
Gastronomia Ingredienti bevande

Further on we will detail about TU. Right now we would like to draw attention to the logical work on tlTerm; in fact on this software product outcome proceeds on the assumption "what you see is what you get". This is very helpful for us in order to decide best settings and avoid mistakes on transliteration.

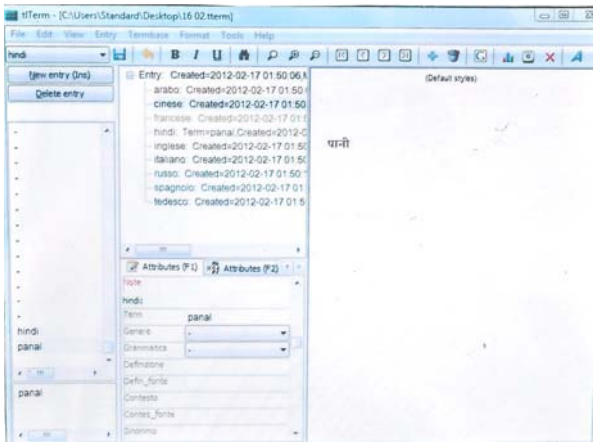


Figure 6: Lemma entry in the database

Now we would like to give an exhaustive review of the TU or DTD as per terminological paper on tlTerm. First of all three fields must be stated: Dominion, Under Dominion, Kind of the Text. The outcomes of our example will be like these:

Dominion: Foods
Under Dominion: Ingredient
Kind of Text: Recipes.

The TU file will be in the format as shown below.

Term: पानी
Grammar: noun/verb/adjective and number (preselected options)
Definition: right now on this field we are using an instrumental one. Later on we will coin our own definition.
Source: of the definition.
Context: where this term is found (i.e. in a sentence) in a text of our corpus. Here is: इसमें पानी और नमक डाल कर और फैंट लीजिये (मिश्रण के अन्दर गुठलियां न रहें).
Source: of the context. (Here: web site and date of consultation) www.nishamadhulika.com:19th Oct 2011
Synonyms:
Source:
Variants: regional, dialectal variants.
Source:
Hyponims: Under categories of TU
Hypheromims: Hyper categories of TU
Meronims: part of TU (if TU have it)
Equivalent: corresponding term in L2 (Sanskrit, Hindustani, others)
Source:
Notes: anything about this term's peculiar usages on L2. It should be in Italian (as par guidelines) but we have preferred to write down few Hindi proverbs in Devanagari. It should be helpful to comprehend linguistic-cultural cross information related to the countries.

Later on we will add more data such as images etc. to enrich our final product. TlTerm could support additions of sound files in the DTD (Document Type Definition) and is reversible in multiple editions. Saved changes are also exportable in file as HTML, RTF, in text or list of words etc. (Figure 7). Researchers could use results in order to create several products offers.

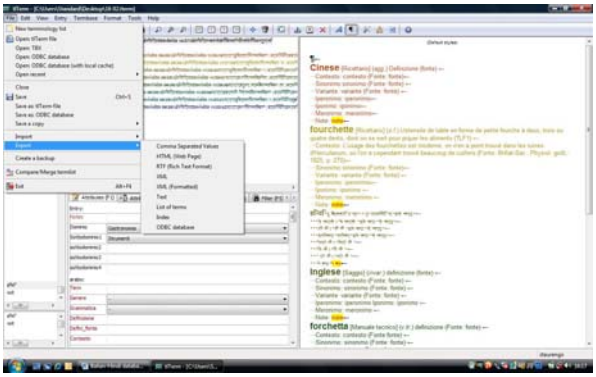


Figure 7: Menu showing file export

3. TLex Italian-Hindi terminological database on Food: A case study

Terminological database is a shared project; however,

Hindi area is facing peculiar problems on it:

1) **words and sentences of a file in HTLM or RTF are improperly translated:** If we would select on tlTerm "Copy file in HTML" the result on HTML will be for instance:

acqua [Ricettario] (s.f.)
paanI [[Ricettario] (s.m.)
vahI paardarSak drava haotaa haO jaisako […]

As you could see below we were expected to have such a final result:

पानी [Ricettario] (s.m.)
Gastronomia Ingredienti Bevande
वही पारदर्शक द्रव होता है जिसके […] ;

in a RTF file TU's elements are fully translated. Terms in brackets are converted in Devanagari script strictly from Italian (Figure 8)

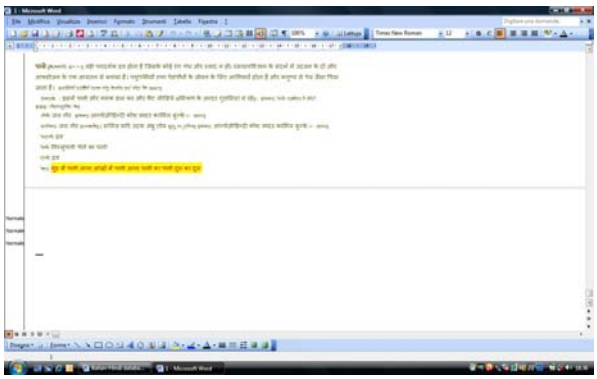


Figure 8: Translation from Italian to Hindi script in the tool

2) **Loading texts on tlCorpus**: As stated before we are buying a good OCR for Hindi. It is necessary to work with texts. Our corpus, however, include web sites and recipes from the web. Some of these converted in .txt lost or changed syllables of Hindi words (i.e. के लिये became के लईठए). Since this happened with a Windows 7, we have tried with Windows XP support and we had good results (i.e. the source and the context of usage of represented TU) as given below (with TU data marked):

भरवां रवा इडली

[...]

आवश्यक सामग्री

- रवा (सूजी) - 300 ग्राम (1 1/2 कप)
- दही - 300 ग्राम (1 1/2 कप)
- पानी - 50 ग्राम से कम (1/4 कप)

[...]

बनाने की विधि

सबसे पहले दही को फैट लीजिये. किसी बर्तन में सूजी छान लीजिये और उसमें दही डाल कर अच्छी तरह मिला लीजिये. इसमें पानी और नमक डाल कर और फैट लीजिये (मिश्रण के अन्दर गुठलियां न रहें).

[...]

पिट्ठी बना लीजिये

आलू को छील कर बारीक तोड़ लीजिये, कढ़ाई में तेल डालकर, हरी मिर्च और अदरक डालिये, पालक डालकर नरम होने तक पकाइये, आलू और नमक डालिये और अच्छी तरह मिला दीजिये, इडली में भरने के लिये पिट्ठी तैयार है. कुकर में 2 छोटे गिलास पानी डालकर आग पर रख दीजिये पानी को गरम होने दीजिये. [...] (www.nishamadhulika.com:19th Oct 2011);

3) Writing settings: TLex supports projects on several languages, in fact this software supplies an huge font and options selection. Since Hindi words in Devanagari require syllabic compositions exceeding what is provided by the keyboard, special settings are of the maximum importance to our work. We have selected and tested three different fonts. Two – Devanagari and Microsoft Himalaya – were available on tlTerm styles formatting and the third one – Shuhsa – was downloaded from the web. End results gave similar errors. We could not set proper options and we were not able to use Microsoft word integration. In addition software started segmenting syllables which presented elements under or above the line. This happened frequently with 'o', 'au', 'ra' and 'ṛi'. E.i.:
वही पारदर्श क द्रव हो ता है जिसके […]

All these inconveniences have delayed our timetable. We have decided to fill some terminological paper/file for effective solutions. We have wrote in Devanagari on gmail and pasted texts on our TU (Figure 9)

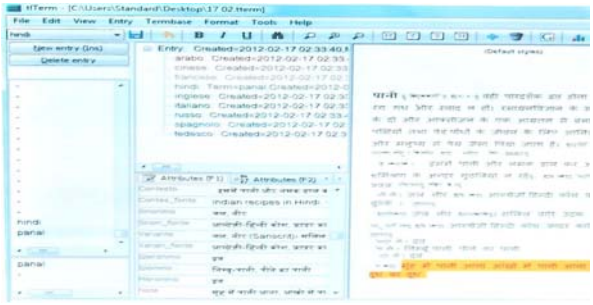


Figure 9: pasted devanagari text from gmail

Case study on our research, right now, is to strike a balance on theory and practice**.** The software provides helpful examples on line; although they clearly state that the applications are not available on our present version. The system is mostly for non technical terms and needs quick basic guidelines in order to have an effective feedback on the research.

Building Large Scale POS Annotated Corpus for Hindi & Urdu

Shahid Mushtaq Bhat, Richa
Linguistic Data Consortium for Indian Languages (LDC-IL)
Central Institute of Indian Languages
Manasagangotri, Mysore-570006
E-mail: shahid.bhat3@gmail.com, rsrishti@gmail.com

Abstract

Creation of annotated corpus is very essential for the technology development in natural languages. In terms of such resources, languages can be resource rich like many English or resource poor like many Indian Languages (ILs). The former have enough technology at their disposal predominantly due to the availability of large scale annotated language resources while the latter have lagged behind due to poor resource scenario. Although, the work to build resources for ILs started very late as compared to their European counterpart, it is gaining momentum now-a-days in the form of various projects. In this paper we describe our experience of developing POS annotated corpus for Hindi & Urdu (69.7K Hindi words & 66.4K Urdu words). Though we carried out annotation both manually as well semi-automatically using LDC-IL tagset, but for building POS annotated corpus with BIS standards, we made use of the existing corpora, tagged as per LDC-IL annotation Scheme, by mapping the tags from this scheme to those of the contemporary BIS Scheme. This resulted in aforementioned quantum of annotated corpus as per BIS Standards.

Keywords: Corpus, POS Tagset, POS annotation, Guidelines, MAT, SPMT, Inter-annotator agreement, Validation

1. Introduction

POS annotation is essentially a classification problem where words are classified & labelled on the basis of some predefined parts-of-speech scheme. For some languages, with split-orthography[1], it is also a mapping problem which involves mismatch in the mapping of the arrays of tokens on the arrays of tags in proper agreement with the syntactic structure of the language. In the entire pipeline of NLP for morphologically rich languages, POS tagging plays a important role of syntactic category disambiguation.

Creating a large scale POS annotated corpus for Indian Languages is very crucial for technology development. So far, large amount of work has been done in this direction but with no unifying standards. Recently, BIS has come up with a standardized scheme for Indian languages that can be customized according to the characteristics of a language. Therefore, it becomes essential for all ongoing annotation projects to follow these standards. Ignoring the already created resources and annotating new corpora from scratch is not only requires tremendous effort, time and money, but also leads to underutilization of existing resources. Therefore, the need of hour is to utilize the already annotated corpus. Hence, mapping from one annotation scheme to another becomes essential.

For the current work, we annotated Hindi & Urdu 10K words, using LDC-IL Manual Annotation Tool (MAT). The interface is shown in Fig.1 Since, we were using fine-grained hierarchical tagset, and annotation was very labour intensive. Therefore, to reduce manual labour, some simple heuristics were devised to automatically predict the tags of the words We shall call this simple heuristic based annotation tool as Simple Pattern Matching Tool (SPMT). The tool was helpful to some extent. We could not use more sophisticated standard machine learning algorithms due to the unavailability of sufficient amount of annotated corpus. Further, the output of SMPT was manually validated and corrected whenever required to achieve gold standard annotated corpus. Finally, the tags of this gold standard annotated corpus were replaced with the appropriate BIS tags using simple mapping rules.

2. Tagsets: An Overview

POS tagset is a minimal set of categories & sub-categories that can be used to classify all the words of a language with maximum precision. The initial efforts in POS annotation resulted in tagsets that were simple inventories of tags corresponding to the

[1]The term split-orthography is used due to the unavailability of any technical term in the existing literature to denote the splitting tendency in the Persio-Arabic script due to which some affixes (bound forms) are written separately from their roots/stems (free forms) & even some roots are written in two tokens. The term is, in a way, new coinage to describe this kind of tokenization problem in Urdu, Kashmiri, etc (See Bhat et al., 2010).

morpho-syntactic features such as Brown tagset & Upenn tagset (Hardie, 2004). It was CLAWS2 tagset (Sartoni, 1987) which is considered a landmark in the history of tagset designing. It marked an important change in the structure of tagsets, from a flat-structure to a hierarchical-structure[2]. So far, various tagsets have been developed for ILs. Some of them are as follows:

1. AU-KBC tagset for Tamil (2001)
2. Hardie's tagset for Urdu (Hardie, 2005)
3. IIIT-Hyderabad tagset for Hindi (Bharati et al. 2006)
4. MSRI IL-POSTS for Hindi & Bangla (Baskaran et al. 2008)
5. MSRI-JNU tagset for Sanskrit (Chandra Shekhar, 2007)
6. CSI-HCU tagset for Telugu (Sree R.J et al. 2008)
7. Nelrlac tagset for Nepali (Hardie et al., 2005)
8. LDCIL tagsets for ILs (Malikkarjun et al., 2010)
9. BIS tag-sets for all ILs (2010/2011)

Many of the aforementioned tagsets (1, 2, 4, 5, and 8) were strictly/loosely based on the guidelines of EAGLES (Expert Advisory Group for Language Engineering Standards) for morpho-syntactic annotation (Leech & Wilson 1999) while some tagsets (3) were inspired by UPenn guidelines. However, BIS standards have taken into consideration EAGLES, UPenn, ILPOSTS, ILMT & LDC-IL tagsets.

Since in this paper we are concerned with LDC-IL tagsets (based on ILPOSTS) and BIS tagsets for Hindi & Urdu, a very brief introduction of the concerned tagsets is given below.

a) ILPOSTS - It is a POS tagset framework designed to cover the fine-grained morphosyntactic details of Indian Languages (Baskaran et al. 2008). It proposes a three-level hierarchy of categories, types and attributes.

b) BIS - It is more recent POS tagset framework recommended by Bureau of Indian Standardization. It is also hierarchical but coarse-grained one, with two-level hierarchy of category and type.

[2]For a tagset, the term "hierarchical" means that the categories in the tag-set are structured relative to one another. A hierarchical tag-set contains a small number of categories, each of which contains a number of sub-categories that may further contain sub-sub-categories, and so on, in a tree-like structure (Hardie, 2003).

3. Process of POS Annotation

POS annotation is the process of labelling words in the running text corpus with their grammatical categories and associated morphological features.

3.1 Background

We started the actual work at LDC-IL, initially by customising version 0.1 tagsets, guidelines as well as annotation tools for Hindi, Urdu and other ILs on the basis of ILPOSTS. All the annotators who worked with this project had a postgraduate degree in linguistics but they did not have any formal training in annotation. Hence, an experimental annotation was needed. At first, annotation of 10k words was carried out by two annotators for Hindi and one for Urdu. After discussing various issues like case syncretism, attributes of proper noun), tagsets and guidelines were updated to version 0.2. During this phase, a user friendly MAT was also developed. Using version 0.2 of tagsets and guidelines, 10k words were reannotated by three annotators for Hindi and two annotators for Urdu. The issues that came up during this phase of annotation resulted in version 0.3 of the tagsets as well as the guidelines.

The process of POS Annotation for the current work was completed in three phases: Manual annotation, semi-automatic annotation by using some heuristics followed by manual validation and annotation by using mapping rules.

3.2 Phase-1: Manual POS Annotation

Initially the manual annotation of 10k words was carried out with the help of the MAT interface. Three annotators carried out the annotation work as per version 0.3 LDC-IL guideline for both Hindi & Urdu.

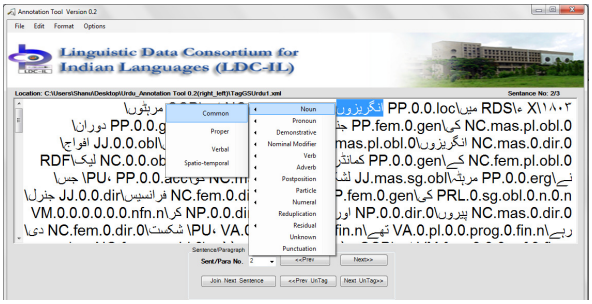


Figure.1: MAT Interface

In any annotation work, evaluation plays an important role. There are two main aspects to evaluating the annotation. One has to do with the extrinsic measures of validity and consistency while the other focuses on the agreement between the annotators (IAA). Since common wisdom treats human annotator agreement measures upper bound for system performance, IAA is of utmost importance.

The question then arises as to what to do with annotator disagreements? Are they just mistakes on the part of one annotator that need to be corrected or they have some other importance? Disagreements can often indicate specially vague or ambiguous phenomenon which might need special handling. Besides, when all the disagreements have been cleared to produce the gold standard data, system developers often want information about the cases which have been difficult to annotate and the original annotator disagreements can be very useful source of information.

