

WILDRE2 - 2nd Workshop on Indian Language Data: Resources and Evaluation

Workshop Programme

27th May 2014

14.00 – 15.15 hrs: Inaugural session

14.00 – 14.10 hrs – Welcome by Workshop Chairs

14.10 – 14.30 hrs – Inaugural Address by Mrs. Swarn Lata, Head, TDIL, Dept of IT, Govt of India

14.30 – 15.15 hrs – Keynote Lecture by Prof. Dr. Dafydd Gibbon, Universität Bielefeld, Germany

15.15 – 16.00 hrs – Paper Session I

Chairperson: **Zygmunt Vetulani**

- Sobha Lalitha Devi, Vijay Sundar Ram and Pattabhi RK Rao, *Anaphora Resolution System for Indian Languages*
- Sobha Lalitha Devi, Sindhuja Gopalan and Lakshmi S, *Automatic Identification of Discourse Relations in Indian Languages*
- Srishti Singh and Esha Banerjee, *Annotating Bhojpuri Corpus using BIS Scheme*

16.00 – 16.30 hrs – Coffee break + Poster

Session Chairperson: **Kalika Bali**

- Niladri Sekhar Dash, *Developing Some Interactive Tools for Web-Based Access of the Digital Bengali Prose Text Corpus*
- Krishna Maya Manger, *Divergences in Machine Translation with reference to the Hindi and Nepali language pair*
- András Kornai and Pushpak Bhattacharyya, *Indian Subcontinent Language Vitalization*
- Niladri Sekhar Dash, *Generation of a Digital Dialect Corpus (DDC): Some Empirical Observations and Theoretical Postulations*
- S Rajendran and Arulmozi Selvaraj, *Augmenting Dravidian WordNet with Context*
- Menaka Sankarlingam, Malarkodi C S and Sobha Lalitha Devi, *A Deep Study on Causal Relations and its Automatic Identification in Tamil*
- Panchanan Mohanty, Ramesh C. Malik & Bhimasena Bhol, *Issues in the Creation of Synsets in Odia: A Report¹*
- Uwe Quasthoff, Ritwik Mitra, Sunny Mitra, Thomas Eckart, Dirk Goldhahn, Pawan Goyal, Animesh Mukherjee, *Large Web Corpora of High Quality for Indian Languages*

- Massimo Moneglia, Susan W. Brown, Aniruddha Kar, Anand Kumar, Atul Kumar Ojha, Heliana Mello, Niharika, Girish Nath Jha, Bhaskar Ray, Annu Sharma, *Mapping Indian Languages onto the IMAGACT Visual Ontology of Action*
- Pinkey Nainwani, *Handling Conflational Divergence in a pair of languages: the case of English and Sindhi*
- Jayendra Rakesh Yeka, Vishnu S G and Dipti Misra Sharma, *Semi automated annotated treebank construction for Hindi and Urdu*
- Saikrishna Srirampur, Deepak Kumar Malladi and Radhika Mamidi, *Improvised and Adaptable Statistical Morph Analyzer (SMA++)*
- K Kabi Khanganba and Girish Nath Jha, *Challenges in Indian Language Transliteration: a case of Devanagari, Bangla and Manipuri*
- Girish Nath Jha, Lars Hellan, Dorothee Beermann, Srishti Singh, Pitambar Behera and Esha Banerjee, *Indian languages on the TypeCraft platform – the case of Hindi and Odia*

16.30 – 17.30 hrs – Paper Session II

Chairperson: **Dr. S. S. Aggarwal**

- Nripendra Pathak and Girish Nath Jha, *Issues in Mapping of Sanskrit-Hindi Verb forms*
- Atul Kr. Ojha, Akanksha Bansal, Sumedh Hadke and Girish Nath Jha, *Evaluation of Hindi-English MT Systems*
- Sreelekha S, Pushpak Bhattacharyya and Malathi D, *Lexical Resources for Hindi Marathi MT*
- Esha Banerjee, Akanksha Bansal and Girish Nath Jha, *Issues in chunking parallel corpora: mapping Hindi-English verb group in ILCI*

17:30 – 18.10 hrs – Panel discussion India and Europe - making a common cause in

LTRs Coordinator: **Hans Uszkoreit**

Panelists - Joseph Mariani, Swaran Lata, Zygmunt Vetulani, Dafydd Gibbon, Panchanan Mohanty

18:10- 18:25 hrs – Valedictory Address by Prof. Nicoletta Calzolari, CNR-ILC, Italy

18:25-18:30 hrs – Vote of Thanks

Editors

Girish Nath Jha
Kalika Bali
Sobha L

Jawaharlal Nehru University, New Delhi
Microsoft Research Lab India, Bangalore
AU-KBC Research Centre, Anna University,
Chennai

Esha Banerjee

Jawaharlal Nehru University, New Delhi

Workshop Organizers/Organizing Committee

Girish Nath Jha
Kalika Bali
Sobha L

Jawaharlal Nehru University, New Delhi
Microsoft Research Lab India, Bangalore
AU-KBC Research Centre, Anna University,
Chennai

Workshop Programme Committee

A. Kumaran
Amba Kulkarni
Chris Cieri, LDC
Dafydd Gibbon
Dipti Mishra Sharma
Girish Nath Jha
Hans Uszkoreit
Indranil Datta
Jopseph Mariani
Jyoti Pawar
Kalika Bali
Karunesh Arora
Malhar Kulkarni
Monojit Choudhary
Nicoletta Calzolari
Niladri Shekhar Dash
Panchanan Mohanty
Pushpak Bhattacharya
S. S. Aggarwal
Sobha L
Umamaheshwar Rao
Zygmunt Vetulani

Microsoft Research, India
University of Hyderabad, India
University of Pennsylvania
Universität Bielefeld, Germany
IIIT, Hyderabad, India
Jawaharlal Nehru University, New Delhi, India
Saarland University, Germany
EFLU, Hyderabad, India
LIMSI-CNRS, France
Goa University, India
MSRI, Bangalore, India
CDAC Noida, India
IIT Bombay, India
Microsoft Research, India
CNR-ILC, Pisa, Italy
ISI Kolkata, India
University of Hyderabad, India
IIT Bombay, India
KIIT, Gurgaon
AU-KBC RC, Anna University, Chennai, India
University of Hyderabad, India
Adam Mickiewicz University, Poznan, Poland

Table of contents

	Introduction	x
1	Anaphora Resolution System for Indian Languages <i>Sobha Lalitha Devi, Vijay Sundar Ram and Pattabhi RK Rao</i>	1
2	Automatic Identification of Discourse Relations in Indian Languages <i>Sobha Lalitha Devi, Sindhuja Gopalan and Lakshmi S</i>	9
3	Annotating Bhojpuri Corpus using BIS Scheme <i>Srishti Singh and Esha Banerjee</i>	16
4	Indian Subcontinent Language Vitalization <i>Andras Kornai and Pushpak Bhattacharyya</i>	24
5	Augmenting Dravidian WordNet with Context <i>Rajendran S and Arulmozi Selvaraj</i>	28
6	A Study on Causal Relations and its Automatic Identification in Tamil <i>Menaka Sankarlingam, Malarkodi C S and Sobha Lalitha Devi</i>	34

7	Issues in the Creation of Synsets in Odia: A Report¹	41
	<i>Panchanan Mohanty, Ramesh C Malik and Bhimasena Bhol</i>	
8	Large Web Corpora of High Quality for Indian Languages	47
	<i>Uwe Quasthoff, Ritwik Mitra, Sunny Mitra, Thomas Eckart, Dirk Goldhahn, Pawan Goyal, Animesh Mukherjee</i>	
9	Mapping Indian Languages onto the IMAGACT Visual Ontology of Action	51
	<i>Massimo Moneglia, Susan Brown, Aniruddha Kar, Anand Kumar, Atul Kumar Ojha, Heliana Mello, Niharika, Girish Nath Jha, Bhaskar Ray and Annu Sharma.</i>	
10	Handling Conflational Divergence in a pair of languages: the case of English and Sindhi	56
	<i>Pinkey Nainwani</i>	
11	Semi-automated annotated treebank construction for Hindi and Urdu	64
	<i>Vishnu S G, Jayendra Rakesh Yeka and Dipti Misra Sharma</i>	

12	Improvised and Adaptable Statistical Morph Analyzer (SMA++)	73
	<i>Saikrishna Srirampur, Deepak Kumar Malladi and Radhika Mamidi</i>	
13	Challenges in Indian Language Transliteration: a case of Devanagari, Bangla and Manipuri	77
	<i>K Kabi Khanganba and Girish Nath Jha</i>	
14	Indian languages on the TypeCraft platform – the case of Hindi and Odia	84
	<i>Girish Nath Jha, Lars Hellan, Dorothee Beermann, Srishti Singh, Pitambar Behera and Esha Banerjee</i>	
15	Issues in Mapping of Sanskrit-Hindi Verb forms	90
	<i>Nripendra Pathak and Girish Nath Jha</i>	
16	Evaluation of Hindi-English MT Systems	94
	<i>Atul Kr. Ojha, Akanksha Bansal, Sumedh Hadke and Girish Nath Jha</i>	
17	Lexical Resources for Hindi Marathi MT	102
	<i>Sreelekha S, Pushpak Bhattacharyya and Malathi D</i>	

18 Issues in chunking parallel corpora: mapping 111

Hindi-English verb group in ILCI

Esha Banerjee, Akanksha Bansal and Girish Nath Jha

Author Index

Arulmozi, S.	29
Banerjee, Esha.....	16, 85, 112
Bansal, Akanksha.....	95, 112
Beermann, Dorothee.....	85
Behera, Pitambar.....	85
Bhattacharyya, Pushpak.....	25, 103
Bhol, Bhimasena.....	42
Brown, Susan.....	52
C.S., Malarkodi.....	35
D, Malathi.....	103
Eckart, Thomas.....	48
Goldhahn, Dirk.....	48
Gopalan, Sindhuja.....	9
Goyal, Pawan.....	48
Hadke, Sumedh.....	95
Hellan, Lars.....	85
Jha, Girish Nath.....	52, 78, 85, 91, 95, 112
Kar, Aniruddha.....	52
Khangamba, K Kabi.....	85
Kornai, András.....	25
Kumar, Anand.....	52
Lalitha Devi, Sobha.....	1, 9, 35
Malik, Ramesh C.	42
Malladi, Deepak Kumar.....	74
Mamidi, Radhika.....	74
Mello, Heliana.....	52
Mishra, Niharika.....	52
Mitra, Ritwik.....	48
Mitra, Sunny Mitra.....	48
Mohanty, Panchanan.....	42
Moneglia, Massimo.....	52
Mukherjee, Animesh.....	48
Nainwani, Pinkey.....	57
Ojha, Atul Kumar.....	52, 95
Pathak, Nripendra.....	91
Quasthoff, Uwe.....	48
R., Vijay Sundar Ram.....	1
Rajendran, S.	29
Rao, Pattabhi RK.....	1
Ray, Bhaskar.....	52
S G, Vishnu.....	65

S, Sreelekha.....	103
S., Lakshmi.....	9
S., Menaka.....	35
Sharma, Annu.....	52
Sharma, Dipti Misra.....	65
Singh, Srishti.....	16, 85
Srirampur, Saikrishna.....	74
Yeka, Jayendra Rakesh.....	65

Introduction

WILDRE – the 2nd workshop on Indian Language Data: Resources and Evaluation is being organized in Reykjavik, Iceland on 27th May, 2014 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 2nd WILDRE will be

- 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 29 full papers received for review, we selected 7 papers for oral, 13 for poster and 1 for demo presentation.

Anaphora Resolution System for Indian Languages

Sobha Lalitha Devi, Vijay Sundar Ram R., and Pattabhi RK Rao

AU-KBC Research Centre, MIT Campus of Anna University,
Chennai, India
sobha@au-kbc.org

Abstract

We describe our work on anaphora resolution engine for all Indian languages. Indian languages are morphologically rich. Morphological richness of the Indian languages is tapped to come up with generic anaphora resolution engine. The system architecture is designed in a simple plug-n-play model. We have used a machine learning technique, Conditional Random Fields (CRFs) to build the core engine. We have tested the engine with Tamil, a Dravidian language and Hindi, an Indo-Aryan language. The results are encouraging.

Keywords: Anaphora resolution system, Multilingual, Machine Learning, CRFs, Tamil, Hindi, Indian languages

1. Introduction

Anaphors are the small words which occur in the text referring to bigger noun phrases called antecedents. These anaphors add elegance to the text and help to overcome monotonicity in reading. In real time natural language applications such as information extraction, question answering, information retrieval, machine translation etc requires anaphora resolution to process the information completely. Any NLP application that requires assimilating the information in the text needs anaphors to be resolved. Though anaphor resolution is studied under syntactic, semantic and discourse level for family of languages, most of the anaphora resolution systems presented by researchers are language dependent. In this paper we present an anaphora resolution system for Indian languages, where we have tried to build a multilingual anaphora resolution system using the common features between Indian languages such as morphological richness. Research in the field of anaphora resolution is active for the past few decades. In the following section we describe various works published in the field of Anaphora resolution.

2. Literature Survey

2.1 Earlier Approaches to Anaphora resolution

The early approaches in the task of anaphora resolution were referred as Knowledge intensive approach. Approaches by Hobbs (1978), Cater, Carbonell and Brown (1988) and Rich and LuperFoy (1988) were knowledge intensive approach in which they used semantic information, world knowledge and case-frame semantics. Hobbs used a syntax based approach for anaphora resolution. Centering theory, a discourse theory based approach by Grosz, Joshi and Kuhn (1979), Joshi and Weinstein (1981), salience factor/ indicators based approach by Lappin and Leass (1994) and Mitkov's (1997) MOA (Mitkov's Original Approach) and MARS (1998) (Mitkov's Anaphora Resolution System).

2.2 Machine Learning Approaches to Anaphora resolution

Machine Learning technique based approaches started with Dagan and Itai (1990). The commonly used machine learning techniques in anaphora resolution task are presented in table 1. With the start of machine learning technique, most of the researcher attempted pronominal resolution and noun phrase resolution simultaneously.

Machine learning Technique	Presented Works
Decision Tree	Aone and Bennett (1995), McCarty and Lehnert, Soon et. al. (2001), Ng and Cardie (2002)
TiMBL (memory based Learning)	Hendrickx, Hoste and Daelemans (2008), Marta Recasen (2009)
SVM (Support Vector Machine)	Yang, Su, and Tan
Linear CRFs	Li Fei et al., (2008), Sobha et al (2011)
Tree CRFs	Sobha et al (2011), Vijay Sundar Ram et al (2012, 2013)
ILP	Denis and Baldridge (2007)
Factorial Hidden Markov Models	Dingcheng Li (2011)
Discriminative Hierarchical Model	Michael Wick (2011), Sameer Singh (2011)

Table 1: Learning Techniques and published works

Hardmeier et al (2012) had published a work on cross-lingual pronoun prediction task, where they predicted the correct French translations of third-person subject pronouns in English discourse, as a prerequisite for machine translation. They used an approach based on neural networks based approach to model anaphoric links as latent variables and showed that its performance was competitive with that of a system with separate anaphora resolution while not requiring any coreference-annotated training data. Iida et al (2013) had presented a cross-lingual solution for zero anaphors between Italian and Japanese.

2.3 Anaphora Resolution in Indian Language

The researcher in the area of anaphora resolution in Indian languages started in the end of the last millennium. The first anaphora resolution system in Indian languages is Vasisth. Vasisth was presented by Sobha et al (2000, 2002) as a platform for Indian language anaphora resolution and demonstrated with Hindi and Malayalam languages. Hindi belongs to Indo-Aryan language family and Malayalam belongs to Dravidian language family. It is a rule based system, where the morphological richness of both languages was exploited to identify the subject, object, indirect object etc. Though Indian constitution has recognized 22 official languages, anaphora resolution systems has been developed in few languages, namely, Hindi, Tamil, Bengali, Malayalam and Urdu. Prasad et al (2000) used centering theory, a discourse theory, for Hindi anaphora resolution. They applied discourse salience ranking to two pronoun resolution algorithms, BFP and S-List algorithm. Uppalpu et al (2009) extended the previous work by using dependency tree relations. Dutta et al (2008) presented Hindi anaphora resolution using Hobb's algorithm as baseline. Probabilistic neural network approach to classify demonstrative pronouns for Indirect anaphor in Hindi was done by Dutta et al (2010). Dakwale et al (2011) demonstrated a hybrid approach for Hindi anaphora resolution in ICON 2011 tool contest and applied the similar approach to Tamil and Bengali. Lakshmi et al (2013) had presented on the issues and challenges in Hindi anaphora resolution and constraints in Hindi resources. Dakwale and Sharma (2013) came up with a hybrid approach to resolve Entity-pronoun reference in Hindi, which combined a rule based system using dependency structures and relations and it was improved using the animacy information.

Narayana Murthy et al (2007) had presented a work on Tamil anaphora resolution, where the authors had described two different approaches, one using salience features and the other using multiple linear regression. Sobha (2007) came up with a salience weight based approach for Tamil anaphora resolution, where the morphological markings were used without using any sophisticated notion of the modern formal linguistic theories. Akilandeswari et al (2012, 2013) used conditional random fields, a machine learning techniques to resolve 3rd person pronoun in Tamil. The features for the machine learning were collected from morphological information and word positional information.

Akilandeswari et al. (2012) had demonstrated a work on resolution of third person neuter pronoun 'atu' in Tamil using CRFs technique. Vijay et al (2013) presented a Tamil anaphora resolution using TreeCRFs, where information from dependency tree was used along with the features obtained from the morphological information. Balaji et al (2011, 2012) had demonstrated a two-step approach to resolve Tamil anaphora using Universal network language. Sanskrit anaphora resolution system based on salience measure was presented by Sobha et al (2008).

Anaphora Resolution in Indian language, a tool contest conducted as a part of ICON 2011 catalyzed the research in Indian language anaphora resolution. The contest became boon to Bengali language as it had four participants. Following are the four different approaches. Sanjay et al (2011) presented to two stage approach using CRFs and random tree classifier. Anaphora resolution using Global discourse knowledge was presented by Senapathi et al (2011). Gosh et al (2011) presented a ruled based approach. Senapathi et al (2013) demonstrated an anaphora resolution system using GuiTAR tool for Bengali. Sikdar et al (2013) presented a Bengali anaphora resolution system by customizing features required for BART anaphora resolution system. Ali et al (2008) had discussed on various factors and optimal order that play important role in resolution of personal anphors in Urdu.

2.4 Multilingual Anaphora Resolution

There are few published works on approaches to come up with multilingual anaphora engines. Following are the published work.

Aone and Mckee (1993) demonstrated a language independent anaphora resolution system with English, Spanish and Japanese languages. They used global discourse world, which holded syntactic, semantic, rhetorical and other information about the input text derived by other part of the system namely knowledge. Knowledge source (ks) had three slots namely ks-function, ks-data and ks-language. Here ks-language slot specifies the language.

Mitkov (1998) with the search for inexpensive, rapid and reliable procedures for anaphora resolution presented a multilingual platform, where he used a set of indicators to identify the antecedents. The indicators were related to salience, structural match, referential distance or preference of terms. Indicators were given scores and the candidate with maximum score was chosen. He evaluated the system with English, Polish and Arabic.

Vasisth, an anaphora resolution system for Indian languages was presented by Sobha et al. Vasisth is a multilingual system, where the morphological richness of Indian languages was exploited. Hindi, an Indo-Aryan language and Malayalam, a Dravidian language were handled in the system to show that, it can handle most of the Indian languages without major modification. It is a rule based system where limited parsing information is used. It used the grammatical rules and morphological marking to identify subject, object, indirect object etc.

The research in the direction of multilingual anaphora resolution was energized by the following tool contests conducted in SemEval-2010 and ICON-2011.

SemEval-2010: Coreference Resolution in Multiple Languages aimed to explore the portability of systems across languages, need for different levels of linguistic information. The contest had six languages namely Catalan, Dutch, English, German, Italian and Spanish (Recasen 2010). There were six submissions. Only two of the participants had submitted results for all the six language. Following is a short description of each participant. RelaxCor used constraint-based graph partitioning approach, where the coreference resolution was solved by relaxation of labeling. Here the rules were generated using Decision tree C4.5 algorithm. They participated in Catalan, English and Spanish. SUCRE used a feature engineering based on relational database and regular feature engineering. Four classifier were integrated in SURCE, namely, Decision-tree, Naïve-Bayes, SVM and Maximum Entropy. They participated in all the six languages. UBIC presented a language independent system and participated in all six languages, using memory based learning and feature model. Corry, participated in English, used two classifiers for coreference and anaphoricity. This system used ILP (integer linear programming) methods. TAN-1 used a binary classifier based on maximum entropy to extract the pair of mentions. They have used lexical features, distance features, syntactic features, count and type features. They have participated in Catalan, English, German and Spanish. BART participated in English, German and Italian. They used maximum entropy based classifier for German and decision tree for Italian. BART tool kit has a language plugin to handle language specific information and it supports pairwise modeling, ranking and semantic tree models for coreference. By customizing engine for different languages, BART is presented as multilingual coreference resolution engine. In this contest, although most of the participants had tried to come up with multilingual systems, their systems have outperformed in one or two languages.

The research in multilingual anaphora engines for Indian languages was motivated by Anaphora Resolution in Indian Languages, NLP Tool Contest conduct as a part of ICON 201 (Sobha et al. 2011). Hindi, Bengali and Tamil were the languages considered for the tool contest. There were four submissions, two participants, viz, IITKgp and IITH attempted for all three languages and other two participants, viz, ISI Kolkata and JU attempted only Bengali. IITKgp tried a two stage approach of identifying the markables using CRFs and identifying the referring entities using random trees approach. They used morphological information based features for Bengali. ISI Kolkata presented a different approach, where they identified the characteristics of nouns whenever it is encountered and accordingly a set of pronouns were attached to it. IITH presented a hybrid approach of using machine learning technique followed by rule-based post-processing. JU group attempted a rule-based engine for Bengali. Even in

this contest the participants came up with system focused to one language and they tried the same systems for the other languages.

The results of the above described two tool contests present the pertinent need for a multilingual anaphora resolution system built using language independent features and supported with required language specific information through a language plugin.

3. Multilingual Anaphora Resolution System – Our Approach

In this work we have developed a generic anaphora resolution system for Indian languages. We have tested this using Hindi and Tamil. The core anaphora engine uses CRFs, a machine learning technique. Since we have used machine learning technique, we have two phases training phase and testing phase. In the training phase the system is provided with annotated data and the features for learning. After the system learns, a model file is generated as output. In the testing phase any unseen text is given for the automatic anaphora resolution. Figure 1 and 2 shows the system architecture diagrams during the training and testing phase respectively. There are mainly five components in this system viz., i) Text Processing module ii) Candidate mentions extraction iii) PNG Agreement Handler iv) Feature Extraction module and v) Anaphora Engine.

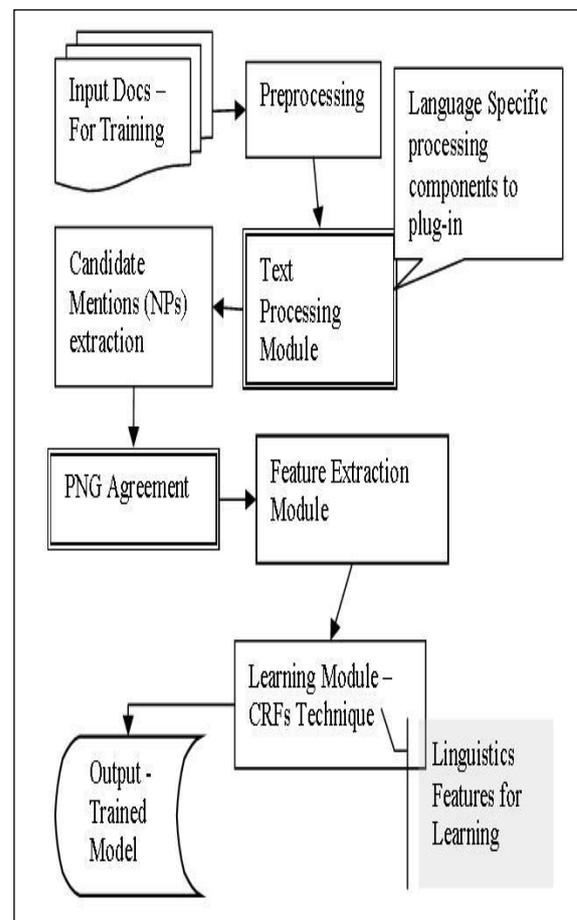


FIGURE 1: System Architecture – Training PhaseText

Processing module and the PNG agreement module are two language dependent modules which can be plugged-in for the respective languages while processing. The Feature extraction Module is a language independent module. This module extracts defined or required features from the processed data to facilitate learning and automatic resolution by the Anaphora Engine. The system is designed to keep the architecture simple and a plug-n-play model. Language specific linguistic analysis sub-components can be added or removed according to the necessity inside the Text Processing module. And this new addition or removal does not affect the rest of the modules.

For example if a specific language requires word segmentation component that can be plugged into the “Text Processing” module and should adhere to the data input-output (I/O) formats as described in the coming sections. Further in this section we describe about the data I/O format specification, features used for learning and PNG agreement handler.

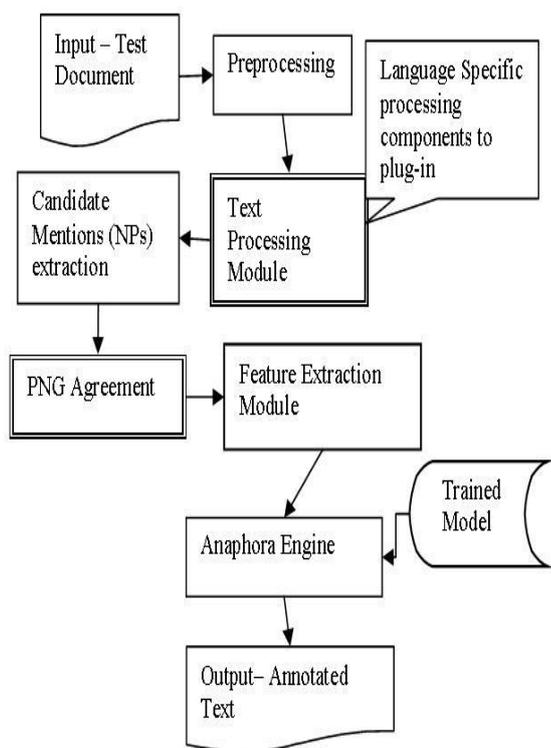


FIGURE 2: System Architecture – Testing Phase

3.1.1 Data Format

In this work we find that it is very important to have specifications of data input and output format well defined. Here we have developed a generic system which takes data input in specified format and gives output in specified format. And this works for any of the morphologically rich language. We have defined the input and output formats for all the modules. We have used the column format of data representation, where each column represents a specific linguistic analysis output. Each module adds its analysis output as a new column to the

existing columns. Column format representation is easy for computational purposes and allows for storing of different kinds of linguistic analyses. Figure 3, shows the data format which contains different linguistic analysis outputs in each column.