IAA: It has been argued that simple IAA is a very good indicator of the reliability of annotated data (Chen & Palmer 2009; Dandapat et. al., 2009). Therefore, for the current work, evaluation was done on the basis of simple Inter-annotator agreement (IAA) measures Total weightage for an agreement case has been taken as 100 which can be distributed over the number of annotators involved. Since, 3 annotators were involved, each annotator was given 33.3 weightage. Similarly, if there would have been only two annotators involved, each would have been assigned 50 weightage. A high IAA denotes that at least two annotators agree on the annotation and therefore, the probability that the annotation is erroneous is very small. For instance, in figure 2 & 3 it is shown that IAA between 3 annotators for Urdu & Hindi each, was 100 when all the 3 annotators agree but it reduces to approx. 66.667 when only 2 annotators agree. It is also shown that category level agreement was good but attribute level agreement was hard to achieve.

S.NO	Annotator-1	Annotator-2	Annotator-3	Category	Attribute
1	JQ.0.0.obl.nnm\سب	PPR.0.pl.0.obl.0.n.0.y\سب	PPR.0.pl.0.obl.0.n.0.y\سب	0	0
2	PP.0.0.ins\سے	PP.0.0.ins\سے	PP.0.0.ins\سے	66.667	100
3	NST.dir\پہلے	JQ.0.sg.obl.ord\پہلے	JQ.0.sg.obl.ord\پہلے	66.667	66.667
4	NP.mas.sg.obl.0.n\خدائے	NP.mas.0.dir.0.n\خدائے	NP.mas.0.dir.0.n\خدائے	100	0
5	JJ.mas.sg.obl\بزرگ	JJ.mas.0.dir\بزرگ	JJ.mas.0.dir\بزرگ	100	0
6	CCD\و	CCD\و	CCD\و	100	100
7	JJ.mas.0.obl\برتر	JJ.mas.0.dir\برتر	JJ.mas.0.dir\برتر	100	0
8	CCD\اور	CCD\اور	CCD\اور	100	100
9	RDF\وحدہ	RDF\وحدہ	RDF\وحدہ	100	100
10	RDF\لاشریک	RDF\لاشریک	RDF\لاشریک	100	100
11	PP.mas.sg.gen\کا	PP.mas.sg.gen\کا	PP.mas.sg.gen\کا	100	100
12	JJ.0.0.dir\مشکور	JJ.0.0.dir\مشکور	JJ.0.0.dir\مشکور	100	100
13	CCD\و	CCD\و	CCD\و	100	100
14	JJ.0.0.dir\ممنون	JJ.0.0.dir\ممنون	JJ.0.0.dir\ممنون	100	100
15	VM.0.sg.1.prs.0.0.fin.n\ہوں	VM.0.sg.1.prs.0.0.fin.n\ہوں	VM.0.sg.1.prs.0.0.fin.n\ہوں	100	100

Figure.2: IAA for Urdu

S.NO	Annotator-1	Annotator-2	Annotator-3	Category	Attribute
1	घाटी\NC.fem.sg.obl.n.n	घाटी\NC.fem.sg.obl.n.n	घाटी\NC.fem.sg.obl.n.n	100	100
2	को\PP.0.0.dat	को\PP.0.0.acc	को\PP.0.0.acc	100	0
3	पार\NC.0.0.dir.n.n	पार\NC.mas.0.dir.n.n	पार\NC.mas.0.dir.n.n	100	0
4	करते\VM.0.0.0.0.0.0.nfn.n	करते\VM.0.0.0.0.0.0.nfn.n	करते\VM.0.0.0.0.0.0.nfn.n	100	100
5	ही\CEMP	ही\CEMP	ही\CEMP	100	100
6	इनमें\PPR.0.pl.3.obl.loc.n.r	इनमें\PPR.0.pl.0.obl.loc.n.n	इनमें\PPR.0.pl.3.obl.loc.n.r	100	0
7	से\PP.0.0.ins	से\PP.0.0.ins	से\PP.0.0.ins	100	100
8	एक\JQ.sg.0.obl.crd.n	एक\JQ.sg.0.obl.crd.n	एक\JQ.sg.0.obl.crd.n	100	100
9	आदमी\NC.mas.sg.obl.n.n	आदमी\NC.mas.sg.obl.n.n	आदमी\NC.mas.0.obl.n.n	100	66.667
10	की\PP.0.fem.gen	की\PP.0.fem.gen	की\PP.0.fem.gen	100	100
11	मशाल\NC.fem.sg.dir.n.n	मशाल\NC.fem.sg.dir.n.n	मशाल\NC.fem.sg.dir.n.n	100	100
12	टीले\NC.mas.sg.obl.n.n	टीले\NC.mas.sg.obl.n.n	टीले\NC.mas.sg.obl.n.n	100	100
13	के\PP.0.mas.gen	के\PP.0.mas.gen	के\PP.0.mas.gen	100	100
14	नीचे\NST.dir.n	नीचे\NST.obl.n	नीचे\NST.obl.n	100	0
15	खड्ड\NC.mas.sg.obl.n.n	खड्ड\NC.mas.sg.obl.n.n	खड्ड\NC.mas.sg.obl.n.n	100	100
16	में\PP.0.0.loc	में\PP.0.0.loc	में\PP.0.0.loc	100	100
17	गिर\VM.0.0.0.0.0.0.nfn.n	गिर\VM.0.0.0.0.0.0.nfn.n	गिर\VM.0.0.0.0.0.0.nfn.n	100	100
18	पड़ी\VA.fem.sg.0.0.prf.0.fin.	पड़ी\VA.fem.sg.3.0.prf.0.fin.n	पड़ी\VA.fem.sg.3.0.prf.0.fin.	100	0

Figure.3: IAA for Hindi

After resolving the inter-annotator disagreement, the manual correction of the annotated corpus was carried out to achieve gold standard POS annotated corpus.

3.3 Phase-2: Annotation using Some Heuristics

In the second phase, simple heuristics were applied to reduce manual labour. On the basis of Gold Standard 10K words, SPMT was used to tag further 25K words. SPMT contained the POS annotated corpus as lexicon that it used as its search space to match the given word and transfer the tag to the word. Since, it was a unigram-based tool; out-of-vocabulary-words were not handled. The output was validated with the help of the MAT interface. Further, 50K words were annotated and validated in the same way on the basis of 35K (10K+25K) annotated words. This resulted in 85K fine grained POS annotated words sufficient enough for developing a tagger using any standard machine learning technique.

However, the gold standard was not used directly; instead, feature/attribute strings of the tags were removed and only category and type labels were retained. For illustration, consider the words:

a) "uThaayaa" उठायाा/VM.mas.sg.3.0.prf.0.fin.n

b) "kaa" का\PP.sg.mas.gen

The tags in the above annotated words were trimmed to remove the feature/attribute strings - (mas.sg.3.0.prf.0.fin.n) & (sg.mas.gen), respectively. The above words were rendered featureless as given below:

a) उठायाा/VM

b) का\PP.

It was observed that along with the attribute level, the accuracy of the automatic tagger is very low and it increases when only the category and the type levels are considered. For instance, the tagging accuracy for Hindi reduces from 87.66% at type level to 69.53% at type plus attribute level (see Dandapat et. al. 2009).

The resultant 85K annotated words (coarse grained) were used as training data for developing hybrid POS tagger (interface shown in fig. 3) for Hindi & Urdu, with accuracy scores - Hindi (Precision: 86.11, Recall: 99.55, F-score: 92.34) & Urdu (Precision: 82.62, Recall: 99.18, F-score: 90.15) (Yoonus & Sinha 2011).

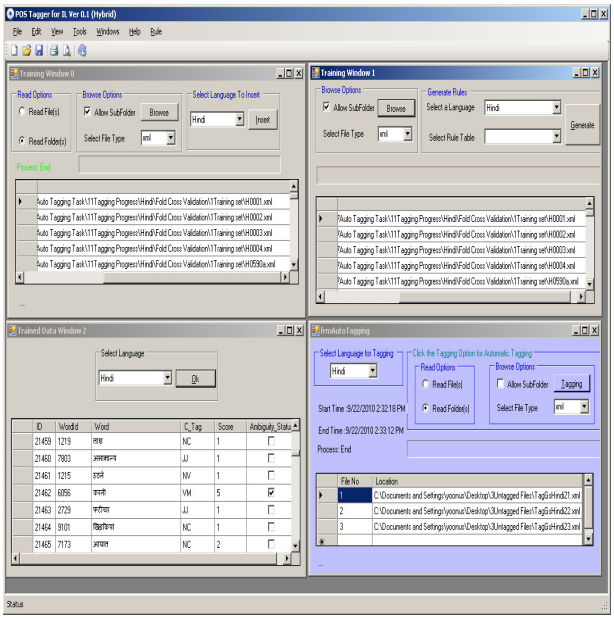


Figure 4: Interface of Hybrid POS Tagger

3.4 Mapping from LDC-IL to BIS

As mentioned above, we have used LDC-IL scheme for POS annotation but the standards were laid down by Bureau of Indian Standardization (BIS) & we had to follow the prescribed standards. It is really a tedious task when you have completed annotation & validation of approximately 85K words and in between you have to change your tagset. However, mapping can be a right solution in these circumstances.

Mapping between two tagsets can be easily done when there are similar sets of categories & subcategories involved. However, if there is lot of mismatch between the two tagsets, the mapping becomes nonproductive. Since, there are similar sets of categories & subcategories involved in LDCIL & BIS tagsets, mapping proved to be very useful.

Therefore, to avoid the manual labour, we formulated simple mapping rules and applied them on 69.7K Hindi & 66.4K Urdu data. Accordingly, the LDC-IL tags were replaced with the BIS tags. The mapping rules are given below in fig. 4:

LDCIL-Hindi v0.3		BIS	
Category	Type	Category	Type
Noun (N)		Noun (N)	
	Common (NC)		Common (NN)
	Proper (NP)		Proper (NNP)
	Verbal (NV)		
	Spatio-temporal (NST)		Nloc (NST)
Pronoun (P)		Pronoun (PR)	
	Pronominal (PPR)		Pronominal (PRP)
	Reflexive (PRF)		Reflexive (PRF)
	Relative (PRL)		Relative (PRL)
	Reciprocal (PRC)		Reciprocal (PRC)
	Wh-pronoun (PWH)		Wh-pronoun (PRQ)
			Indefinite (PRI)
Demonstrative (D)		Demonstrative (DM)	
	Absolutive (DAB)		Deictic (DMD)
	Relative Demonstrative (DRL)		Relative (DMR)
	Wh-demonstrative (DWH)		Wh-word (DMQ)
			Indefinite (DMI)
Verb (V)			Verb (V)
	Main Verb (VM)		Main Verb (VM)
	Auxiliary Verb (VA)		Auxiliary Verb (VAUX)
Nominal Modifier (J)			
	Adjective (JJ)		Adjective (JJ)
Adverb(A)	Manner (AMN)	Adverb (RB)	
Post- position(PP)		Post- position (PP)	
Particle (C)		Particle (RP)	
	Interjection (CIN		Interjection (INJ)
Residual(RD)		Residual (RD)	
	Foreign Word (RDF)		Foreign Word (RDF)
	Symbol (RDS)		Symbol (SYM)
Reduplication (RDP)		Reduplication (RDP)	

Figure 5: Mapping Rules LDC-IL to BIS

However, there were some exceptions to the mapping rules. Some LDC-IL tags were not having the corresponding BIS tags. For example, LDC-IL tag NUMR has no direct corresponding BIS tag. Therefore, additional rules (ref. to fig. 5) were formulated to take care of this mismatch in the tagsets.

JQ.non-numeral → QT_QTF
JQ.cardinal → QT_QTC
JQ.ordinal → QT_QTO
NV → VM
JINT → RP_INTF
CCD → CC_CCD
CSB → CC_CCS
UNK → RD_UNK
PU → RD_PUNC
CIN/CEMP/CAGR/CDLIM/CHON/CDED/CTOP/CEXCL/CDUB/CSIM/CINT/CX → RP_RPD
NUMR/NUMS/NUMC→ QT_QTC
ORD → QT_QTO
If a token X1 is tagged as xx and the following token is also X (i.e. X2) tagged as RDP → tag X2 as xx If a token X1 is tagged as xx and the following token is Y (not the same as X1) tagged as RDP → tag Y as RD_ECH

Figure 6: Additional rules (Exceptions)

3.5 . Final Validation

Mapping did not solve the problem because the categories of LDC-IL tagset couldn't be unambiguously mapped on the BIS categories completely even after formulating additional mapping rules. There were some cases, e.g., the type PRI in BIS tagset had no corresponding tag in LDC-IL tagset. The indefinite pronouns were tagged under the cover term PPR in the LDC-IL tagset. Similarly, DMI tag in BIS has no corresponding tag in LDC-IL tagset. So, such cases had to be corrected manually.

4. Hindi & Urdu Data

Since, POS tagging is essentially a classification problem, 69.7K words of Hindi & 66.4K words of Urdu were classified and graded as per decreasing order of their respective frequencies and percentages. The classification of Hindi & Urdu words with their respective percentages & frequencies are given below in Figs. 6 & 7.

S.No.	Tag	Freq Count	Percentage
01	N_NN	14618	20.9658
02	PSP	9531	13.6698
03	RD_PUNC	8794	12.6128
04	V_VM	8757	12.5597
05	V_VAUX	5486	7.8683
06	JJ	4134	5.9292
07	PR_PRP	3560	5.1059
08	N_NNP	2349	3.3690
09	RP_RPD	2054	2.9459
10	CC_CCD	1676	2.4038
11	N_NST	1296	1.8588
12	CC_CCS	962	1.3797
13	QT_QTF	865	1.2406
14	QT_QTC	844	1.2105
15	RP_NEG	809	1.1603
16	DM_DMD	766	1.0986
17	RB	641	0.9194
18	PR_PRF	505	0.7243
19	RD_UNK	382	0.5479
20	PR_PRL	275	0.3944
21	PR_PRQ	254	0.3643
22	QT_QTO	248	0.3557
23	DM_DMI	216	0.3098
24	RD_RDF	207	0.2969
25	PR_PRI	176	0.2524
26	RP_INTF	125	0.1793
27	DM_DMR	105	0.1506
28	DM_DMQ	37	0.0531
29	RP_INJ	25	0.0359
30	PR_PRC	15	0.0215
31	RP_UNK	9	0.0129
32	RD_SYM	2	0.0029
	Total	69723	100.0000

Figure 7: Classification of Hindi words

S.No	Tag	Frequency	Percentage
01	PSP	11597	17.44225
02	N_NN	9324	14.02358
03	V_VM	8218	12.36013
04	RD_PUNC	6669	10.03038
05	JJ	5325	8.008964
06	V_VAUX	4786	7.198291
07	PR_PRP	3348	5.035495
08	RP_RPD	3270	4.918181
09	CC_CCD	2753	4.140597
10	N_NST	1448	2.177837
11	DM_DMD	1431	2.152268
12	CC_CCS	1316	1.979305
13	RB	1125	1.692035
14	Q_QTF	1040	1.564192
15	Q_QTC	771	1.159608
16	V_VM	717	1.07839
17	PR_PRF	559	0.840753
18	PR_PRL	467	0.702382
19	N_NNP	433	0.651245
20	RD_RDF	380	0.571532
21	Q_QTO	278	0.418121
22	PR_PRQ	269	0.404584
23	RP_INTF	191	0.28727
24	QT_QTF	143	0.215076
25	UNK	141	0.212068
26	DM_DMR	125	0.188004
27	QT_QTC	123	0.184996
28	RP_NEG	109	0.163939
29	DM_DMQ	80	0.120322
30	QT_QTO	40	0.060161
31	RD_SYM	12	0.018048
	Total	66488	100

Figure 8: Classification of Urdu words

5. POS Annotation Issues

It is usual experience for any annotator to face lots of problems while annotating data. Such problems need to be documented and discussed properly. It takes time to resolve them. In many cases there will be easy solution. But in some cases we need to take adhoc decisions instead of finding clear cut solutions as there are fuzzy areas in natural language where there is no categorical answer. Moreover, binary logic doesn't work in such cases.

In this section, we highlight some major issue that we encountered during the annotation of Hindi, Urdu data.

i) Paradox of Corpus Annotation: Does form determine function or function determines form? Corpus linguistics is a methodology which tries to capture the functional aspect of corpus rather the formal one. But there is no categorical decision on the form-function aspect in any POS schemes for ILs, e.g. Verbal Nouns/Gerunds for Urdu & Hindi. They play a clear cut nominal function but as per BIS recommendations we have to classify them under verb. Similarly, reduplicated verbs and participles are to be classified under verbs but they function as adverbs and adjectives respectively. Contrary to this, nouns such as karaN, vajah, taaluq etc. are to be tagged as PSP on the functional basis. It is often argued that there should be a trade-off between form and function but it is still not clear how to maintain this trade-off in the annotation process, i.e. where it is necessary to annotate on the basis of function and where to annotate it on the basis of form?

ii) Hindi-Urdu Complex predicates: These are generally composed of "NN/JJ + Light Verb." For example, khush honaa, haasil karnaa.