The column description is as follows: 1st column - Word, 2nd column – POS Tag, 3rd Column – Chunk Tag, 4th Column – Morph Analysis, 5th Column – Chunk Head information. The columns are separated by tab-space. At the outset though this seems to be rigid or fixed, it is flexible in sense that the individual modules can have other columns to be added as their interim output for processing purposes without disturbing original input and finally give the desired output in the fixed format. Other advantage of this representation is that in future if a new linguistic analysis module is to be used; it will be easy to plug-in without affecting the operation of other modules.

3.1.2 Features for Learning

The features for the task of anaphora resolution are identified from shallow parsed input sentences. The input sentences are preprocessed with morphanalyser, part-of-speech (POS) tagger, and chunker and clause boundary identifier, which is optional. The identified features are based on following preferences or characteristics

a) Positional Characteristics:

Candidate antecedent’s place of occurrence, whether within the same sentence or recency of sentence occurrence.

b) Grammatical Role:

Candidate antecedents grammatical roles such as “subject markings”, “object marking”. This is inferred from the case markers suffixed to the nouns, such as noun with nominative case can be a subject, noun with accusative case can be object.

c) Linguistic Characteristics:

Candidate antecedents POS tag, Case markers/suffixes

d) Repetition:

Occurrence of the Candidate antecedents in the span of 5 sentence window prior to the occurrence of the anaphor

e) Type of the NP:

Whether the candidate antecedent (probable antecedent) is Possessive or Existential or a Prepositional Compliment.

f) Combination of the above said features.

We have used Conditional Random Fields (CRFs), a machine learning technique. In our approach we have modeled this as a binary classification task. The machine

has to classify whether the given candidate antecedent is the real antecedent or not based on the features presented with the candidate antecedents. The features for learning are extracted from the shallow parsed data. The feature extraction module extracts these features for all possible candidate antecedents from shallow parsed data. For an anaphors, the “Candidate Antecedent Extraction” component collects all the NPs/mentions occurring prior to the occurrence of anaphor in the window span of 5 sentences above the sentence in which anaphor occurs. The “PNG Agreement Handler” filters these candidate antecedents based on the PNG features.

3.1.3 PNG Agreement Handler

In most of the Indian languages there are strong Person, Number and Gender (PNG) agreement constraints. Person agreement is for the purpose to distinguish between 1st person, 2nd person and 3rd person. Also for the purpose of distinguishing between subject positions, object position, and genitive position. Number agreement is to distinguish between singular and plural references with respect to the anaphors number. Gender agreement is to distinguish male, female, and non-personal genders of the reference with respect to the anaphors gender. Each language has its own set of PNG agreement constraints. This PNG agreement handler takes care of the agreement rules for each specific language. This module takes a configuration file as input, where the constraints for the specific language are define. This configuration file can be plugged into the system for the language data under consideration. The following examples explain the PNG features for Tamil and Hindi. Example 1, 2 and 3 are Tamil examples. Example 4, 5 and 6 are Hindi examples.

(1) Ta: kamala_i kadaikkuc cenraal. avaL_i niraya
 Kamala shop+dat go+past+3sf she lot
 poorutkaL vaangkinal.
 things buy+past+3sf
 (Kamala went to shop. She bought lot of things.)

In the above example ‘avaL’ is a 3rd person pronoun and it is referring to the noun phrase ‘kamala’, which is the feminine noun phrase.

(2) Ta: raamu paLLikkuc cenraan. avan
 Ramu school+dat go+past+3sm he
 naRRaaka padikkiraan.
 good Study+present+3sm
 (Ramu went to school. He studies well.)

In example 2, the pronoun ‘avan’ (3rd masculine pronoun), it refers to the masculine noun phrase ‘Ramu’.

(3) Ta: periyakovil thancaiyil ullawu. athu mika
 big_temple Thanjai+loc is It very
 pukalpeRRathu.
 famous
 (Periyakovil is there in Thanjai. It is very famous.)

In example 3, ‘athu’ (3rd neuter singular pronoun). It is refers to a neuter noun phrase ‘periyakovil’ in the previous sentence.

Similarly in Hindi, the following examples describe the usage of pronouns.

(4) Hi: kamala_i dukaan gayii thi. usne_i
 Kamala shop go+fem+past copula. she
 bahut sii chiije kharidii.
 lot of items buy+past+fem.
 (Kamala went to the shop. She bought lot of items)

In this example “usne” is the 3rd person singular pronoun referring to Kamala. This pronoun does not have the gender information. The gender marker in the first sentence is available in the verb “thi”, gives the gender of “kamala”. In Hindi when the subject noun occurs with the ergative suffix, the verb agrees with the object as in second sentence. This 3rd person pronoun “usne” can be used for both masculine and feminine forms. This is illustrated in the next example.

(5) Hi: raama_i paaTashaala gayaa tha. veh_i
 Ram school go+past copula. He
 aChChaa padataa hai.
 good study+present+mas present+copula.
 (Ram went to school. He studies well.)

In the example (5), the 3rd person pronoun “veh” is referring to “Ram”, which is masculine.

(6) Hi: brihadesvara_i mandir tanjavur me hai.
 Brihadesvar temple Thanjavur at is.
 yeh_i bahut badaa mandir hai.
 It very big temple is.
 (Brihadeswar temple is at Thanjavur. It is a very big temple.)

In example (6), “yeh” is the 3rd person neuter gender pronoun referring to the Brihadeswar temple.

In Tamil, 1st person pronouns have singular-plural distinction 'naan' “I” and 'naangkal' “I honorophic” are singular and plural forms. Similarly 'nii' and 'niingkal' are singular and plural forms of 2nd person pronoun respectively. Both 1st and 2nd person pronouns do not have masculine-feminine distinction. There is singular-plural and masculine-feminine distinction in all 3rd person pronouns. 3rd person pronouns are presented in table 2.

	Singular	Plural
Masculine	avan	avarkal
Feminine	aval	avarkal
Honorific	avar	avarkal
Neuter	athu	avai

Table 2: 3rd Person Pronouns in Tamil

In Hindi pronominal anaphors are ambiguous; especially

the third person pronouns have no masculine-feminine distinction. The 3rd person pronouns such as “veh” (he/she), “ve” (he/she), “us” (he/she/it), “unhon-ne” (he/she honorific), “inhon-ne” (he/she honorific), “yeh” (this), “unko” (them), “unse” do not have any gender information associated with them and is used for both masculine and feminine genders. The gender is known from the verb form. The pronoun “veh” is also used for demonstrative pronoun to mean “that”. This makes the automatic resolution more challenging. Hence PNG agreement handler plays an important role.

4. Experiments, Results and Discussion

We have performed experiments to test the anaphora resolution system using two Indian languages Hindi and Tamil. We have chosen two languages which are representative of these two families. Hindi belongs to the Indo-Aryan family of languages and Tamil belongs to the Dravidian family. The Corpus for these experiments is from Tourism domain. The corpus was collected from various tourism web pages. The corpus was divided into training and testing partitions. The training partition is used for developing the system and building the CRFs trained model. The distribution of pronouns in the training and testing partitions for each language is given in the table 3. The corpus was annotated and stored in column format as explained in the previous section.

The corpus is processed with POS, Chunking and morphological analysis for both the languages. These linguistic processors are obtained from IL-ILMT consortium. Using the features explained in previous section the system is trained on the training data and a trained CRFs model is obtained as output. CRF++ open source tool is used for the CRFs algorithm implementation. The features for each candidate antecedent is extracted using the Feature extraction module and using the PNG agreement handler, the candidate antecedents not having matching PNG with the respective anaphors are filtered out.

Language	Training Partition No. of Pronouns	Testing Partition No. of Pronouns	Total
Hindi	835	507	1342
Tamil	925	609	1534

Table 3: Distribution of Pronouns in the Corpus

After this filtering, the pair of pronoun and corresponding candidate antecedents is given as input to the anaphora engine. Anaphora engine uses the output trained CRFs model to automatically resolve the correct antecedent. We have done the evaluation using three popularly known metrics in the community viz., MUC, B-Cubed, CEAF_E. The average of the scores of all these metrics is considered. Each metrics tries to overcome the short comings of the other metric. Hence an average of all these scores has to be taken into consideration.

Metric	Prec	Recall	F1
MUC	31.3	42.4	36.01
B-CUBED	76.5	68.3	72.16
CEAF _E	58.2	49.5	53.39
Average	55.3	53.4	53.85

Table 4: Evaluation Results in Hindi

Metric	Prec	Recall	F1
MUC	33.2	48.6	39.45
B-CUBED	75.7	65.2	70.05
CEAF _E	58.8	47.2	52.36
Average	55.9	53.6	53.95

Table 5: Evaluation Results in Tamil

The evaluation results are shown in the table 4 for both the languages.

We observe that the MUC score are lower than other scores. This is due to the fact that MUC is more focused towards the chain or link formation. Here in our system we are not intending to form chains of all entities but instead on the resolution of anaphors. The other metrics are focused on the entities resolved. After analyzing the output data of the system, in Tamil we observe that third person honorific pronouns are not being resolved properly due to them being more ambiguous. The ambiguity is due to their suitability for both genders.

(7)Ta: aaciriyar kamala villavil kalanthukkoNtaar.
Teacher Kamala function+loc join+past+3SH

avar athil uraiyaRRinaal.
He/she this gave lecture

(Teacher Kamala joined the junction. She gave the lecture.)

In example 7, ‘avar’ (3rd honorific singular pronoun) in the second sentence, refers to a feminine noun phrase ‘aaciriyar kamala’ in the previous sentence. The honorific pronoun can also refer to a masculine pronoun. This possibility of referring to both the gender introduces more errors.

In Hindi, we observe that there is necessity of having verb analysis and animicity identifier sub-modules in the text processing component. If we have this information to the resolution engine, it can reduce ambiguity and improve the anaphora resolution.

5. Conclusion

In this paper we have presented anaphora resolution system for all Indian languages. Here we have described the system architecture and its implementation. We have chosen two Indian languages representing two major families of languages of India. They are Hindi and Tamil belonging to Indo-Aryan and Dravidian families respectively. The system is specifically designed to be scalable and allows plug-n-play architecture. The core anaphora resolution engine uses CRFs a machine learning technique. This uses feature based learning and we have

provided syntactic and positional based features. We have obtained encouraging evaluation results. The evaluation metrics used are standard measures MUC, B-CUBED and CEAF_E. We have obtained an average of 53.85% F-measure for Hindi and 53.95% of F-measure for Tamil.

These results are comparable with other reported systems. We plan to test the system design by implementing it with other Indian languages. In this work we have attempted to build an anaphora resolution framework which can be used for all Indian languages.

isI	DEM	B-NP	<fs af='yaha.pn.any.sg.3.o.0.0' name=isI emph=y>	<fs af='blca.n.m.sg.3.d.0.0' poslcat="NM" head=blca>
blca	NST	I-NP	<fs af='blca.n.m.sg.3.d.0.0' name=blca >	o
xUsare	QO	B-NP	<fs af='xUsarA.adj.m.sg.,o.,' name=xUsare>	<fs af='javAna.n.m.sg.3.d.0.0' head=javAna>
javAna	NN	I-NP	<fs af='javAna.n.m.sg.3.d.0.0' name=javAna>	o
trena	NN	B-NP	<fs af='trena.n.f.sg.3.d.0.0' name=trena>	<fs af='trena.n.f.sg.3.d.0.0' head=trena>
cena	NN	I-NP	<fs af='cena.n.f.sg.3.d.0.0' name=cena>	<fs af='cena.n.f.sg.3.d.0.0' head=cena>
KIMca	VM	B-VGF	<fs af='KIMca.v.any.any.,0.0' name=KIMca>	<fs af='KIMca.v.any.any.,0.0' head=KIMca>
.	SYM	I-VGF	<fs af='.,punc,,,,,' name=.>	o

Figure 3: Data I/O Format – An example Hindi sentence (the script used in WX-notation)

6. References

- Akilandeswari A., Sobha Lalitha Devi. (2013). Conditional Random Fields Based Pronominal Resolution in Tamil. *International Journal on Computer Science and Engineering*, Vol. 5 Issue 6 pp 601 - 610
- Akilandeswari A., Sobha, Lalitha Devi. (2012). Resolution for Pronouns in Tamil Using CRF. *In Proceedings of Workshop on Machine Translation and Parsing in Indian Languages, COLING 2012*, Mumbai, India.
- Akilandeswari A, Bakiyavathi T and Sobha Lalitha Devi, (2012), "atu Difficult Pronominal in Tamil", *In Proceedings of Lrec 2012*, Istanbul
- Ali M.N., Khan M. A., Khan, M.A., (2008). An Optimal Order of Factors for the Computational Treatment of Personal Anaphoric Devices in Urdu Discourse. *In Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, Hyderabad, India, pp 81–90
- Aone C., and McKee D. (1993). A Language-Independent Anaphora Resolution System for Understanding Multilingual Texts. *In proceeding of ACL 1993*, pp 156-163.
- Aone C., and Bennett S. (1995). Evaluating automated and manual acquisition of anaphora resolution strategies. *In: 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 122-129.
- Balaji J., Geetha T. V., Parthasarathi R., Karky M.. (2011). Anaphora Resolution in Tamil using Universal Networking Language. *In: Proceedings of ICAI 2011*, pp. 1405-1415
- Balaji J., Geetha T.V., Ranjani Parthasarathi R., Karky M. (2012). Two-Stage Bootstrapping for Anaphora Resolution *In: Proceedings of COLING 2012*, pp 507–516
- Carbonell J. G., and Brown R. D. (1988). Anaphora resolution: A multi- strategy approach. *In: 12th International Conference on Computational Linguistics*, 1988, pp. 96-101
- Carter D. (1987). Interpreting anaphors in natural language texts. Chisester: *Ellis Horwood* ltd.
- Dagan I., and Itai. A. (1990). Automatic processing of large corpora for the resolution of anaphora references. *In: 13th conference on Computational linguistics*, Vol. 3, Helsinki, Finland, pp.330-332.
- Dakwale. P., Mujadia. V., Sharma. D.M. (2013). A Hybrid Approach for Anaphora Resolution in Hindi. *In: Proc of International Joint Conference on Natural Language Processing*, Nagoya, Japan, pages 977–981
- Dakwale. P., and Sharma. H. 2011. Anaphora resolution in Indian languages using hybrid approaches. *In proceedings of ICON Tool Contest, 9th International Conference on Natural Language Processing*, Chennai, India
- Dutta. K., Prakash N. and Kaushik. S. Probabilistic Neural Network Approach to the Classification of Demonstrative Pronouns for Indirect Anaphora in Hindi," *Expert Systems with Applications: An International Journal*, Volume 37, Issue 8, pp. 5607-5613, Elsevier, 2010.
- Dutta. K., Prakash. N. and Kaushik. S. (2008). Resolving Pronominal Anaphora in Hindi using Hobbs" algorithm," *Web Journal of Formal Computation and Cognitive Linguistics*, Issue 10, 2008.
- Fei, Li., Shuicai, Shi., Yuzhong, Chen., and Xueqiang, Lv. (2008). Chinese Pronominal Anaphora Resolution Based on Conditional Random Fields. *In: International Conference on Computer Science and Software Engineering*, Washington, DC, USA, pp. 731-734.
- Ghosh A., Neogi S., Chakrabarty S., and Bandyopadhyay S. (2011). Anaphora resolution in Bengali. *In proceeding of ICON Tool Contest, 9th International Conference on Natural Language Processing*, Chennai, India
- Hardmeier C., Tiedemann J., Nivre J. (2013). Latent Anaphora Resolution for Cross-Lingual Pronoun Prediction *In: Proceedings of Conference on Empirical Methods in Natural Language Processing*, pages 380–391, Seattle, Washington, USA
- Hendrickx I., Hoste V., and Daelemans W. (2008). Semantic and syntactic features for Dutch coreference resolution. In Gelbukh A. (Ed.), *CICLing-2008 conference*, Vol. 4919 LNCS, Berlin, Springer Verlag, pp. 731-734.
- Hobbs J. (1978). Resolving pronoun references. *Lingua* 44, pp. 339-352.
- Iida R., and Poesio M. (2011). A Cross-Lingual ILP Solution to Zero Anaphora Resolution. *In: Proceedings of ACL 2011*
- 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.
- Lakhmani. P., and Singh. S. (2013) Anaphora Resolution in Hindi Language. *International Journal of Information and Computation Technology* Vol. 3, pp. 609-616
- Lappin S., and Leass H. J. (1994). An algorithm for pronominal anaphora resolution. *Computational Linguistics* 20 (4), pp. 535-561.
- Li D., Miller T., and Schuler W. (2011). A Pronoun Anaphora Resolution System based on Factorial Hidden Markov Models. In: *Proceedings of ACL 2011*
- 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.
- Mitkov, R. (1997). "Factors in anaphora resolution: they are not the only things that matter. A case study based on two different approaches". In *Proceedings of the ACL’97/EACL’97 workshop on Operational factors in practical, robust anaphora resolution*, 14-21. Madrid, Spain.
- Murthi N.K.N., L. Sobha, B. 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.
- Ng V., and Cardie C. (2002). Improving machine learning approaches to coreference resolution. In. *40th Annual Meeting of the Association for Computational Linguistics*, pp. 104-111.
- Prasad R., and Strube, M. (2000). Discourse Saliency and Pronoun Resolution in Hindi, *Penn Working Papers in Linguistics*, Vol 6.3, pp. 189-208,
- Rafiya Begum, Samar Husain, Arun Dhawaj, Dipti Misra Sharma, Lakshmi Bai, Rajeev Sangal. (2008). Dependency Annotation Scheme for Indian Languages. In *proceedings of IJCNLP 2008*. pp: 721-726
- Rajesh Bhatt., Bhuvana Narasimhan., Martha Palmer., Owen Rambow., Dipti Misra Sharma., Fei Xia. (2009). A Multi-Representational and Multi-Layered Treebank for Hindi/Urdu. In: *Linguistic Annotation Workshop* pp: 186-189
- Recasens M., Marquez L., Sapena E., Martı M.A., Taulé M., Hoste V., Poesio M., Versley Y. (2010). SemEval-2010 Task 1: Coreference Resolution in Multiple Languages. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, Uppsala, Sweden, .pages 1–8.
- Recasens M., Hovy E. (2009). A Deeper Look into Features for Coreference Resolution. Lalitha Devi, S., Branco, A. and Mitkov, R. (eds.), *Anaphora Processing and Applications (DAARC 2009)*, LNAI 5847, Springer-Verlag Berlin Heidelberg, pp.535-561.
- Sameer Singh, Amarnag Subramanya, Fernando Pereira and Andrew McCallum. (2011). Large-Scale Cross-Document Coreference Using Distributed Inference and Hierarchical Models. In: *Proc of ACL 2011*
- Senapati A., Garain U. (2013). GuiTAR-based Pronominal Anaphora Resolution in Bengal. In: *Proceedings of 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria pp 126–130
- Senapati A., and Garain. U. (2011) Anaphora resolution system for Bengali by pronoun emitting approach. In *proceeding of ICON Tool Contest, 9th International Conference on Natural Language Processing*, Chennai, India
- Sikdar U.K, Ekbal A., Saha S., Uryupina O., Poesio M. (2013). Adapting a State-of-the-art Anaphora Resolution System for Resource-poor Language. In *proceedings of International Joint Conference on Natural Language Processing*, Nagoya, Japan pp 815–821
- Sobha L. and Patnaik B. N. (2000). Vasisth: An Anaphora Resolution System for Indian Languages. In *Proceedings of International Conference on Artificial and Computational Intelligence for Decision, Control and Automation in Engineering and Industrial Applications*, Monastir, Tunisia.
- Sobha L. and Patnaik, B.N. (2002). Vasisth: An anaphora resolution system for Malayalam and Hindi. In *Proceedings of Symposium on Translation Support Systems*.
- Sobha L. (2007). Resolution of Pronominals in Tamil. Computing Theory and Application, *The IEEE Computer Society Press*, Los Alamitos, CA, pp. 475-79.
- Sobha L., Pralayankar P. (2008). Algorithm for Anaphora Resolution in Sanskrit. In *Proceedings of 2nd Sanskrit Computational Linguistics Symposium*, Brown University, USA, 2008.
- Sobha, Lalitha Devi., Vijay Sundar Ram and Pattabhi RK Rao. (2011). Resolution of Pronominal Anaphors using Linear and Tree CRFs. In. *8th DAARC*, Faro, Portugal, 2011.
- Sobha L., Sivaji Bandyopadhyay, Vijay Sundar Ram R., and Akilandeswari A. (2011). NLP Tool Contest @ICON2011 on Anaphora Resolution in Indian Languages. In: *Proceedings of ICON 2011*.
- Soon W. H., Ng, and Lim D. (2001). A machine learning approach to coreference resolution of noun phrases. *Computational Linguistics* 27 (4), pp.521-544
- Taku Kudo. 2005. CRF++, an open source toolkit for CRF, <http://crfpp.sourceforge.net> .
- Uppalapu. B., and Sharma, D.M. (2009). Pronoun Resolution For Hindi. In: *Proceedings of 7th Discourse Anaphora and Anaphora Resolution Colloquium (DAARC 09)*, pp. 123-134.
- Vijay Sundar Ram, R. and Sobha Lalitha Devi. (2013). "Pronominal Resolution in Tamil Using Tree CRFs", In *Proceedings of 6th Language and Technology Conference, Human Language Technologies as a challenge for Computer Science and Linguistics - 2013*, Poznan, Poland
- Vijay Sundar Ram and Sobha, Lalitha Devi (2012). Coreference Resolution using Tree-CRF. In A. Gelbukh (ed), *Computational Linguistics and Intelligent Text Processing, Springer LNCS* Vol. 7181/2012, pp.285-296
- Wick M., Singh S., and McCallum A. (2012). A Discriminative Hierarchical Model for Fast Coreference At Large Scale. In: *Proceedings of ACL 2012*.

Automatic Identification of Discourse Relations in Indian Languages

Sobha Lalitha Devi, Sindhuja Gopalan, Lakshmi S

AU-KBC Research Centre,
MIT Campus of Anna University,
Chennai, India.
sobha@au-kbc.org

Abstract

This paper describes the first effort on automatic identification of connectives and their arguments for three Indian languages Hindi, Malayalam and Tamil. We have adopted machine learning technique Conditional Random Fields (CRFs) for our work. We have used a corpus of 3000 sentences belonging to health domain. Domain independent features were extracted to improve the performance of the system. We mainly concentrated on the identification of explicit connectives and their arguments. Two sets of experiments were performed. First set of experiment was performed for the identification of connectives and next for the identification of argument boundaries. Using this approach we obtained encouraging results for all the three languages. Error analysis shows the presence of different structural patterns of discourse relations among three languages.

Keywords: Discourse relation, CRFs, Connectives, arguments

1. Introduction

Discourse relations link clauses and sentences in the text and compose the overall text structure. The study of discourse is concerned with analysing how clause or sentence level units of text are related to each other within a larger unit of text (Wellner & Pustejovsky, 2007). The two basic units of discourse relations are discourse markers and their arguments. The discourse markers are the words or phrases that connect two clauses or sentences and establish a relation between two discourse units. Consider the Example 1

(1) Raju likes sweets **but** his friend prefers chocolates.

In the above example the connective “but” establishes a relation between two clauses thus making the text coherent.

Discourse relations are extensively used in NLP applications including text summarization, natural language generation, generation of speech from text, dialogue generation and question answering systems. Hence the automatic detection of discourse relation is important. Automatic identification of discourse relation is a challenging task. Discourse connectives, despite their common function of linking the contents of two different clauses, also acts as a conjunction (Versely, 2010). Hence it is difficult to distinguish discourse and non-discourse markers. The automatic identification of argument boundaries is even more difficult. The argument for a discourse marker can be clauses or sentences. The discourse relation can be inferred intra sententially or inter sententially. This makes the automatic identification of arguments difficult as the boundaries of the arguments may lie within a sentence or adjacent sentences. Sometimes the relation inferred by the discourse marker can be between non adjacent sentences.

Instead of identifying all possible discourse relations, we concentrated on identifying the explicit discourse connective and their arguments for Indian languages like Hindi, Malayalam and Tamil. To the best of our knowledge this is the first ever approach in the identification of connectives and their arguments in Hindi, Malayalam and Tamil. We adopted the machine learning technique CRFs to build models for the identification of discourse connectives and the boundaries of the

arguments. The error patterns were analyzed and the existence of different structure pattern of discourse relations in three languages was observed.

Work on the annotation of discourse connectives and their arguments have been explored in various languages like Turkish (Zeyrek et al., 2008), Czech (Mladová et al., 2008), Arabic (Al-Saif and Markert, 2010), English (Prasad et al., 2008), etc. Penn Discourse Tree Bank (PDTB) is the large scale annotated corpora of linguistic phenomena in English (Miltsakaki et al., 2004). Roze et al., (2010) has presented a French lexicon of 328 discourse connectives and discourse relations they convey. Published works on discourse relation annotations in Indian languages is available for Hindi (Kolachina et al., 2012; Prasad et al., 2008; Oza et al., 2009) and Tamil (Rachakonda and Sharma, 2011).

Automatic identification of discourse connectives and arguments has been carried out for various languages like English, Arabic, Chinese and German. Marcu and Echihabi (2002) have focused on recognition of discourse relation using cue phrases, but not extraction of arguments. Wellner and Pustejovsky (2007) used maximum entropy rankers combined with a re ranking step to jointly select the two arguments of each connective in the PDTB. Rather than identifying the full extent of the argument, they re-cast the problem to that of identifying the argument heads, effectively sidestepping the problem of discourse segmentation. Robert Elwell and Jason Baldrige (2008) used maximum entropy rankers and showed that using models for specific connectives and the types of connectives and interpolating them with a general model improved the system performance. Their best model achieves a 3.6% absolute improvement over the state of art on identifying both arguments of discourse connectives. Versley (2010) presented his work on tagging German discourse connectives using English training data and a German – English parallel corpus. He has presented an approach to transfer a tagger for English discourse connectives by annotation projection using a freely accessible list of connectives as the only German resources. He achieved an F-score of 68.7% for the identification of discourse connectives.

Ghosh (2012) used a data driven approach to identify arguments of explicit discourse connectives in the PDTB corpus. Lin et al., (2012) developed a PDTB styled

end-to-end parser consisting of multiple components joined in a sequential pipeline architecture, which includes a connective classifier, argument labeler, explicit classifier, non-explicit classifier, and attribution span labeler. The parser gave an overall system F_1 -score of 46.80% in partial matching utilizing gold standard parses, and 38.18% with full automation. Al Saif, (2012) used machine learning algorithms for automatically identifying explicit discourse connectives and its relations in Arabic language. Wang et al., (2012) used sub-trees as features and achieved a significant improvement in identifying arguments and explicit and implicit discourse relations in one go. Zhou et al., (2012) presented the first effort towards cross lingual identification of the ambiguities of discourse connectives. A language independent framework is proposed utilizing bilingual dictionaries, PDTB and parallel data between English and Chinese. The results showed that, this method not only built a high quality connective lexicon for Chinese, but also achieved a high performance in recognizing the ambiguities. Fan et al., (2012) has presented a unified framework for discourse argument identification using shallow semantic parsing. Faiz et al., (2013) improved explicit discourse connectives identification in the PDTB and the Biomedical Discourse Relation Bank (BDRB) by combining certain aspects of the surface level and syntactic feature sets. Ramesh et al., (2010) developed supervised machine-learning classifiers with CRFs for automatically identifying discourse connectives in full-text biomedical articles. They have trained the classifier with Penn Discourse tree bank and obtained a 0.55 F_1 -score for identifying the discourse connectives. Cross validation results in the biomedical text attained 0.69 F_1 -score. In our work, we have identified both connectives and their arguments using CRFs techniques by extracting various features.

The rest of the paper is organised as follows. The following section gives an overview of discourse relations. Third section explains the method used for the automatic identification of discourse relation. The results are presented in section 4. In section 5 the errors obtained in automatic identification of discourse relation is analyzed. The paper ends with the conclusion.

2. Overview of discourse relations

Our work concentrates mainly on automatic identification of explicit discourse connectives and their arguments. Hence, in this section we give an overview of discourse relation realized explicitly by the connectives. The relation inferred can be explicit or implicit. While tagging the corpus, we also observed other types of relations such as anaphoric relations and Noun-Noun relations.

2.1 Explicit Connectives

Explicit connectives connect two discourse units and trigger discourse relation. They occur as free words or bound morphemes. They occur intra sententially or inter sententially. It can occur at the initial, medial and final position in an argument. Different types of explicit connectives are explained below with examples.

2.1.1. Subordinators

Subordinators connect the main clause with the subordinate or dependent clause. It provides a necessary

transition between two ideas in a sentence. The transition indicates a time, cause effect relationship, etc., It also reduces the importance of one clause, thereby giving more importance to another clause.

(2) Hi: [dera raata meM jaagne se baceM]/_{arg1}, **kyoMki**
late night in awake by avoid because
[Shariira ko niiMda kii jaruurata
body for sleep gen necessary
hotii hai]/_{arg2}.
is
(Avoid being awake at late night, because sleep is necessary for the body.)

In the above Example 2, the connective “kyoMki” act as subordinator inferring a relation between two clauses.

2.1.2. Coordinators

Coordinators connect two or more items of equal syntactic importance. They connect two independent clauses. In the Example 3 given below “aanaal” is the coordinator that connects two independent clauses.

(3) Ta: [paal vayirrai mantam aaki jiiraNa caktiyai
milk stomach sloth make digestion power
kuraikkum]/_{arg1}. **aanaal** [tayir paalai
decrease but curd milk
vita viraiVaaka jiiraNam aakum]/_{arg2}.
than fast will digest
(Milk sloth the stomach and will decrease the digestive power. But curd will digest faster than milk.)

2.1.3. Conjunct adverbs

A conjunct adverb is an adverb that connects two independent clauses. They are said to modify the clauses or sentences in which they occur.

(4) MI: [naam kazhikkunna aahaaramaaN nammute
we eating food_is our
Sareeram]/_{arg1}. **athinaal** [nalla
body therefore good
aaharaSeelam namme rogangaLil ninnum
eating_habits us diseases from
akattum]/_{arg2}.
away
(The food what we eat is our body. Therefore good eating habits will keep us away from diseases.)

In the above Example 4 “athinaal” is the conjunct adverb that connects the two sentences.

2.1.4. Correlative conjunction

Correlative conjunctions are paired conjunctions. They link words or group of words of equal weights in a sentence.

(5) Hi: [khuuba paanii piine se]/_{arg1} na kevala [khanee ke
lot water drink by not only food
tukade saafa ho jaate hai]/_{arg1}, **balki**
particles get cleans but
[laara bhii banatii hai]/_{arg2}.
saliva also produce
(Drinking lot of water not only cleans the food particles, but also produces saliva.)

In Example 5 the correlative conjunction “na kevala balkii” occur as a paired connective.

2.1.5. Complementizer and Relativizer

A complementizer is a conjunction which marks a complement clause. A complementizer is a lexical category, including those words which can be used to turn a clause into subject or object of a sentence.

(6) MI: [2050-ote lokathil 85 peril oraaLkk
2050-by world 85 people one_person
alsheemers rogam vyaapikkum]/arg1 **enn** [ron
alzheimer's disease spread that Ron
brookmeyarum kootarum
Bookmyer colleague's
choontikaaNikkunnu]/arg2.
points_out
(Ron Brookmeyer and colleagues points out that,
by 2050 at least one person among 85 people will
get Alzheimer's disease.)

Relativizer pronoun does not refer to a preceding noun, rather it comments on the whole preceding clause or sentence. It provides a link between main clause and relative clause.

(7) Hi: [kuch loga behatara aadiyo larnara hote
some people better audio learner will
haiN]/arg1, **jo** [sunakara siikhte haiN]/arg2.
be who listen learn is
(Some people will be better audio learners, who
can learn by listening.)

3. Method

3.1 Corpus collection and Annotation

The corpus used for our work belongs to the health domain, as a large number of relations were seen to exist in health corpus. The corpus consisted of health related articles from blogs, Wikipedia, etc. The inconsistencies like hyperlinks were removed and we obtained a corpus of 3000 sentences for all three languages. The statistics of the corpora for Hindi (Hi), Malayalam (MI) and Tamil (Ta) is given in Table 1.

Corpus	Total sentences	Total tokens	Total Discourse relations observed
Hindi	3000	50335	841
Malayalam	3000	35865	1192
Tamil	3000	39714	936

Table 1: Corpora Statistics.

Annotation of the corpus with various types of discourse relation like explicit discourse relation, implicit discourse relation and other relations was done. The annotation was purely syntactic. We used the following guidelines to annotate the discourse relations in the text. The initial argument was marked as arg1 and the second argument as arg2. arg2 always follows the connective. The argument's span and connectives are denoted by following notation. The connectives are shown in bold. The square brackets subscripted with arg1 and arg2 are used for denoting the argument's span. The linear order of the arguments and markers are not always the basic order arg1-connective-arg2. It varies according to the marker. We observed that all relations can either be between sentences or clauses and not with phrases. One single sentence can serve the purpose of connective and their arguments. Sometimes, one of the preceding sentence acts as an argument. Also the argument can be a non-adjacent

sentence. But the text span follows the minimality-principle. Multiple sentences can also act as an argument. The connectives in Hindi occur as free words and connectives in Malayalam and Tamil occur as bound morphemes and free words. When free words occur, we tag them separately and the discourse unit between which the relation is inferred is marked as arg1 and arg2. When the connectives exist as bound morphemes we keep them along with the word to which it is attached. Annotated corpus is used to train the system and the models are built for the identification of connectives and arguments.

3.2 Implementation

This section describes the implementation of discourse relations using machine learning technique CRFs.

3.2.1. CRFs

CRFs is a framework for building probabilistic models to segment and label sequence data. When an observation sequence is given, the conditional model can specify the probabilities of possible label sequences. It is a form of undirected graphical model. The advantage of CRFs is its conditional nature and it avoids label bias problems. In our work we have used CRF++ tool (Kudo, 2005), an open source implementation of CRFs. It is a simple and customizable tool. It can be used for generic purpose and can be applied to a variety of NLP tasks.

3.2.2. Feature selection

The performance of the machine leaning technique depends on the features used in learning. For the identification of connectives, we have used Parts of Speech (POS) tagging information, morphological suffixes and clause information as features for Malayalam and Tamil. Morphological suffixes such as conditional markers, causal markers, relative participle (RP) marker followed by postposition (PSP) and coordination markers were used. For connective identification in Hindi, word and POS tagging information were used. For argument identification we have taken POS tagging information, morphological suffixes, clause information, combination of POS and chunk information and connectives as features. The features used for Hindi, Malayalam and Tamil are listed below.

Connectives	Arguments
1. Word	1. POS tagging
2. POS tagging	2. Chunk information
	3. Combination of 1& 2
	4. Connectives

Table 2: List of Features used for Hindi.

Connectives	Arguments
1. POS tagging	1. POS tagging
2. Clause information	2. Chunk information
3. Morphological suffixes	3. Combination of 1& 2
a. Conditional markers	
b. Causal markers	

c. RP marker followed by PSP	4. Connectives
d. Coordination markers	5. Clause information

Table 3: List of Features used for Malayalam and Tamil.

3.2.3. Preprocessing

Preprocessing is an important task, where the corpus is preprocessed with tokenizer, Morph analyzer, POS tagger, chunker, pruner and clause identifier. First, tokenizer splits the sentences into words. Sentence boundary is denoted by an empty line. Then, morphological analyzer provides grammatical information of the word. POS tagger assigns POS information for each word. Chunker identifies the boundary of the chunks. Pruner prunes the morph analyzers multiple analysis. Clause identifier identifies the clauses in the sentence. This preprocessed data are provided as input to train the CRFs.

3.2.4. Experiment

The method used by Menaka et al., (2011) for identifying cause-effect relations have been adopted by us for the identification of discourse relations. In our work we developed discourse relation models for three languages like Hindi, Malayalam and Tamil. The models were built using the health corpus with 3000 sentences for three languages. The features mentioned in table [2] and table [3] was extracted to train the system. We have performed two sets of experiments in our work for the identification of discourse markers and arguments. First, the system was trained for the identification of discourse markers. In the second set of experiments, the system was trained for the identification of arguments. The output from the first experiment is fed as input to the second experiment. In this step four models were built for the identification of each boundary of the arguments. We identified the argument boundaries in the following order – arg1 end, arg2 start, arg1 start and arg2 end. The system was tested with the test corpus. We found that the features such as POS and chunk play a significant role in the identification of discourse markers. Features like POS, chunk, morph, clause and connectives have significant impact on the identification of their arguments.

4. Results

The evaluation of the system was carried out in the terms of precision, Recall and F-measure, where precision is the number of correctly tagged connectives or arguments by the system to the total number of connectives or arguments tagged by the system. Recall is defined as the number of correctly tagged connectives or arguments by the system to the total number of connectives or arguments present in the corpus. F-score is the weighted average of the precision and recall. To evaluate the performance of the system we divided the corpus in the ratio 80:20 for all three languages. 80% of the corpus was used to train the system and 20% of the corpus was used as test data. The results for connective and argument identification are tabulated below.

	Precision	Recall	F-score
Hindi	96.33	91.3	93.8
Malayalam	96.29	91.23	93.68
Tamil	95.35	93.18	94.25

Table 4: Results for connective identification.

The precision, recall and F-score for connective identification are given in above table [4].

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 5: Results for arg1 start identification.

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 6: Results for arg1 end identification.

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 7: Results for arg2 start identification.

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 8: Results for arg2 end identification.

The connectives are identified by the system initially and the output from this model is given as input for the identification of argument boundaries. We have built 4 models for the boundary identification. The above tables show the results for the identification of argument boundaries. Each boundary of the arguments is evaluated and the measures for argument boundaries are mentioned in table [5, 6, 7, 8]. The F-score for arg1 end and arg2 start is found to be high in all 3 languages because of its close proximity to the markers in most cases.

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 9: Results for arg1 start and end identification.

	Precision	Recall	F-score
Hindi	75.4	71.2	73.3
Malayalam	78.57	70	74.04
Tamil	81.53	71.6	76.57

Table 10: Results for arg2 start and end identification.

The evaluation measures for complete identification of argument boundaries are given in table [9, 10]. We have analysed the errors in identification of connectives and arguments by the system and have explained it in the following section.

5. Error Analysis

Error analysis shows that there are certain ambiguities regarding the identification of connectives, because all connectives do not act as discourse markers. Consider the Example 8 given below.

- (8) Ta: muTTai **maRRum** elumiccai caaRu ularnta
 egg and lemon juice dry
 carumattiRku nallatu.
 skin good
 (Egg and lemon juice are good for dry skin.)

Here “maRRum” acts as a conjunction that connects two words egg and lemon juice. In this case “maRRum” is not a discourse marker.

The errors in the identification of argument boundaries are due to the presence of paired connectives, overlapping structures, shared arguments between discourse relations. Paired connectives exist in Hindi. For example, connectives like “agar-to”, “na keval-balki” occur as pairs in a sentence.

- (9) Hi: **agar** [100 ef se jyaadaa fiivara ho]/arg1 **to**
 If 100 F than more fever is then
 [pairaasiitaamola kii golii deN]/arg2.
 paracetamol -gen tablet give
 (If the fever is more than 100 F, then give paracetamol tablets.)

In the above Example 9, paired connectives “agar” and “to” will share the same arguments. We found that the error is generated by the system in such cases.

While analysing the corpus, we found that multiple connectives occurred within the same sentence. In such cases the arguments are shared by these connectives. Errors are generated by the system, while identifying these types of shared arguments. In the Example 10 given below, there are two connectives “aur” and “isliye”. The arg2 of first discourse relation is shared as arg1 by the second discourse relation realized by connective “isliye”. The linear order of this pattern is arg1-Con_i-arg2_{ij}-con_j-arg2_j

- (10) Hi: [[boojana kaa paacana Tiika se hotaa hai]/arg1_i
 food gen digestion properly is

aur_i [shariira ko urjaa milatii hai]/arg2_i/arg1_j,
 and body for energy get is
isliye_j [reshedaar padaartha bhojana meM jaruuri
 because fibrous material food in necessary
 hote haiM]/arg2_j.
 is

(Food is properly digested and the body gets energy, so fibrous material is needed in food.)

While analysing the errors that occurred in the identification of arguments in all three languages, we observed various patterns of interactions between the arguments of discourse relations. Unique pattern of interactions also exists between three languages. We have discussed the observed variation in argument structures with examples.

The Example 11 in Hindi shows that the connective “na keval balki” occurs as paired connective. Connective “na keval” is found to be embedded in the medial position of arg1. “na keval balki” in Malayalam and Tamil is “maathramalla pakshe” and “maTTumalla aanaal” respectively. In Malayalam and Tamil “maathramalla” and “maTTumalla” occur at the final position of arg1.

- (11) Hi: [machalii khane se]/arg1 **na keval** [aapakii,
 fish eat by not only your's
 tvaca, baal, dimaag svastha rahataa
 skin hair brain healthy keeps
 hai]/arg1 **balki** [najar bhii teja hotii hai]/arg2.
 is but sight also sharp is
 (Eating fish, not only keeps your skin, hair, brain healthy, but also keeps the eyesight sharp.)

In this case we observed that the system could not identify the arg1 properly as the connective “na keval” is sandwiched in the middle of arg1 clause.

The connective “and” in Tamil is “um” and maTTtrum” and in Malayalam “um” and “mattum”. In Malayalam and Tamil “um” act both as connective and clitic marker. When two similar constituents in a sentence are affixed with “um”, it acts as conjunction “and”. It can occur between two phrases, clauses and sentences. “um” is considered as discourse connective only when it occurs between two clauses or sentences. They always occurs as paired connective as shown in the Example 12. Hence there are ambiguities in the identification of “um” as a discourse marker by the system.

- (12) Ta: [maaraTaippu eRpaTTa anaivarukkumme
 Heart_attack develop everyone
 muuccu vituvatil ciramam
 breathing_in difficulty
 eRpaTTiruppatum]/arg1,[atuvum 4 vaara
 will develop_and that_too 4 week
 iTaiveLikkul 80 cataviitam perukku intha
 gap_within 80 percentage people this
 aRikuRi eRpaTTiruppathum]/arg2 aayvill
 symptom develop_and study
 teriya vantullaLtu.
 got_to_know
 (A study reveals that everyone who got heart attack had developed difficulty in breathing and that too in between a gap of 4 weeks this symptom has developed.)

The above example shows that the connective “um” occurs in the final position of arg1 and arg2. Whereas this is not the case in Hindi, where the connective “and” occur as free lexicon “aur”.

In the Example 13 given below, there are two types of

discourse relations in the consecutive sentences. The first discourse relation is similar to the type explained in example 11. In the second discourse relation, the relation between the arguments is realized by the connective “lekin”. Here, we observe that the whole of the first discourse relation realized by the connective “yadi-to” acts as the argument for the second discourse relation. This pattern is true for Malayalam and Tamil. Hence it is difficult for the system to identify the arguments correctly.

(13) Hi: [[koyii suucanaa yaa rahasya]/arg1i **yadi**i [ek any information or secret if one klika kii doorii para ho]/arg1i, click distance on is **to**i [koyii kyoN apane dimaaga ko kasTa then any why their mind for suffer denaa caahegaa]/arg2i/arg1j. **lekin**j give want but [aage calakara ye aadateN visheSha ruup se further go these habits particular form mastiSka kii yaada rakhane kii kshamataa brain gen remember gen capacity para ghaataka asar Daal saktii haiN]/arg2j. on fatal impact can_put (If any information or secrets are a click away, then why would any want to give sufferings for their mind. But later these habits particularly have fatal impact on memory power.)

(14) Mi: [[maathapithaakkaL kootuthalum parents mostly jolikkaaraay**athinaal**]/arg1 [dhaaraaLam employed_hence lots_of budhimuttukaL uNtaayekkaam]/arg2]/arg1 problems happen **enkilum** [kuttiyute uthama bhaavikk vendi but child's better future for rakshakarthaakkaL chila vittuveezhchakaLkk parents some compromises thayyaraakaNam]/arg2 ready (Most of the parents are employed, hence lots of difficulties happen. But for their child's better future parents must be ready for compromises.)

In the above Example 14 the first discourse relation as a whole acts as arg1 for the second discourse relation. The discourse markers are usually different for the two relations which are structurally interdependent.

As in the Example 15, the relations are also realized between non-adjacent sentences. In such cases the identification of arguments becomes difficult for the system. Given below is an example in Malayalam where non adjacent sentences act as arguments.

(15) Mi: [nilavil rogamuLLavaril ninnu maathrame presently patients from only kshayarogam pakarunnuLLu.]/arg1 athaayath tuberculosis spread that_is rogaaNukkaL Sareerathil uNtegilum nilavil germs body is_there presently rogam illaathavaril ninn rogam pakarunnilla. disease not_present from disease not_spread oraalil ninn matoraalilekk rogam one_person from another_person disease pakarunnath niravadhi GatakangaLe spread many aspects

aaSrayichirikkunnu **athinaal** [rogam vanna aaLe depends therefore disease came person mattuLLavaril ninn akatti niRthiyaal ee other_people from away keep this rogam thatayaam.]/arg2 prevent disease

(Transmission of tuberculosis occurs only from people with active TB. Those with latent infection are not thought to be contagious. The transmission of the disease from one person to another depends upon several factors. Therefore segregating those with active TB from others can stop the spread of this disease.)

6. Conclusion

In this paper, we have presented our approach in identifying discourse connectives and their arguments in Indian languages – Hindi, Malayalam and Tamil. We observed that dividing the identification task into two sub-phases (connective identification and arguments identification) have really improved the system performance. To the best of our knowledge, we have provided the first results on automatic identification of discourse connectives and arguments in three languages. The results obtained are encouraging. The existence of structural dependencies influenced the performance of the system. Our future work is to resolve the embedded structure relation identification to improve the performance of the system.

7. References

- AlSaif, A. (2012). Human and automatic annotation of discourse relations for Arabic. Ph.D. thesis, University of Leeds.
- Al-Saif, A., and Markert, K. (2010). The Leeds Arabic Discourse Treebank: Annotating Discourse Connectives for Arabic. In *Proceedings of Language Resources and Evaluation Conference*. Valletta, Malta.
- Castor, A., and Pollux, L. E. (1992). The use of user modelling to guide inference and learning. *Applied Intelligence*, 2(1), pp. 37-53.
- R. Elwell and J. Baldrige. (2008). Discourse connective argument identification with connective specific rankers. In *Proceedings of the International Conference on Semantic Computing*. Santa Clara, CA.
- Faiz, S. I., and Mercer, R. E. (2013). Identifying explicit discourse connectives in text. *Advances in Artificial Intelligence*, Springer Berlin Heidelberg, pp. 64-76.
- Ghosh, S. (2012). End-to-End Discourse Parsing with Cascaded Structured Prediction. Doctoral dissertation, University of Trento.
- Kolachina, S., Prasad, R., Sharma, D. M., and Joshi, A. K. (2012). Evaluation of Discourse Relation Annotation in the Hindi Discourse Relation Bank. In *Proceedings of Language Resources and Evaluation Conference*. pp. 823-828.
- Kudo, T. (2005). CRF++, an open source toolkit for CRF, <http://crfpp.sourceforge.net>.
- Lin, Z., Ng, H. T., and Kan, M. Y. (2012). A PDTB-styled end-to-end discourse parser. *Natural Language Engineering*, 1(1), pp. 1-34.
- Marcu, D., and Echiabi, A. (2002). An unsupervised

- approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics. pp. 368-375
- Menaka, S., Pattabhi RK Rao, and Sobha Lalitha Devi. (2011). Automatic identification of cause-effect relations in tamil using CRFs. In *Computational Linguistics and Intelligent Text Processing*. Lecture Notes in Computer Science, 6608, pp 316-327.
- Ming, X. F. Z. Q., and Dong, Z. G. (2012). A Unified Framework for Discourse Argument Identification via Shallow Semantic Parsing. In *Proceedings of International Conference on Computational Linguistics*, Mumbai, India, pp.1331-1340.
- Miltsakaki, E., Prasad, R., Joshi, A. K., and Webber, B. L. (2004). The Penn Discourse Treebank. In *Proceedings of Language Resources and Evaluation Conference*. Lisbon, Portugal.
- Mladová, L., Zikanova, S., and Hajicová, E. (2008). From Sentence to Discourse: Building an Annotation Scheme for Discourse Based on Prague Dependency Treebank. In *Proceedings of Language Resources and Evaluation Conference*. Marrakech, Morocco.
- Oza, U., Prasad, R., Kolachina, S., Sharma, D. M., and Joshi, A. (2009). The Hindi discourse relation bank. In *Proceedings of the Third Linguistic Annotation Workshop* pp. 158-161.
- Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A. K., and Webber, B. L. (2008). The Penn Discourse TreeBank 2.0. In *Proceedings of Language Resources and Evaluation Conference*. Marrakech, Morocco.
- Prasad, R., Husain, S., Sharma, D. M., and Joshi, A. K. (2008). Towards an Annotated Corpus of Discourse Relations in Hindi. In *International Joint Conference on Natural Language Processing*, Hyderabad, India, pp. 73-80.
- Prasad, R., Miltsakaki, E., Dinesh, N., Lee, A., Joshi, A., Robaldo, L., and Webber, B. L. (2007). The penn discourse Treebank 2.0 annotation manual. Technical report. <http://www.seas.upenn.edu/pdtb/PDTBAPI/pdtbannotation-manual.pdf>.
- Rachakonda, R. T., and Sharma, D. M. (2011). Creating an annotated Tamil corpus as a discourse resource. In *Proceedings of the 5th Linguistic Annotation Workshop* pp. 119-123.
- Ramesh, B. P., and Yu, H. (2010). Identifying discourse connectives in biomedical text. In *AMIA Annual Symposium Proceedings*, pp. 657-661.
- Roze, C., Danlos, L., and Muller, P. (2010). LEXCONN: a French lexicon of discourse connectives. *Proceedings of Multidisciplinary Approaches to Discourse*, Moissac, France.
- Versley, Y. (2010). Discovery of ambiguous and unambiguous discourse connectives via annotation projection. In *Proceedings of Workshop on Annotation and Exploitation of Parallel Corpora (AEPC)* pp. 83-82.
- Wang, X., Li, S., Li, J., and Li, W. (2012). Implicit Discourse Relation Recognition by Selecting Typical Training Examples. In *Proceedings of International Conference on Computational Linguistics*, Mumbai, India, pp. 2757-2772.
- Webber, B., Knott, A., and Joshi, A. (2001). Multiple discourse connectives in a lexicalized grammar for discourse. *Computing Meaning*, 2, pp. 229-249.
- Wellner, B., and Pustejovsky, J. (2007). Automatically Identifying the Arguments of Discourse Connectives. In *Proceedings of EMNLP-CoNLL* pp. 92-101.
- Zeyrek, D., and Webber, B. L. (2008). A Discourse Resource for Turkish: Annotating Discourse Connectives in the METU Corpus. In *International Joint Conference on Natural Language Processing*, Hyderabad, India, pp. 65-72.
- Zhou, L., Gao, W., Li, B., Wei, Z., and Wong, K. F. (2012). Cross-Lingual Identification of Ambiguous Discourse Connectives for Resource-Poor Language. In *Proceedings of International Conference on Computational Linguistics*, Mumbai, India, pp. 1409-1418.

Annotating Bhojpuri Corpus using BIS Scheme

Srishti Singh and Esha Banerjee

Jawaharlal Nehru University
New Delhi, India
{singhsriss, esha.jnu}@gmail.com

Abstract

The present paper talks about the application of the Bureau of Indian Standards (BIS) scheme for one of the most widely spoken Indian languages 'Bhojpuri'. Bhojpuri has claimed for its inclusion in the Eighth Schedule of the Indian Constitution, where currently 22 major Indian languages are already enlisted. Recently through Indian government initiatives these scheduled languages have received the attention from Computational aspect, but unfortunately this non-scheduled language still lacks such attention for its development in the field of NLP. The present work is possibly the first of its kind. The BIS tagset is an Indian standard designed for tagging almost all the Indian languages. Annotated corpora in Bhojpuri and the simplified annotation guideline to this tagset will serve as an important tool for such well-known NLP tasks as POS- Tagger, Phrase Chunker, Parser, Structural Transfer, Word Sense Disambiguation (WSD), etc.

Keywords: Bhojpuri annotation, BIS, Classifiers, ergativity, and word formations.

1. INTRODUCTION

Bhojpuri is one of the major Indo-Aryan languages of north India which has been given code ISO 639-3 among world languages. It is spoken in the Uttar Pradesh province of India mainly in Mirzapur, Ghazipur, Jaunpur, Ballia Gorakhpur Deoria, Basti, Azamgarh and Varanasi districts whereas in the Bihar province Rohtas, Eastern Champaran, Saran, Siwan, Ranchi and Bhojpur districts are major Bhojpuri speaking belt. Besides this, it is one of the official Languages in Nepal and Mauritius. It is also spoken in Guyana, Fiji, Uganda and in some parts of Burma. According to 2001 census the number of Bhojpuri speakers was 33,099,497 in India. This big population speaks the language with some regional differences as has been reported in Upadhyay, 1988:

1. More prestigious variety: Bhojpur and Rohtas in Bihar; and Ballia and Ghazipur in U.P.
2. Western Bhojpuri: Varanasi and nearby districts in U.P.
3. Madheshi spoken in Tihari and Gorakhpur.

The intelligibility ratio among different varieties of the language is very high (their syntactic structure and basic lexicons are common) but other elements such as affixes, auxiliaries, address terms, kinship terms and domain specific terms differs a lot.

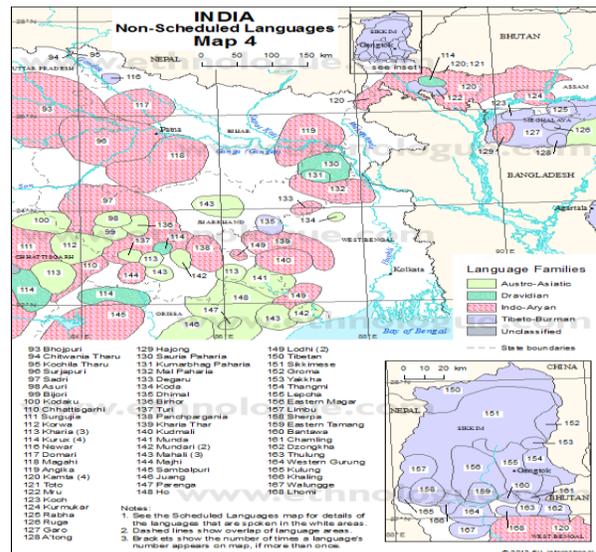


Figure 1: Map of non-schedule languages of Eastern India¹

Bhojpuri entertains SOV word order, postpositions and final noun head; word formation may have 1 prefix, up to 5 suffixes; clause constituents indicated by both case-marking and word order; verbal affixation marks person, number and genders of subject and object. It is an ergative less non-tonal language with 34 consonant and 6 vowel phonemes, about 4 diphthongs. Bhojpuri writing system follows Devanagari and Kaithi script. Some magazines, newspapers, radio programs, dictionaries, are available in the language. Nowadays, local TV channels and films are the most popular resource for Bhojpuri. The domain specific Bhojpuri

¹ <https://www.ethnologue.com/>

corpus is also being developed for different fields especially media and entertainment. The Tagset of Bhojpuri will enable Bhojpuri corpus and other tools such as parser, chunker, morphological analyzer etc. to work better.