In such constructions, it is very hard to find out the POS category of the first element.

iii) Split-Orthography: Certain bound forms (Persian borrowed) in Urdu are written as separate tokens. They are generally adjectival (yaftah, paziir, etc) or participle suffixes (e.g. shudah, kardah, etc) & have purely inflectional status. Therefore, they do not belong to any POS category. Whether, it should be considered a tokenisation problem & handled at pre-processing level by joining them to their preceding

categories or should it be handled at POS level by introducing null tag (See Hardie, 2005). It is not clear in the present scheme.

6. Conclusion

In this paper, we have shared our experience of developing 85k fine grained & 85k coarse grained annotated corpus of Hindi & Urdu based on LDCIL standards. We have also shared our experience of developing approx.70k words coarse grained annotated corpus of Hindi & Urdu based on BIS standards. Further, we have shown how we have made a transition from LDC-IL to BIS annotation scheme. Finally, we highlighted some major issues that we came across during the annotation process.

References

AU-KBC Tagset. AU-KBC POS Tagset for Tamil. http://nrcfosshelpline.in/smedia/images/downloads/Tamil_Tagset-opensource.odt

Bali, K., Choudhary, M., Biswas, P., Jha G. N., Choudhary, N. K. and Sharma M. (2008). Indian Language Part-of-Speech Tagset: Hindi. LDC2010T24.

Baskaran S. et al. (2007). Framework for a Common. Parts-of-Speech Tagset for Indic Languages. (Draft) http://research.microsoft.com/~baskaran/POSTagset/

Baskaran, S., Bali, K., Bhattacharya, T., Bhattacharya, P., Choudhury, M., Jha, G. N., S. Rajendran, K. Saravanan, L. Sobha and Subbarao, K.V. (2008). A Common Parts-of-Speech Tagset Framework for Indian Languages. In *Proceedings of 6th Language Resources and Evaluation Conference LREC*.

Bhat. Shahid Mushtaq. Richa & Farooq. (2010). Developing Hierarchical POS Tag-set for Kashmiri. In *Proceedings of International Conference on Language Development & Computing Methods ICLDCM, Karunya Institute of Technology, Coimbatore*.

Chandrashekar R., (2007). POS Tagger for Sanskrit. Doctoral Thesis, Jawaharlal Nehru University

Cloeren, J. 1999. Tagsets. In Hans van Halteren (Ed.), *Syntactic Word-class Tagging*. Dordrecht: Kluwer Academic.

Dandapat S., Biswas P., Choudhury, M., Bali, K. (2009). Complex Linguistic Annotation – No Easy Way Out! A Case from Bangla and Hindi POS Labeling Tasks. In *Proceedings of the Third Linguistic Annotation Workshop ACL-IJCNLP* 20, pp, 10--18.

Garside, R. (1987). The CLAWS word-tagging system. In Garside, Leech & Sampson (Eds.), *The Computational Analysis of English*. London: Longman.

Habash, N. & Owen Rambow. (2005). Arabic Tokenization, Morphological Analysis, and Part-of-Speech Tagging in One Fell Swoop. In *Proceedings of the Conference of American Association for Computational Linguistics (ACL05)*.

Hardie, Andrew, et al. (2005). Categorisation for Automated Morphosyntactic Analysis of Nepali: Introducing the Nelralec Tagset (NT-01). Nelralec/Bhasha Sanchar Working Paper 2.

Hardie, A. (2004). The Computational Analysis of Morpho-syntactic Categories in Urdu. PhD Thesis submitted to Lancaster University.

Hardie. A. (2003). Developing a Tag-set for Automated Part-of-Speech Tagging in Urdu. In *Proceedings of the Corpus Linguistics Conference*.

IIT-tagset. A Parts-of-Speech Tagset for Indian Languages. http://shiva.iiit.ac.in/SPSAL2007/iiit_tagset_guidelines.pdf

Leech, G. & Wilson, A. (1999). Recommendations for the Morpho-syntactic Annotation of Corpora. EAGLES Report EAG-TCWG-MAC/R.

Leech, G. & Wilson, A. (1999). Standards for Tag-sets. In Hans van Halteren (Ed.), *Syntactic Word-class Tagging*. Dordrecht: Kluwer Academic.

Leech, G. & Wilson, A. (1996). Recommendations for the Morpho-syntactic Annotation of Corpora. EAGLES Report EAG-TCWG-MAC/R.

Richa. (2010). LDC-IL POS Annotation Guidelines for Hindi. Ms.

Rushda. (2010). LDC-IL POS Annotation Guidelines for Urdu. Ms.

Santorini, B. (1990). Part-of-Speech Tagging Guidelines for the Penn Treebank Project. Technical report MS-CIS- 90-47, Department of Computer and Information Science, University of Pennsylvania

The Nelralec Tagset (NT-01) and Guidelines for Manual Tagging. Ms. Nelralec/Bhasha Sanchar.

Yoonus, Mohamed & Samar Sinha. (2011). A Hybrid POS Tagger for Indian Languages. In *Language in India*. Vol.11.

Tamil Clause Boundary Identification: Annotation and Evaluation

Vijay Sundar Ram, R., Bakiyavathi, T., Sindhuja Gopalan, Amudha, K. and

Sobha, Lalitha Devi

AU-KBC Research Centre,
MIT Campus of Anna University,
Chennai, India
sobha@au-kbc.org

Abstract

Clause boundary identification has a significant role in NLP applications. It has been used to improve the performance of different practical NLP systems. In this paper, we present various types of clausal structures that exist in Tamil, a Dravidian language. The clausal sentences from Newspapers, Novels and Tourism domain were collected and variations in the clausal structures across these domains are analysed. Here we discuss about the annotation of tags used for various clauses and have focussed on the Inter-annotator agreement. Inter-annotator agreement is the relative level of agreement between annotators. We have used kappa coefficient as the agreement statistic, which is the measure of inter-annotator agreement. We also present the Automatic Clause Boundary Identification System developed using CRFs technique. We have evaluated and discussed on the performance of the system.

1. Introduction

Syntactic structure information, given by the clause boundaries in a sentence helps in improving the NLP applications. The clause boundary identification task is employed in various applications such as machine translation, text-to-speech, information extraction, question answering system and bilingual corpus alignment tools. A clause is defined as a word sequence which contains a subject and a predicate. This subject can be explicit or implied. In the task of automatic clause boundary identification, the boundaries of these clauses in a sentence are automatically marked. In the present work, we discuss about the annotation of the clause boundaries in Tamil sentences and automatic clause boundary identification task. The clause boundary annotation is done on three different domain corpus. The automatic clause boundary identification tool is built using machine learning technique and grammatical rules as its features.

The early works towards the clause boundary identification evidently shows the need of this information. Eva Ejerhed (1988) used clause boundaries to improve AT & T's text-to-speech system by introducing pause, lengthening at the clausal boundaries. Papgeorgiou (1997) used hand crafted rules in identifying the clause boundaries. This information was used to improve the bilingual alignment task. The idea behind the rule-based method used by Leffa (1998) was, clauses can be ultimately be reduced to a noun, an adjective or an adverb regardless of their length or the number of embedded clauses they may contain. This was used to improve the English/Portuguese machine translation system. After different rule-based approaches, different machine learning techniques, hybrid techniques were used to come up with clause boundary identification systems. Orasan (2000) did a hybrid approach, where the clauses were tagged initially using memory-based learning system and post-processed using a set of rules. The clause boundary identification task was well boosted with CoNLL 2001 Shared task. The shared task had a huge clause tagged English corpus (corpus taken from WSJ corpus). The corpus was preprocessed with part-of-speech and chunking information. The clause boundaries were marked with S* and *S for start and end boundaries respectively. From this shared task six different systems using different machine learning techniques came out. Patrick and Goyal (2001) used Ada boost algorithm for boosting the performance of decisions graph. Dejean (2001) used Symbolic learning ALLis, specialized HMM was used by Molina and Pla (2001), Hammerton (2001) used long short-term memory, a recurrent neural network architecture, Tjong Kim Sang (2001) used memory based learning approach and Carreras's (2001) did an approach of decomposing clause into combination of binary decisions using Ada boost learning algorithm. Carreras (2002) made a global inference on a local classifier on partial parsing of sentence and used a dynamic programming for choosing the best decomposition of sentence to clauses. Carreras (2003) used phrase recognition using perceptrons and an online learning algorithm. A multilingual method for clause splitting was done by Georgiana Puscasu (2004). Here he used the information of coordination and subordination with machine learning technique. Vinh Van Ngugen (2007) did clause boundary identification for English using Conditional Random Fields (CRFs). They have also used linguistic information and a bottom-up dynamic algorithm for decoding to split a sentence into clauses. Ram and Sobha (2008) did a hybrid approach for English clause boundary identification using CRFs, where the output of the CRFs was passed through an error analyser and post processed using a set of rules. Alegria et al. (2008) has done the clause identification task in Basque, where he has combined the rule-based grammar with filtering-ranking perception. They used EPEC corpus of about 25000 words. Palavers parser produced a syntactic tree for

Portuguese texts, which include clause information. They have tagged three tags finite (fcl), non-finite (icl) and averbal (acl) using Entropy guided transformational learning.

In Indian languages, there are very few works in clause boundary identification task. Research is active in this area for the past few years. In Tamil, Ram et al (2008) has come with CRFs based approach, where the syntactic rules are used as the major feature. They have used a small corpus of 219 sentences for training and 75 sentences for testing. Daraksha Parveen et al. (2011) has done clause boundary identification task for Urdu using classifiers. They have used the machine learning technique, which has linguistic rules as features, to identify the clausal boundaries first and the misclassified clause boundaries were corrected using additional linguistic rules. They have annotated the Urdu corpus with following clause tags, complimentizer, relative particle, particles, temporal, manner, causality, condition and nominal. Ariruddha Ghosh et al. (2010) had worked for Bengali, where they have used rules for identifying the clause boundaries and CRFs to classify the clause. In Bengali, corpus used is from NLP Tool Contest: ICON 2009, they have annotated the clause information. The tags used by them are principle clause, noun clause, adjective clause and adverbial clause. Hindi clause boundary information is automatically tagged using the Stage 1 parser for Hindi developed by Husain et al. (2009). This uses MST for identifying the inter-clausal relations.

The paper is arranged as follows, in the next section we discuss about the various clausal structures in Tamil. In the third section, we have dealt with variation in the clausal structures in the three domains (newspapers, tourism and novels). In the fourth section, we have explained about the annotation of clause boundary markers and inter-annotation agreements. The automatic clause boundary identification system and its evaluation are explained in fifth section. The paper concludes with a discussion on future works.

2. Clause Structures

We describe about the clauses and the clausal structures that occurs in Tamil, one of the South Dravidian language. It is a verb final language and allows scrambling. It has postpositions, the genitive precedes the head noun in the genitive phrase and the complementizer follows the embedded clause. It is a nominative-accusative language like the other Dravidian languages. Here, due to rich inflection, the part-of-speech tagging of the words contribute more information than in English. The clause boundaries are indicated by suffixes attached to the verb. The subject of a Tamil sentence is mostly nominative, although there are constructions with certain verbs that require dative subject. Tamil has png agreement (Ram, 2009).

We have considered the following clauses for analysis, relative participle clause (RP), conditional clause (CON), infinitive clause (INF), non-finite clause (NF), complementizer (COM) and main clause (MCL). The clause is identified by the type of non-finite verb present in the sentence. Different structures in each clause are described below.

2.1 Relative Participle Clause

The relative participle clause is identified by the relative participle verb in a sentence. The relative participle (RP) verb will occur in three tense and 'a' is the relative participle suffix for present and past tense and 'um' for future tense. Based on the words that follow the relative participle verb, following types of RP clause occurs.

2.1.1. RP verb + Noun

This is one of the common structures of RP clause, RP verb followed by a noun phrase, which will take all the case markers. This NP can be preceded by a genitive noun.

Example 1:
{RP}uuttikku varum currulaap payaNikalai{/RP}
ooty+dat come+future+rp tourist+pl+acc
{MCL} rojaa thoottangkal kavarkinrana.{/MCL}
rose garden+pl attract+present+3pn.
(Tourists who come to ooty are attracted by rose gardens.)
In the above example 'varum' (vaa+future+RP) is the future relative participle verb. It is followed by noun inflected with accusative case marker.

2.1.2. RP verb + Noun + PSP

In this structure RP verb is followed by a noun phrase and a post position (PSP). The noun phrase will be inflected with case markers depending on the PSP that follows.

Example 2:
{RP}eeraththaaza 300 aantukal thamizakaththai
approximately 300 year+pl tamilnadu+acc
thatumaarac ceytha kalappirar parri{/RP}
shake do+past+rp kalappirar about (PSP)
{MCL}arinjarkalitaiyee karuththu veerrumai untu{/MCL}
sholar+pl+emp difference of opinion have.
(About kalappirar, who shook tamilnadu approximately for 300 years, scholars have difference of opinion.)
Here the RP verb is ' ceytha' (cey+past+RP). It is followed by 'kalappirar' (NP) and 'parri' (PSP).

2.1.3. RP verb + PSP

The RP clause can also have RP verb followed by PSP without NP in between. PSPs such as 'pothu', 'pothum', 'pozhuthu', 'mun', 'pin', 'piraku' etc. will follow the RP verb.

Example 3:
{RP}raamar 14 aantukal vanavaacam cenra
Rama 14 year+pl exile go+past+rp
poothu{/RP}{MCL}intha vaziyaaka ceVnYrYAr.{/MCL}
psp this way+adv went+past+3sh.
(When Rama went to exile for 14 years, he went by this

way.)
Example 3 has the RP verb 'cenra' (cel+past+RP). This is followed by the PSP 'poothu'. This PSP 'poothu' introduces the time.

2.1.4. RP verb + Noun + Adv

In this structure of RP clause, RP verb is followed by a dative noun and an adverb.

Example 4:
{RP} inthiyaavin anaiththu paakangkalilirunthu
india+gen all part+pl+abl
varum payanikalukku vacathiyaaka{/RP}
come+future+rp tourist+pl+dat facility
{MCL}raamanaathapuraththirkum, raameesvaraththirkum
raamanaathapuram+dat+inc, raameeswaram+dat+inc
rayil vacathi untu.{/MCL}
train facility have.
(For convenience of the tourists who come from all parts of India, there is train facility between Ramanathapuram and Rameswaram.)
In the above Example (4) the RP verb 'varum' (vaa+future+RP) is followed by a dative noun ' payanikalukku' (tourists+dative) and an adverb 'vacathiyaaka' (facility+adv marker).

2.1.5. RP verb + pronoun

The RP verb can be followed by a pronoun, similar to RP verb followed by NP. Here the pronoun can be agglutinated with the RP verb.

Example 5**:**
{RP}neRRu ingu vanthavan{/RP}{MCL} vaalviiccu
Yestrday here come+past+rp+pn sword playing
kaRRukkontaan.{/MCL}
learn+past+3sn
(He who came here yesterday learnt sword playing.)
Here the RP verb 'vantha' (vaa+past+RP) and the pronoun 'avan' have agglutinated to form 'vanthavan'.

2.2 Infinitive Clause

Infinitive (INF) verb does not take tense markers and the infinitive marker is 'a'. The infinitive clause in the sentence is identified using the infinitive verb. The infinitive verb that occurs before the finite verb will not be considered for infinitive clause.

Example 6:
{INF} ithuvee kuccanuur ena peyar pera {/INF}
it+emph kuccanur as name to get
{MCL} kaaranam aayirru. {/MCL}
the reason became.
(This became the reason to get the name as kuccanur.)
In the above Example (6), the 'pera' (to get) is an infinitive verb in this sentence.
The different structures of Infinitive clause are discussed below:

2.2.1. INF+Inclusive marker

Example 7:
{INF}vivacaayam ceyyavum {/INF}
Farming do+inc
{MCL}avarkalukku nilangkal tharappatukinrana. {/MCL}
he+pl+dat land+pl are give+present+3pn.
(Also to do farming, lands are given to them.)
Here, in this sentence (7) the infinitive verb is further inflected with the inclusive marker.

2.2.2. INF +Reduplication

Example 8:
{INF} inthap puththakam patikkap patikka {/INF}
This book read+inf read+inf
{MCL}enakku ithu pitikkath thotangkiyathu.{/MCL}
i it like start+past+3sn.
(Kept on reading this book, i started liking it.)
In Example 8, the infinitive verb occurs twice (reduplicating) to show the continuation of the action. Here the infinitive verb 'patikka' has occurred twice.

2.2.3. INF+emphatic

The infinitive verb which helps to identify the infinitive clause also gets inflected with emphatic marker.

Example 9:
{INF}aval thappikka muyalavee,{/INF}
she escape try+inf+emp
{MCL}naan avalai irukap parrineen.{/MCL}
i her+acc tightly hold+past+1sg.
(She tried to escape, i hold her tightly.)
In the above example the infinitive verb ‘muyala’ (try+INF) gets inflected with the emphatic marker 'e' to form 'muyalavee'.

2.3 Non finite Clause

The verb which occur in verbal participle form is used to identify the non-finite (NF) clause. The verbal participle markers are 'u', 'i'. The verbal participle form that occurs before the finite verb will not be considered for identifying the NF clause.