2. BIS TAGSET

The BIS tagset 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 is a hierarchical tagset 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 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. The BIS scheme is comprehensive and extensible; captures appropriate linguistic information, and also ensures the sharing, interchangeability and reusability of linguistic resources (Gopal, 2012).

2.1 POS TAGGING

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). 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.

The description of the tagset is given in table 1.

Sl. No.	Category	Subtype (level 1)	Annotation Convention**	Examples	Remarks
1	Noun		N	Darwaza,	
1.1		Common	N NN	Daru:AzA,	
1.2		Proper	N NNP	Bharbittan	
1.3		Nloc	N NST	agwan,	
2	Pronoun		PR	jamun, je ka,	
2.1		Personal	PR_PRP	Okar, u, tu,	
2.2		Reflexive	PR_PRF	Apan, apun	
2.3		Relative	PR_PRL	Jamun, je ka	
2.4		Reciprocal	PR_PRC	Ek dusar, .	
2.5		Wh-word	PR_PRQ	Kaun, kabhar	
2.6		Indefinite	PR_PRI	Koi, kisi	
3	Demonstrative		DM	Ehar, hoary,	
3.1		Disjunct	DM_DMD	Ehar, ohar,	
3.2		Relative	DM_DMR	Jamun, je	
3.3		Wh-word	DM_DMQ	kA, kaun	
3.4		Indefinite	DM_DMI	Koi, kisi	
4	Verb		V	roya, nahwa	
4.1		Main	V_VM	roya, nahwa	
4.2		Auxiliary	V_VAUX	bhasel, lagal	
5	adjective		IJ	sarab	
6	Adverbs		RB	bari, dhire	
7	prepositions		PSP	kA, ka, kar	
8	Conjunctions		CC	jabki, ki	
8.1		co-ordinator	CC_CCD	ya, balki	
8.2		Subordinator	CC_CCS	Magar, to,	
9	Particles		RP	To, hi, bhi,	
9.1		Classifier	RP_CL	Tho, The,	
9.2		Default	RP_RPD	To, hi, bhi,	
9.3		Interjection	RP_DN	Ara, ha, a,	
9.4		Intensifier	RP_DNIF	kHoob,	
9.5		Negation	RP_NEG	nAhi, nA	
10	Quantifiers		QT	Pura, sab,	
10.1		General	QT_QTF	purA, sab,	
10.2		Cardinals	QT_QTC	Ek, du	
10.3		Ordinals	QT_QTO	dusar, tisar	
11	Residuals		RD		
11.1		Foreign word	RD_FW		A word written in script other than the script of the original text
11.2		Symbol	RD_SYM	\$, &, %, (,)	for symbols such as \$, &, etc
11.3		Punctuation	RD_PUNC	!, ?, :, :	only for punctuations
11.4		Unknown	RD_UNK		
11.5		Echowords	RD_ECH	(chup-chAp)	

Table 1: Bhojpuri Tagset

2.2 Corpus and Data

The data for the present experiment is a collection of 9 folk stories currently having approximately 5,300 tagged words (3 stories). The data has been collected in spoken form and then transcribed. It includes two major dialects: ‘Bhojpuri’ spoken in Bhojpur and ‘Benarasi’ spoken in Varanasi. The examples below contain the interlinear glossing, free translation and PsOS tagged data (if required) of the sentences.

2.3 About TAGSET

The present tagset includes 33 categories divided into eleven major Parts of Speech categories which are further sub divided among some lower level categories designed in accordance to the utility of the tool. A word belonging to a particular lexical category may function differently in a given context. Besides adjective, adverb and preposition all other categories have some further distributions. For guidelines see appendices.

3. Characteristics of Bhojpuri

This section discusses the characteristics of Bhojpuri in detail with examples:

3.1 Classifiers

Like Bangla and Maithali, Bhojpuri also heavily use classifiers with numerals. The marker for classifiers in different dialects of language is different as: ‘Tho’, ‘go’, ‘The’, Kho etc

1. There were two brothers.

du	go	bhAI	rahasan
Two	CL	brother	live.PST.HON

This confirms to the shifting paradigm among Indian languages where there is Awadhi on one hand (classifier less language) and Bengali (classifier rich language) on the other hand giving place to Bhojpuri somewhere in between.

3.2 Ergative

In Bhojpuri, there is no overt ergative case marker available with the nouns but the inflection of the verb represents the perfective aspect of the sentence which makes Bhojpuri a Ergative less Language like Awadhi. This construction is quite different from Hindi which is a Ergative language. In some dialects of Bhojpuri the Perfective is marked with the -I- or -les- forms. Following are examples :

2. Ganesu said that he would do everything.

ganesu	kahales	
Ganesu.3MSg.	say.ERG.PST	
Ham	sab	kAm
1MSg	all	work.ACC
Karab		

do.FUT.1MSg

Tagged:

ganesu\N_NNP kahales\V_VM ham\PR_PRP
 sab\QT_QTF kAm\N_NN karab\V_VM

3. How much sugarcane did that old man eat.

kitnA	U	buddhA
How much	that	old-man
U.nkh	chuhales	
sugarcane	eat.ERG.PST	

Tagged:

kitnA\RP_INJ U\DM_DMD buDDHA\N_NN
 U.nkh\N_NN chuhales\V_VM I

The ergativity in Bhojpuri is absent and the aspectual marker is there to make the construction sensible, therefore, we can not include it to any of the categories at this level of annotation.

3.3 Word-Formation Process

With respect to Hindi morphology, Bhojpuri morphology differs greatly in certain categories. For instance, in Hindi the particle for emphatic ‘hi’ and ‘bhi’ are placed under Default Particle whereas in Bhojpuri these elements sometimes occur independently as a separate unit and most of the times these are merged with the host giving rise to a new subcategory under some major categories of the tagset.

For sample, some data paired with its respective variants are presented here to make the point more clear:

- a. biswAs hI → biswAse
 Belief EMPH → belief-EMPH
- b. koI bhI → kauno
 Anybody EMPH → anybody-EMP
- c. tabhI to → tabbe/tabbae
 then EMPH → then-EMPH
- d. kabhI to → kabbo
 Sometime EMPH → sometime-EMPH

4. Nobody believed.

kaunoM	biswAse
Anybody.3MSg.EMPH	believe.EMPH
nahIM	kareM
not	do.PRF.PST

Tagged:

kaunoM\PR_PRI biswAse\N_NN
 nahIM\RP_NEG kareM\V_VM I\RD_PUNC

5. They went and knocked the door still nobody woke up.

jA	ke	kiwADI
----	----	--------

Go do door
khaT-khaTAwat huan tabbo
knock.RDP aux.PRS.PI still

Tagged:

jA\|V_VM ke\|V_VAUX kiwADI\|N_NN khaT-
khaTAwat\|V_VM huan\|V_VAUX tabbo\|RB
nA\|RP_NEG koI\|PR_PRI utHat\|V_VM hau\|V_VAUX
\|RD_PUNC

3.4 Determiners

Determiners are such unique feature of the language which is not there in Hindi. These are basically the discourse particles but the present work is restricted to its syntactic utility only.

Like Maithili, Awadhi, and Bengali, Bhojpuri also gives a wide space for determiners to set in. Determiners in Bhojpuri can occur with almost all the common and proper nouns. Although these determiners are not found with honorific nouns, most of the address terms are suffixed heavily with Determiners. But like emphatic category in Bhojpuri the determiners also get merged with their host nouns and they need to be excluded from the main category of the tagset and also a new category under noun can be proposed where those nouns containing a demonstrative can be put.

Word final sounds	-a/-A	-i/-I	-u
Determiners	-vA	-jA	-A

From the tale above, we get the notion that the word final sound of a noun is responsible for the occurrence of determiners in Benarasi. A word ending with -a or -A sound will take -vA suffix, -i/-I will take -jA suffix and -u will take an -A suffix. Similar constructions are found in Maitheli, Magahi, Awadhi and other related languages.(Kachru, 1980)

Generally, such constructions determines which noun is talked about as inferred from the examples below:

- If BhARBittan would be there,he would have offered us abundance of guavas.
bharbittanwA
Bharbittan.3MSg.DEM
bhAI rahat ta
brother.3MSg/NOM be.PRF EMPH
amrood kHoob toD-toD ke
guava very pluck.RDP PP
khiyAwat
eat.CAUS.PST

Tagged:

bharbittanwA\|N_NN bhAI\|N_NN rahat\|V_VM
ta\|CC_CCS amrood\|N_NN khoob\|RP_INTF toD -
toD\|V_VM ke\|V_VAUX khiyAwat\|V_VAUX

- He had no money to buy the garland.
paisawe nAhI rahal
Money.DEM not live.PST
ki mAIA kHarIde
that garland buy.3MSg.PRS

Tagged:

paisawe\|N_NN nAhI\|RP_NEG rahal\|V_VM
ki\|CC_CCS mAIA\|N_NN kHarIde\|V_VM

- Jackal said to lamb to go to the well along to see who is fairer.
siyarA lahalas
Jackal.3MSg.DEM say.PST
memaNI se chala-chalA
lamb PP lets go.RDP

Tagged:

siyarA\|N_NN lahalas\|V_VM memaNI\|N_NN se\|PSP
chala\|V_VAUX -\|RD_SYM CHALa\|V_VAUX
kua.n\|N_NN meM\|PSP dekHal\|V_VM jAI\|V_VAUX
ke\|PR_PRQ gor\|JJ ba\|V_VAUX |

But the synthetic nature of the language makes it again difficult to give a separate tag for a bound morpheme. Therefore, this issue is presently left to be sorted out with the help of a more robust morphological analyzer and the Determiners are not given a separate category by the time.

3.5 Homophonous

In Bhojpuri homophonous cases are prevalent and this makes annotation task difficult. A human annotator needs to see the tokens in the given context and according to their function assign a proper tag. Some samples:

- The lion started to roar loudly.
ser mAre dahADe lagal
Lion loudly roar started.aux.PST

Tagged:

ser\|N_NN mAre\|RP_INTF dahADe\|V_VM
lagal\|V_VAUX

- Then they all were about to beat him.
ta sab mAre jAt
CONJ all beat go.PST live
rahalan
aux.PROG.PST

Tagged:

ta\|RP_RPD sab\|N_NN mAre\|V_VM jAt\|V_VM
rahalan\|V_VAUX

- He put the bundle on the ground with a that.

U le jA ke
 He.3MSg take go do
 gaTHariyA paTak delas
 bundle throw give.PRF.PST

Tagged:

U\PR_PRP le\V_VM ja\V_VM ke\V_VAUX
 gaTHariyA\N_NN patak\V_VM delas\V_VM

12. He used to purchase the garlands with that money

ta U paisa ke
 CONN. that money PP
 rojAnA mAIA
 daily garland.3FSg
 kHaride
 buy.PRF.PST

Tagged:

ta\RP_RPD U\PR_PRP paisa\N_NN ke\PSP
 nA\RP_RPD ,\RD_PUNC rojAnA\N_NST
 mAIA\N_NN-|RD_PUNC phool\N_NN kharide
 \V_VM

3.5 Different Realizations in Spoken Bhojpuri

Since no natural language is free from ambiguities, Bhojpuri morphology also carries functional ambiguities where one form can be interpreted in more than one functions. It shows great variations in the spoken form of the language. A single entity is pronounced differently in different context and places of their occurrences. Within a variety, these variations can be easily noticed at conjunction, particle and preposition's end. These make it difficult for the tool (a POS tagger or analyzer) to process efficiently on the data and sometimes gives bad results.

As explained below:

13. And he too ate.

aur U bhI Apan
 And he EMPH he.REFL
 khsayiles
 seat.PRF.PST .

Tagged:

aur\CC_CCD U\PR_PRP bhI\RP_RPD Apan\RP_PRF
 kHayiles\V_VM |\RD_PUNC

14. And he climbed up.

Ta U chaDH gael .
 And he.3MSg climb go.PRF.PST

Tagged:

a\CC_CCD U\PR_PRP chaDH\V_VM gael\V_VM

4. Annotation Challenges

The very first challenge with digitizing any language is to have the available data in the desired domain and format. The corpus data here is an extraction from the

spoken data which is transcribed into written corpus, for the task. Bhojpuri is an Ergativeless and classifier rich language which places it somewhere in between the two languages 'Hindi' and 'Awadhi'. The different realizations of the same lexicon and their correct categorization is as challenging as finding out the homophonous words in the data and differentiating their meanings contextually. Determiners inflected with the nouns and the floating particles occurring in between the proper nouns, though not so big a problem at this level, might seek significant attention at other levels of annotations.

5. Conclusion

Bhojpuri, being a developing language, requires more attention which can be attained by generating more NLP resources. However, the corpus annotated with this tagset would be more useful as it is tagged by a standard tagset/scheme. This will maximize the usage of sharing tagged data. The initiative for tagging less resourced Indian languages with the present standard tagset is a promising effort in this direction with the hope that all Indian languages corpora annotation programmes will follow these linguistic standards for enriching their linguistic resources.

6. References

- Bhaskaran, S., Bali, K., Bhattacharya, T., Bhattacharya, P., Choudhury, M., Jha, G.N., S, Rajendran, K., Sravanan, L. Sobha and Subbarao KVS. (2008). A Common Parts-of-Speech Tagset Framework for Indian languages. In *Proceeding of 6th Language Resources and Evaluation Conference (LREC, '08)*.
- Bhaskaran, S., Bali, K., Bhattacharya, T., Bhattacharya, P., Choudhury, M., Jha, G.N., Rajendran S., Sravanan, K., Sobha, L. and Subbarao K.V.S. (2008). Designing a common POS-Tagset Framework for Indian Languages. In *Proceeding of VI workshop on Asian Language Resources, IIIT, Hyderabad*.
- Gopal, M. (2012). Annotating Bundeli Corpus Using the BIS POS Tagset. In *Proceeding of workshop on Indian Language Data: Resources and Evaluation, under (LREC'12)*, pp. 50-56.
- Hardie, A. (2003). *The Computational Analysis of Morphosyntactic Categories in Urdu*. PhD Thesis Lancaster University.

Jha, G. N., Gopal, M. and Mishra, D. (2009). *Annoating Sanskrit Corpus: Adapting IL-POSTS*. Spriger Heidelberg Dordrecht London Ney York.

Kachru, Yamuna.(1980) *Aspects of Hindi Grmmar*. Manohar, niversty if Michigan.

Mitkov, R. (2003). *The Oxford Handbook of Computational Linguistics*. Oxford University Press, New York.

Upadhyay, H. S. (1988). *Bhojpuri folksongs from Ballia*. India Enterprises Encorporated.

<http://www.ethnologue.com/language/bho>

http://shiva.iiit.ac.in/SPASAL2007/iiit_tagset_guidelines.pdf

Appendix Guidelines to Bhojpuri Tagset

1. NOUN (N)

The first major category in the Tagset is ‘nouns’. Though the categorization is made keeping in view that it can broadly cover all the dialects and sub dialects of Bhojpuri, therefore it has three sub types in it.

1.1 Common noun (N_NN)

The nouns that simply function as noun and are content words should be marked as the common noun. This includes the general variety of all the nouns, e.g. darawAzA, samay, log, sAdhU etc.

For example:

- 1) Ab rAnI\N_NN tiyAr hot hain
Now the queen is getting ready.

1.2 Proper noun (N_NNP)

Proper nouns are generally names that stands for some particular person or place. For example Bhabbitan, GanesU, Chunamun etc.

- 2) Cunmun\N_NNP DerAyal
Cunmun got afraid.

1.3 Spatio-temporal noun (N_NST)

There are a specific set of words that functions both as preposition and argument of a verb. Such words are marked as spatio-temporal irrespective of their function in a given context. Some of them are agaweM, pacHaweM, upaM, nichaweM etc.

- 3) Phir Age\N_NST Gael
Then he went forward

2. PRONOUNS (PR)

The category of pronoun has been divided into six sub-categories. These include personal, reflexive, relative, reciprocal, wh-word and indefinite. These categories should be self-explanatory and follows the same definitions as posited in common linguistic literature.

2.1 Personal Pronouns (PR_PRP)

Personal pronouns cover all the pronouns that denotes a person, place or thing. This includeS all their cases as well for example: okar, U, toke, tU, ham etc.

- 4) ser kahalA - ab maiM\PR_PRP tumko khAungA
Lion said, now I will eat you.

2.2 Reflexive Pronouns (PR_PRF)

Reflexive pronouns are the ones that denote to ownership to its antecedent which can be either a noun or a pronoun. There are only a few words in this category, namely Apan, apan, khud etc.

- 5) tab tU Apan\PR_PRF dHolak le ke bhAg jAe

Then you run away with your Dholak.

2.3 Relative Pronouns (PR_PRL)

The relative pronouns are those pronouns whose antecedent can be either a noun or a pronoun. However, these pronouns do not make any difference in number or gender as in the case of personal pronouns. The relative pronoun in Bhojpuri are jaun, jeke, jahAM etc.

2.4 Reciprocal Pronouns (PR_PRC)

Reciprocal pronouns denote some reciprocity. This is commonly denoted by ek dusar, Apas, apne meM etc.

2.5 Wh-pronouns (PR_PRQ)

The wh-word pronouns are typically the pronouns that are used to ask questions. These words are kaun/ke, kab, kehar etc.

2.6 Indefinite Pronouns (PR_PRI)

The indefinite pronouns refer to unspecified objects, places or things. These words are koi, kisietc.

3. DEMONSTRATIVES (DM)

The category of demonstrative has been separated from the category of pronouns as the demonstratives mainly indicate about a noun and does not act as anaphora. The demonstratives have been sub-categorized into four divisions- deictic, relative, wh-words and indefinite.

3.1 Deictic (DM_DMD)

The deictic demonstratives are default demonstratives that demonstrate the noun it modifies. The deictic demonstratives in Hindi are typically I, U, ehar, ohar, je etc. They generally occur before a noun.

3.2 Relative Demonstrative (DM_DMR)

The relative demonstrative occur in the same form as the relative pronoun. The difference is only that these relatives are always followed by a noun that it modifies. For example jaun, je etc.

3.3 Wh-Word Demonstrative (DM_DMQ)

The wh-demonstratives are the same question words as wh-pronouns. The difference is that in their demonstrative function they do not ask question, rather only demonstrates. The wh-word demonstratives in Bhojpuri are kA, kaun etc.

3.4 Indefinite Demonstratives (DM_DMI)

Like indefinite pronouns, the indefinite demonstratives refer to unspecified objects, places or things. These words are koi, kisi etc.

4. VERBS (V)

The verbs are sub- divided into two only- Main (V_VM) and Auxiliary(V_VAUX) Verbs. While the

auxiliary verb is a closed set of verb, the main verb can be anything from a root verb to any of its inflected forms. Each sentence or clause must have a main verb. A sentence can have one more auxiliary verbs.

- 6) kaunoM biswAse nahiM kareM\V_VM
Nobody believed.

5. ADJECTIVES (JJ)

Adjective is a single whole category. There is one definition for an adjective which is self-explanatory. These are mostly attributive adjectives.

- 7) bahut garIb\JJ rahal
He was very poor.

6. ADVERB (RB)

Adverb also is mono-category part-of-speech. The standards document says that the category of adverb (RB) is only for manner adverbs. For example, words like chAhe jaise bhi, tabhiM, bArI, dhIre etc.

- 8) bAbA ke saNge , jaise\RB tU log tAs khelalA waise\RB U pAsA kheleM
The way you play cards, he used to play 'pasa'.

7. POSTPOSITION (PSP)

Postpositions are all the parts-of-speech that work as case marker. Words like meM, se, kA, ke, kar etc. are examples of postposition.

- 9) hAr, bhagwAn jI ka\PSP, herA gael
The god's necklace was lost.

8. CONJUNCTION (CC)

Conjunctions words act as joiners of phrases or clauses within a sentence. The category of conjunction has been divided into two sub-categories of coordinator and subordinator.

8.1 Co-ordinating conjunctions (CC_CCD)

Coordinators are typically the words that join two phrases (noun or verb), of the same category or a clause. Some common conjunctions are aur, par, yA, balkI etc.

- 10) Amrood khUb toD –toD ke khAe
aur\CC_CCD Apan sab bhaiyan ke delas
Plucking the guavas he ate it and gave to all his brothers.

8.2 Subordinating conjunctions (CC_CCS)

Subordinator typically conjoins two clauses and the second clause is subordinated. Some of the subordinate conjunctions are magar, to ki etc

- 11) ta dhobi kahe ki\CC_CCD tU kAm bahut kaile hauA hamAr
The washer man said **that** you have worked a lot for me.

9. PARTICLES (RP)

Particles are words that do not decline and also do not fall into any other categories described above and elsewhere. Bhojpuri particles includes following five sub-categories

9.1 Classifiers (RP_CL)

A classifier sometimes called a **measure word** is a word or morpheme used in some languages to classify the referent of a countable noun according to its meaning.

- 12) dU\QT_QTC go\RP_CL bhaI rahasan
There were two brothers.

9.2 Default Particle (RP_RPD)

Default Particle is a category that includes all those element of the language which though do not have any lexical important but are auspicious functionally. Some of the Bhojpuri default particles are to hI bhI, nA, jI etc.

- 13) to ganes jI\RP_RPD ek The bAlak kA rUP
rakh ke ayilan
Then lord Ganesa appeared in the form of a boy.

9.3 Interjection (RP_INJ)

Interjections are particles which denote exclamation utterances. The common exclamatory marks in Bhojpuri are अरे, हे, ए, हो etc.

- 14) are\RP_INJ ! sab tarapf se band hai
It is closed from all sides

9.4 Intensifier (RP_INTF)

Intensifiers are words that intensify the adjectives or adverbs. The common intensifiers in Bhojpuri are arkhoob, itnA, bahut, mAre, itl etc.

- 15) aum bahut\RP_INTF cHoTe\JJ hokahAM
jaoge
You are too young to go with them.

9.5 Negation (RP_NEG)

The negation particles are the words that indicate negation. These include nAhi, nA, mat, binA, bagair etc.

- 16) ab mat\RP_NEG Aye, nAhiM\RP_NEG ta
bahut mArab
Now don't follow or I will beat you.

10. QUANTIFIERS (QT)

Quantifiers are the words that indicate quantity and modify nouns or adjectives. These have been sub-categorized into three parts- general, cardinals and ordinals.

10.1 General (QT_QTF)

The general quantifiers do not indicate any precise quantity, e.g, purA, sab, ek etc.

- 17) Rahat ta khUb\QT_QTF seb khiyAwat

If he would be there, must have brought us a lot of apples.

10.2 Cardinals (QT_QTC)

The cardinal quantifiers are absolute numbers, either in digits or in words such as 1, 2, 3, ek, do, tIn etc.

- 18) ek\QT_QTC The dhobi ke ghar rahal
There was a washerman's house.

10.3 Ordinals (QT_QTO)

The ordinals denote the order part of the digits such as pahilA, dusar, tIsar etc.

- 19) duno\QT_QTO log gayilan khet meM
Both went to the fields.

11. RESIDUALS (RD)

The category of residuals has been demarcated for the words that are usually not intrinsic part of the language/speech. Divided into five parts, these include foreign words, symbols, punctuations, unknown words and echo-words.

11.1 Foreign Words (RD_RDF)

The foreign words are all the words that are not written in the Devanagari script.

11.2 Symbols (RD_SYM)

The symbols are the characters that are not part of the regular Devanagari script such as *, @, #, \$, % etc.

11.3 Punctuations (RD_PUNC)

Punctuations include the characters that are considered as the regular punctuation marks in Hindi, e.g. (,),,!,?,- etc.

11.4 Unknown (RD_UNK)

Unknown words would the words for which a category cannot be decided by the annotator. These may include words from phrases or sentences from a foreign language written in Devanagari.

11.5 Echo-Words (RD_ECH)

The echo-words are the words that are formed by the morphological process known as echo-formation e.g. (chup-chaap), (sach-much), (kAT-kUT) etc.

Indian Subcontinent Language Vitalization

Andras Kornai, Pushpak Bhattacharyya

Department of Computer Science and Engineering, Department of Algebra
Indian Institute of Technology, Budapest Institute of Technology
kornai@math.bme.hu, pb@cse.iitb.ac.in

Abstract

We describe the planned Indian Subcontinent Language Vitalization (ISLV) project, which aims at turning as many languages and dialects of the subcontinent into digitally viable languages as feasible.

Keywords: digital vitality, language vitalization, Indian subcontinent

In this position paper we describe the planned Indian Subcontinent Language Vitalization (ISLV) project. In Section 1 we provide the rationale why such a project is called for and some background on the language situation on the subcontinent. Sections 2-5 describe the main phases of the planned project: Survey, Triage, Build, and Apply, offering some preliminary estimates of the difficulties at each phase.

1. Background

The linguistic diversity of the Indian Subcontinent is remarkable, and in what follows we include here not just the Indo-Aryan family, but all other families like Dravidian and individual languages spoken in the broad geographic area, ranging from Kannada and Telugu with tens of millions of speakers to the languages of scheduled tribes which may be spoken by only a few hundred people. We define the Subcontinent broadly, so as to include not just India, Pakistan, and Bangladesh, but also Nepal, Bhutan, Sri Lanka, Afghanistan, and the Maldives, because the languages spoken in this geographic area often form cross-border continua. The simple question of exactly how many languages/dialects we need to consider is already fraught with difficulty, with estimates ranging from over 1,600 in the 1961 Census, see <http://www.languageinindia.com/aug2002/indianmother tongues1961aug2002.html>, to less than 500 in the *Ethnologue* (Lewis et al. 2013).

Kornai (2013) divided languages in four major categories: digitally Thriving, Vital, Heritage, and Still. Without prejudging matters, it is clear that on the subcontinent all four possibilities obtain: English is thriving, Hindi is vital, Sanskrit is heritage, and Bagata (the language of a scheduled tribe in Andhra Pradesh, not even listed in the *Ethnologue*) is still. ISLV puts the emphasis on the borderline cases between digitally viable (T and V) languages on the one hand, and digitally dead (H and S) languages on the other. The goal is not just to enhance scholarly knowledge in this area, but also to inform decisionmakers where the limited resources available to language vitalization are best applied. This requires not just a detailed *survey* of the languages in question (see Section 2) but also an objective *triage* mechanism (see Section 3).

We will be paying considerably less attention to languages like English and Hindi that are thriving or nearly so, sug-

gesting that efforts aimed at *building* language technology (see Section 4) are best concentrated on the less vital (but still vital or at the very least borderline) cases at the expense of the obviously moribund ones. To find this borderline we need to distinguish the heritage class of languages, typically understood only by priests and scholars, from the still class, which is understood by native speakers from all walks of life. For heritage language like Sanskrit considerable digital resources already exist, both in terms of online available material (in translations as well as in the original) and in terms of lexicographical and grammatical resources of which we single out the Koln Sanskrit Lexicon at <http://www.sanskrit-lexicon.uni-koeln.de/monier> and the INRIA Sanskrit Heritage site at <http://sanskrit.inria.fr>. For still languages, there is practically nothing, and conservation efforts are very justified.

We emphasize at the outset that we do not advocate the wholesale abandonment of still languages. Unlike in a field hospital, where triage really means the abandonment of the likely fatally wounded so that those who can still be saved get a better chance, here still languages can receive a different kind of treatment, heritage preservation. This is a very worthy goal, and there are already significant societal efforts in this direction such as the Endangered Languages Project at <http://www.endangeredlanguages.com>. This should be kept in mind as we *apply* our findings, especially as the preservation effort is in a substantively different direction, requiring very different resources, than vitalization proper. As we shall see, preservation is primarily the work of anthropologists and linguists trained in fieldwork, while digital vitalization requires machine learning techniques.

2. Survey

The purpose of the first stage of ISLV is to collect a broad range of facts and opinion that covers not just branches of Indic in the strict sense but also languages and cultures deeply influenced by Indic vocabulary and script on the subcontinent. We use the directed crawling technique described in Zseder et al. (2012) to collect as much data for each dialect as possible. An important intermediate result of this stage is the development of robust dialect-identification models along the lines of the well known TextCat (see

<http://odur.let.rug.nl/~vannoord/TextCat>) and CLD2 (see <https://code.google.com/p/cld2>) models, taking into account various encodings ranging from legacy schemes such as ISCII to varieties of Unicode and even latinized writing (still common in text messaging) and scholarly systems such as IPA.

At the current stage, our database covers 634 languages and dialects of the subcontinent, excluding English. Table 1 gives the breakdown per primary country.

Country	Language
Afghanistan	30
Bangladesh	16
Bhutan	20
India	397
Maldives	1
Nepal	107
Pakistan	59
Sri Lanka	4

Table 1: Current coverage

It is, of course, quite debatable whether languages of Nepal or Afghanistan should all be all included, and we welcome any cogent argument in this regard, especially as it is unclear whether the same funding sources that support efforts in India would be equally available in other countries. That said, we lean toward inclusion, rather than exclusion.

3. Triage

Once the data is collected, we apply the methodology of Kornai (2013) to decide which varieties can be classed as vital, heritage, or still. There are no digitally thriving languages as defined originally, though we acknowledge that Hindi may be classified as such. Table 2 summarizes the breakdown is as follows:

Status	Language
vital	36
borderline	21
heritage	1
still	576

Table 2: Main classes

Only one language, Pali, is listed as heritage, since a considerable number of people (over 10,000) are listed as native (L1) speakers of Sanskrit. Be it as it may, the data is dominated by digitally still languages (over 90% of the languages considered, see the Appendix), and we are left with some 50-60 languages that have a chance to take root digitally. It should be noted that our classification of vital vs. borderline was already highly optimistic (see Kornai 2013 for a more detailed description of the conservative methodology chosen so as not to raise false alarms), with languages like Dogri (dgo) listed as digitally vital, which is quite debatable.

In the full study, we are likely to retain the positive outlook that characterized the earlier work, so as not to hinder the

digital ascent of any language that has a fighting chance. With five million speakers, Dogri may very well not be a lost cause. A good first step toward demonstrating the vitality of a language could be the collection of a BLARK (Krauwert 2003).

Since these lists are so small, we provide them in full below, but with the clear understanding that *our results are preliminary*, and more sophisticated data gathering in Phase 1 may still change them.

Vital: Angika, Assamese, Bengali, Bishnupriya, Brahui, Chakma, Dogri, Maldivian, Dzongkha, Gujarati, Goan Konkani, Gujarati, Hindi, Kannada, Kashmiri, Kachchi, Khasi, Khowar, Lushai, Maithili, Malayalam, Marathi, Nepali, Newari, Oriya, Western Panjabi, Dari, Rangpuri, Sanskrit, Sinhala, Seraiki, Sindhi, Tamil, Tulu, Telugu, Urdu.

Borderline: Awadhi, Baluchi, Southern Balochi, Badaga, Bhojpuri, Halbi, Chhattisgarhi, Kukna, Konkani, Manipuri, Naga Pidgin, Ao Naga, Adivasi Oriya, Punjabi, Southern Pashto, Pashto, Rajasthani, Santali, Saurashtra, Sylheti, Kok Borok.

We welcome criticisms both of the data (if a language of the subcontinent that you are working with does not appear on any of the above lists nor in the Appendix this obviously points at an error in the data gathering process) and of the classification. We are particularly interested in cases where languages should evidently be moved from the still to the vital category, or the other way round. Again, we emphasize that these results are preliminary, and we welcome scholarly debate and discussion.

The ISLV plan is to make the final results, and the data it is based on, publicly accessible, either hosted directly at a dedicated website or by means of pointing to or replicating data already available elsewhere.

We should add here that a policy recommendation based on an assessment of digital death should go beyond a simple exhortation to concentrate all effort regarding this language on heritage conservation. As we already noted in Kornai (2013), such efforts, while obviously necessary for preserving the cultural heritage of humankind, contribute practically nothing to language vitality. To mitigate the human cost of digital language death it is therefore suggested that we expend effort on identifying, if at all feasible, for each digitally still language a vital ‘champion’ of similar vocabulary, script, and grammar, with the idea that this champion can become a medium of access to the digital realm that is easier to acquire than English.

In certain cases, such as Andaman Creole (hca), bilingualism is so strong that the choice of the champion is obvious, but in many cases the task is far from trivial. This effort, which should be undertaken primarily by scholars intimately familiar with the language and the sociolinguistic/dialect situation, requires only user-level knowledge of language technology. This is very different from our recommendations for vital/borderline languages, to which we turn now.

4. Build

The build stage no longer considers all dialects and languages of the Indian Subcontinent, just those deemed vi-

tal/thriving in Stage 2. These we can hope to endow with a full computational toolchain composed of the following stages.

Tool	Effort
script	0.1
normalization	1
language detector	1
word list	2
bilingual dictionary	6
morphology	12
spellchecker	6

Table 3: Word-level tools

Table 3 lists the effort (in person-months) associated with building the tool or resource in question *after* the data-gathering phase is complete, but assuming significant online material is found, for if there is no material online the digital vitality of the language is in grave doubt. It is not assumed that the tools so obtained will be of quality comparable to those available for English (or for MT, say English-French). Nevertheless, such tools are already useful for a broad variety of purposes, and their incremental refinement and their higher level tools built on top of them are left to the last Phase of the ISLV project (see Section 5 below).

Special funds may be obtained from the NSF Documenting Endangered Languages program, the Endangered Languages Project, or other similar preservation efforts, but the only direct contribution of ISLV in this regard would be the recommendation that these dialects are indeed in need of such an effort. The main focus of building would initially be on the word-level technologies, including spellchecking (standardized orthography), stemming (prefix- and suffix-removal, but not necessarily deeper morphological analysis), glyph analysis, and building a common multilingual dictionary of basic vocabulary similar to Ács et al (2013). Such efforts obviously require better, engineer-level understanding of language technology, and may serve as a training ground for a new generation of native computational linguists.

It is the task of this phase to determine to what extent standard (two-tape) finite-state transducer technology is usable for providing cross-transliteration among the vital varieties on the one hand, and between the vital champions and their satellites on the other. Only a limited amount of parallel (synchronized) grammar writing is envisioned at this stage.

5. Apply

The main applications we envision extend the text and image-based work to speech (and if resources permit, to sign languages). Of particular interest are cross-language speech translators which do not assume users to have great familiarity with standard Hindi, text to speech systems capable of synthesizing speech in any vital language, and perhaps captioning systems that would extend the reach of broadcast operations. As can be seen from the following Table 4, the effort of building these tools is considerably larger, and it is only with computational linguists who are

both native speakers and already skilled in the design and application of the word-level tools that most of these can be attempted.

Tool	Effort
light parser	12
NER	6
OCR	12
ASR	12
MT	12

Table 4: Higher tools

For a significant subset (over half) of the thriving/vital languages, in particular Assamese, Bengali, Bodo, English, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Malayalam, Manipuri, Marathi, Nepali, Oriya, Panjabi, Sanskrit, Tamil, Telugu, and Urdu, there is already a concentrated effort under way, see the IndoWordNet site at <http://www.cfilt.iitb.ac.in/indowordnet>, and that a practical application with font transcoding and other critical parallelization technology is available in a 5-way parallel tourism site (Bengali, Hindi, Marathi, Tamil, Telugu), see <http://www.tdil-dc.in/sandhan>. The build phase can increment these systems for the languages not yet considered.

It is expected that orphaned dialects without a near champion can only be documented in the sense of heritage preservation, while dialects close to champions will participate in (sub)koine formation. The beneficial effects may even extend to some of the languages and dialects outside the Subcontinent. It will require a great deal of care to select the champion dialects, a matter of particular importance in Hungary, where several Roma dialects, some obviously close to main Indian languages, some less visibly so, are spoken. It is not expected that such dialects would be vital in and of themselves, but finding a champion they could attach to would significantly enhance their chances of digital survival.

6. Acknowledgement

Work supported in part by the European Union and the European Social Fund through project FuturICT.hu grant #TÁMOP-4.2.2.C-11/1/KONV-2012-0013.

7. References

- Judit Ács, Katalin Pajkossy, and András Kornai. 2013. Building basic vocabulary across 40 languages. In *Proceedings of the Sixth Workshop on Building and Using Comparable Corpora*, pages 52–58, Sofia, Bulgaria, August. Association for Computational Linguistics.
- András Kornai. 2013. Digital language death. *PloS one*, 8(10):e77056.
- Steven Krauwer. 2003. The Basic Language Resource Kit (BLARK) as the first milestone for the language resources roadmap. In *Proceedings of the 2003 International Workshop on Speech and Computers (SPECOM 2003)*, pages 8–15, Moscow, Russia.
- Paul Lewis, Gary Simons, and Charles Fennig, editors. 2013. *The Ethnologue*. Summer Institute of Linguistics.

Attila Zséder, Gábor Recski, Dániel Varga, and András Kornai. 2012. Rapid creation of large-scale corpora and frequency dictionaries. In *Proceedings to LREC 2012*, pages 1462–1465.

Appendix: digitally still languages

A'tong, A-Pucikwar, Adap, Adi, Adiwasi Garasia, Aer, Afghan Sign Language, Agariya, Ahirani, Ahom, Aimaq, Aimol, Aiton, Aka-Bea, Aka-Bo, Aka-Cari, Aka-Jeru, Aka-Kede, Aka-Kol, Aka-Kora, Akar-Bale, Allar, Alu Kurumba, Amri Karbi, Anal, Andaman Creole Hindi, Andh, Angami Naga, Ange, Apatani, Aranadan, Ashkun, Asuri, Athpariya, Attapady Kurumba, Badeshi, Bagheli, Bagri, Bahing, Balti, Bantawa, Baraamu, Bateri, Bauria, Bawm Chin, Bazigar, Belhariya, Bellari, Betta Kurumba, Bhadrawahi, Bhalay, Bharia, Bhatola, Bhatri, Bhattiyali, Bhaya, Bhilali, Bhili, Bhoti Kinnauri, Bhujel, Bhunjia, Biete, Bijori, Bilaspuri, Birhor, Bodo Gadaba, Bodo Parja, Bodo, Bondo, Bote-Majhi, Braj, Brokkat, Brokpake, Brokskat, Bugun, Buksa, Bumthangkha, Bundeli, Burushaski, Byangsi, Camling, Car Nicobarese, Central Nicobarese, Central Pashto, Chalikha, Chamari, Chambeali, Chang Naga, Changthang, Chantyal, Chaudangsi, Chaura, Chenchu, Chepang, Chhintange, Chhulung, Chillisso, Chinali, Chiru, Chitkuli Kinnauri, Chittagonian, Chitwania Tharu, Chocangacakha, Chodri, Chokri Naga, Chothe Naga, Chug, Chukwa, Churahi, Dakpakha, Dameli, Dandami Maria, Dangaura Tharu, Darai, Darlong, Darmiya, Deccan, Degaru, Dehwari, Deori, Desiya, Dhanki, Dhanwar, Dhatki, Dhimal, Dhodia, Dhundari, Digaro-Mishmi, Dimasa, Dolpo, Domaaki, Dotyali, Dubli, Dumi, Dungmali, Dungra Bhil, Dura, Duruwa, Dzalakha, Eastern Balochi, Eastern Gorkha Tamang, Eastern Gurung, Eastern Magar, Eastern Meohang, Eastern Muria, Eastern Parbate Kham, Eastern Tamang, Eravallan, Far Western Muria, Gadaba, Gadaba, Gaddi, Gade Lohar, Gahri, Galo, Gamale Kham, Gamit, Gangte, Garhwali, Garo, Gata', Gawar-Bati, Ghandruk Sign Language, Ghera, Goaria, Godwari, Gondi, Gongduk, Gowlan, Gowli, Gowro, Grangali, Gurgula, Hajong, Harijan Kinnauri, Haroti, Haryanvi, Hazaragi, Helambu Sherpa, Hinduri, Hmar, Ho, Holiya, Hrangkhoh, Hruso, Humla, Idu-Mishmi, Indian Sign Language, Indo-Portuguese, Indus Kohistani, Inpui Naga, Irula, Ishkashimi, Jad, Jadvani, Jandavra, Jangshung, Jarawa, Jatapu, Jaunsari, Jennu Kurumba, Jerung, Jhankot Sign Language, Jirel, Juang, Jumla Sign Language, Jumli, Juray, Kabutra, Kachari, Kachi Koli, Kadar, Kagate, Kaikadi, Kaike, Kalaktang Monpa, Kalami, Kalanadi, Kalasha, Kalkoti, Kamar, Kamviri, Kanashi, Kanauji, Kangri, Kanikkaran, Kanjari, Kannada Kurumba, Karbi, Kathoriya Tharu, Kati, Katkari, Kayort, Khaling, Khamba, Khamyang, Khandesi, Kharam Naga, Kharia Thar, Kharia, Khengkha, Khetrani, Khezha Naga, Khi-amniungan Naga, Khirwar, Khoibu Naga, Kinnauri, Koch, Kochila Tharu, Koda, Kodaku, Kodava, Kohistani Shina, Koi, Koiheng, Kol, Kom, Konyak Naga, Korku, Korlai Creole Portuguese, Koro, Korra Koraga, Korwa, Kota, Koya, Kudiya, Kudmali, Kui, Kullu Pahari, Kulung, Kumaoni, Kumarbhag Paharia, Kumbaran, Kumhali, Kundal Shahi, Kunduvadi, Kupia, Kurichiya, Kurmukar, Kurtokha, Ku-

rukha, Kusunda, Kutang Ghale, Ladakhi, Lahnda, Lahul Lohar, Lakha, Lambadi, Lambichhong, Lamkang, Lasi, Layakha, Lepcha, Lhokpu, Lhomi, Liangmai Naga, Limbu, Lingkhim, Lish, Loarki, Lodhi, Lohorung, Loke, Lotha Naga, Lui, Lumba-Yakkha, Lunanakha, Lyngngam, Magahi, Mahali, Mahasu Pahari, Majhi, Majhwar, Mal Paharia, Mala Malasar, Malankuravan, Malapandaram, Malaryan, Malavedan, Malvi, Manangba, Manda, Mandeali, Manna-Dora, Mannan, Mao Naga, Mara Chin, Maram Naga, Maria, Maring Naga, Marma, Marwari, Marwari, Marwari, Mawchi, Megam, Memoni, Merwari, Mewari, Mewati, Miju-Mishmi, Mina, Mirgan, Mirpur Panjabi, Mising, Mixed Great Andamanese, Mogholi, Monsang Naga, Moyon Naga, Mru, Mudu Koraga, Muduga, Muggom, Mukha-Dora, Mullu Kurumba, Munda, Mundari, Munji, Musasa, Muthuvan, Mzieme Naga, Na, Naaba, Nachering, Nagarchal, Nahali, Nahari, Nar Phu, Nefamese, Nepalese Sign Language, Nepali Kurux, Nepali), Nihali, Nimadi, Nocte Naga, Noiri, Norra, Northeast Pashayi, Northern Ghale, Northern Gondi, Northern Hindko, Northern Pashto, Northern Rengma Naga, Northwest Pashayi, Northwestern Kolami, Northwestern Tamang, Nubri, Nupbikha, Nyenkha, Nyishi, Od, Oko-Juwai, Olekha, Oraon Sadri, Ormuri, Pahari-Potwari, Pahlavani, Paite Chin, Pakistan Sign Language, Paliyan, Palpa, Palya Bareli, Panchpargania, Pangwali, Paniya, Pankhu, Pao, Parachi, Pardhan, Pardhi, Parenga, Parkari Koli, Parsi, Pathiya, Pattani, Pauri Bareli, Pengo, Phake, Phalura, Phangduwali, Phom Naga, Phudagi, Pnar, Pochuri Naga, Porja, Poumei Naga, Powari, Prasuni, Puimei Naga, Puma, Purik, Puroik, Purum Naga, Purum, Rabha, Rajbanshi, Raji, Rajput Garasia, Ralte, Rana Tharu, Rangkas, Ranglong, Rathawi, Rathwi Bareli, Raute, Ravula, Rawat, Reli, Riang, Rongmei Naga, Rongpo, Ruga, Saam, Sadri, Sajalong, Sakachep, Sambalpuri, Sampang, Samvedi, Sanglechi, Sangtam Naga, Sansi, Sartang, Sauria Paharia, Savara, Savi, Seke, Sentinel, Shekhawati, Shendu, Sherdukpen, Sherpa, Sheshi Kham, Shina, Sholaga, Shom Peng, Shumashti, Shumcho, Sikkimese, Simte, Sindhi Bhil, Singpho, Sirmauri, Sonha, Sora, Southeast Pashayi, Southeastern Kolami, Southern Ghale, Southern Gondi, Southern Hindko, Southern Nicobarese, Southern Rengma Naga, Southern Uzbek, Southern Yamphu, Southwest Pashayi, Southwestern Tamang, Spiti Bhoti, Sri Lankan Creole Malay, Sri Lankan Sign Language, Stod Bhoti, Sumi Naga, Sunam, Sunwar, Surgujia, Surjapuri, Tagin, Tangchangya, Tangkhul Naga, Tarao Naga, Tawang Monpa, Teressa, Thachanadan, Thado Chin, Thakali, Thangal Naga, Thangmi, Thudam, Thulung, Tichurong, Tilung, Tinani, Tippera, Tirahi, Tiwa, Toda, Torwali, Toto, Tregami, Tshangla, Tsum, Tukpa, Turi, Turung, Tutsa Naga, Ullatan, Urali, Ushojo, Usui, Vaagri Booli, Vaghri, Vaiphei, Varhadi-Nagpuri, Varli, Vasavi, Veddah, Vishavan, Waddar, Wadiyara Koli, Wagdi, Waigali, Wakhi, Waling, Walungge, Wambule, Wancho Naga, Waneci, War-Jaintia, Warduji, Wayanad Chetti, Wayu, Western Balochi, Western Gurung, Western Magar, Western Meohang, Western Muria, Western Parbate Kham, Western Tamang, Wotapuri-Katarqalai, Yakha, Yamphu, Yidgha, Zangskari, Zeme Naga.

Augmenting Dravidian WordNet with Context

S. Rajendran and S. Arulmozi

Amrita Vishwa Vidyapeetham, Coimbatore and University of Hyderabad, Hyderabad
(rajushush, arulmozi)@gmail.com

Abstract

It is difficult to interpret the meaning of a lexical item without context. WordNet lists different senses of a word and provides definition and usage example for each sense. But like any sense enumerative lexicon it also does not provide any mechanism for the novel usage of a word. The polysemy found in verbs and adjectives convincingly tell us that we have to augment WordNet with context. Such mechanism will help us to condense senses listed under a word and allow us to interpret the senses of a word creatively or generatively.

Keywords: lexicon, synset, polysemy

1. Introduction

WordNet as we understand is made up of synsets which are linked to each other by lexical and semantic relations in the background of ontology. Each synset represents a concept or sense and the sense is given as a description along with usage examples. For an end user word net serves both as a thesaurus as well as a dictionary. A user by typing a word in the interface slot can have a list of all the senses for the word. English WordNet for example lists 35 senses for the word *go* which includes 4 nominal senses, 30 verbal senses and one adjectival sense. Hindi on the other hand lists 2 nominal senses, 16 verbal senses and 2 adjectival senses for the word *chalna* 'go'. Tamil lists 9 verbal senses for the word *poo* 'go'. There is no guarantee that only these are the possible senses for the word under consideration. As we know language is dynamic and not static. So there is always a possibility of expansion of the meaning of a word (i.e. addition of new senses) as the word may be used in new contexts. A static list of senses cannot capture this meaning expansion or generative use of words. The senses are also not compartmentalized; they are overlapping with one another. The lexicon which lists the senses of words can be called sense enumerative lexicon (SEL). SEL may not be able to capture the dynamic use of a word. It is in this respect argued in this paper that WordNet need to be complemented or augmented by a mechanism of condensing the senses listed under a word in the WordNet and providing a mechanism for interpreting novel senses in new contexts in which the word is being used.

Pike Vossen (2001) points out the need for condensing meaning in WordNet. He states.

“The matching of meanings across the WordNets makes it necessary to account for polysemy in a generative way and to establish a notion of equivalence at a more global level.” A context sensitive framework for lexical ontology like WordNet has been proposed by Velae and Hao (2007).

This paper is purely a theoretical one based on certain assumptions and there by points out or proposes a methodology to augment WordNet.

2. Limitations of Sense Enumerative Lexicon

Pustejovsky (1995:5) who argues for a generative framework for a lexicon points out that lexical semantics should address the following issues:

- (a) Explaining the polymorphic nature of language;
- (b) Characterizing semanticity of natural language utterances;
- (c) Capturing the creative use of words in novel contexts;
- (d) Developing a richer, co-compositional semantic representation.

SELs are inadequate to account for the description of natural language semantics. Pustejovsky points out that there are three basic arguments showing the inadequacies of SELs for the semantic description of language (Pustejovsky, 1995:39).

- (1) THE CREATIVE USE OF WORDS: Words assume new senses in novel contexts.
- (2) THE PERMEABILITY OF WORD SENSES: Word senses are not atomic definitions but

overlap and make reference to other senses of the word.

- (3) THE EXPRESSION OF MULTIPLE SYNTACTIC FORMS: A single word sense can have multiple syntactic realizations.

Each of these consideration points to the inability of sense enumerative models to adequately express the nature of lexical knowledge and polysemy. Taken together, it would seem that the frameworks incorporating SELs are poor models of natural language semantics. A word may have contrastive or complementary senses. SEL lists contrastive senses as belonging to different words (i.e. under separate words) and complementary senses as belonging to the same word (i.e. under same word). Pustejovsky (1995: 38) restate the SEL's account of contrastive and complementary senses as follows:

A Lexicon L is a Sense Enumeration Lexicon if and only if for every word w in L , having multiple senses s_1, \dots, s_n associate with that word, then:

- (i) if s_1, \dots, s_n are contrastive senses, the lexical entries expressing these senses are stored as w_{s_1}, \dots, w_{s_n} .
(ii) if s_1, \dots, s_n are complementary senses, the lexical entry expressing these senses is stored as $w\{s_1, \dots, s_n\}$.

Every ambiguity is either represented by (i) or (ii) above.

Though Pustejovsky points out the advantage of this model of lexical description, he also points out that the SEL model is inadequate for the purpose of linguistic theory.

3. Problem of Polysemy in Verbal Semantics

We are not going to adopt Pustejovsky's (1919) model of generative lexicon for our purpose. We are planning for a different strategy that will suit WordNet and there by the contexts responsible for different senses of a particular word can be represented.

We are going to take up verbal polysemy to start with. As Tamil WordNet is only at its infant stage, we are going to make use of a representative SEL for Tamil (i.e. *kriyaavin taRkaalat tamiz akaraathi* (KTTA) (Dictionary of Contemporary Tamil) to serve our purpose. We will also make use of Generative Lexicon for Tamil (in manuscript form) written by

Rajendran under a UGC sponsored project (Rajendran, 2010).

If we look at KTTA, we find out that the number of senses enumerated under a verb vary from three to thirty approximately. Some verbs like *aTi* 'beat' and *pooTu* 'drop' acquire an enormous list of senses as they can collocate with a number of nouns forming different verbal senses. The different senses interpreted for them is based on the object noun with which they collocate with (for example *kaapi aTi* 'copy', *accu aTi* 'print' etc.). Such verbs behave like light verbs. So they show various senses based on the object-noun with which they collocate with. If we go through the dictionary and analyse the different senses listed under each verb, we will come to know that the context represented by the arguments of the verb (such as subject or agent-argument, object or patient-argument, indirect-object-argument, instrument-argument, location-argument) play a vital role in the interpretation of different senses for the concerned verbs. For the sake of illustration and to discuss the issue at hand a less polysemous verb *uTai* 'break' has been taken as an example. The different senses denoted by the verb are presented in the Table-1 along with the sense description and usage examples. The usage examples are analyzed for argument structure of the concerned verb.

4. Problem of Meaning Interpretation of Compound Verbs

There are compound verbs in Tamil which are formed from a base by the addition of a verb which function as the verbalizer or whose function is to verbalize the base. The bases are generally nouns. Even a verb can be compounded with a verbalizing verb to form another verb. There are a number of verbs which are used to form verbs from nouns. Not all nouns can be added to a verbalizer and conversely not all verbalizers can be added to a noun; only a closed set of nouns can be collocated with a particular verbalizer. The compounds could be overlapping in their meaning as same nouns can be collocated with overlapping group of verbs. This leads to synonymy among compound verbs. Though the formation of verbs from $N + V$ combination is a productive process, the nouns involved in the formation of compound verbs with reference to a particular verbalizer appear to be a closed set rather than an open set. But it is possible to recruit new members to a closed set which makes the process productive.