Example 10:
{NF}avarkal arici kontuvanthu {/NF}
he+pl The rice bring+past+vbp
{MCL} malaippatikalil thuuvukinranar. {/MCL}
mountain step+pl+il scatter+present+3ph.
(They bring the rice,scatter on the mountain step.)
In this Example (10), ' kontuvanthu' (bring+past+vbp) is the verbal participle which indicates the NF clause in the sentence.

2.3.1. NF+Emphatic

Verbal participle can also be inflected with emphatic marker.

Example 11:
{NF}avaraip paarththuthaan {/NF}
he+acc see+past+vbp+emph

{MCL} naan ooviyam varaiyak karrukkonteen.{/MCL}
i art+inf draw learn+past+1sg.
(Only by seeing him, i learned to draw art.)
In the above Example (11), 'paarththuthaan' (see+past+vbp+emph) is a verbal participle with emphatic marker.

2.4 Conditional Clause

Conditional clause is identified by conditional (CON) verb. The suffixes for conditional verb are 'aal', 'athaal'. CON verbs take tense markers. It occurs in present, past and future tense.

Example 12:
{CON}mazai peythaal {/CON}
Rain rain+past+con
{MCL} viLaiccal nanRaaka irukkum.{/MCL}
harvest good+adv be+future+3sg
(If it rains, the harvest will be good)
The sentence in Example 12 has 'peythaal' (rain+past+con) is the CON verb.
The CON verbs can be inflected with emphatic markers such as 'e' and 'thaan'.

Example 13:
{CON} muulavarai vazipattaalee {/CON}
The moolavar worship+past+con+emp
{MCL}palan kitaikkum.{/MCL}
benefits get+future+3sn.
(Just by worshipping the Moolavar,[we] will get benefits.)
In the above sentence (Example 13) the CON verb is inflected with emphatic marker 'e'.
CON verb is also inflected with inclusive marker 'um' which shows concession.

Example 14:
{CON} naan kuRRaalaththiRkuc cenRaalum {/CON}
I kuRRaalam+dat go+past+cond+um
{MCL} aruviyil kuLikka maatteen.{/MCL}
falls+loc bath+past+1sg (neg)
(Even if i go to kuRRaalam, I will not take bath in falls.)
Here in example (14), the conditional verb occurs with conditional marker ‘aal’ and emphatic marker ‘um’. ‘aalum’ acts a marker for clause of condition, concession and contrast .

2.5 Complementizer Clause

'enru', 'ena' are the Complementizer (COM) markers in Tamil, which is similar to 'that' in English. The Complementizer Clause can occur in three different positions in a sentence. It can be before the Main clause embedded between the subject and the finite verb and after the Main clause.
This is explained in Example 15.

Example 15:
(1) {COM} raamu varuvaar enRu {/COM}
raamu come+future+3sh that
{MCL}coomu connaar . {/MCL}
somu say+past+3sh
(2) {MCL}coomu {COM} raamu varuvaar
somu raamu come+future+3sh
enRu {/COM} connaar .{/MCL}
that say+past+3sh
(3) {MCL}coomu connaar {/MCL}
somu say+past+3sh
{COM} raamu varuvaar enRu. {/COM}
raamu come+future+3sh that
(Ramu said that somu will come).
The above sentences in Example 15 show how the Complementizer clause moves across the sentence.

3. Variation of Clausal Structures across Domain

We have collected clausal sentences from three different domains such as Newspaper articles, Novels and Tourism. For the sentences pertaining to the Newspaper articles it is collected from daily Newspapers available on the web. We have used two novels, Kalki's Ponniyin Selvan and Dr. M.Varadharajan's Akal Vilaku to collect clausal sentences. Tourism related web pages were used for collecting clausal sentences belonging to tourism domain. The size of each set of clausal sentences is given in table 1.

Domain	Total number of Sentence
Novel	7386
Newspaper	786
Tourism	2042

Table 1: Statistics of the Corpus.

While analysing the sentences from these three domains, the clausal sentences in each domain vary with the complexity (number of clauses in a sentence), distribution of different clauses used and deviation from commonly used clausal structure. Following are the examples from various domains with explanations.

Example 16:
{CON} peNkaLukku eetheenum itaiyuuRu
woman+pl+dat any hindrance
eeRpatin {/CON} {MCL} sri aanjcaneeyarukku
happen+past+con sri Anjaneyar+dat
maalai caaththukiRaarkaL.{/MCL}
garland decorate+present+3ph
(If hindrance happens to women, they decorate Sri Anjaneyar with garland.)
In Example 16, 'eeRpatin' occurs in the meaning 'eeRpattal' (if it happens). Conditional verb commonly occurs with 'aal' as a suffix which is inflected to the verb. Here the conditional clause is introduced by verb 'eeRpatin' (eeRpatu+in – happen+'in' suffix)**.** This kind of structure is common in tourism domain.

Example 17:
{RP} avan kathaiyaip patiththup paarkkaiyil {/RP}
he story+acc read+past+vbp see+past+rp

{MCL} athil onRum puriyavillai. {/MCL}
it+loc one not understand.
(When reading the story, he did not understand anything.)
Here a verbal noun with locative case ' paarkkaiyil' (when seeing) occurs in the sentence to introduce RP+PSP structure. This gives the meaning similar to 'paarkkum poothu' (see+future+RP PSP). This kind of RP clause is common in tourism domain.

Example 18:
{CON} canthiran nakaraththukku vanthathanaal {/CON}
chandiran city+dat come+past+con
{MCL} mikavum ciramppattaan.{/MCL}
more suffer+past+3sn
(Since chandran came to city, [he] suffered a lot.)

Example (18) sentence occurs in novel, here the author has used the 'athanaal' as a conditional suffix, instead of 'athaal', which is more commonly used as conditional marker. The author has introduced this to present the sentence in a stylistic manner.

Example 19:
{COM} avan nanRaaka irukkaveeNtum enRuthaan
he good+adv be+finitte that +emp
{/COM} {MCL} naan virumpukiReen.{/MCL}
i like+present+1sg
(I liked only that he should be good.)

Complementizer clause is introduced by the complementizer marker 'ennru' as seen before. In novels, the authors use emphatic markers with the complementizer marker, which is not usual in News papers and tourism domain. Here in the above example 19, 'enRuthaan' has occurred, which contains 'enRu – complementizer thaan- emphatic marker'.

In the following table (2) the distribution of number of clauses in each sentences present in each domain is given in percentage.

Number of Clauses	Novel %	Newspaper %	Tourism %
2	78.70	66.16	73.80
3	18.41	24.55	22.09
4	1.69	5.34	3.33
5	0.11	0.25	0.59

Table 2: Percentage of Clausal sentences based on the number of clauses present in the sentences

From the above table (2), it is evident that two clause sentences are high in novels, whereas sentences having more than two clauses are high in newspaper and tourism domain.
The clausal distribution in the sentences in various domains is presented in the table 3. Here it is represented in percentage.

Clause	Novel %	Newspaper %	Tourism %
INF	2.93	5.66	9.90
NF	30.13	18.64	30.61
RP	41.25	34.68	42.49
COM	15.23	31.66	2.86
CON	10.45	9.37	14.11

Table 3: Percentage of Clause distribution in various domain corpus..

The stylistic writing in novels, use lengthy sentences using more NF and RP clauses. The complementizer clause is more in newspapers as they have more reported speech. In tourism domain, as it describes the different places using RP clause. Thus RP clause is more in tourism domain. These are evident from the percentages of the clauses in each domain given in the table above (table 3).

4. Annotation and Inter-annotator Agreement

4.1 Annotation

The clause tagged corpus used in the CoNLL Shared task 2001, is one of the first available corpus. In that corpus, they have used "S*' to indicate clause start and "*S" for indicating clause end. The corpus was presented in column format, which has word, part-of-speech tag, chunk tag and the clause boundary tag. The column format and annotation of clause tags with S* and *S was adopted in creating a Basque clause tagged corpus. In this style of annotation the type of clause is not mentioned.
As mentioned earlier, we have tagged relative participle clause, conditional clause, infinitive clause, non-finite clause, complementizer and main clause. We have used the tags {RP} and {/RP} for RP clause start and End respectively. Similarly we have used the following tags to represent the start and end tags, {INF} and {/INF} for INF clause, {NF} and {/NF} for NF clause, {CON} and {/CON} for CON clause, {COM} and {/COM} for COM clause and {MCL} and {/MCL} for main clause.
We have also used column format of representation. The columns have word, part-of-speech tag, chunk tag, morphological analysis and the clause boundary tag. As Tamil is morphological rich, we require the morphological analysis also. A sample of the clause tagged sentence is given in figure 1.

4.2 Inter-annotator agreement:

Inter-annotator agreement is the degree of agreement among annotators. It is the percentage of judgments on which the two analysts agree when coding the same data independently. There are different statistics for different types of measurement. Some are joint-probability of agreement, Cohen's kappa and the related Fleiss' kappa, inter-rater correlation, concordance correlation coefficient, Cochran's Q test, intra-class correlation and Krippendorff's Alpha. We use Cohen's kappa as the agreement statistics. The kappa coefficient is generally

regarded as the statistic of choice for measuring agreement on ratings made on a nominal scale. It is relatively easy to calculate, can be applied across a wide range of study designs, and has an extensive history of use.

The kappa statistic k is a better measure of inter-annotator agreement which takes into account the effect of chance agreement (Ng et al., 1999).

$$K = (p0 - pc)/(1- pc)$$

where p0 is agreement rate between two human annotators and pc is chance agreement between two annotators.

The results of kappa-like agreement measurements are interpreted in six categories as follows (Yalçınkaya et al., 2010).

1- Measurement > 0.8: Perfect agreement
2- 0.8 > Measurement > 0.6: Substantial agreement
3- 0.6 > Measurement > 0.4: Moderate agreement
4- 0.4 > Measurement > 0.2: Fair agreement
5- 0.2 > Measurement > 0.0: Slight agreement
6- 0.0 > Measurement: Poor agreement

We calculated the kappa score for each clause start and end and are presented in the following table:

Clause	Start (**K**)	End (**K**)
RP	0.98	0.97
COM	1	1
INF	1	0.83
MCL	0.94	0.98
NF	1	0.96
CON	1	1

Table 4: kappa scores for each clause start and end tag

As clause end is identified using the clausal markers, the agreement between the annotators should be more for the clause end tags. But the scores show that the agreement is more for clause start tag. The overall kappa score is 0.97. This shows there is a perfect agreement between the annotators.

```
varutaththirku NN      B-NP    <fs af='varutam,n,any,sg,any,d,ku,ku' case_name="dat">          {RP}
orumurai       NN      B-NP    <fs af='orumurai,n,any,sg,any,d,,' case_name="nom">              O
kuurai         NN      B-NP    <fs af='kuurai,n,any,sg,any,d,,' case_name="nom">                O
veeyum         VM      B-VGNF  <fs af='veey,v,any,any,any,,um_0,um_0' tense="FUTURE" rp="Y">    O
poothu         PSP     B-BLK   <fs af='poothu,psp,n,sg,any,,,'>                               {/RP}
paampu         NN      B-NP    <fs af='paampu,n,any,sg,any,d,,' case_name="nom">               {MCL}
varuvathu      VM      B-VGNN  <fs af='vaa,v,n,sg,3,,v_a,v_a_awu' tense="FUTURE" ndrd="Y" rp="Y">  O
vazhakkam      NN      B-NP    <fs af='vazhakkam,n,any,sg,any,d,,' case_name="nom">             O
.              SYM     I-NP    <fs af='&dot;,punc,,,,,,'>                                      {/MCL}
```

Figure 1: Clause annotated example sentence

5 Automatic Clause Boundary Identifier

We have built an automatic clause boundary identification system using Conditional Random Fields (CRFs) technique. We have trained CRFs with different clause sentences separately to get language models for each individual clause. This we have done to avoid ambiguities in learning.

CRFs are an undirected graphical model. Here conditional probabilities of the output are maximized for given input sequence (Lafferty, 2001). This technique is used for various tasks in NLP. Here we have used CRF++ tool which is available on the web (Kudo, 2005).

The performance of the machine leaning technique depends on the features used in learning. The training sentences are presented in column format. The first column contains the word, the following columns contain part-of-speech, chunk information and morphological analysis .The last column contains the clause tag information.

We have used word level and structural level features. In word level feature, word, its PoS and chunk information are considered. Here we have used window of size five. Using grammatical rules, the structural features are represented. Based on the grammatical rules, a column representing the structural features is introduced before the last column.

Grammatical Rules

The grammatical rules used in the clause boundary identification work are as follows.

Rule 1: To get the relative participle clause boundary end

-1 VM+RP = 1
0 NP = 1 RP
1 PSP = 0

If the current token is a np, the previous is a relative participle verb and next word is not a PSP then the current np is marked as probable RP clause end.

Rule 2: To get the relative participle clause boundary end

-1 VM+RP = 1
0 PSP = 1 RP

If the current token is a post position, the previous is a relative participle verb then the current post position is marked as probable RP clause end.

Rule 3: To get the conditional clause boundary

0 VM+CON = 1 CON

If the current verb has a conditional marking suffix, then the current verb is marked for probable conditional clause end.

Rule 4: To get the infinitive clause boundary end
0 VM+INF=1 INF
1 AUX = 0

If the current verb has the infinitive suffix then the current verb is marked for probable conditional clause end.

Rule 5: To get the non-finite clause boundary end

0 VM+NF= 1 NF
1 AUX=0

If the verb is a non-finite verb and not followed by another auxiliary verb then it is marked as probable NF clause boundary end.

Rule 6: To get the complimentizer clause boundary end

-1 VGF = 1
0 Complimentizer=1 COM
1 NP=1

If the current word is a complimentizer such "enna ", "ennru", the previous word is a finite verb and followed by a noun phrase. The COM clause end boundary is marked.
Once these rules are run, the probable clause start positions are marked based on the probable clause ends marked.
Consider the following sentence in example 20 as the input sentence to the clause identifier system.

Example 20
rayilil cendraal na:n makilveen.
train+loc go+future+cond I (will be happy)
(If I go in train I will be happy.)

After preprocessing for the part-of-speech and chunking information, and analyzing the words with morpanalyser is as follow.
rayilil NN B-NP n+loc
cendraal VM B-VGNF v+COND
na:n PN B-NP pn
makilveen VM B-VGF v+FUTURE+3sm
. SYM I-VGF punc

After preprocessing the text the noun phrase is replaced with 'np' and the head noun morphological information is maintained. The PoS, chunk and morphanalyser output is also changed to a format, which presents more distinctly the features in the input to the clause identifier engine. The altered input is shown below.

np np n_loc
cendraal VM_COND V_COND
np np pn_nom
makilveen VM_VGF V_VGF
. SYM I-VGF

To this altered input, the information from the grammatical rules, which points the possible start and end of each clause, is added as the column next to it. Each clause is represented by different numbers to avoid ambiguity and for better learning. After this step the input will be as follows.

np np n_loc 2
cendraal VM_COND V_COND -2
np np pn_nom 6
makilveen VM_VGF V_VGF -6
. SYM I-VGF o

5.1 Evaluation and Discussion

We have trained the system with 5K sentences containing 2000 sentences from tourism domain, 2500 sentences from novel and 500 sentences from daily online newspaper articles. This is tested with 450 sentences containing 150 sentences from each domain. The performance of the clause boundary system is presented in the table 5, table 6 and table 7 for sentences taken from tourism, newspaper and novels respectively.

Clause	Total no of clauses	Clause open		Clause close		Total correct %
		Correct %	Wrong %	Correct %	Wrong %	
INF	14	100	0	100.00	0	100
NF	19	84.21	15.79	84.21	15.79	78.95
RP	114	90.35	9.65	82.46	17.54	79.82
COM	5	20	80	80	20	20
MCL	126	85.71	7.94	97.62	2.38	82.54
CON	12	91.67	8.33	100.00	0	91.67
Total	290	78.65	20.28	90.72	9.28	75.49

Table 5: Performance of clause boundary identification for tourism domain sentences

Clause	Total	Clause open	Clause close	Total

	no of clauses	Correct %	Wrong %	Correct %	Wrong %	correct %
INF	14	85.71	14.29	85.71	14.29	85.71
RP	80	72.50	27.50	62.50	37.50	55.00
NF	57	82.46	17.54	85.96	14.04	80.70
COM	12	33.33	66.67	58.33	41.67	33.33
MCL	139	63.31	36.69	97.12	2.88	63.31
CON	21	80.95	19.05	90.48	9.52	80.95
Total	323	69.71	30.29	80.02	19.98	66.5

Table 6: Performance of clause boundary identification for Newspaper article sentences

Clause	Total no of clauses	Clause open		Clause close		Total correct %
		Correct %	Wrong %	Correct %	Wrong %	
INF	5	100	0	100	0	100
NF	63	90.48	9.52	93.65	6.35	85.71
RP	62	88.71	11.29	79.03	20.97	72.58
COM	20	55.00	45.00	95.00	5.00	55.00
MCL	116	78.45	14.66	99.14	0.86	72.41
CON	24	87.50	12.50	87.50	12.50	75.00
Total	290	83.36	15.49	92.39	7.61	76.79

Table 7: Performance of clause boundary identification for sentences collected from Novels

On analysing the performance tables, it is clear that the propagation of errors from the prior modules affect the performance, as this identification tasks requires all the three analysis, morph analysis, PoS and chunk information to be correct, to introduce the tag at the correct chunk. Identification of COM clause is poor compared to the other clauses. This affects the clause identification in Newspaper domain.