Table 1: Different senses of the verb *uTai*

S.No.	Sense	Usage examples	Arguments	
			Subject	Object
1	துண்டாதல், பிளத்தல் 'break; split'	1. திருடன் பூட்டை உடைத்து உள்ளே நுழைந்திருக்கிறான். 'The thief has entered the house by breaking open the lock'	திருடன் 'thief'	பூட்டு 'lock'
2	(கட்டப்பட்டிருப்பதை அல்லது ஒட்டப்பட்டிருப்பதை) பிரித்தல், மூடியாகப் பொருத்தப்படிருப்பதை திறத்தல் 'break open (a bundle by snapping the string tied around), open (an envelope, a bottle, etc.)	1. துணிக்கட்டை உடைத்துப் ஒவ்வொன்றாக வெளியே எடுத்து விலை போட்டார். 'He opened the cloth bundle, took out the cloth and wrote the prices' 2. தபாலில் வந்த கடித்ததை உடைத்துப் படித்தார். He opened the envelope of the letter and read it. 3. அவருக்குச் சோடா உடைத்துக் கொடு 'Open the soda bottle and give him'	அவர் 'he' அவர் 'he' நீ 'you'	துணிகட்டு 'cloth bundle' கடிதம் 'letter' சோடா 'soda'
3	(கட்சி நிறுவனம் போன்றவற்றைப் பிளத்தல், பிரிவுபடுத்துதல் Split; break up (a party, an organization, etc)	கூட்டுறவு சங்கத்தை உடைக்க அவர் செய்த முயற்சிகள் வீணாயின. 'The efforts he had taken to break up the society failed'	அவர் 'he'	கூட்டுறவு சங்கம் 'society'
4	(ரகசியத்தை, உண்மையை) வெளியாக்குதல் Make public (a secret, hidden facts, etc.); disclose.	1. அவர் யாருக்கும் தெரியாமல் வைத்திருந்த விஷயத்தை இப்படி உடைக்கலாமா? 'How can you disclose the secret he has kept to himself?' 2. உண்மையை உடைத்துச் சொல்லிவிட வேண்டியதுதான் 'I have to disclose the secret.'	நீ 'you' நான் 'I'	விஷயம் 'matter' உண்மை 'truth'
5.	(கோடாலியால் மரத்துண்டுகளைப்) பிளத்தல் Split (logs)	விறகு உடைக்க ஆள் வரவில்லை 'The person to split the log has not come so far.'	ஆள் 'person'	விறகு 'log'

Table-1 (above) reveals the fact that the object-noun of the verb determines the different senses assigned to the verb.

There are thirty nine verbs which can be claimed to function as verbalizers to form compound verbs from nominal bases.

Table-2: Verbalizers

S.No	Verbalizers with core meaning	Examples of Compound verbs in which the verbalizers form a part
1	<i>ati</i> 'beat'	<i>kan</i> 'eye' + <i>ati</i> > <i>kannati</i> 'wink'
2	<i>atai</i> 'get'	<i>mutivu</i> 'end' + <i>atai</i> > <i>mutivatai</i> 'come to an end'
3	<i>ali</i> 'give'	<i>paricu</i> 'prize' + <i>ali</i> > <i>paricali</i> 'award'

4	<i>aku</i> 'become'	<i>veli</i> 'outside + <i>aku</i> > <i>veliyaku</i> 'come out'
5	<i>akku</i> 'produce'	<i>coru</i> 'cooked rice' + <i>akku</i> > <i>corakku</i> 'cook rice'
6	<i>atu</i> 'move'	<i>kuttu</i> 'drama' + <i>atu</i> > <i>kuttatu</i> 'act'
7	<i>attu</i> 'swing'	<i>cir</i> 'orderliness' + <i>attu</i> > <i>cirattu</i> 'tend lovingly'
8	<i>arru</i> 'perform'	<i>pani</i> 'work' + <i>arru</i> > <i>paniyarru</i> 'work'
9	<i>itu</i> 'put'	<i>parvai</i> 'look' + <i>itu</i> > <i>parvaiyitu</i> 'inspect'
10	<i>uru</i> 'obtain'	<i>kelvi</i> 'hearsay' + <i>uru</i> > <i>kelviyuru</i>

		'get to know'
11	<i>uruttu</i> 'trouble'	<i>tunpam</i> 'suffering' + <i>uruttu</i> > <i>tunpuruttu</i> 'cause suffering'
12	<i>uttu</i> 'give'	<i>ninaivu</i> 'remembrance' + <i>uttu</i> > <i>ninaivuttu</i> 'remind'
13	<i>etu</i> 'take'	<i>oyvu</i> 'rest' + <i>etu</i> > <i>oyvetu</i> 'take rest'
14	<i>eytu</i> 'obtain'	<i>maranam</i> 'death' + <i>eytu</i> > <i>maranameytu</i> 'die'
15	<i>el</i> 'accept'	<i>patavi</i> 'position' + <i>el</i> > <i>pataviyel</i> 'take office'
16	<i>eru</i> 'rise'	<i>cutu</i> 'heat' + <i>eru</i> > <i>cuteru</i> 'become hot'
17	<i>erru</i> 'raise'	<i>veli</i> 'outside' + <i>erru</i> > <i>veliyerru</i> 'expel'
18	<i>kattu</i> 'tie'	<i>itu</i> 'compensation' + <i>kattu</i> > <i>itukattu</i> 'make up'
19	<i>kattu</i> 'show'	<i>acai</i> 'desire' + <i>kattu</i> 'show' > <i>acaikattu</i> 'lure; tempt'
20	<i>kuru</i> 'say'	<i>puram</i> 'back' + <i>kuru</i> > <i>purankuru</i> 'backbite'
21	<i>kotu</i> 'give'	<i>peeccu</i> 'conversation' + <i>kotu</i> > <i>peccukkotu</i> 'initiate a talk'
22	<i>kol</i> 'get'	<i>totarpu</i> 'contact' + <i>kol</i> > <i>totarpu kol</i> 'contact'
23	<i>cey</i> 'do'	<i>vicaranai</i> 'investigation' + <i>cey</i> > <i>vicaranai cey</i> 'investigate'
24	<i>col</i> 'say'	<i>kol</i> 'lie' + <i>col</i> > <i>kol col</i> 'tell tale'
25	<i>tattu</i> 'pat'	<i>mattam</i>

		'substandard' + <i>tattu</i> > <i>mattam tattu</i> 'degrade'
26	<i>patu</i> 'experience'	<i>vetkam</i> 'shyness' + <i>patu</i> > <i>vetkappatu</i> 'feel shy'
27	<i>patuttu</i> 'cause to experience'	<i>tunpam</i> 'suffering' + <i>patuttu</i> > <i>tunpappatuttu</i> 'cause to suffer'
28	<i>pannu</i> 'do'	<i>yocanai</i> 'thinking' + <i>pannu</i> > <i>yocanai pannu</i> 'think'
29	<i>par</i> 'see'	<i>vevu</i> 'spying' + <i>par</i> > <i>vevupar</i> 'spy'
30	<i>piti</i> 'catch'	<i>atam</i> 'obstinacy' + <i>piti</i> > <i>atampiti</i> 'become obstinate'
31	<i>puri</i> 'do'	<i>manam</i> 'marriage' + <i>puri</i> > <i>manampuri</i> 'marry'
32	<i>peru</i> 'get'	<i>oyvu</i> 'rest' + <i>peru</i> > <i>oyvu peru</i> 'retire (from service)'
33	<i>po</i> 'go'	<i>coram</i> 'adultery' + <i>po</i> > <i>corampo</i> 'commit adultery'
34	<i>potu</i> 'drop'	<i>cattam</i> 'sound' + <i>potu</i> > <i>cattam poTu</i> 'shout'
35	<i>muuTTu</i> 'make'	<i>kopam</i> 'anger' + <i>muuttu</i> > <i>kopamuttu</i> 'cause anger'
36	<i>va</i> 'come'	<i>valam</i> 'right' + <i>va</i> > <i>valamva</i> 'go round'
37	<i>vanku</i> 'get'	<i>velai</i> + <i>vanku</i> > <i>velaivanku</i> 'extract work'
38	<i>vitu</i> 'leave'	<i>muccu</i> 'breath' + <i>vitu</i> > <i>muccuvitu</i> 'breathe'
39	<i>vai</i> 'keep'	<i>ataku</i> 'pledge' + <i>vai</i> > <i>atakuvai</i> 'pledge'

It has to be noted here that all the verbalizing verbs are native Tamil words. Not all the verbs listed above are actually used as verbalizers. The number of compound verbs formed from each verbalizer also varies.

As inferred from the table the verbalizers or the light verbs depend on the preceding noun for the interpretation of the compounded meaning. Some of compounds formed thus

find their place in the Tamil dictionary. But most of them are not listed in the dictionary as the process of this formation is productive.

The question raised here is how are we going to list these verbs in the WordNet? Here again we need a generative mechanism to capture the polysemy in the light verbs.

5. Problem of Meaning Interpretation of Adjectives

A set of adjectives depend upon the nouns they attribute for the interpretation of their meaning. The following examples in Tamil will illustrate this issue:

paccai poy (green lie) ‘extreme lie’
paccai irattam (green blood) ‘raw blood’
paccai kaaykaRi (green vegetable) ‘raw vegetable’
paccai arici (raw rice) ‘raw rice’
paccai miLakaay ‘green chilly’
paccai taNNiir (green water) ‘water (in general as opposed to cold water and hot water)’

Though *nalla* ‘good’ can attribute any noun, its interpretation depends on the noun which follows it.

nalla peenaa ‘good pen’
nalla peN ‘good woman’
nalla katti ‘good knife/sharp knife’
nalla aaciriyar ‘good teacher/efficient teacher’

Here again we need a generative mechanism for the interpretation of adjectives.

6. Pustejovsky’ Approach

A formalized structure of contexts is not given for a dictionary or lexicon. Of course the dictionaries in a formal way try to tell its users the context of a particular usage. The context has been widely talked about in the field of semantics, but an explicit account of it for the interpretation of senses has eluded so far.

It is Pustejovsky who has given a formal representation to account for different senses for a word. Pustejovsky (1995:61) characterize a generative lexicon as a computational system involving at least the following levels of representation:

1. ARGUMENT STRUCTURE: Specification of number and type of logical arguments
2. EVENT STRUCTURE: Definition of the event type of an expression and its sub-eventual structure
3. QUALIA STRUCTURE: A structural differentiation of the predicative force for a lexical item
4. LEXICAL INHERITANCE STRUCTURE: Identification of how a lexical structure is related to other structures in this type of lattice

Pustejovsky (1995:76, 2001:56) assumes that word meaning is structured on the basis of four generative factors, or qualia roles, that capture how humans understand objects and relations in the world and provide the minimal explanation for the linguistic behaviour of lexical items.

CONSTITUTIVE: The relation between an object and its constituent parts.

FORMAL: The basic category that distinguishes the object within a larger domain.

TELIC: The object’s purpose and function

AGENTIVE: Factors involved in the object’s origin or “coming into being.”

A set of generative devices connects these four levels, providing for the compositional interpretation of words in context. Included in these generative operations are the following semantic transformations, all involving well-formedness conditions on type combinations:

TYPE COERCION: where a lexical item or phrase is coerced to a semantic interpretation by a governing item in the phrase, without change of its syntactic type.

SELECTIVE BINDING: where a lexical item or phrase operates specifically on the substructure of a phrase, without changing the overall type in the composition.

CO-COMPOSITION: where multiple elements within a phrase behave as functors, generating new non-lexicalized sense for the words in composition. This also includes cases of underspecified semantic forms becoming contextually enriched, such as manner co-composition, feature transcription, and light verb specification.

7. An Alternative Approach for Tackling Polysemy

Adopting the methodology dealt by Pustejovsky to account for the polysemous structure found in the WordNet is difficult. If we again look at the table discussed above, we may presume that the semantic features of the nominal object determine the senses to be enumerated. Probably it may be inferred that a set of items belonging to a domain of objects gives one sense to the verb and another set of items of object another meaning and so on. If it is possible to link these domains in an ontological tree, we may be able to infer the difference in the nominal object and there by assign different senses to the concerned verb. The nearness to the ontological hierarchy

(which again is difficult to measure) may give rise to overlapping of senses. The distance in the ontological hierarchy may tell us how much the senses are apart. This will help us decide whether a particular group of sense is complementary to one another demanding a common entry in the lexicon or contrastive demanding separate entries. Probably we require a fine-grained ontological tree. This methodology may reduce the subjectivity in grouping senses under one or more entries in a lexicon. The context provided by the ontological tree can be exploited for the interpretation or generation various senses for a particular word.

8. Conclusion

Human categorization is neither a binary nor a context-free process. Rather, the criteria that govern the use and recognition of certain concepts may be satisfied to different degree in different contexts. Much work remains to be done on the current framework with the aim of a more formal treatment of how our approach serves to augment WordNet (or similar resources) with concept descriptions that can be used both to categorize in context and to reason about those categorizations. WordNet is itself a little more than a classification hierarchy, and the conceptual functions we assign to its lexical entries serve much the same purpose (i.e. categorization and introspective reasoning).

References

- Bouquet, P., et.al. (2003). C-OWL: Contextualizing Ontologies. In: *Proceedings of the 2nd International Semantic Web Conference*. LNCS, Vol.2870, pp. 164-76. Springer verlag.
- Brijesh Bhat and Pushpak Bhattacharyya. (2012). Domain Specific Ontology Extractor for Indian Languages, 10th Workshop on Asian Language Resources, *COLING 2012*. Pp.75-84.
- Fellbaum, C. (Ed.). (1998). *WordNet: An Electronic Lexical Database*. Cambridge: The MIT Press.
- Levin, B. (1993). *English Verb Classes and Alternations: A Preliminary Investigation*. Chicago: University of Chicago Press.
- Pease, Adam. (2011). *Ontology: A Practical Guide*. Angwin, CA: Articulate Software Press.
- Pustejovsky, J. (1991). *The Generative Lexicon*. Cambridge: The MIT Press.
- Rajendran, S. (2000). *taRkaalat tamiz coRkkaLanjiciyam* (Thesaurus for Contemporary Tamil). Thanjavur: Tamil University Press.
- Rajendran, S. (2000). Strategies in the Formation of Compound Verbs in Tamil. *International Journal of Dravidian Linguistics*. Vol.29:2, pp.107-26.
- Rajendran, S. (2010). *Creating Generative Lexicon from MRDs in Tamil*. University Grants Commission Project Report. Thanjavur: Tamil University.
- Subramanian, P. R. (1992/1998). *kriyavin taRkaalat tamiz akaraati*. Chennai: Cre-A.
- Vossen, Piek. (2001). Condensed Meaning in EuroWordNet. In: Pierrette B & Frederica Busa (Ed). *The Language for Word Meaning*. Cambridge: CUP.
- Veale, Tomy & Hao Yanfen. (2007). A Context-Sensitive Framework for Lexical Ontologies. In: *The Knowledge Engineering Review*. Vol. 23:1, pp. 101-15.

A Study on Causal Relations and its Automatic Identification in Tamil

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

Abstract

The objective of the present work is to identify the cause-effect expressions in Tamil text. We have classified the causal markers into different types for computational purposes. The linear order of cause-effect markers and arguments are explained with examples. Tamil corpora consisting of 31,741 sentences were annotated manually for this task. To overcome the structural interdependencies existing in cause-effect relations, we came up with the separate set of features for each type and the type specific models were generated. We have introduced the Sliding window type specific testing approach. Post-processing using linguistic and heuristic rules improves the system performance. We have conducted performance distribution and ten-fold cross validation experiments and the results are encouraging.

Keywords: cause-effect, causal marker, type specific, structural dependencies

1. Introduction

The cause is an event that is the reason for the other event called an effect to happen. Cause-Effect relations have been studied through two different approaches - Cue-based and statistical. Cue-based approach is to start with discourse markers or cue phrases and search for a pattern while statistical approach is to use machine learning techniques to identify relations.

(Khoo et al. 1998) developed an automatic cue-based method, using linguistic cues and pattern matching for identifying and extracting explicitly mentioned cause-effect relations. (Girju & Moldovan, 2002) provide a syntactic and semantic classification of cause-effect lexico-syntactic patterns for automatic detection and extraction of causal relationships in English texts.

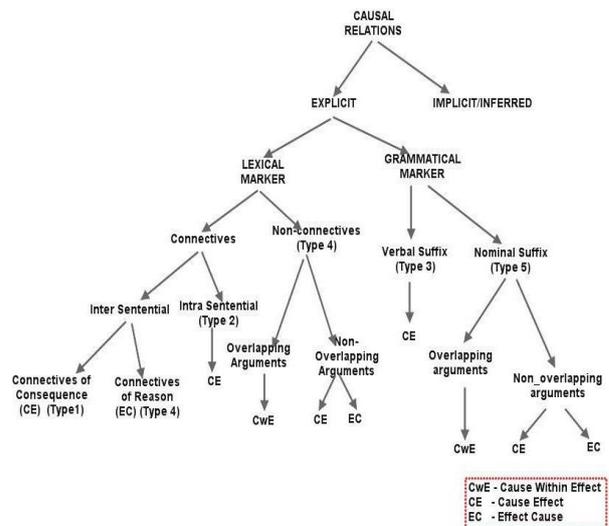
(Saito et al. 2006) proposed a method to identify discourse relations between two successive sentences. (Pechsiri et al. 2008) attempted to automatically extract Know-Why relations from Thai documents on the website to support Question-Answering system for disease treatment. (Blanco et al. 2008) attempted a supervised method for detection of explicit cause-effect relations from open domain text.

(Sobha et. al 2010) identified and classified the causal makers in Tamil. They also described the semantic and syntactic properties of cause-effect relations in such a way that would be useful for IR and QA systems. (Menaka et. al 2012) developed the cause-effect relation identification system to automatically identify the causal relations in Tamil text using CRF. From the error analysis they claim that the errors can be attributed to structural interdependencies between and within the cause-effect relations. To overcome those structural dependencies in (Menaka et. al 2012), we worked further and came up with type-specific models.

2. Hierarchical Classification of Cause-effect Relations

The cause-effect relation or the causal relation is a semantic relation between two events. The cause is the event, E1 that causes the event, E2. In other words E1 is the reason for E2 to occur. The event, E2 is the effect which is the result or consequence of the cause. From (Menaka et. al 2012), we have pictured the hierarchical classification of cause-effect markers which is in figure 1. The cause-effect relation can be broadly classified as explicit and implicit cause-effect relations based on the presence or absence of a marker.

Figure1: Hierarchical Classification of Cause-effect Relations in Tamil



2.1. Implicit Cause-Effect Relation

An Implicit cause-effect relation is typically inferred in two adjacent sentences or clauses without any explicit grammatical or lexical signal to make the cause-effect connection. In Tamil, this can happen in two adjacent sentences (Ex.1).

- (1) mazai pey-t-atu. tuNikaL kaaya-villai.
rain rain-Pst-3sn clothes dry-Neg
'It rained. The clothes did not dry.'

2.2. Explicit Cause-Effect Relation

Explicit cause-effect relations can be marked by a grammatical marker or a lexical marker. Explicit cause-effect markers can be intra-sentential or inter-sentential. Intra-sentential markers are mostly grammatical markers. Grammatical markers get inflected with a noun or a verb. In Tamil, the grammatical marker can be a verbal suffix or a nominal suffix. Verbal suffixes are -ataal -ati^naal -ata^naal -apaTiyaal -amaiyaal and -aamaiyaal.

The only nominal suffix is -aal (Ex.2). Inter-sentential discourse connectives like ata^naal, ita^naal, aa^napaTiyaal, aakaiyaal, aakaiyi^naal, aatalaal, aakavee, e^navee are lexical markers denoting cause-effect. Certain inter-sentential connectives like ee^ne^nil and ee^ne^nRaal behave differently from others by explaining the reason for an event rather than explaining the consequence of an event. Other non-connective lexical markers are kaaraNam, kaaraNamaaka and kaaraNattaal and they occur in complex patterns.

- (2) [avar maaraTaipp-aal]C [kaalamaa^naar]E
he heartattack-Cause expired.
'He died of heart attack.'

3. Computational Classification of Cause-Effect Relations

Based on the hierarchical classification shown in (Figure. 1), we have classified the cause-effect markers into five types for computational purposes. Features used in developing the language models were extracted based on this computational classification.

3.1. Type 1 Markers

Inter-sentential connectives of consequence are classified as Type1 markers. They are lexical markers with arguments that span across sentences. Type1 markers are markers of consequence and it occurs in the sentence-initial position and the cause always precede the effect.

3.2. Type 2 Markers

Connectives that occur intra-sentential to conjoin two sentences are Type 2 markers. These markers are a subset of Type 1 markers, but they occur sentence-medial. They are always markers of consequence.

3.3. Type 3 Markers

Type 3 markers are verbal suffixes indicating cause-effect

relations. They are subordinating suffixes. The subordinate clause is the cause and the matrix clause is the effect. kaaraNattaal, one of the inflections of kaaraNam also falls under this type because it follows a verb (e.g. vanta kaaraNattaal) and behaves like a verbal suffix of cause-effect.

3.4. Type 4 Markers

Type 4 markers are non-connective lexical markers. Most of these markers occur in several syntactic patterns. So, this type is very challenging for machine learning. They have arguments that span across sentences. Inter-sentential connectives of reason have been combined with Type 4 markers.

3.5. Type 5 Markers

Type 5 markers are nominal suffixes. The nominal phrase containing the marker is the cause. The arguments can be overlapping or non-overlapping. Overlapping arguments have the cause contained within the effect. When the arguments are non-overlapping, the cause can precede or follow the effect. The arguments of these markers never span across sentences. Computational classification of causal marker is shown in Table 1.

Computational Classification Type	Cause-Effect Marker
Type 1	ata^naal ita^naal aakaiyaal aakaiyi^naal aatalaal/aatali^naal aa^napaTiyaal e^navee
Type 2	aakaiyaal/ aakaiyi^naal/ aatalaal/ aa^napaTiyaal
Type 3	ataal -ati^naal -ata^naal -apaTiyaal -amaiyaal and -aamaiyaal
Type 4	kaaraNam and its inflected forms, ee^ ne^nil ee^ne^nRaal
Type 5	-aal

Table 1: Computational Classification of Causal Marker

4. Corpus Annotation

Unlike English (Penn Discourse Tree Bank for English (Prasad et al., 2008)) and Hindi (Hindi Discourse Relation Bank for Hindi (Oza et al., 2009)) Tamil has no existing Discourse Tree Bank, so we have developed a suitable annotated corpus. We have chosen three different corpora. The first corpus is *akal viLakku*, a novel written by M. Varatharajan(1961). The second corpus is a historical novel, *chivakaamiyi^n chapatam*, authored by Kalki(1944). *kuRinchi malar*, a social novel written by N. Parthasarathi(1960), was chosen as the third Corpus. The corpus statistics is shown in Table 2.

Corpus	Corpus 1	Corpus 2	Corpus 3	Total
Total Sentences	9094	13635	9012	31741
Total Tokens	72968	166821	94851	72968
Total Single CE relations	376	702	269	1347
Total Multi CE relations	8	5	12	25

Table 2: Computational Classification of Causal Marker

5. Linear Order of Markers and Arguments

The linear order of the arguments and markers are not always the basic order of Arg1-Marker-Arg2. It varies depending on the marker and its properties.

Type 1 markers occur in sentence-initial, though there may be other connectives preceding them ((Ex. 3) has the contrast marker aa^naal before the cause-effect marker). They have their Arg1 preceding Arg2 in the canonical form (Ex.3). The cause-effect marker is always tagged as part of Arg2.

- (3) [ava^nai varavaittee^n]Arg1 . aa^naal [ata^naal
he come-Cause But because
ava^n keTTaa^n]Arg2.
he spoil
'I made him come. But, he got spoilt because of it.'

In Type 2, the Arg1 usually immediately precedes the marker. If there is agglutination, the marker is annotated as part of Arg1 (Ex.4). In all other cases, the marker is part of Arg2 (Ex.5).

- (4) [maturai pazamaiyaa^na uur-aakaiyaal]Arg1
Madurai old city-Cause
[pirachittiyaa^natu]Arg2
famous
'Because Madurai is an old city, it is famous.'
- (5) [maturai pazamaiyaa^na uur]Arg1.[aakaiyaal atu
Madurai old city Cause it
pirachittiyaa^natu]Arg2.
famous
'Madurai is an old city. So it is famous.'

Type 3 markers are subordinating suffixes. So the Arg1 immediately precede the marker. The arguments of subordinators are necessarily adjacent and show Arg1-Marker-Arg2 linear order (Ex.6).

- (6) [appaa cho^n^ntaal]Arg1 [naa^n poo^nee^n]Arg2.
father say-Cause I went
'I went because father said.'

Type 4 markers are lexical markers and therefore there is no fixed pattern to the arguments. kaaraNam has a dative/genitive NP and a nominative NP as arguments. The dative/genitive NP is usually the Arg2 and it precedes

the marker. The Arg1 may precede Arg2 or follow the marker. So, the linear order is either Arg1-Arg2-Marker (Ex.7) or Arg2-Marker-Arg1 (Ex.8) with the Marker attaching to Arg2.

- (7) [nii-taa^n]Arg1 [ataRkuk kaaraNam]Arg2
you-Emph that-Dat reason
'You are the reason for that.'
- (8) [ataRkuk kaaraNam]Arg2 [nii-taa^n]Arg1.
that-Dat reason you-Emph
'The reason for that is you.'

Type 5 marker is a nominal suffix and hence the marker is a part of Arg1, which is the nominal phrase it attaches to. Arg2 may follow Arg1 (Ex.9) or Arg2 may have discontinuous constituents on either side of Arg1 (Ex.10).

- (9) [mutukuk kaTTiy-aal]Arg1 [appaa
back boil-Cause father
varuntikkoNTiruntaar]Arg2.
suffer-Pst-Prg
'Because of boils on his back, father was suffering.'
- (10) [appaa [mutukuk kaTTiy-aal]Arg1
father back boil-Cause
varuntikkoNTiruntaar]Arg2.
suffer-Pst-Prg
'Father was suffering because of boils on his back.'

6. Identification of Causal Relations

6.1. Our Approach

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 (Lafferty et al., 2001). CRF++ (Kudo, 2005), an open source toolkit for linear chain CRFs, was used for the experiments performed. CRF++ is a simple, customizable, and open source implementation of Conditional Random Fields (CRFs) for segmenting or labeling sequential data.

6.2. Preprocessing Of Text

The corpus was preprocessed for sentence splitting, tokenizing, morph analysis, part-of-speech tagging (POS), chunking, pruning and clause tagging. The tokenizer splits the raw text into sentences based on cues like a period (.) or a question mark (?). These sentences are further split into tokens. Tamil Morphological Analyzer (Viswanathan et al., 2003) was used to find the root words. POS Tagger for Tamil (Arulmozhi & Sobha, 2006) was used to tag the Part of Speech for each token. Chunker for Tamil (Sobha & Vijay Sundar Ram, 2006) was used for identifying phrase. Connectives, Vocatives, Relative Participles, and Complements are marked.

For Type 4, the chunks and the clauses ending in the clitics taa^n or ee and those ending in dative or genitive case markers need to be marked. For Type 5, the verb roots with paradigms are needed to eliminate instrumental markers and the cases where verbs of ability are used.

Ontology-based disambiguation of the Type 5 marker -aal is necessary to disambiguate the causal sense from the instrumental sense in which the marker occurs.

6.3. Type-Specific Models

From our analysis, it was noted that features apply to certain cause-effect markers do not make any significant impact for other markers. So the type-specific models were created and the features used for machine learning were identified separately for each type of markers.

Four sets of the training corpus were made and in each of these sets, instances of cause-effect relations one type of marker were retained and the rest of the relations were ignored. Thus the overall problem was split into 4 sub-problems. More features were identified specific to each type and these were marked for each type separately. The order of testing of the boundaries depended on the type of the marker and it is shown in Table 3. Training models were created for each boundary for each type. So, considering the 4 boundaries and 4 types, there were 16 models created.

TYPE	Order of Boundary Identification Of Arguments
Type 1 & 2	ARG-START,Arg1-END, Arg1-START, Arg2-END
Type 3	Arg2-START,Arg1-END, Arg1-START, Arg2-END
Type 4	Arg1-START,Arg1-END, Arg2-END, Arg2-START
Type 5	Arg1-END,Arg1-START, Arg2-START, Arg2-END

Table 3: Order of Testing Boundaries

6.3.1. Feature Selection

The features identified for each of the types are listed in Table 4.