Clause start of the complementizer clause identification is tougher. As complementizer clause can occur in three different structures. In these different structures, the clause end remains in the same position, whereas the clause start varies. This affects the identification of complementizer clause.

6. Conclusion

In this paper, we have discussed about the clausal structures in Tamil, described about annotation of clause boundaries in Tamil sentences. Finally we have explained about the automatic clause boundary identifier using CRFs. We have discussed about the factors affecting the identification task. We will further work on COM clause to overcome the ambiguity in tagging the clause start marker, while tagging with the machine learning technique.

7. References

Alegria, I., Arrieta, B., Carreras, X., Ilarraza, A.D., Uria, L., (2008). Chunk and Clause Identification for Basque by Filtering and Ranking with Perceptrons . *In proceedings of Natural Language Processing, 41*, pp. 5-12.

Artstein, R., Poesio, M., (2008). Inter-Coder Agreement for Computational Linguistics . *Journal of Computational Linguistics,* 34(4), pp. 555-596.

Carreras, X., Màrquez, L., (2001). Boosting trees for clause splitting. *In proceedings of the 2001 workshop on Computational Natural Language Learning*. Toulouse, France, pp. 73-75.

Carreras, X., Màrquez, L., Punyakanok, V., Roth, D., (2002). Learning and Inference for Clause Identification. *In proceedings of the 13th European Conference on Machine Learning*. Helsinki Finland, pp. 35-47.

Carreras, X., Màrquez, L., (2003). Phrase Recognition by Filtering and Ranking with Percep-trons. *In Proceedings of RANLP-2003.* Borovets Bulgaria, pp. 205-216.

Déjean, H., (2001). Using ALLiS for clausing. *In proceedings of the 2001 workshop on Computational Natural Language Learning*. Toulouse, France, pp. 1-3.

Ejerhed, E., (1988). Finding clauses in unrestricted text by finitary and stochastic methods. *In proceedings of the second conference on Applied natural language processing.* Austin, Texas, pp. 219–227.

Ghosh, A., Das, A., Bandyopadhyay, S., (2010). Clause Identification and Classification in Bengali. *In Proceedings of the 1st Workshop on South and Southeast Asian Natural Language Processing, the 23rd International Conference on Computational Linguistics (COLING).* Beijing, pp. 17-25.

Hammerton, J., (2001). Clause identification with long short-term memory. *In proceedings of the 2001 workshop on Computational Natural Language Learning.* Toulouse, France, pp. 61-63.

Harris, V.P., (1997). Clause Recognition in the Framework of Alignment. In R. Mitkov, N. Nicolov (Eds.), *Recent Advances in Natural Language Processing.* Amsterdam/Philadelphia, John Benjamins Publishing Company, pp. 417-425.

Husain, S., Gadde, P., Ambati, B., Sharma, D.M., Sangal, R., (2009). A modular cascaded approach to complete parsing. In proceedings of COLIPS In ternational Conference on Asian Language Processing. Singapore.

Jon, D.P., Goyal, I., (2001). Boosted decision graphs for NLP learning tasks. I*n proceedings of the 2001 workshop on Computational Natural Language Learning.* Toulouse, France, pp. 58-60.

Kudo, T., (2005). CRF++, an Open Source Toolkit for CRF. *http://crfpp.sourceforge.net*

Lafferty, J.D., McCallum, A., Pereira, F.C.N., (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. *In proceedings of the Eighteenth International Conference on Machine Learning.* Morgan Kaufmann, San Francisco, pp. 282-289.

Molina, A., Pla, F., (2001). Clause detection using HMM. *In proceedings of the 2001 workshop on Computational Natural Language Learning.* Toulouse, France, pp. 70-72.

Ng, H.T., Lim, C.Y., Foo, S.K., (1999). A Case Study on Inter-Annotator Agreement for Word Sense Disambiguation. *In Proceedings of the {ACL} {SIGLEX} Workshop on Standardizing Lexical Resources {(SIGLEX99)}.* Maryland, pp. 9-13.

Nguyen,V., (2007). Using Conditional Random Fields for Clause Splitting. *In proceedings of The Pacific Association for Computational Linguistics.* Melbourne, Australia.

Orasan, C., (2000). A Hybrid Method for Clause Splitting in Unrestricted English Text. I*n proceedings of ACIDCA 2000 Corpora Processing*. Monastir, Tunisia, pp. 129-134.

Parveen, D., Sanyal, R., Ansari, A., (2011). Clause Boundary Identification using Classifier and Clause Markers in Urdu Language . *Polibits Research Journal on Computer Science*, 43, pp. 61-65.

Puscasu, G., (2004). A Multilingual Method for Clause Splitting. *In proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics.* Birmingham, UK, pp. 199-206.

Tjong, K.S., Déjean, H., (2001). Introduction to the CoNLL-2001 shared task: clause identification. *In proceedings of the 2001 workshop on Computational Natural Language Learning.* Toulouse, France, pp. 53-57.

Uebersax, J.S., (1987). Diversity of Decision-Making Models and the Measurement of Interrater Agreement. *Psychological Bulletin*, 101(1), pp. 140-146.

Vijay Sundar Ram, R., Sobha, Lalitha Devi., (2008). "Clause Boundary Identification Using Conditional Random Fields". In A. Gelbukh (Eds), Computational Linguistics and Intelligent Text Processing, Springer LNCS. Berlin Heidelberg, pp. 140-150.

Vijay Sundar Ram, R., Bakiyavathi, T., Sobha, L., (2009). "Tamil Clause Identifier". *PIMT Journal of Research*, 2(1), pp.42-46.

Vilson, J.L., (1998). Clause Processing in Complex Sentences. *In Proceedings of the First International Conference on Language Resource & Evaluation.* Granada, Spain, pp. 937-943.

Yalçınkaya., Ihsan, S., (2010). An Inter-Annotator Agreement Measurement Methodology for the Turkish Discourse Bank (TDB). A thesis submitted to the Graduate school of informatics of the Middle East Technical University

A Complex Network Analysis of Syllables in Bangla through SyllableNet

Manjira Sinha[1], Tirthankar Dasgupta[1, 2], Anupam Basu[1]
[1] Indian Institute of Technology Kharagpur, India
[2]Society for Natural Language Technology Research, Kolkata
E-mail: manjira87@gmail.com, iamtirthankar@gmail.com, anupambas@gmail.com

Abstract

In this paper we present a development of a SyllableNet for Bangla language. Here, nodes of a network are the syllables and an edge between two syllables signifies that the two syllables have occurred within a same word. Number of times the two syllables occurred in a word reflects the edge weight of the graph. We use two different data sets viz. the online Rabindra Rachanabali from the web and the standard Bangla Banan Obhidhan, to perform the analysis of the network. Critical analysis of the syllabic network shows a low distance and a high clustering coefficient when compared with an associated Erdos – Renyi graph and with a random network with the same distribution of connectivity. Our comparison of network numeric with that of the Portuguese and Chinese syllabic networks reveals that despite having different origins, all of these networks have shown similar structural properties in terms of average path length, clustering coefficient and distribution of connectivity.

Keywords: SyllableNet, complex network, language, phonology, phonological neighbourhood

1. Introduction

Language is one of the most important and exciting inventions by the Human race. The different forms and manifestations in which we perceive language at present are results of the changes influenced by evolving cognitive and social dynamics of human beings over the ages. Therefore, a language is not just a tool of communication; a language also reflects the socio-cultural, geo-political and economic environment of its users. For the past few decades a plethora of work has been carried out towards developing computational models related to different aspects of language. Computational approach towards modelling of language is necessary in several ways like, to study processes like language acquisition, communication and evolution. A large amount of data has to be handled for this purpose. Thus, computational modelling provides an effective way to investigate the events and to simulate plausible mechanisms involved under different constraints and criterion. Further, computational modelling helps to incorporate the knowledge obtained regarding language in developing practical applications.

A numbers of modelling approaches have been employed to understand the language phenomena. One of the recent additions in the list is the complex network paradigm. Complex network have been used to study natural languages both from an evolutionary perspective and from the point of view of practical NLP applications like information retrieval and spell-checker (Mukherjee et al., 2011). Analyzing important network properties such as degree distribution, distribution of connectivity, clustering coefficient, component size and community structure we can develop useful insights in the organization and behaviour of language or complex cognitive system such as the mental lexicon which is defined as the representation and organization of words in the human brain (Vitevitch, 2008). Networks can be built at different levels like, word co-occurrence, semantic, syntactical, and phonological. Detailed studies of the applicability of networks on these areas are discussed in Bienmann & Quasthoff (2009) and Choudhury & Mukherjee (2009).

One of the important aspects of any spoken language is its syllables. Syllables are considered as the smallest unit of pronunciation articulated without interruption. It serves an important interface between lower level (phonetic and phonological) and higher level (morphological) representational tires of language (Shastri, Chang & Greenberg, 1999). Researchers have given syllables a very crucial role in language processing, in the perception of both speech and print (Spoehr, 1981). Cutler et al. (1986) and Content, Kearns & Frauenfelder (2001) stated importance of syllables in mental representation of words and speech production. Yup & Balota (2009) also holds the view that syllables are significant in visual recognition of words. Although, syllables are considered an important part of a language and network models of languages have been explored at phonological level, using syllables as units of phonology have not occurred until recently (Soares, Corso & Lucena, 2005; Peng, Minett & Wang, 2009)..

Bangla is an Indo-Aryan language and it is typologically agglutinative. Its phonology consists of 35 segmental phonemes and 5 non-segmental phonemes[1]. A syllable is a macro-level phonological unit of a language. Different languages have different phonological constraints which shape their syllable structures. Bangla has borrowed a large part of its vocabulary from various languages across diverse origins. Therefore, it is possible that the diversity of the vocabulary is well reflected in the pattern of phonological constructions. However, not much work has been done towards building a computational model of Bangla in terms of its syllabic structure. This can be accounted due to the following two reasons:

a) Syllable inventory of Bangla is not exhaustive and it

[1] Information obtained from www.lisindia.net/Bengali/Bengali.html

is increasing as new words are constantly added into the Bangla lexicon.

b) There is scarcity of a large syllable inventory that can be used for computational analysis.

In order to address the above mentioned issues, the primary objective of this paper is to develop a large inventory of Bangla syllables and to analyze the topology of syllable co-occurrence in Bangla words. To achieve this, we have developed a Bangla SyllableNet. The construction of the SyllableNet follows the same design techniques as discussed in the literature (Soares, Corso & Lucena, 2005; Peng, Minett & Wang, 2009). Here, syllables are considered as the representatives to study the phonetic structure of Bangla as they are the basic perceptual unit in speech production and recognition. We have used two distinct linguistic data sets namely, Bangla Banan Obhidhan and complete prose collection of Rabindranath Tagore available in the web. At each step we have compared our results with an associated Erdös-Renyi graph and a regular network and also with the network of syllables in Portuguese (Soares, Corso & Lucena, 2005) and Chinese (Peng, Minett & Wang, 2009). Our initial shows that Bangla, despite being a language of different origin, shows striking structural similarities with the other two languages.

The rest of the paper is organized as follows: section 2 contains a brief survey of related works. Corpus collection and network building have been discussed in section 3. Analysis and interpretations of network characteristics has been presented in section 4. Section 5 contains a general discussion followed by conclusion in section 6 and scope of future works in section 7.

2. Related Works

The network structures based on phonological similarities and syllables can be used to model the phenomena of language learning, lexical processing and retrieval, speech errors, new word formations (Vitevitch, 2008; Steyvers & Tenenbaum, 2005; Strokel, Armbruster & Hogan, 2006; Luce & Pisoni, 1998). The nature of organization of a network has important implications for the type of processing in the system represented by the network (Strogatz, 2001; Ward, 2002). Linguists believe that phonetics is the main element in language particularly speech processing. Vitevitch(1997) observed that malapropisms were more common in high frequency words with dense phonological neighbourhood. It has been found that word clustering coefficient varies inversely with the repetition latency and thus recognition accuracy (Gruenenfelder & Pisoni, 2005; Chan & Vitevitch, 2010). Arbesman, Strogatz and Vitevitch (2010) have studied the structure of phonological word-forms across seven different languages from different origins. Their study suggests that the linguistic networks have different property than other social networks. Although the languages show surface differences, they exhibit similar network structure. Modelling of sound inventories across different languages has exhibited surprising regularity in patterns. Self-organization of vowel inventories has been studied in Boer (2000) and Schwartz et al. (1997). Mukherjee et al. (2007) have analysed the consonant inventories of different languages through PhoNet and PlaNet. PlaNet is a bipartite where one partition consists of language nodes and the other partition consist of consonant nodes. PhoNet is the one-mode projection of PlaNet on the consonant nodes. The authors have built the networks based on the co-occurrence of consonants among languages. According to their findings the comparative analysis of consonant inventories across languages has physical significance in the process of language evolution. Bipartite spectral graph partitioning of Dutch dialects have been done to cluster varieties of different dialects of the Dutch language and identify the most distinctive features among them (Wieling & Nerbonne, 2011).All these mentioned networks are built at the word-level based on phonological similarity measures or at the phoneme level. Another method of construction of network is by taking syllables as the nodes of the network. Syllable structure in Portuguese has been explored to analyze and model the evolution of the language by phonetic elements (Soares, Corso & Lucena, 2005). Chinese language has very different structure and organization than Portuguese. However, Peng, Minett & Wang (2009) have shown that a syllable level network of Chinese shows similar structural qualities with that of the Portuguese one. According to them, the growth in the syllable structure closely resembles the way children grow their vocabulary. These findings can lay the path for studying universal qualities across the languages (Soares, Corso & Lucena, 2005; Arbesman, Strogatz and Vitevitch 2010).

3. Building the Bangla SyllableNet

As mentioned earlier, the primary objective of this paper is to build and analyze the Bangla syllable network. We formally define a network as a graph G = (V, E) containing a pair of sets (V, E). V is a set of nodes (or vertices) and E is a set of undirected edges (or links) connecting two nodes of V. There can be several criteria by which two nodes interact. In the present work we have constructed a graph whose edges are not directed but have weights among the edges. This is one of the simplest versions of a graph. For our present study we use the Bangla language to build our network.

The syllabic network consists of a graph where,

1. The set of nodes or vertices V are defined as the syllables in a given corpus.
2. $(i, j) \in E$, if the syllables i and j co-occur in a word.
3. $weight\ (i, j) = d$, here d is the total number of times a Bangla word shares the syllables i and j in a given corpus..

The network projects the nature of connectivity among the syllables of the Bangla language. Figure 1 illustrates a small portion of the original syllabic networks in Bangla. We use the following nine syllables of Bangla: (O)n, ne, rA, ge, hA, no, sho, rot, kAl. Table 1. Illustrates the meaning and syllabification of the 5 randomly chosen words used to construct the example network. The network connections among these syllables are indicated by the links between the syllables. There are 11 links in this given example network. The edge weight d_i signifies the number of times the connecting syllables occur in a

given word of a corpus. The basic input to our network is words and there syllabic form in Bangla language. However, getting such a large volume of unique word corpus along with their syllabic forms is not a trivial task particularly for resource poor languages like Bangla. In the following subsection we will discuss in details about the corpus and the technique used to extract syllables from the individual words.

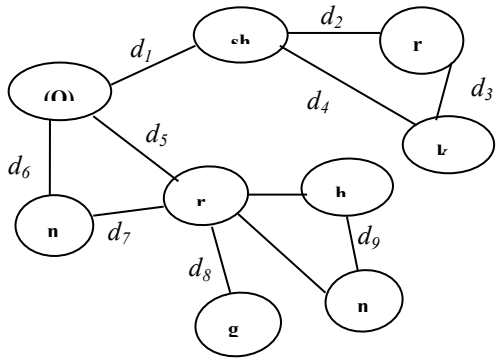


Figure 1: A small portion of the original Bangla syllable networks consisting of 5 randomly selected words

Words in the syllabified form (two syllables of a word are connected by a hyphen)	**Meaning**
(O)n - sho	Part
shO - rot - kAl	Autumn season
rA - ge	In anger
(O)n-ne-rA	Others
hA - rA - no	To loose

Table 1: Syllabified forms and meanings of the five randomly chosen words to construct the example network

3.1 Corpus Collection and Syllabification

We have used two distinct corpuses in our analysis. The first is 'Bangla Banan Obhidhan'[2](S_{BA}) and the second is complete works (prose) of Rabindra Nath Tagore[3](S_{RR}). The details of the datasets are given in table 2. We have taken these two datasets because they differ in their nature. S_{BA} is not a conventional dictionary, it contains the correct spellings of all types of words which are mispelled frequently, whereas, S_{RR} contains many derived and inflected words used only in classical literature. Therefore, the network built from S_{BA} represents the day to day language more closely than that built from S_{RR}. Yet, it is very interesting that this dissimilarity do not create significant differences in the network statistics as we will see in the following section.