- For Type 1 and Type 2 model Potential Arg1 feature is previous clause and Potential Arg2 is current clause.
- For Type 3 model potential Arg1 was marked on the tokens in the current sentence occurring before the marker and on the marker. Potential Arg2 was marked on the tokens in the current sentence occurring after the marker.
- For Type 4 model Potential Arg1 was marked on the chunks and clauses ending in the case markers. Potential Arg2 was marked on the chunks and clauses ending in the clitics taa^n or ee.
- For Type 5 Markers potential Arg1 is first marked on the noun phrase containing the marker. The rest of the tokens in the sentence before the chunk and after the chunk are marked for potential Arg2.

ID	Features	TYPE 1 & 2	TYPE 3	TYPE 4	TYPE 5
A	Token	1	1	1	1
B	POS	1	1	1	1
C	Chunk	1	1	1	1
D	Word Suffix	1	1	1	1
E	Marker	1	1	1	1
F	Marker Type	1	1	1	1
G	Quotes	1	1	1	1
H	Connectives/ Vocatives	1	1	1	1
I	Complementizer/Relative Participle	1	1	1	1
J	Potential Arg2	1	1	1	1
K	Potential Arg1	1	1	1	1
L	Arg2-START	1	1	0	1
M	Arg1-END	1	1	1	1
N	Arg1-START	1	1	1	1
O	Arg2-END	0	0	1	0
P	Combination of b, k	1	1	1	0
Q	Combination of b, i and j	0	1	1	0
R	Combination of e, k	0	0	0	1
S	Combination of k, j	0	0	0	1
T	Combination of i, j	0	0	0	1

Table 4: Features Used For Each Type

6.3.2. Rule-based Post-Processing

After, the machine tags the boundaries, if there are one or more missing tag, certain linguistic and heuristic rules are used in the post-processing phase to obtain the complete causal relation. The rules identified for post-processing are given below.

6.3.2.1. Linguistic Rules

Some of the linguistic rules identified to improve the identification of the argument boundaries are presented below.

(1) If Arg1-END is marked at a certain index and the token at the next index is a clitic like taa^n or maTTum, then move the Arg1-END to the next token. If Arg2-START is at the next token, move it by one position. If Arg2-START is not at the next token, no action needs to be taken.

Rule 1:

If	i	=	Arg1-START
and	i+x	=	Arg1-END
and	i+x+1	=	Arg2-START
and	tokenat(i+x+1)	=	taa^n/maTTum
then	i+x	=	\varnothing
	i+x+1	=	Arg1-END
	i+x+2	=	Arg2-START

(2) If Arg1-START starts with the token e^nRu preceded by ending quotes, trace to the beginning of the quotes and then move to the beginning of the sentence and move Arg1-START to this position. Apply a similar rule to Arg2-START as well.

Rule 2:

If	i	=	Begin of Sentence
and	tokenat(i+w)	=	“
and	tokenat(i+x)	=	”
and	tokenat(i+x+1)	=	e^nRu
and	i+x+1	=	Arg1-START
then	i+x+1	=	\varnothing
	i	=	Arg1-START

(3) If Arg2 is not tagged at all and the marker is of Type 1, Type 2 or Type 3, then position the Arg2-START at the next token of Arg1-END and position Arg2-END at the end of the sentence containing Arg2-START.

Rule 3:

If	i	=	Arg1-START
and	i+w	=	Arg1-END
and	Arg2-START	=	?
and	Arg2-END	=	?
and	i+w+x	=	End of Sentence
then	i+w+1	=	Arg2-START
	i+w+x	=	Arg2-END

(4) If Arg1-END or Arg2-END ends in e^nRu or e^npatu when there is no agglutination with the verb of the inner clause (like vantaane^nRu) move the tag to the previous token.

Rule 4:

If	tokenat(i+w+1)	=	e^nRu
and	i+w+1	=	Arg1-END
then	i+w+1	=	\varnothing
	i+w	=	Arg1-END

(5) If there are several Arg1-STARTs for a particular Arg1-END, remove all those Arg1-STARTs which are not at the same level of quotes as Arg1-END. This rule can be applied to the other three combinations of multiple Arg1-ENDs for single Arg1-START, multiple Arg2-STARTs for single Arg2-END and multiple Arg2-ENDs for single Arg2-START.

6.3.2.1. Heuristic Rules

Few heuristic rules were added to the above set of linguistic rules to improve the performance. These rules boosted the identification of the START of an argument, if the END was already identified and vice versa.

(1) If Arg1-START is tagged and Arg1-END is missing, find the end of the sentence containing Arg1-START and mark it as Arg1-END. This end of the sentence is the token preceding quotes if Arg1-START is within quotes. Apply a similar rule for Arg2.

(2) If Arg1-END is tagged and Arg1-START is missing, go up to the beginning of the sentence containing Arg1-END and mark it as Arg1-START. This start of the sentence is the token after quotes if Arg1-END is within quotes. Apply a similar rule for Arg2.

(3) If Arg1-END and Arg2-START coincide, move Arg2-START to the next token. Similarly, if Arg2-END and Arg1-START coincide, move Arg1-START to the next token.

7. Experiments Performed

A set of experiments was performed where an improved set of features were used to train the model and type-specific, sliding window testing was done.

7.1. Sliding Window Testing

To overcome the problems associated with the interdependencies of cause-effect relations, a sliding window testing approach was made. In this approach, the testing of the whole test corpus is not done at the same time. Instead, an outer system was developed which looks for a marker, classified the marker into one of the types, takes a window of tokens for each of these markers and then performs testing on it. In essence, at any point of testing, only one marker and its context are taken into account by the machine learning system. The features marked based on the other markers in the vicinity or even the results or the argument boundaries marked for other markers do not interfere when the testing for a particular marker is done.

7.2. Individual Corpus Based Results

Type-specific sliding window testing followed by post-processing was done on models trained with an improved set of type-specific features. These experiments were conducted on the three corpora. The results for Corpus 1, Corpus 2 and Corpus 3 are shown in (Table. 5) where MACH O/P denotes the results for machine-tagged output and PP/OP denotes the final post-processed output. For all the three corpora, the post-processing using linguistic and heuristic rules have given a very good increase in the recall. Arg2-END seems to give the least results in all cases. It can be noted that post-processing has indeed increased the accuracy of identification of the individual boundaries.

CORPUS		Arg1-START		Arg1-END		Arg2-START		Arg2-END	
		MACH O/P	PP O/P	MACH O/P	PP O/P	MACH O/P	PP O/P	MACH O/P	PP O/P
CORPUS 1	PRE	79.36	79.74	91.47	92.27	87.50	89.39	70.89	79.52
	REC	75.51	78.32	87.50	88.27	82.03	87.76	68.49	77.86
CORPUS 2	PRE	80.52	82.55	94.35	94.37	85.79	86.96	79.72	80.00
	REC	79.49	82.91	92.74	93.16	73.80	87.34	75.55	80.35
CORPUS 3	PRE	81.88	84.41	93.06	93.73	82.86	85.66	78.52	79.00
	REC	80.20	84.98	91.47	91.81	72.24	85.05	75.44	79.00

Table 5: Results for Individual Corpus

TYPE	Arg1-START		Arg1-END		Arg2-START		Arg2-END	
	PRE	REC	PRE	REC	PRE	REC	PRE	REC
Type1 & 2	87.50	84.85	95.31	92.42	98.44	95.45	92.19	89.39
Type 3	58.76	91.94	72.62	98.39	69.41	95.16	61.18	83.87
Type 4	66.67	57.14	58.33	50.00	61.54	57.14	61.54	57.14
Type 5	60.50	78.26	72.95	96.74	58.82	80.46	54.62	74.71

Table 6: Results for Type Specific Models

7.3. Performance-distribution results

To understand the contribution of each type of markers to the overall system's performance, experiments were done to test markers of only a particular type. This is also done using the same algorithm of sliding window testing. The only difference is that for a particular run, only one type was tested. The test corpus of Corpus 2 was subject to a performance-distribution testing one-type at a time. The results are given in Table 6.

It was observed that types 1, 2 and 3 gave the best results due to the fairly consistent patterns in which they occur and the total number of instances of such markers. Due to the high number of markers of this type, the training is quite effective. As against these types, type 4 occurs in several different patterns and the number of instances is also very less. This leads to sparse data which affects the performance. Nevertheless, due to the very low number of occurrences, the contribution of Type 4 markers to the overall performance is also less.

Another noteworthy point is that in all types, the performance reduces according to the order of identification of the boundaries and the dependency of a boundary on a previous boundary. This is due to the error propagation of a previously identified boundary to the next boundary. The case of type 5 markers is different from the above cases in that the number of occurrences is more, but the performance is not comparable to Types 1, 2, and 3. This is firstly, owing to its heavy dependence on preprocessing. Since the cause is a nominal chunk, it depends on the efficiency of chunk identification. In addition, embedding is very common in this type.

Detection of embedding depends on the chunk identification. If this detection goes wrong, the starting point of both the arguments is affected. It was found that the performance of both the START tags is

comparable in this type. The third factor which affects the performance of this type is the disambiguation of polysemous markers.

S.NO.	PRECISION	RECALL
1	81.38	73.75
2	81.13	78.18
3	82.35	79.03
4	63.35	66.23
5	68.67	70.55
6	65.53	68.18
7	74.69	73.78
8	81.69	76.32
9	77.01	69.07
10	76.85	70.33
Total	73.89	72.09

Table 7: 10-fold Cross-Validation Results

7.4. Results of 10-fold Cross Validation

To statistically validate the performance, 10-fold cross-validation was performed on all the three corpora put together. All three corpora were concatenated and divided into 10 approximately equal parts in terms of the number of sentences. Sometimes this had to be compromised because of quoted text and other factors. The cause-effect relations were not equally distributed among these parts. 10-fold cross validation was performed and the results were all added up and the average precision and recall for a complete single cause-effect relations were computed. It can be noted that the system performs with an overall precision of 73.89 % and Recall of 72.09 %.

8. Conclusion

In this work, we classified the causal markers based on the computational classification. Machine learning features of each type were identified and language models have been developed specifically for each type. We have explained how the order of identification of the boundaries varied according to the type. A sliding window method of testing was employed. Linguistic and heuristic post-processing rules yields good precision and recall. Performance-distribution testing for each type was made. Ten-fold cross validation was performed to validate the system.

9. References

- Arulmozhi, P., Sobha Lalitha Devi. (2006). HMM based POS Tagger for a Relatively Free Word Order Language, *Journal of Research on Computing Science*. Mexico, vol. 18, pp. 37--48.
- Blanco., Eduardo., Nuria Castell., and Dan I., Moldovan. (2008). Causal Relation Extraction. In *Proceedings of the LREC*.
- Elwell, R., Baldridge, J. (2008). Discourse Connective Argument Identification with Connective Specific Rankers. In *Proceedings of the IEEE International Conference on Semantic Computing*. pp.198--205.
- Girju, R. (2003). Automatic Detection of Causal Relations for Question Answering. In *proceedings of the Workshop on Multilingual Summarization and Question Answering - Machine Learning and Beyond*. Association for Computational Linguistics, Vol. 12, pp. 76--83.
- Girju, R., Moldovan, D. I. (2002). Text Mining for Causal Relations. In *proceedings of the FLAIRS Conference*. pp. 360--364.
- Kalki. (1944) Chivakaamiyin Chapatam.
- Khoo, C., Kornfilt, J., Oddy, R., and Myaeng, S. H. (1998). Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Literary & Linguistic Computing*, vol. 13(4), pp.177--186.
- Kudo, T. (2005) CRF++, an open source toolkit for CRF. <http://crfpp.sourceforge.net>
- Lafferty, J., McCallum, A., Pereira, F. (2001). Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*. pp. 282--289.
- Lee, A., Prasad, R., Joshi, A., Dinesh, N., Webber, B.: Complexity of Dependencies in Discourse: Are Dependencies in Discourse More Complex Than in Syntax?. In *Proceedings of the 5th International Workshop on Treebanks and Linguistic Theories*. Prague, Czech Republic.
- Mann, W. C., Thompson, S.A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text & Talk*. vol. 8(3), pp. 243--281.
- Marcu, D., Echihiabi, A. (2002). An Unsupervised Approach to Recognizing Discourse Relations. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. ACL, Philadelphia.
- Menaka, S., Patabhi, RK Rao., Sobha Lalitha Devi. Automatic identification of cause-effect relations in tamil using CRFs. In *Proceedings of the Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, pp. 316--327.
- Oza, U., Prasad, R., Kolachina, S., Sharma, D. M., and Joshi, A. (2009). The Hindi Discourse Relation Bank. In *Proceedings of the Third Linguistic Annotation Workshop*. Association for Computational Linguistics. Morristown, NJ, pp. 158-161.
- Parthasarathi, N. (1960). kuRinchi Malar.
- Pechsiri, C., Sroison, P., and Janviriyasopak, U. (2008). Know-why extraction from textual data for supporting what question. In *Coling: Proceedings of the Workshop on Knowledge and Reasoning For Answering Questions*. Association for Computational Linguistics, Morristown, NJ, pp. 17-24.
- Prasad, R., Dinesh, N., Lee, A.,Miltakaki, E., Robaldo, L., Joshi, A., Webber, B. (2008) The Penn Discourse TreeBank 2.0. In *Proceedings of the Language Resources and Evaluation (LREC)*.
- Saito, M., Yamamoto, K., Sekine, S. (2006). Using phrasal patterns to identify discourse relations. In *Proceedings of the Human Language Technology Conference of the NAACL*. Association for Computational Linguistics, pp. 133-136.
- Sobha, L., Vijay Sundar Ram, R. (2006). Noun Phrase Chunker for Tamil. In *Proceedings of the First National Symposium on Modeling and Shallow Parsing of Indian Languages (MSPIL)*. IIT Mumbai, India, pp. 194--198.
- Sobha Lalitha Devi., Menaka S. (2010). Semantic Representation of Causality. In *Proceedings of the National Seminar on Lexical Resources and Applied Computational Techniques on Indian Languages*. Pondicherry University.
- Varatharajan, M. (1962) Akal ViLakku.
- Vijay Sundar Ram, R. and Sobha Lalitha Devi. (2008). Clause Boundary Identification Using Conditional Random Fields. In *Computational Linguistics and Intelligent Text Processing*, Springer LNCS, pp.140-150.
- Viswanathan, S., Ramesh Kumar, S., Kumara Shanmugam, B., Arulmozi, S., Vijay Shanker, K. (2003). A Tamil Morphological Analyser. In *Proceedings of the International Conference on Natural Language Processing (ICON)*, CIIL,Mysore, India.
- Wellner, B., Pustejovsky, J. (2007). Automatically Identifying the Arguments of Discourse Connectives. In *Proceedings of the EMNLP-CoNLL*.

Issues in the Creation of Synsets in Odia: A Report¹

Panchanan Mohanty, Ramesh C. Malik & Bhimasena Bhol

Centre for Applied Linguistics & Translation Studies, University of Hyderabad, India

Email: panchananmohanty@gmail.com

Abstract

Since languages differ from each other, it is difficult to find equivalents for the words and expressions of one language in another. So creating an interlingual WordNet in Odia vis-à-vis Hindi has been a challenging task. While dealing with the equivalence problems in Odia, creation of new expressions dominates the synsets involving various kinds of wage, derivation of nouns from nouns and adjectives, adjectives derived from nouns, and single-word Hindi synsets expressing complex ideas and kinship synsets. The other important procedure is borrowing that has been used widely in the domains of historical events, geographical locations, socio-cultural practices, place names, personal names, flora and fauna, ecological entities, gods and goddesses, culture-specific items, etc.. Apart from these, certain problematic issues of the Hindi WordNet, viz. wrong categorization of synsets, concepts with inadequate information and description, mismatch between concepts and synsets, and imprecise concepts have also been discussed with a view to sensitizing other Indian language WordNet developers regarding these deficiencies. At the same time, we expect the quality of the Hindi WordNet to improve if these problems are taken care of.

Keywords: WordNet, Synset, Odia, New Expressions, Borrowing, Kinship term, Compound Verb, Conjunctive Participle

1. Introduction

“WordNet is an on-line lexical reference system whose design is inspired by current psycholinguistic theories of human lexical memory” and also “it is a proposal for a more effective combination of traditional lexicographic information and modern high-speed computation” (Miller et al., 1993: 1). The main objective of WordNet is to describe the lexical and componential features of words and their use by the concerned linguistic community. WordNet contains linguistic data, such as syntactic, semantic and pragmatic features of words which are important factors for designing a WordNet in a language. In order to create a WordNet for Odia on the basis of the Hindi WordNet, the major problems faced were of equivalence. While looking for Odia equivalents of Hindi words, the problems were of various kinds. We would like to highlight some of these issues in this paper.

All the synsets of Hindi WordNet have been divided into four lexical categories: nouns, adjectives, verbs, and adverbs. These cover different domains like historical events, geographical locations, socio-cultural practices, place names, personal names, flora and fauna, ecological entities, gods and goddesses, culture-specific items, etc. In fact, creating equivalent synsets in Odia has been quite a challenging task. But one thing which we have religiously left untouched is the category of the synsets. In other words, the categories of the Hindi synsets have been always retained as such in Odia. So this discussion will focus on the problems of synset creation in Odia and their solutions.

2. Problems and Procedures in Odia Synset Creation

Translation is the common method used for creating Odia synsets. While translating, the major problems were both linguistic and cultural. Since “no two languages are ever sufficiently similar to be considered as representing the same social reality” (Sapir 1959: 69), translating from one language to another definitely poses problems as it is very hard to find one to one correspondence between two languages. Jakobson (1971: 235) states, “Languages differ essentially in what they must convey and not in what they can convey. Each verb of a given language imperatively raises a set of specific yes/no questions, as for instance: Is the narrated event presented prior to the speech event or not? Naturally the attention of native speakers and listeners will be constantly focused on such items as they are compulsory in their verbal code”. That is why, translators often try to find out proper equivalents in the target language and ultimately this helps to solve many translation problems. According to Nida (1964: 91), “Translation-problems, which are essentially problems of equivalence, may be conveniently treated under (1) ecology, (2) material culture, (3) social culture, (4) religious culture, and (5) linguistic culture”. In order to create synsets in Odia on the basis of the Hindi synsets, similar problems have been encountered and efforts have been made to bridge the gap between the two languages. Let us now discuss these problems and the ways in which we have tried to solve them.

3. New Expressions

Since languages and cultures differ from one another, it is difficult to find lexical equivalents of one language in

another. That is why creation of new expressions becomes an important procedure to represent the source language (SL) concept in the target language (TL). Following this, new expressions have been created by combining the existing Odia words in order to express the meanings of the Hindi synsets which do not have equivalents in Odia.

Hindi uses single words to represent wages for various kinds, but these are not commonly used in Odia. So while providing equivalents for these, expressions involving more words have been used in Odia. Therefore, this procedure has been very useful especially in creating synsets for different kinds of wage. For example:

SL : Hindi
ID : 20708
CAT : NOUN
CONCEPT : छिड़कने की मज़दूरी
EXAMPLE : "उसने प्रति एकड़ दवा छिड़काई पचास रुपए लिए"

SYNSET-HINDI : छिड़काई
TL : Odia
ID : 20708
CAT : NOUN
CONCEPT : କିଛି ଛିଞ୍ଚିବାପାଇଁ ଦିଆଯାଇଥିବା ମଜୁରି
EXAMPLE : "ସେ ଏକର ପ୍ରତି ଓପିଏସ ଛିଞ୍ଚିବା ମଜୁରି ପଚାଶ ଟଙ୍କା ନେଲେ"

SYNSET-ODIA : ଛିଞ୍ଚିବା ମଜୁରି, ଛିଞ୍ଚିବା ମୂଲ, ଛିଞ୍ଚା ମଜୁରି, ଛିଞ୍ଚା ମୂଲ

In the above example¹, the SL word *chiRka:i:* (छिड़काई) refers to *chiRka:ne ki mazdu:ri:* (छिड़काने की मज़दूरी) 'the wage paid for sprinkling'. In Odia, there is no single word to express this concept. Therefore, it has been rendered as *chinciba: majuri* (ଛିଞ୍ଚିବା ମଜୁରି) or *chinca: majuri* (ଛିଞ୍ଚା ମଜୁରି) 'sprinkling wage' and *chinciba: mu:la* (ଛିଞ୍ଚିବା ମୂଲ) or *chinca: mu:la* (ଛିଞ୍ଚା ମୂଲ) 'sprinkling wage'. The words *majuri* (ମଜୁରି) and *mu:la* (ମୂଲ) refer to 'wage' and *chinciba:* (ଛିଞ୍ଚିବା) and *chinca:* (ଛିଞ୍ଚା) mean 'sprinkling'. When used together, they express the same meaning as *chiRka:i:* (छिड़काई) in Hindi.

There are cultural and occupational differences between the Hindi and Odia speech communities for which it is difficult to find equivalents for such Hindi words in Odia. The above terms used in Hindi for paying

¹ In this paper, [T, Th, D, Dh, R, Rh, L] have been used for the voiceless unaspirated retroflex stop, voiceless aspirated retroflex stop, voiced unaspirated retroflex stop, voiced aspirated retroflex stop, voiced unaspirated retroflex flap, voiced aspirated retroflex flap, and retroflex lateral respectively.

wages for different types of jobs are not found in Odia and that is why this difficulty.

3.1. Derivation of Nouns from Nouns and Adjectives

We find a number of Hindi nouns that have been derived by adding prefixes or suffixes. But the same is not possible in Odia even though both these languages belong to the Indo-Aryan family. When we follow the Hindi way, the outcomes do not give appropriate meanings. So to create such equivalents in Odia, the Odia morphological practices have been followed. For example:

SL : Hindi
ID : 22283
CAT : NOUN
CONCEPT : उपशमन न होने या न दबाने की क्रिया
EXAMPLE : "इंद्रियों की दासता अप्रतिक्रिया के ही परिणाम हैं"

SYNSET-HINDI : अप्रतिक्रिया
TL : Odia
ID : 22283
CAT : NOUN
CONCEPT : ଶାନ୍ତ ନ ହେବା ବା ପ୍ରଶମିତ ନ ହେବା ପ୍ରକ୍ରିୟା
EXAMPLE : "ଇନ୍ଦ୍ରିୟଗୁଡ଼ିକର ଦାସତ୍ୱ ହେଲେ ପ୍ରତିକ୍ରିୟାହୀନତାର ପରିଣାମ"

SYNSET-ODIA : ପ୍ରତିକ୍ରିୟାହୀନତା
This example shows how nouns in Hindi have been created by adding prefixes to other nouns. In this example, the Hindi word *apratikriya:* (अप्रतिक्रिया) as a noun refers to *upashaman na hone ya: na daban:ne ki: kriya:* (उपशमन न होने या न दबाने की क्रिया) which means 'the state of remaining unabated'. It is a noun which has been formed by adding the negative prefix *a-* (अ-) to the noun *pratikriya:* (प्रतिक्रिया) 'reaction' whereas there is no word *apratikriya:* (अप्रतिक्रिया) in Odia. In order to create an equivalent in Odia, the negative suffix *-hi:nata:* (-ହୀନତା) 'absence of something' has been added to *pratikriya:* (ପ୍ରତିକ୍ରିୟା). The product *pratikriya:hi:nata:* (ପ୍ରତିକ୍ରିୟାହୀନତା) is an equivalent to Hindi *apratikriya:* (अप्रतिक्रिया).

We find that the above words are not used in Odia the way they are used in Hindi. Though the above Hindi nouns have been created by adding a prefix, only a suffix has been added in Odia.

Hindi adjectives are usually formed by adding prefixes or suffixes to nouns whereas in Odia, if a noun does not have an adjective, it can also be used as an adjective. The word (ID No. 7323) *cama:ri:* (चमारी) refers to *cama:r ka: ya: cama:r se sambandhit* (चमार का

या चमार से संबंधित) in Hindi. As there is no adjective of *cama:r* (चमार) in Odia, the same *cama:ra* (ଚମାର) has been used as an adjective.

3.2. On Adjectives Derived from Nouns

There are some adjectives in Hindi that have been created by adding certain suffixes to nouns. Since such suffixes are not usually used in Odia, we have created Odia equivalents by adding other suffixes to the nouns. For example:

SL	:	Hindi
ID	:	17292
CAT	:	ADJECTIVE
CONCEPT	:	विद्यालय का या विद्यालय से संबंधित
EXAMPLE	:	"विद्यालयी वातावरण को गुरुकुल के वातावरण के समकक्ष बनाने का प्रयत्न किया जाना चाहिए"
SYNSET-HINDI	:	विद्यालयी, विद्यालयीन, विद्यालयीय, स्कूली, पाठशालेय
TL	:	Odia
ID	:	17292
CAT	:	ADJECTIVE
CONCEPT	:	ବିଦ୍ୟାଳୟ ବା ବିଦ୍ୟାଳୟ ସହ ସମ୍ବନ୍ଧିତ
EXAMPLE	:	"ବିଦ୍ୟାଳୟର ବାତାବରଣକୁ ଗୁରୁକୂଳର ବାତାବରଣ ସହ ସମାନ କରିବାକୁ ଆମେ ଚେଷ୍ଟା କରିବା ଭବିଷ୍ୟ"
SYNSET-ODIA	:	ବିଦ୍ୟାଳୟର, ସ୍କୁଲର, ବିଦ୍ୟାଳୟ ସମ୍ବନ୍ଧୀୟ, ସ୍କୁଲ ସମ୍ବନ୍ଧୀୟ

In the above example, the Hindi synonyms *vidya:layi:* (विद्यालयी), *vidya:layi:n* (विद्यालयीन), *vidya:layi:y* (विद्यालयीय), *sku:li:* (स्कूली), and *pa:Thsha:ley* (पाठशालेय) mean 'of a school or related to a school'. These adjectives have been created by adding the suffixes *-i:* (-ई), *-i:n* (-ईन), *-i:y* (-ईय) to the noun *vidya:lay* (विद्यालय) and *-i:* (-ई) to the noun *sku:l* (स्कूल), *-ey* (-एय) to the noun *pa:Thsha:la:* (पाठशाला) 'school'. As these adjectives are not used in Odia, the possessive case suffix *-ra* (-ର) and the word *sambandhi:ya* (ସମ୍ବନ୍ଧୀୟ) 'related' have been added to *bidya:Laya* (ବିଦ୍ୟାଳୟ), and *skul* (ସ୍କୁଲ) leading to creation of the adjectives *bidya:Layara* (ବିଦ୍ୟାଳୟର), *skulra* (ସ୍କୁଲର) which are equivalents of Hindi *vidya:layi:* (विद्यालयी), *vidya:layi:n* (विद्यालयीन), *vidya:layiy* (विद्यालयीय), *sku:li:* (स्कूली), and *pa:Thsha:ley* (पाठशालेय).

It is clear from the above example that the formation of adjectives from nouns by adding suffixes in

Hindi and Odia is not the same though both of them belong to the Indo-Aryan language family.