[2] Available at www.snltr.org

In order to get the syllabified representation of each of the Bangla words, we have used the Grapheme-to-Phoneme (G2P) converter [4] . The given converter takes a Bangla word as input and returns its syllabified form.

	Bangla Banan Abhidhan (S_{BA})	Rabindra Rachanabali (prose) (S_{RR})
Number of words	75418	173138
Number of unique syllables	5676	7036
Number of syllables in the largest connected component	5236	6531
Number of islands (isolated connected components)	15	14
Number of isolated nodes (having zero degree)or hermits	392	462

Table 2: Details of the two corpuses used to build the networks

In order to get the syllabified representation of each of the Bangla words, we have used the Grapheme-to-Phoneme (G2P) converter[5]. The given converter takes a Bangla word as input and returns its syllabified form. However, the given G2P takes the input in the form of iTrans transcription[6]. On the other hand the corpus that we have collected is in the form of Unicode (utf-8) encoding. Thus, we have created a wrapper[7] over the G2P engine that will take any Unicode word as input, convert the Unicode word into the G2P familiar iTrans format and then extract the syllables from the iTrans formatted Bangla word using the G2P. Finally, using the syllables and the words from the given datasets we have constructed two different Bangla SyllableNets. The networks thus constructed were provided as an inputs to the Pajek (Batageji & Mvrar, 1998) and Cytoscape [8]tools for better visualization and analysis.

[4] Downloaded from www.cel.iitkgp.ernet.in
[5] Downloaded from www.cel.iitkgp.ernet.in
[6] http://www.aczoom.com/itrans/
[7] Resource uploaded at www.mla.iitkgp.ernet.in
[8] www.cytoscape.org/documentation_users.html

Language	Networks	N	<k>	L_{real}	L_{ER}	L_R	D	C_{real}	C_{ER}	CR	γ
Chinese	Putonghua B.S.	393	104	1.77	1.29	2.39	4	0.61	0.265	0.74	0.21
	Cantonese B.S.	614	109	1.91	1.37	3.31	4	0.54	0.178	0.74	0.40
	Putonghua T.S.	1240	54.3	2.40	1.78	11.91	5	0.32	0.044	0.74	0.91
	Cantonese T.S.	1671	60.8	2.34	1.81	14.23	5	0.27	0.036	0.74	0.97
	Putonghua Character	3773	21.1	3.07	2.71	89.88	8	0.23	0.006	0.71	1.40
	Cantonese Character	4942	21.2	3.04	2.79	117	10	0.19	0.004	0.71	1.49
Portuguese	Portuguese SDIC	2285	27.6	2.44	2.33	41.88	6	0.65	0.012	0.72	1.35
	Portuguese SMA	3188	28.2	2.61	3.40	57.01	8	0.50	0.009	0.72	1.36
Bangla	Bangla Banan Academy	5676	36.8	2.64	2.77	77.12	8	0.53	0.006	0.72	1.04
	Rabindra Rachanabali (prose)	7036	58.6	2.56	2.59	60.34	9	0.60	0.008	0.73	0.98

Table 3. Summary information of syllabic networks in Bangla, Chinese and Portuguese built using different datasets. For each network, results for number of nodes N, average connectivity <k>, the average distance L, the diameter D, the clustering coefficients C and C*, and the exponent γ of the best-fitting power-law distribution are included. $L_{ER,}$ and C_{ER} stands for the corresponding Erdös-Renyi networks. L_R and C_R denote the values for the corresponding regular networks.

4. Characteristics of the SyllableNet

In this section we analyze the different structural properties of our network and compare them with the network of syllables in Chinese (Peng, Minett & Wang, 2009) and Portuguese (Soares, Corso & Lucena, 2005), and also with that of random networks having same number of nodes and degree distribution (see table 3). Every result is subsequently followed by the possible implications

4.1 Small-World Properties

Small-world structure of a graph is of special relevance as it can explain the interactions between the nodes of a network leading to the robustness of the overall structure. This property depends mostly on two basic statistics namely, the average path length and the clustering coefficients (Watts & Strogatz, 1998). If N is total number of syllables in the network and k_i stands for the degree i.e. number of neighbors of node i, then the average connectivity of the network is computed as:

$$\langle k \rangle = \sum_i k_i / N. \quad \ldots(1)$$

From Table 3 it can be observed that for our given network, $\langle k \rangle \ll N$, which means the network is sparse and therefore comparable with the main works in the complex network domain (Soares, Corso & Lucena, 2005) . Let us call $l_{i,j}$ the minimum path length between vertex i and vertex j. Thus, the average path length of a syllable i to all other syllables is given as:

$$l_i = \sum_j l_{ij} / (N-1), \quad \ldots(2)$$

And the average path length of the whole network is:

$$L = \sum_i l_i / N. \quad \ldots(3)$$

The diameter of a network is the longest shortest path and it is defined as:

$$\text{Diameter } (D) = \max(l_i) \quad \ldots(4)$$

The diameter of S_{BA} and S_{RR} are 8 (between (do(u)(ri)) and (to(ri))) and 9 (between (do(u)(ri)) and (dho(ri)(A))) respectively.

The L values for associated Erdös-Renyi (Bollobas, 1985) and regular networks (Watts, 1999) are respectively:

$$L_{ER} = \ln(N)/\ln\langle k \rangle \text{ and, } \ldots(5)$$

$$L_R = ((N(N + \langle k \rangle - 2))) \,/\, ((2\,\langle k \rangle\,(N-1))) \quad \ldots(6)$$

The clustering coefficient of a node is a measure of network density. It indicates how well neighbors of a node are connected among themselves. Let E_i be the actual number of connections among the neighbors of a node i against the $k_i(k_{i-1})/2$ number of connections in a fully connected situation. Then, the clustering coefficient of the node i is:

$$C_i = 2E_i / k_i(k_i - 1) \quad \ldots(7)$$

And the clustering coefficient of a network is defined as:

$$C = \sum_i C_i / N.$$

Another way to compute clustering coefficient C^* using a weighted sum method introduced by Bollobas and Riordan (2003) is:

$$C^* = 2\sum_i E_i / \sum_i k_i(k_i - 1) \quad \ldots(8)$$

The clustering coefficient of the associated Erdös-Renyi random network (Bollobas, 1985) is:

$$C_{ER} = \langle k \rangle \,/\, (N) \quad \ldots(9)$$

and that of the corresponding regular network (Watts, 1999) is:

$$C_R = (3(\langle k \rangle - 2)) \,/\, (4(\langle k \rangle - 1)) \quad \ldots(10)$$

A graph with small-world structure has high clustering coefficient and low vertex to vertex distance. Random graphs like Erdös-Renyi have short average path length as well as low clustering coefficient, whereas, on the other hand regular lattices have high clustering coefficient and long path lengths. This means a small degree of disorder generates short paths but preserves the local topology (Watts & Strogatz, 1998).

As can be seen from Table 3, L_{BA} and L_{RR} have very short average path lengths (less than 3) similar to the associated random networks and have a high clustering coefficient which is comparable to the corresponding regular networks. Therefore, we can conclude that the syllable network of Bangla exhibit a small-world structure similar to the syllabic networks in Portuguese (Soares, Corso & Lucena, 2005) and Chinese (Peng, Minett & Wang, 2009).

Small-world structure often accounts for a rapid and robust navigation inside a network (Cancho & Sole, 2001). Despite the huge number of syllables, any syllable can be reached with fewer than three intermediate syllables on average. This information is particularly useful for the studies of lexical retrieval. It is well established that reaction time during word recognition and word production plays an important factor in retrieval of words. Thus, a word can be easily recognized if the underline structure places the syllables closer to each other. Pronunciation times in visual word recognition also get affected by the path length variations within a phonological network (Yarkoni, Balota & Yap, 2008). According to Chan and Vitevitch(2010), an analysis of a network based on phoneme distance among English words shows that neighborhood connectivity of a word influences speech production. They have found that speech error rates vary inversely with the clustering coefficient of words. Based upon the above factors, from our present SyllableNet, we can draw an analogy that if the clustering coefficient of a given Bangla syllable is high, then there will be an increase in tendency to drop the syllable during speech production. However, a detailed study on this aspect has to be done in order to get a concrete result.

4.2 Distribution of connectivity

The degree distribution *p(k)* of a network is the fraction of nodes having degree *k*. *p(k)* is particularly significant in modeling how a network grows over time. Figure 2.a. and Figure 2.b. shows the variation of *p(k)* with *k* in S_{BA} and S_{RR} in log-log scale. Both of the curves are approximately straight lines. A graph is said to follow a power law degree distribution if, $(k) \propto k^{-\gamma}$, where γ is the exponent of the distribution. One of the interesting properties of the power law is that it makes a network scale invariant (Barabási & Albert, 1999). Therefore, networks having this distribution are said to be scale free. This behavior is responsible for the straight-line nature in the log-log plot of *p(k)* versus *k*. Thus, we can say that degree distributions of the two Bangla SyllableNet roughly follow the power law. In our networks the exponent γ for the best fit is γ_{BA} = 1.041 and γ_{RR} = 0.983 respectively (refer table 3). These values are comparable to that of the Chinese (Peng, Minett & Wang, 2009) and Portuguese (Soares, Corso & Lucena, 2005) networks. Power-law behavior of the degree distribution indicates the fact that when a new node arrives in the network it has high frequency will require less entropy to get activated and will be more available for production. If we define the frequency in terms of degree of a syllable, then this can account for the preferential attachment phenomena. This finding is important in the study of language evolution.

The scale-free property of a network provides immunity from external disturbances directed towards randomly chosen nodes but makes the network vulnerable when targets are the high degree nodes (Albert, Jeong & Barabási, 2000). In table 4, degree-wise the top ten syllables are shown from S_{BA} and S_{RR} respectively. It can be seen that most of the high degree syllables are common in both the datasets (S_{BA} and S_{RR}). This can be accounted by the fact that these syllables are the most frequently used in daily communication.

Banan Obhidhan (S_{BA})		**Rabindra Rachanabali (S_{RR})**	
Syllable	Degree	Syllable	Degree
"ri"	1578	"ke"	2179
"ni"	1556	"ro"	1959
"to"	1445	"to"	1934
"ti"	1392	"ni"	1843
"ro"	1360	"rA"	1827
"no"	1325	"ri"	1800
"nA"	1158	"no"	1776
"kA"	1157	"te"	1761
"rA"	1141	"re"	1712
"tA"	1111	"bA"	1658

Table 4: Syllables having top ten degrees from S_{BA} and S_{RR}

Assortativity defines the mixing pattern of nodes in a network. It is measured by the correlation between nodes having similar degrees. Consider a network having *M* edges and j_i and k_i are the degrees of the two end vertices of an edge *i*. Then the assortativity *r* of the network is computed as:

$$r = \frac{(M^{-1}\sum_i j_i k_i - [M^{-1}\sum_i 1/2(j_i + k_i)]^2)}{(M^{-1}\sum_i 1/2({j_i}^2 + {k_i}^2) - [M^{-1}\sum_i 1/2(j_i + k_i)]^2)} \quad \ldots(11)$$

If *r* is positive then it is said that the network has assortative mixing and it implies that the high degree nodes tend to connect with the other high degree nodes. On the other hand if *r* is negative then the network shows disassortative mixing i.e. high degree nodes prefer to get

attached with the low degree nodes (Newman, 2002).
Both the networks show a negative correlation between degrees with r_{BA} = -0.2964 and r_{RR} = -0.3408. These results go well with the disassortative mixing shown by high proportion of empirical networks like biological, technological or linguistic (Johnson et al., 2005). This property along with the scale-free nature of our networks can point to the fact that the total entropy of the network is restricted to a small finite value (Bianconi, 2009; Johnson et al., 2010). If we take the degree of a node as a indicator for its frequency, then from a speech production and recognition perspective, this mixing pattern can be considered beneficial as even the least used syllables are at a short distance from the most frequently used ones. Thus, the uncommon syllables can be accessed during communication with a little increase in effort (Cancho & Sole, 2001).

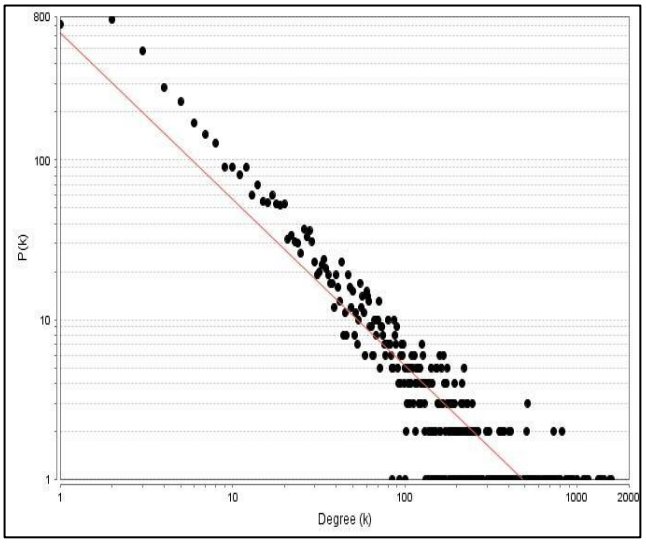


Figure 2.a. Degree distribution plot of Bangla Banan Obhidhan in log-log scale

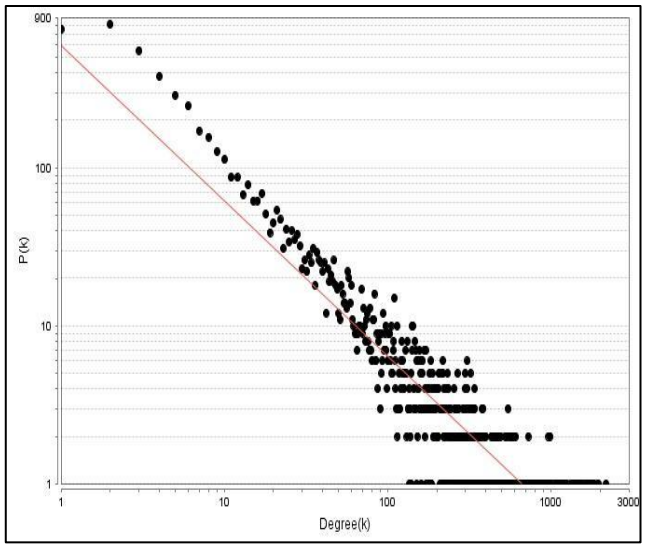


Figure 2.b. Degree distribution plot of Rabindra Rachanabali in log-log scale

5. General Discussion

It is well established that language is a complex cognitive system. A network approach can facilitate the deeper understanding of the underlying structure and topology of the language. The mental lexicon has been explored with the complex network analysis at different levels based on syntax, orthography, morphology and phonology (Bienmann & Quasthoff, 2009; Choudhury and Mukherjee, 2009). It has been argued that network modeling allows a less complicated and convenient way to study the structural dynamics and growth processes of a system. Thus, looking at a real-world complex system like language through network perspective can help to gain useful insights regarding language acquisition and lexical decision-making (Callaway et al., 2001; Vitevitch, 2008).

As discussed earlier, we have built and analyzed the network of syllables in Bangla through two different SyllableNet. Here, syllables are used as the basic phonetic units of Bangla. We have used two distinct linguistic data sets namely, Bangla Banan Obhidhan and complete prose collection of Rabindranath Tagore available in the web. Our results have been compared with an associated Erdös-Renyi graph and a regular network and also with the network of syllables in Portuguese (Soares, Corso & Lucena, 2005) and Chinese (Peng, Minett & Wang, 2009) at each step.

Our observations show that co-occurrence structure of syllables from both the corpuses form small-world network and have scale-free properties as well as disassortative mixing by degrees. The small world feature helps in the robust and rapid movement in the network, whereas, the scale-free properties provide insights about the growth of the network. The preferential attachment concept states if a new node is inserted in the network at any given point of time, it will tend to be connected to the already existing nodes with the highest number of links (Albert, Barabási, 2002).

This phenomenon can also be used to understand the language acquisition process. As newly formed sound units are likely to get associated with the high degree ones, it may help in getting acquaintance with the new words by referring it phonologically to a frequently used word. We have already shown in section 4.2. how negative correlation in degrees possibly leads to efficient sound production and recognition. In addition to the above mentioned properties, each of our networks has a large interconnected component along with many syllable islands and hermits (refer table 2). This can be due to the morphological and phonotactic constraints on a new word formation.

6. Conclusion and Future Work

Our comparison of network numeric with that of the Portuguese and Chinese syllabic networks display that all of these networks have similar structural properties in terms of average path length, clustering coefficient and distribution of connectivity (refer table 3) . Study regarding assortativity is absent in the other two languages. This observation is very interesting: Portuguese, Chinese and Bangla, three languages are

from three different backgrounds. Portuguese is a Romance language, Chinese descends from the Sino-Tibetan family of languages and on the other hand, Bangla is of Indo-Aryan origin. Yet all of them show universal structural qualities in terms of syllables. This can point to the hypotheses that different languages actually evolved from a common ancestor.

The Bangla SyllableNet can be used to study the language acquisition process. Instead of taking an average vocabulary, if individual's vocabulary over different time points can be modeled and the network growth is studied then we can possibly infer about the path of language acquisition. This can further help to gain insights in language related disorders. Modeling a child's vocabulary can help to identify the propagation of sound change through the lexicon (Gierut, 2001).

If the challenges of preparing a well organized database documenting the change of sounds of a language over a long time period and across geographic regions can be overcome, then the network approach can be applied to explain the processes of evolution and modifications of different dialects of a language (e.g. Wieling and Nerbonne, 2011). At present, to study these patterns, one possible attempt can be to analyze different Bangla news and blog corpuses with the help of SyllableNet. As news and blogs are written in more colloquial languages to reach a greater number of people, analyzing them can help studying the demographic patterns of syllables of Bangla.

Choudhury et al. (2007) have shown that the probability of real word error in a language is directly proportionate to the network structure of the words in the language based on orthographic similarity. Similarly, we can investigate if such a relation exists between the errors during speech production and recognition and the Bangla syllableNet. This can have application in the field of automatic speech recognition as it is established that syllable level error corrections in automatic speech recognition gives better results (Jeong, Kim & Lee, 2004). Another scope of this graph theoretic approach can be to simultaneously study along different dimensions of a language. For example, a network can be developed where there can exist multiple type of links like based on morphology, semantics, phonology etc. between a pair of nodes (Kohely & Pattison, 2005). From this we can explore how overlap of different relations shapes the overall lexical retrieval and acquisition process. Overall, the complex network way provides a very useful and convenient tool to develop a better and deeper understanding of language.

Acknowledgement

We are thankful to the Society for Natural Language Technology Research, Kolkata for providing us the full Rabindra Rachanabali corpus and the Unicode to iTrans converter.

7. References

Albert, R., Barabási, A.L. (2002). Statistical Mechanics of Complex Networks. *In Review of Modern Physics,* 74, pp.47—97.

Albert, R., Jeong, H., Barabási, A.L. (1999). The Diameter of The World Wide Web. *In Nature,* 401, pp.130-131.

Arbesman, S., Strogatz, S.H., Vitevitch, M.S. (2010). The Structure of Phonological Networks across Multiple Languages. *In International Journal of Bifurcation and Chaos,* 20(3), pp.679--685.

Barabási, A.L., Albert, R. (1999). Emergence of Scaling in Random Networks. *In Science,* 286, pp.509--512.

Batageji, V., Mrvar, A. (1998). Pajek : A Program for Large Network Analysis. *In Connections,* 21, pp.47--57.

Bianconi, G. (2007). Entropy of Network Ensembles. *In Physical Review E,* 79.

Bienmann, C., Quasthoff, U. (2009). Networks Generated from Natural Language Text. *In Dynamics On and Of Complex Network, Modelling and Simulation Science, Enginnering and Technology*, part 2, pp.167--185.

Bollobas, B. (1985). Random Graphs. London : Academic Press.

Callaway, D.S., Hopcroft, J.E., Kleinberg, J.M., Newman M.E.J., Strogatz, S.H. (2001). Are Randomly Grown Graphs Really Random ? *In Physical ReviewE : Statistical, Nonlinear, and Soft Matter Physics,* 64(4 Pt1).

Cancho, R.F., Solé, R.V. (2001). The Small World of Human Languages. *In Proceedings of The Royal Society : Biological Sciences,* 268, pp.2261--2265.

Chan, K.Y., Vitevitch, M.S. (2010). Network Structure Influences Speech Production. *In Cognitive Science,* 34, pp.685--697.

Choudhury, M., Mukherjee, A. (2009). The Structure and Dynamics of Linguistics Networks. *In Dynamics On and Of Complex Network, Modelling and Simulation Science, Enginnering and Technology,* part 2, pp.145--166.

Content, A., Kearns, R., & Frauenfelder, U.H. (2001). Boundaries Versus Onsets in Syllabic Segmentation. *In Journal of Memory and Language,* 45, pp.177--199.

Cutler, A., Mehler, J., Norris, D., & Segui, J. (1986). The Syllables's Differing Role in the segmentation of French and English. *In Journal of Memory and Language,* 25, pp.385--400.

de Boer, B. (2000). Self-organization in Vowel Systems. *In Journal of Phonetics,* 28(4), pp.441--465.

Gierut, J.A. (2001). A Model of Lexical Diffusion in Phonological Aquisition. *In Clinical Linguistics & Phonetics,* 15, pp.19--22.

Gruenenfelder, T.M., Pisoni, D.B. (2005). Modelling the Mental Lexicon as Complex System : Some Preliminary Results using Graph Theoretic Ananlysis. *In Research on Spoken Language Processing,* progress report no. 27. Indiana University.

Jeong, M., Kim, B., Lee, G.G. (2004). Using Higher-Level Linguistic Knowldge for Speech Recognition Error Correction in a Spoken Q/A Dialog. *In Proceedings of th HLT-NAACL Special Workshop on Higher-Level Linguistic Information for Speech Processing,* Boston, USA : pp.48--55.

Johnson, S., Torres, J.J., Marro, J., & Muňoz, M.A. (2005). Shannon Entropy and Degree Correlations in Complex Networks. *In Physical Review Letters,* 104.

Koehly, L.M., Pattison, P. (2005). Random Graph Models for Social Networks : Multiple Relations or Multiple Raters. *Models and Methods in Social Network Analysis,* NY : Cambridge University Press, pp 162--192.

Luce, P.A., Pisoni, D.B. (1998). Recognizing Spoken Word : The Neighborhood Activation Model. *In Ear and Hearing,* 19, pp.1--36.

Mukherjee, A., Choudhury, M., Hassan, S. & Muresan, S. (eds.) (2011). Networks Models for Cognitive and Social Dynamics of Language. *In Elsevier Journal of Computer Speech and Language.* Volume 25(3), pp. 635--638.

Mukherjee, A., Choudhury, M., Basu, A., Ganguly, N. (2007). Modelling the Co-occurrence Principles of the Consonant Inventories : A Complex Network Approach. *In International Journal of Modern Physics C,* 18(2), pp.281-265.

Mukherjee, A., Choudhury, M., Basu, A., Ganguly, N. (2007). Self-organization of the sound inventories : Analysis and Synthesis of the Occurrence and Co-occurrence Networks of Consonants. *In Journal of Quantative Linguistics,* 16(2), pp.157-184.

Newman, M.E.J. (2002). Assortative Mixing in Networks. *In Physical Review Letters,* 89(20).

Peng, G., Minett, J.W., & Wang, W.S--Y. (2009). The Networks of Syllables and Characters in Chinese. *In Journal of Quantitative Linguistics,* 15(3), pp.243--255.

Schwartz, J.-L., Boë, L.-J., Vallée, N., & Abry, C. (1997). The Dispersion-Focalization Theory of Vowel Systems. *In Journal of Phonetics,* 25, pp.255-286.

Shastri, L., Chang, S., Greenberg, S. (1999). Syllable Detection and Segementation Using Temporal Flow Neural Networks (Postscripts only). *In Proceedings of the Fourteenth International Congress of phonetic Sciences.* San Francisco, USA.

Soares, M.M., Corso, G., & Lucena, L.S. (2005). The Network Syllables in Portuguese. *In Physica A,* 355, pp.678--684.

Spoehr, K.T. (1981). Word Recognition in Speech and Reading : Toward a Single Theory of Language Processing. In P.D. Eimas & J.L. Miller (Eds.), *Perspective on the study of speech.* Hillsdale,NJ : Erlbaum.

Steyvers, M., Tenenbaum, J. (2005). The Large Scale Structure of Semantic Networks : Statistical Analyses and a Model of Semantic Growth. *In Cognitive Science,* 29, pp.41--78.

Strokel, H.L., Armbruster, J., Hogan, T.P. (2006). Differentiating Phonotactic Probability and Neighborhood Density in Adult Word Learning. *In Journal of Speech Language and Hearing Research,* 49, pp.1175--1192.

Vitevitch, M.S. (1997). The Neighborhood Charcteristics of Malapropisms. *In Language and Speech,* 40, pp.211--228.

Vitevitch, M.S. (2008). What can Graph Theory Tell Us About Word Learning and Lexical Retrieval ? *In Journal of Speech Language and Hearing Research,* 51(2), pp. 408--422.

Watts, D.J. (1999). Small Worlds : The Dynamics of Networks between Order and Randomness. Princeton, NJ : Princeton University Press.

Watts, D.J., Strogatz, S.H. (1998). Collective Dynamics of « Small--World » Networks. *In Nature,* 393, pp. 440--442.

Wieling, M., Nerbonne, J. (2011). Bipartite Spectral Graph Partitioning for Clustering Dialect VArities and Detecting Their Linguistic Features. *In Computer Speech and Language,* 25, pp.700--715.

Yap, M.J., Balota, D.A. (2009). Visual Word Recognition of Multisyllabic Words. *In Journal of Memory and Language,* 60(4), pp.502--529.

Yarkoni, T., Balota, D., Yap, M. (2008). Moving beyond Coltheart's N : A New Measure of Orthographic Similarity. *In Psychonomic Bulletin & Review,* 5, pp.971--979.

Blurring the demarcation between Machine Assisted Translation (MAT) and Machine Translation (MT): the case of English and Sindhi

Pinkey Nainwani

Centre for Linguistics, Jawaharlal Nehru University

New Delhi 110067

pinkeybhu39@gmail.com

Abstract

Dealing with divergences, at present, is a major concern of any Machine Translation (MT) system. This paper is an attempt to classify and analyze different types of translation divergences between English-Sindhi on the lines of Dorr ('90, '93, and '94) and further suggests some strategies to handle divergences with rule-based methods. The paper also discusses some of the recent studies on classification and handling of divergences involving Indian languages (Gupta, '09, Sinha, '05).

Keywords: Language divergences, "effective" machine translation, rule-based methods

1. Introduction

The multilinguality of India is well known. There are 4 language families: Indo Aryan (76.87 % speakers), Dravidian (20.82 % speakers), Austro-Asiatic (1.11 %), and Tibeto-Burman (1%). Cognate Languages are very close at the morphological and syntactic level. With over 452 languages[1], some have been widely explored from a computational perspective. Sindhi is an exception to it.

Sindhi[2] is an Indo-Aryan language of the Indo-Iranian branch of the Indo-European language family and it is one of the scheduled languages as recognized by constitution of India. The major linguistic differences between Sindhi and English is that the former has verb final (or syntactically, head final), postpositional, free word ordering, adjectives precede and follow the head nouns, partial null subject language (only in the case of first person) whereas the latter has verb medial (or head initial), prepositional, fixed word ordering and mostly adjectives precede the nouns, non null subject language. These linguistic differences are difficult to handle via machines. Therefore, language understanding, language generation and mapping between language pairs are linguistic challenges in developing an MT system. This motivates the development of an "effective" MT system so that a larger population (the population which cannot understand English) can access them. MT is relatively young in India, in its around 20 years of history, few MT systems and projects are being developed such as Anglabharti, Anussaraka, Shakti, and many others.

The idea behind development of MT system is to produce high-quality translation; in practice the output of most MT systems is post edited. These systems are not capable to give the desired output. Other problems are accounted for that there is no equivalent word in the target language, word order differences, difficulties in translating idiomatic, metaphoric and cultural expressions, etc.

To achieve a desirable output, MT systems must be able to process language-specific phenomena for each individual language pair. Languages are highly ambiguous and each language has its own peculiarities. Therefore, the study of divergences becomes crucial because it helps us to build an "effective" machine translation. Divergence occurs "when structurally similar sentences of the source language do not translate into sentences that are similar in structures in the target language", (Dorr, 1993). Divergence is a purely language-dependent phenomenon, its nature may change along with the source and target language taken under consideration. Moreover, the identification of divergences and handling them would blur the demarcation between Machine Assisted Translation (MAT) and MT.

The rest of the paper is organized as follows: Section 2 describes the related work. The classifications of the divergences between English-Sindhi are described in section 3. The proposal of catering these divergences with specific rules is discussed in section 4. Finally, section 5 concludes the paper and also elaborates the future work.

[1] http://www.ethnologue.com

[2] It is spoken by 53,410,910 people in Pakistan, according to the National Government Statistics Division and there are around 5,820,485 speakers in India.

2. Related Work

In this section, we present a broad overview of closely related work.

2.1 Classifications of divergences

Among the works of Divergences, Dorr ('90, '93, and '94) has a significant place. She discusses two types of classifications of divergences: syntactic divergences, characterized by each language's syntactic properties independent of the actual lexical items that are used, and lexical-semantic divergences, characterized by properties that are entirely lexically determined. Syntactic translation divergences are accounted for syntactic parameterization of principles of Government and Binding theory. Basically, they are constituent order divergence (divergent constituent order in the concerned languages), preposition stranding divergence (differences in proper governor), long-distance movement divergence (the choice of bounding nodes varies in language-pairs), null-subject divergence (whether a subject position can be empty), dative construction (does or doesn't allow alternation of dative construction). She also has defined seven types of lexical-semantic divergences based on English-Spanish and English-German translations; thematic divergence (changes in argument structure), promotional divergence (head swapping), demotional divergence (head swapping, here lexical category becomes functional category while translating from source language into target language), structural divergence (the verbal object is realized as a noun phrase in one language and as a prepositional phrase in other language), conflational divergence (the sense conveyed by a single word in one language requires at least two words in other language), categorical divergence (change in category), lexical divergence (the event is lexically realized as the main verb in one language but as different verb in other language).

Gupta et al. ('03) and Sinha et al. ('05), remarked that the classifications of translation divergences as proposed by Dorr (('90, '93, and '94) are not sufficient to capture translation divergences for MT between English and Hindi. They have observed the following list of translation divergences (though the list is not very exhaustive) needs to be taken care of as far as English-Hindi MT is concerned: 1) Reduplicative Words (repetition of root words to emphasize the context, very common phenomenon in Indian languages, difficult to find in European languages such as dʰɪre-dʰɪre "slowly (*slow-slow)), 2) Determiner System (Hindi lacks an overt article system whereas English has (in)definite articles that mark the (in)definiteness of the noun phrase exactly, eg. ləṭka aya → the boy came), 3) Morphological gaps (Hindi uses certain types of passive constructions to mark a certain kind of modality function, the exact counterpart is difficult to find in English, e.g. ram se gələtı ho gəyı → "Ram made the mistake unintentionally". To capture the intended sense, English has to resort to other devices to fill the gaps, 4) Conjunctions and Particles (Hindi has different types of particles like "wala, nə", there is no exact counterpart in English), 5) Gerunds and participle clauses (The adjunct verbal clauses and complement verbal clauses in Hindi is realized by infinitival clauses in English), and 6) Honorific (Hindi employs several linguistic markers such as plural pronouns and plural verbal inflections which is missing in English).

3. Classification of English-Sindhi divergences

Divergence is purely language dependent phenomenon. The present study lists and discusses the divergences between English-Sindhi (E-S). In this section, first I will discuss the divergences identified by Dorr ('90, '93, and '94) and others with reference to English-Sindhi and Sindhi-English and it also presents further cases of divergences between English and Sindhi which are not discussed earlier.

3.1 Constituent Order divergence

Unlike English rigid constructions, (S)ubject-(V)erb-(O)bject, Sindhi has relatively free word order, though the standard order is SOV.

(1) E: I love India
S: mu bharət-sã pyar kəd ɪ ahiyã
I India-PSP love do be-PRES

English word order patterns do not match Sindhi word order and therefore, the problem of divergence arises.

3.2 Pleonastic divergence

In English, pleonastic subjects ("it", "there") have no semantic content but they are needed for the grammaticality of the sentences. On the other hand, Sindhi does not require these kinds of elements as is illustrated in (2).

(2) E: It is raining
S: mĩ to pɛ
rain be-PROG fall

There is no equivalent word of "it" in Sindhi.

3.3 Adjunction divergence

An adjunct construction that diverges between English and Sindhi is prepositional phrase adjunction with respect to a verb phrase. In English, prepositional phrase occurs to the right side of the verb, at the maximal level (i.e., not between the verb and its object), whereas in Sindhi prepositional phrase occurs to the left side of the verb and also between the verb and its object, but it cannot adjoin maximally on the right side.

The following example illustrates these distinctions:

(3) E: I met John [$_{P\text{-}Max}$ in the garden]
S: mu jɔn-sã [$_{P\text{-}Max}$ bəgɪce me] miluə̃m
I John-PSP garden-in met-be-PST

The above example also shows the **head order divergence** (not listed by Dorr and Sinha), in English,

the head governor occurs to the left of the complement and, in Sindhi, it occurs to the right of the complement.

I have not come across other syntactic divergences (null subject divergence, dative divergence and long-distance movement divergence) between English and Sindhi.

The lexical-semantic divergences classifications discussed by Dorr based on the lexical semantic properties of the source-language/target language.

3.4 Thematic divergence

Here, the theme is realized as the verbal object in one language but as the subject of the main verb in another. Consider the following example:

(4) E: John please Mary
S_1: mɛri jɔn-k^{h}e pəsənd kãdı ahe
Mary John-acc like do-3SF be-PRES
S_2: mɛri-k^{h}e jɔn pəsənd ahe

There are two possible Sindhi translations of the above English sentence. The verbal object *Mary* in English becomes the subject of the main verb in Sindhi. Thematic divergence occurs not so often in English-Sindhi translation.

3.5 Promotional divergence

It is a type of head swapping divergence. The modifier is realized as an adverbial phrase in one language but as the main verb in another. For example:

(5) E: Fan is on
S: pãk^{h}o həle to
fan on be-PRES-PROG

English modifier, *on* (adverbial phrase) is realized as the main verb, *həle* in Sindhi. In Chomskyan Syntax, noun, verb, adjective, and prepositions are considered as higher level categories whereas adverbs and conjunctions are known as lower level category. In this divergence, lower level category (adverbial modifier) gets promoted to higher level category (verb).

3.6 Structural divergence

The verbal object is realized as a noun phrase in one language and as a prepositional phrase in other language. Example:

(6) E:Raam attended the marriage
S: ram wyawəh- te ayo
raam marriage-PSP attended

In English, *the marriage* is a noun phrase but it becomes a prepositional phrase *wyawəh te* in Sindhi.

3.7 Conflational divergence

Conflation is the lexical incorporation of necessary components of meaning (or arguments) of a given action. This kind of divergence results when two or more words are required in one language to convey a sense which is expressed by a single word in another language as given in (7):

(7) E: He stabbed me
S: hu mu-k^{h}e curi-sã mare
he I-PSP knife-PSP kill

There is no one-word equivalent of *stab* in Sindhi, the intended sense of *stab* cannot be obtained unless we do not introduce the word *curi* (knife) in Sindhi.

3.8 Categorial divergence

Categorial divergence is the mismatch between parts of the speech of the pair of translation languages. The predicate is adjectival in one language but nominal in the other. For example:

(8) E: She is jealous of me
S: huə mu-sã saɽdı ahe
she me-PSP jealous be-PRES

In English, *jealous* is an adjective whereas in Sindhi, it has a verbal mapping.

3.9 Lexical divergence

This kind of divergence arises due to the unavailability of the exact translation for a construction in one language into another language. In other words, two different verbs are chosen for a single expression in the language pairs. For example:

(9) E: John broke into the room
S: jɔn daɗayı-kəre kəməre-me g^{h}usı ayo
John force- put room-PSP enter be-PST

To get the intended sense, Sindhi has to use other devices like an adverbial element *daɗayı-kəre* (put the force), and on the other hand, *ghusəɖ* is the literal translation of *enter*.

3.10 Morphological divergence

3.10.1 Adjective-Noun agreement

This type of divergence illustrates the gaps between the system of English and Sindhi. For instance, in (10a), (10b) and (10c), Sindhi counterparts of English adjectival phrases show that in Sindhi adjective inflects for gender and number whereas English lacks this.

(10a) E: good boy
S: suʈ^{h}o cokro
(10b) E: good girl
S: suʈ^{h}ı cokrı
(10c) E: good girls
S: suʈ^{h}ıyə̃ cokrıyə̃

In addition to the above, not all the adjectives in Sindhi inflect for gender and number, therefore, these kinds of adjectives need to be handled separately because of their different behaviours. The handling of these kinds of adjectives is discussed in the following section. On the other hand, there are other adjectives which are loanwords from Hindi such as *sərvʃeʂəʈ^h* or *utəm*, do not inflect for person, number and gender.

3.10.2 Object-Verb agreement

Due to its rich morphology, Sindhi has the property where objects also agree with verb.

(11) E: I have eaten many breads
S: mu daɗa pʰulka kʰada

In (11), *pʰulka* (breads, object) agrees with the verb *kʰada* (eaten).

The following four divergences i.e. causative construction, reduplicative words, echo constructions, and "wara" divergence arise while translating from Sindhi to English. On these lines, it will be evident that Sindhi and Hindi syntactically behave similarly to large extent.

3.10.3 Causative construction

Under this section, the cases of divergences related to verb causitivity are discussed:

(12) S: mohən kʰilo
E: Mohan laughed

(13) S: mohən sita-kʰe kʰilaye
E: Mohan Sita-psp made-to-laugh

The *kʰilaye* (made-to-laugh) form of Sindhi is morphologically derived from *kʰilo*.

3.10.4 Reduplicative words

Indian languages are known for its reduplicative features (where the whole or part of the previous word is repeated). In Sindhi, almost all type of words can be reduplicated to denote number of functions. This presents potential area of divergence between Sindhi and English MT (14), the same cannot be true in the case of English-Sindhi MT (15):

(14) S: mu kare-kare thaki payo ahiyã
I do-do tried went be-PRES
E: I am tired of doing it (repeatedly)

(15) E: The very small girls of this village are beautiful.
S_1: hinə gãv-ji ɗaɖʰiyũ nəɗĩyũ cokriyũ
this village-psp very small girls
tʰokiyũ ahẽ
beautiful be-PRES
S_2: hinə gãv-ji nəɗĩyũ nəɗĩyu cokriyũ tʰokiyũ ahẽ

There is no exact counterpart of reduplicative verb in English. To manifest the intended sense of Sindhi in English, the only way out is to use of other categories (in (14) the use of adverbs). The example in (15) shows that in the E->S translation, there is no divergence.

3.10.5 Echo construction

Echo words are partially reduplicated to denote wide range of meanings with slight semantic constraints:

(16) S: təhã kʰadəv wadəv ki nə
E: you ate etc or not
"Have you taken something for eating?"

These are typical features of Indian languages, therefore, the effective MT systems need to be built to capture all these kinds of phenomenon.

3.11 "wara" divergence

Sindhi *wara* is the counterpart of Hindi *wala*. In literature (Sinha '05), *wala* has been considered as particle. According to the definition of particle, it does not change its form, but, *wala* does change the form depending on the context. The word *wara* can occur with any syntactic category and denote number of functions which are mapped in English by different linguistic devices.

(17) S: kəm kərəɽ wara maɽu
work do people
E_1: Working people
E_2: The people who do the work

The above sentence has two correspondents in English. In first translation, English preserves the participial construction as it is in Sindhi whereas in second, it gets translated into relative clause.

3.12 Honorific Marker

Indian languages employ several linguistic markers like the use of plural pronouns and plural verbal inflections for honorificity:

(18) E: your father
S_1: tunjo piu (for younger)
S_2: təhanja piə (for elders)

The plural forms in Sindhi make the distinction between elder and younger and also the following nouns get affected.

The infinitive constructions (3.12) and tense mismatch (3.13) are newly found lexical- semantic type of divergences between English and Sindhi which have not been studied with reference to English and Hindi.

3.13 Infinitive Constructions

The infinitive constructions show different types of behaviour while translated from English to Sindhi. These constructions are not widely explored. Under this section, we present the rich case of infinitive constructions (bare form of verb with or without to) with multiple functions:

To-infinitive modifying words indicating time:

(19) E: It is time to go home
S: gʰər vənəɽ jo səməy tʰi wəyo
home go psp time be went

To-infinitive as a modifier of main verb:

(20) E: I would love to live in Delhi
S: mu dili me rahaɽ pəsand kəndum
I delhi in live love would

To-infinitive related with adjective modified by "too":

(21) E: This stuff is too expensive to buy
S: hiə ciz kʰəridəɽ waste məhəngi aheN

this stuff to buy for expensive be-PRES

It can be observed that wherever infinitive form of verb occurs in English, it takes gerundial form of the verb in Sindhi.

The next example shows the semantic differences between gerundial and infinitive constructions:

Gerundial Construction:
(22) E: I like dancing
S: mu-k^{h}e nəcəɽ pəsənd ahẽ
I-PSP to dance like be-PRES

The infinitival construction:
(23) E: I like to dance
S: mu-k^{h}e nəcəɽ sut^{h}o ləgədo ahẽ
I-PSP to dance like feel be-PRES

The semantic difference between the above two sentences is in (22) the person likes dancing whether (s)he knows it or not whereas in (23), the person knows dance and (s)he usually does it in leisure time.

ECM constructions:
(24) E: I want him to go
S: mu cahiðo ahiya ki tu vən
I want be-PRES COMP you go

Exceptional Case Marking (ECM) constructions are those constructions where the subject of the lower infinitive gets its case from the main verb of the higher clause. Now, the mapping of infinitival form of verb of English into Sindhi differs, it takes the root form of verb itself in Sindhi. Moreover, Sindhi translation doesn't retain ECM construction any more, "mu" gets the case from its local verb "cahiðo" and "tu" get the case from "vən". Therefore, all the ECM constructions in English while translating into Sindhi become complementizer constructions.

3.14 Tense Mismatch

The tense feature in a language gives the tense and aspectual information of the verb. In English, the tense is always the same in main and subordinating clause, the mismatching of tenses in main and subordinating clause leads to the ungrammaticality of the structure. On the contrary, Sindhi does not have this kind of restriction. Consider the following example:
(25) E: I thought he was here
S: mu soco (ki) hu hite ahe
I think-PST he here be-PRES

In Sindhi, the tense of the subordinating clause does not match with the tense of the main clause, even so, the sentence is perfectly acceptable. Though this difference looks very minor at the outset, it poses major linguistic challenges for English-Sindhi and Sindhi-English MT.

4. Strategies for handling divergences

Dorr ('93, '94) has further explained interlingual representation technique (the extended version of Lexical Conceptual Structure (LCS)) to resolve the translation divergences. This interlingual approach treats translation as a process of extracting the meaning of the input and then expressing that meaning in the target language. This representation is suitable to the task of translating between divergent structures for two reasons: (1) it provides an abstraction of language-independent properties from structural idiosyncrasies and (2) it is compositional in nature. On the other hand, Gupta (2009) has proposed the solution of divergences between English-Hindi language pair with the help of Example Based Machine Translation (EBMT). EBMT generates a translation for the input sentence with the help of the retrieved example database. Sometimes, this retrieved example database may not be helpful in generating the correct translation of the given input sentence. Consequently, developing an efficient adaptation scheme becomes extremely difficult. A possible solution could be building two separate Example Bases (EBs): Divergence EB and Normal EB so that given input sentence retrieval can be made from the appropriate of the example base. This scheme can work successfully only if the EBMT system has the capability to judge from the input sentence itself whether its translation will involve any divergence. To deal with this issue, two following major evidences are used in succession: 1) to check the Functional Tags (FTs) of the constituent words, and, 2) to check the semantic similarities of the constituent words.

The techniques for handling divergence are complex phenomenon. The MT systems which have been built in India till date do not capture divergences. Our approach is to handle the divergences between English-Sindhi is rule-based methods. Statistical methods rely on large parallel example database which would be slightly difficult to create for Sindhi (less-resource language).

The rule-base MT system requires large volume of computational resources such as:

Creating a Rich Lexical database: with noun verb paradigm;

Creating a Rule base for handling: normal translations and divergences

Besides developing above lexical resources, the following mapping rules are being created to handle specific divergences as shown in table 1:

Types of Divergences	**Mapping Rules**
Constituent Order divergence	SVO → SOV
Adjunction divergence	SVO+[adverbial modifier]→SO[adverbial modifier]+V

Table 1: Mapping for constituent order and adjunction divergence:

To handle the Lexical semantic divergences in which the semantic mapping of the words gets altered between the language pairs. We have come up with some rules which can be incorporated in the process of building MT:

Two types of lexical entries can be done for the words which have more than one semantic mapping in the target language as given in table 2:

ON : [P] : [$_{Sindhi}$ te]
ON: [V]: [$_{Sindhi}$ hɔle]

Table 2: Lexical entry for polysemous words

When this word "ON" is followed by noun in source language, fetch the meaning of [P] in target language and when it is preceded by verb, fetch the meaning of verb. This rule can accommodate promotional divergence.

In Sindhi, most of the adjectives (discussed in section 3) inflect for number and gender. To reduce the human efforts for post editing, we can put a check for "Adjective+Noun construction" whenever an adjective precedes the noun, it must check its gender and number information, and accordingly takes the respective adjective.

The notion of causatives is very specific to Indian languages. There are Indian language in which double causative (two morphological from derived from single verb root) is also present (e.g. Hindi) and some Indian language (like Sindhi, Punjabi) has only one morphological form to get the meaning of double causative while translating from Hindi to Sindhi or Hindi to Punjabi. English lacks this feature and introduces the verb "made" except for some pairs like eat-feed, kill-die.except. In order to cater this type of divergence, the verbs of Sindhi which do not have exact equivalents in English, the list of those verbs can be generated with all possible meanings, and, later can be plugged into the rule-base. For instance, the case of the lexical entry for causative verb *k^{h}ilaye:*

Lexical entry	k^{h}ilaye [V] [$_{English}$ made to laugh]
Input	mohən sita-k^{h}e k^{h}ilaye
Rule-base	[S O V] --→ [SVO]
Output	*Mohan made to laugh Sita*
Correct Output	*Mohan made Sita to laugh*

Table 3: the case of causative verbs

The above example shows that only the exhaustive lexical entries for causative verbs would not suffice to capture these kinds of divergences, once these constructions are in place, they again need to be filtered out with comprehensive rules.

5. Conclusions and Future Work

The motivation behind the study of divergence is to develop robust MT systems with little or no human editing. This paper discusses the types of divergences found between pairs of languages and provides some basic rules to accommodate these divergences in order to reduce the errors of MT.

The list of divergences is very elaborate, not exhaustive which could be promising avenue for further research. Rules which are discussed above needs to be further explored. We also plan to build a preliminary MT system for English-Sindhi based on AnglaBharti[3] architecture (ANGLABHARTI is a translation methodology developed by R.M.K. Sinha at IIT, Kanpur for translation from English to Indian languages. At present, the AnglaBharti architect systems are performing translations for 5 Indian languages and those are Bangali, Hindi, Malayalam, Punjabi, and Urdu. The approach is rule-base).

6. Acknowledgements

I would like to thank Dr. Girish Nath Jha for encouraging me to do multidisciplinary research and CDAC (Centre for Development of Advanced Computing) Noida for assisting me to develop English-Sindhi MT.

7. References

Asamidinova, A. (2007). Knowledge Base For Russian-English Machine Translation Divergences. Ph.D. Thesis. Jawaharlal Nehru University.

Bharati, A., Chaitanya, V. & Sangal, R. (1995). *Natural Language Processing: A Paninian Perspective*. Prentice Hall, India.

[3]http://www.tdil-dc.in/components/com_mtsystem/CommonUI/homeMT.php

Dave, S., Parikh, J. & Bhattacharya, P. (2001). Interlingua Based English-Hindi Machine Translation and Language Divergence. In *Journal of Machine Translation*, 16 (4), pp. 251-304.

Dorr, B. J. (1990). Solving Thematic Divergences in Machine Translation. In *Proceedings of the 28th Annual Conference of the Association for Computational Linguistics*. University of Pittsburgh, Pittsburgh, PA, pp. 127-134.

Dorr, B. J. (1993). *Machine Translation: A View from the Lexicon*. MIT Press, Cambridge, MA.

Dorr, B. J. (1994). Machine Translation Divergences: A Formal Description and Proposed Solution. *Journal of Computational Linguistics* 20(4), pp. 597-633.

Gupta, D. (2005). Contributions to English to Hindi Machine Translation using Example-Based Approach. Ph.D. Thesis. IIT Delhi.

Gupta, D. & Chatterjee, N. (2003). Identification of divergence for English to Hindi EBMT. In *Proceedings of MT Summit IX*. New Orleans, USA, pp.141-148.

Gupta, D. (2009). Will Sentences Have Divergence Upon Translation? A Corpus Evidence Based Solution for Example Based Approach. *Language in India* 9, pp. 316-363.

Haegeman, L. (1991). *Introduction to Government and Binding Theory*. Oxford: Basil Blackwell.

Johannessen, J. B., Nordgård, T. & Nygaard, L. (2008). Evaluation of Linguistics-Based Translation**.** In *Proceedings of the sixth conference on International Language Resource and Evaluation (LREC'08)*, pp. 396-402.

Shukla, P., Shukla, D. & Kulkarni, A. P. (2010). Vibhakti Divergence between Snaskrit and Hindi. In G.N.Jha (ed.), *Sanskrit Computational Linguistics*, LNCS 6465, pp. 198-208.

Sinha, R. M. K. & Thakur, A. (2005). Translation Divergence in English-Hindi MT. *In Proceedings of EAMT*. pp. 245-254.