3.3. Single-Word Synsets Expressing Complex Ideas

There are certain Hindi synsets which consist of single words, but express relatively complex ideas. Such synsets cannot be expressed in single words in Odia. More words have to be used together to express the appropriate meanings. For example:

SL	:	Hindi
ID	:	9562
CAT	:	NOUN
CONCEPT	:	सफेद शरीर और काली चोंच वाला कबूतर
EXAMPLE	:	"छत पर कलटोरे का एक जोड़ा बैठा हुआ है"
SYNSET-HINDI	:	कलटोरा
TL	:	Odia
ID	:	9562
CAT	:	NOUN
CONCEPT	:	ଧଳା ଶରୀର ଓ କଳା ଥଣ୍ଡୁଣିବା ପାଠ
EXAMPLE	:	"ଛାତ ଉପରେ ଯୋଡ଼େ କଳା ଥଣ୍ଡୁଣିବା ଧଳାପାଠ ବସିଛନ୍ତି"
SYNSET-ODIA	:	କଳା ଥଣ୍ଡୁଣିବା ଧଳାପାଠ, କଳା ଚଞ୍ଚୁଣିବା ଧଳାପାଠ, କଳାଥଣ୍ଡୁଣିଆ ଧଳାପାଠ

In the above example, the Hindi word *kalTora:* (कलटोरा) refers to 'a pigeon whose body is white and beak is black'. This is an example of how a single word in Hindi expresses complex concept. There is no word in Odia which can convey the same meaning. So while creating the expression in Odia, it has been rendered as *kaLa: thaNTathiba: dhaLa: pa:ra:* (କଳା ଥଣ୍ଡୁଣିବା ଧଳାପାଠ), *kaLa: cancuthiba: dhaLa: pa:ra:* (କଳା ଚଞ୍ଚୁଣିବା ଧଳାପାଠ), and *kaLa: thaNTia: dhaLa: pa:ra:* (କଳାଥଣ୍ଡୁଣିଆ ଧଳାପାଠ) where *kaLa:* (କଳା) means 'black', *thaNTathiba:* (ଥଣ୍ଡୁଣିବା), or *cancuthiba:* (ଚଞ୍ଚୁଣିବା) and *thaNTia:* (ଥଣ୍ଡୁଣିଆ) mean 'having a beak', *dhaLa:* (ଧଳା) means 'white', and *pa:ra:* (ପାଠ) means 'pigeon'.

4. Kinship Relations

Kinship terms denote some definite relations and are culture-specific. So these are very difficult to translate from one language to another. For example, in ID No. 6992, the SL kinship adjective *phuphera:* (फुफेरा) means 'father's sister's husband'. *Phuphi:* in Hindi refers to the kin *pusi:* (ପିଉସୀ) 'father's sister' in Odia. So the word *phuphera: bha:i:* (फुफेरा भाई) has been translated as *pui:i pua bha:i* (ପିଉସୀ ପୁଅ ଭାଇ) and *pi:si pua bhai* (ପିଉସୀ ପୁଅ ଭାଇ) "father's sister's son".

Similarly, in ID No. 7011, the kinship adjective *mamera*: (ममेरा) is ‘related to mother’s brother’. While translating *mamera*: *bha:i*: (ममेरा भाई), i.e. ‘mother’s brother’s son’ to Odia, it has been rendered as *ma:muM pua bha:i* (ମାମୁ ପୁଅ ଭାଇ) or *mauLa: pua bha:i* (ମାଉଳା ପୁଅ ଭାଇ) ‘a cousin who is the son of mother’s brother’. In ID No. 9876, the kinship adjective *mausera* (मौसेरा) means ‘related to mother’s sister’s husband’ which has been used to denote the relationship with ‘mother’s sister’s husband’. So *mausera sasur* (मौसेरे ससुर) has been translated as *mausa: sasura* (ମାଉସା ଶଶୁର) ‘wife/husband’s mother’s sister’s husband’ as this is widely used in Odia.

5. Borrowing of Synsets

A number of concepts in Hindi WordNet are language specific, e.g. name of persons, places, geographical locations, food items, historical places and events, culture-specific items, occupational instruments, different kinds of dress and dress materials, domestic articles, gods and goddesses, rituals, festivals, and names of religion-related items which do not have equivalents in Odia. So they have been borrowed in Odia- sometimes directly and sometimes with phonological adjustments. In fact, whenever we could not find an equivalent in Odia for a Hindi item in spite of all our efforts, we resorted to borrowing. Consider the following examples:

Example 1

SL : Hindi
ID : 20073
CAT : NOUN
CONCEPT : एक पेड़ से प्राप्त गहरे लाल रंग का गोल फल

EXAMPLE : "शीला आलूबुखारा खा रही है"

SYNSEM-HINDI : आलूबुखारा

SL : Odia
ID : 20073
CAT : NOUN
CONCEPT : ଗୋଟିଏ ଗଛରୁ ପ୍ରାପ୍ତ ଗାଢ଼ ନାଲି ରଙ୍ଗର ଗୋଲ ଫଳ

EXAMPLE : "ଶୀଳା ଆଲୁବଖାରା ଖାଉଛି"

SYNSEM-ODIA : ଆଲୁବଖାରା, ଆଲୁବୁଖାରା, ଆଲୁବୋଖାରା

In this example, the SL word *a:lu:bukha:ra*: (आलूबुखारा) does not have an equivalent in Odia. Actually, *a:Lubakhara*: (ଆଲୁବଖାରା) is “a kind of plum that has acid taste”. It is not grown in Odisha; so there is no equivalent for it in Odia. In this situation, the Hindi word has been borrowed and naturalized as *a:Lubakhara*: (ଆଲୁବଖାରା), *a:Lubukha:ra*: (ଆଲୁବୁଖାରା), *a:Lubokha:ra*: (ଆଲୁବୋଖାରା) in Odia.

A number of synsets in Hindi refer to a variety of animals, birds, trees, and some mythological characters

from the *Puranas*. Such words also do not have equivalents in Odia. Take an example:

Example 2

SL : Hindi
ID : 26503
CAT : NOUN
CONCEPT : इंडोनेशिया में पाई जानेवाली एक प्रकार की भैंस

EXAMPLE : "सूबा अच्छा दूध देती है"

SYNSEM-HINDI : सूबा,सूबा भैंस

TL : Odia
ID : 26503
CAT : NOUN
CONCEPT : ଇଣ୍ଡୋନେସିଆରେ ମିଳୁଥିବା ଏକ ପ୍ରକାର ମଇଁଷି

EXAMPLE : "ସୁବାମଇଁଷି ଭଲ ଖାଏ ଦିଏ"

SYNSEM-ODIA : ସୁବାମଇଁଷି

In this example, the SL word *su:ba*: (सूबा) refers to ‘a type of buffalo found in Indonesia’ (इंडोनेशिया में पाई जानेवाली एक प्रकार की भैंस). This buffalo is not found in India. Though only *su:ba*: is also given as a synonym in Hindi, it is unlikely to be understood by Odia speakers. So *maiMsi* (ମଇଁଷି) has been added to it so that the users can decode *suba:maiMsi* (ସୁବାମଇଁଷି) easily.

6. Wrong Categorisation of Synsets

All synsets in Hindi WordNet have been categorized according to their lexical categories, but some of them are found to be in the wrong grammatical category. We have not changed the categories of these incorrectly categorized synsets. For example:

SL : Hindi
ID : 1105
CAT : ADJECTIVE
CONCEPT : जो करनेवाला हो

EXAMPLE : "भगवान ही सब कामों के कर्ता हैं, हम तो निमित्त मात्र हैं"

SYNSEM-HINDI : कर्ता, कर्ता, कर्तार, करनहार, करणहार

TL : Odia
ID : 1105
CAT : ADJECTIVE
CONCEPT : କରିବାବାଲ

EXAMPLE : "ଭଗବାନ ହିଁ ସବୁ କାମର କର୍ତ୍ତା, ଆମେ ତ ନିମିତ୍ତ ମାତ୍ର"

SYNSEM-ODIA : କର୍ତ୍ତା, ପ୍ରଭୁ, ମାଲିକ

In this example, the Hindi word *kartta*: (कर्ता) has been categorized as an adjective. But it refers to a ‘doer’ or ‘a person who does something’ and thus it is a ‘noun’. The example sentence also confirms that the word has

been used as a noun because after *sab ka:moM ke* (सब कामों के) only a noun can occur in Hindi. However, the Odia synset retains *kartta:* (कर्ता) and new synonyms like *prabhu* (ପ୍ରଭୁ) and *ma:lika* (ମାଲିକ) are added to it.

7. Concepts with Inadequate Description

Generally, the concepts are expected to give a detailed description of the synsets. But some Hindi synsets do not provide sufficient information in the concepts. So it has been very difficult to deal with those while creating equivalents in Odia. Let us take an example:

SL : Hindi
ID : 23390
CAT : NOUN
CONCEPT : एक प्रकार का फूल
EXAMPLE : "गुलदस्ते में रंग-बिरंगी लिलियाँ सजी हैं"
SYNSET-HINDI : लिलि
TL : Odia
ID : 23390
CAT : NOUN
CONCEPT : ଏକ ପ୍ରକାର ଫୁଲ
EXAMPLE : "ଫୁଲତୋଡ଼ାଟି ରଙ୍ଗବେରଙ୍ଗ ଲିଲିରେ ସଜ୍ଜିତ ହୋଇଛି"
SYNSET-ODIA : ଲିଲି

In the above example, the SL concept *ek praka:ra ka: phu:l* (एक प्रकार का फूल) refers to 'a kind of flower'. There are many kinds of flower found in the world and the concept is not a proper description of the flower named *lili* (लिलि). It would have been much better, if the concept had described its colour, shape, size as well as the kind of plant that bears it. However, the concept in Odia has been given as *eka praka:ra phula* (एक प्रकार फूल) like the SL concept.

8. Mismatch between Concepts and Synsets

There are some IDs in Hindi which convey ambiguous meanings where the relation among the concept, example, and synset is mismatching. For example:

Example
SL : Hindi
ID : 34533
CAT : NOUN
CONCEPT : लेकर जाना
EXAMPLE : "इतना सामान उठाकर ले जाना मुझे भारी पड़ रहा है/आप इन्हें भी अपने साथ ले जाएँ"
SYNSET-HINDI : ले जाना
TL : Odia
ID : 34533
CAT : NOUN

CONCEPT : ନେଇକରି ଯିବା
EXAMPLE : "ଏତେଗୁଡ଼ିଏ ଜିନିଷ ନେଇକରି ଯିବା ମୋପାଇଁ କଷ୍ଟକର"
SYNSET-ODIA : ନେଇକରି ଯିବା, ନେଇକି ଯିବା

In this example, the synset *le ja:na:* (ले जाना) refers to the concept *lekar ja:na:* (लेकर जाना) in Hindi. In fact, the concept *lekar ja:na:* (लेकर जाना) does not at all mean *le ja:na:* (ले जाना). For example, *ho gaya:* (हो गया), *kha: gaya:* (खा गया), and even *le gaya:* (ले गया) from a popular Hindi song *le gayi: dil guRiya: ja:pa:n ki:* (ले गयी दिल गुड़िया जापान की) do not express the meaning of *ho kar gaya:* (हो कर गया), *kha: kar gaya:* (खा कर गया), and *le kar gayi: dil guRiya: ja:pa:n ki:* (ले कर गयी दिल गुड़िया जापान की). It is because *le ja:na:* (ले जाना), *ho ja:na:* (हो जाना), and *kha: ja:na:* (खा जाना) are compound verbs whereas *le kar ja:na:* (लेकर जाना), *ho kar ja:na:* (हो कर जाना), and *kha: kar ja:na:* (खा कर जाना) are conjunctive participles. So the above mentioned concept and the synset do not match with each other. Considering these, the Odia synset has been made *neikari jiba:* (ନେଇକରି ଯିବା) and *nei ki jiba:* (ନେଇକି ଯିବା).

The other important point to be mentioned here is that the second example sentence make it evident that *le ja:eM* (ले जाएँ) is a verb though the category of the synset is stated to be a noun.

9. Imprecise Concepts

There are certain concepts which do not convey a clear and precise meaning. It was difficult to understand such concepts. As a result, creating equivalents in Odia was difficult. For Example:

SL : Hindi
ID : 195
CAT : ADJECTIVE
CONCEPT : सामनेवाले का
EXAMPLE : "तुम्हारा घर भी बहु तप्यारा है"
SYNSET-HINDI : तुम्हारा,तेरा
TL : Odia
ID : 195
CAT : ADJECTIVE
CONCEPT : ସାମନା ଲୋକର
EXAMPLE : "ତୁମ ଘର ବହୁତ ସୁନ୍ଦର"
SYNSET-ODIA : ତୁମ, ତମ, ତୁମର, ତମର, ତୋ, ତୋର, ଆପଣଙ୍କ, ଆପଣଙ୍କର

In the above example, the concept is *sa:mneva:le ka*: (सामनेवाले का) ‘of the person present in the front’ whereas the synset has the pronouns *tumha:ra*: (तुम्हारा) ‘your’ (semi-honorific) and *tera*: (तेरा) ‘your’ (non-honorific). The concept does not match with the synset because it may also refer to a person present in the front for whom a third person pronoun can be used, e.g., *ta:ra kalama achi* (ତାର କଲମ ଅଛି) ‘he/she has a pen’. Notice that the third person possessive pronominal form *ta:ra* (ତାର) is used here. Again, the pronouns *a:paNanka* (ଆପଣଙ୍କ) and *a:paNankara* (ଆପଣଙ୍କର) ‘your’ (honorific) have the same possessive function. Considering all these, it would have been better if the concept had been *sa:mna:re upasthita o sambodhita heuthiba: lokara* (ସାମ୍ନାରେ ଉପସ୍ଥିତ ଓ ସମ୍ବୋଧିତ ହେଉଥିବା ଲୋକର) ‘of the person present in the front and being addressed to’. In Hindi, it will be *sa:mne upasthit aur sambodhit honevale vyakti ka*: (सामने उपस्थित और संबोधित होनेवाले व्यक्ति का). This concept will express the intended meaning more faithfully than the existing one.

10. Conclusion

The present paper has dealt with the problems encountered while creating synsets in Odia based on the Hindi WordNet. In order to create Odia synsets, the procedures of new expressions and borrowing have been used widely. The former has been used mostly to resolve the problems of synsets dealing with different kinds of wage and kinship relations. The other important issues discussed in this paper are wrong categorization of synsets, concepts with inadequate description, mismatch between concepts and synsets, imprecise concepts in Hindi with suggestions as to how to improve these.

Acknowledgement

The authors are thankful to Dr. Smita Mohanty, Dr. H.K. Patra, Ms. Gouri Sahoo and Dr. R.R. Mohapatra for their help during the preparation of various drafts of this paper.

References

- Jakobson, Roman. (1971). On Linguistic Aspects of Translation. Selected Writings, Vol. 2, The Hague: Mouton, pp. 232-239.
- Miller, George A, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine Miller. (1990). Introduction to WordNet: An On-line Lexical Database. International Journal of Lexicography. Vol. 3 . No. 4. pp. 235-244.
- Nida, E.A. (1964). Linguistics and Ethnology in Translation Problems. Language in Culture and Society, Ed. by Dell Hymes. Bombay: Allied Publishers Private Ltd. Indian reprint), pp. 90-100.
- Sapir, Edward (1956). Culture, Language and Personality: Selected Essays. (Ed). David G. Mandelbaum. Berkeley: University of California Press.

Large Web Corpora of High Quality for Indian Languages

Uwe Quasthoff¹, Ritwik Mitra², Sunny Mitra², Thomas Eckart¹, Dirk Goldhahn¹,
Pawan Goyal², Animesh Mukherjee²

¹ Natural Language Processing Group, University of Leipzig, Germany

² Indian Institute of Technology Kharagpur, Kharagpur, India

Email: ¹ {quasthoff, teckart, dgoldhahn}@informatik.uni-leipzig.de

² {ritwik.mitra, sunny.mitra, pawang, animeshm}@cse.iitkgp.ernet.in

Abstract

Large textual resources are the basis for a variety of applications in the field of corpus linguistics. For most languages spoken by large user groups a comprehensive set of these corpora are constantly generated and exploited. Unfortunately for modern Indian languages there are still shortcomings that interfere with systematic text analysis. This paper describes the Indian part of the Leipzig Corpora Collection which is a provider of freely available resources for more than 200 languages. This project focuses on providing modern text corpora and wordlists via web-based interfaces for the academic community. As an example for the exploitation of these resources it will be shown that they can be used for the visualization of semantic contexts of terms and for language comparison.

Keywords: corpus generation, text acquisition, comparative corpus analysis

1. Availability of Indian text resources

Many applications in the field of corpus linguistics require the availability of large corpora. Well-known examples for corpora based on Indian languages are the web corpora built by Kilgariff and Duvuru (2011) or the large Sanskrit corpus by GRETIL that is provided for free. The Technology Development for Indian Languages (TDIL) Programme also provides some corpora for Indian languages for download and further resources are available from the Central Institute of Indian Languages (CIIL).

Unfortunately many of the existing corpora or resources lack features that are strongly desirable for their use in the scientific context. These shortcomings include problems with availability (in some cases the use of very specific interfaces is required), lack of currentness (a problem especially when dealing with ongoing political developments), high costs or strict licences that permit reuse and data aggregation. As some of these problems can't be eliminated in general (like in the context of copyright and personality rights) it would be desirable to have more resources that can be used with as less restrictions as possible and that can be useful for further progress in the exploitation of Indian language corpora and other text-based resources .

2. Indian Resources

The Leipzig Corpora Collection (LCC)¹ has been collecting digital text material for more than 20 years. Starting with a focus on European languages it became apparent that a lot of the developed strategies and tools could be reused for other languages as well. Over the last years the established tool chain for text acquisition and text processing was adopted to deal with non-Latin scripts

and used to create and improve resources based on Indian text material.

2.1. Text Acquisition Strategies

Different strategies are combined for collecting textual data from the WWW. The main goal is to ensure that corpora of large extent and high diversity concerning topics or genres can be created for specific languages. Especially, Indian languages that are spoken in many countries require a variety of approaches to achieve this objective.

2.1.1. Generic Web Crawling

A framework for massively parallel Web crawling is applied that utilizes the standard Web crawler and archiver *Heritrix*² of the Internet Archive. Among other enhancements, it was enriched with means for the automatic generation of crawling jobs.

Heritrix is used in several ways. On one hand whole Top Level Domains are crawled. In this case a small list of domains of a country of interest is used as an input. *Heritrix* is then configured to follow links within this TLD. This has been conducted for several countries.

On the other hand News sources are downloaded using the *Heritrix* based Web crawler. Basis is a list of more than 32,000 news sources in about 120 languages provided by *ABYZ News Links*³. This service offers URLs and information regarding country and language. This way, news texts for several Indian languages were collected. This includes text data excluded in the TLD crawling because of non-country TLDs used such as “.com”.

2.1.2. Distributed Web Crawling

FindLinks (Heyer and Quasthoff, 2004) is a distributed Web crawler using a client-server architecture. The

¹ <http://corpora.uni-leipzig.de>

² <http://webarchive.jira.com/wiki/display/Heritrix>

³ <http://www.abyznewslinks.com>

Java-based client runs on standard PCs and processes a list of URLs, which it receives from the *FindLinks*-server. FindLinks has been used with community support for several years and allowed us to crawl the WWW to a large extent.

2.1.3. Bootstrapping Corpora

In addition, an approach similar to Baroni (2004) and Sharoff (2006) was applied. Frequent terms of any specific language are combined to form Google search queries and the resulting URLs are retrieved as basis for the default crawling system.

A small set of frequent terms is needed for languages in question. Therefore existing corpora of the LCC or other sources such as the *Universal Declaration of Human Rights (UDHR)*⁴ were utilized as a resource.

Based on these lists, tuples of three to five high frequent words are generated. These tuples are then used to query Google and to collect the retrieved URLs, which are then downloaded.

2.1.4. Crawling of Special Domains

Certain domains are beneficial sources for Web corpora since they contain a large amount of text in predefined languages.

One example is the free Internet encyclopedia Wikipedia, which is available in more than 200 languages and of course also contains entries for Indian languages.

Wikipedia dumps for these Indian languages, were downloaded. *Wikipedia Preprocessor*⁵ was used for further processing and text extraction.

2.2. Corpus Creation Toolchain

Necessary steps for the creation of dictionaries are text extraction (mostly based on HTML as input material), language identification (Pollmacher, 2011), sentence segmentation, cleaning, sentence scrambling, conversion into a text database and statistical evaluation.

An automatic and mainly language independent tool chain has been implemented. It is easily configurable and only few language-dependent adjustments, concerning e.g. abbreviations or sentence boundaries, have to be made.

In a final step statistics-based quality assurance is applied to achieve a satisfying quality of the resulting dictionaries (Quasthoff, 2006b) (Eckart, 2012). Using features such as character statistics, typical length distributions, typical character or n-gram distributions, or tests for conformity to well-known empirical language laws during corpora creation can be detected and corrected.

The processing of Indian language text required several changes to the existing toolchain. Most of the developed tools could be reused but specific configurations had to be changed. This includes changes to components like sentence segmentation or quality assurance procedures. Besides some minor problems the general system again proved to be stable enough.

2.3. Sentence Scrambling

For all corpora the sentences had to be "scrambled" to destroy the original structure of the documents due to copyright restrictions. This inhibits the reconstruction of the original documents. With respect to German copyright legislation this approach is considered safe.

2.4. Available Resources

Corpora of this collection are typically grouped regarding the dimensions language, country of origin, text type (newspaper text, governmental text, generic Web material, religious texts etc.) and time of acquisition. Table 1 gives an introduction to currently available resources. It contains the number of sentences for different languages and genres. For comparison, the size of the corresponding Emille⁶ corpora is given. As the crawling is an ongoing process new corpora are added at least every year.

To do a sanity check over the corpora, 1000 high frequency words from Bengali as well as Hindi corpora were taken. These words were then manually checked by native speakers of Bengali and Hindi. For both Bengali and Hindi, more than 94% of the words turned out to be valid. The remaining 6% contained spelling errors, mathematical symbols etc.

2.5. Available Interfaces

The corpora are available via Web-based interfaces. There is a freely available web portal where a variety of information can be accessed based on a word level (like sample sentences, word co-occurrences, co-occurrence graphs etc.). Furthermore many corpora can be downloaded for free in different formats. These include plain text versions of the textual material and also MySQL databases. For the later a platform-independent browsing tool is provided which allows examining the corpus locally.

3. Applications

3.1. POS-Tagging

For many Indian languages (like Hindi, Telugu, Tamil, Kannada, Punjabi, Urdu, Bengali and Marathi) POS taggers are available. They all use Latin transliteration and most of them use the WX transliteration scheme. Therefore the used tag sets are comparable.

For testing purposes some of the Hindi sentences were tagged using the Hindi shallow parser⁷. In addition to morphological analysis and chunking, this tool also gives the POS tagging analysis of a sentence.

4 <http://www.ohchr.org>

5 <http://sourceforge.net/projects/wikiprep/>

6 <http://www.emille.lancs.ac.uk/>

7 http://ltrc.iit.ac.in/showfile.php?filename=downloads/shallow_parser.php

As an example the Hindi sentence (in Devanagari)

राजुका उत्तर राज ने अपने पत्र में लिखा कि सत्यम को
तंगी से उबारने के लिए उन्होंने ऐसा किया ।

(English translation: *Raju answered this in his letter that
he did it to rescue Satyam from scarcity.*)

is transliterated (using roman transliteration) to:

isakā uttara rājū ne apane patra meṃ diyā ki satyama ko
taṅgī se ubārane ke lie unhoṃne aisā kiyā .

The output from the POS tagger is:

isakā/PRP uttara/NNPC rājū/NNP ne/PSP apane/PRP
patra/NN meṃ/PSP diyā/VM ki/CC satyama/NNP
ko/PSP taṅgī/NN se/PSP ubārane/VM ke/PSP lie/PSP
unhoṃne/PRP aisā/PRP kiyā/VM ./SYM

3.2. Diachronic Comparisons

Newspaper corpora collected on yearly basis can be used to investigate changes in the frequency of words. These changes may reflect different types of modern developments.

3.3. Lexicography

Frequency ordered word lists help identifying lemmas for dictionaries, especially for the enlargement of existing dictionaries. Especially neologisms (i.e. words found in the corpus of the current year with a certain frequency and not seen before) are interesting for the study of language change.

3.4. Semantic Relations given by Word Co-occurrences

Significant word co-occurrences often show semantic relations between those words. The investigation of word co-occurrence graphs also helps identifying the most prominent topics in the corpus. Figure 1 shows two clusters of words that significantly often occur in the context of “Mumbai”: one cluster corresponding to sports events and the other corresponding to the terror attacks in Mumbai 2008.

3.5. Comparisons between Countries and Regions

For languages spoken in more than one country, the corpora will reflect their differences. For the languages spoken in India, the corpora can be used for:

- Comparison of Urdu, Sindhi and Punjabi in India and Pakistan,
- Comparison of Tamil in India and Sri Lanka
- Comparison of Bengali in India and Bangladesh

4. Outlook

The Leipzig Corpora Collection will continue in aggregating Web-based text material to extend the amount and quality of available resources. Currently 220 GB of

additional raw material crawled from the Indian TLD are processed and will be available in spring 2014. Furthermore, the result of these efforts will be provided to all the interested users.

Until mid of 2014 a new Web portal will be deployed that provides extended functionality and a more user-friendly interface. The underlying REST-based web services are also freely available and can be used for external applications as well. As a next step in exploiting word lists as a valuable resource in information extraction and language comparison, it is planned to publish a volume in the series of frequency dictionaries focusing on word frequency information in specific Indian languages.

5. References

- Baroni, M.; Bernardini, S. (2004). BootCaT: Bootstrapping corpora and terms from the web. Proceedings of LREC 2004.
- Eckart, T.; Quasthoff, U.; Goldhahn, D. (2012). Language Statistics-Based Quality Assurance for Large Corpora. Proceedings of Asia Pacific Corpus Linguistics Conference 2012, Auckland, New Zealand.
- Heyer, G.; Quasthoff, U. (2004). Calculating Communities by Link Analysis of URLs. Proceedings of IICS-04, Guadalajara, Mexico and Springer LNCS 3473.
- Kilgariff, A.; Duvuru, G. (2011). Large Web Corpora for Indian Languages. Information Systems for Indian Languages. Communications in Computer and Information Science. Volume 139, pp 312-313.
- Pollmächer, J. (2011). Separierung mit FindLinks gecrawlerter Texte nach Sprachen. Bachelor Thesis, University of Leipzig.
- Quasthoff, U.; Biemann, C. (2006). Measuring Monolinguality. Proceedings of LREC-06 workshop on Quality assurance and quality measurement for language and speech resources.
- Sharoff, S. (2006). Creating general-purpose corpora using automated search engine queries. In M. Baroni and S. Bernardini, editors, WaCky! Working papers on the Web as Corpus. Geddit, Bologna.

Language (ISO 639-3)	News	Wikipedia	For comparison: Emille
asm	-	-	100,095
ben	109,855	240,128	153,948
guj	848,723	-	601,947
hin	5,162,167	727,882	469,395
kan	-	389,395	76,445
kas	-	-	11,858
mal	216,788	185,928	75,645
mar	774,201	149,420	96,296
ori	-	-	80,262
pan	507,059	-	429,948
pnb	-	39,606	8,587
sin	-	-	287,554
tam	1,341,954	-	1,298,802
tel	326,233	430,723	198,669
urd	1,733,995	144,312	60,903

Table 1: Amount of available text resources in number of sentences

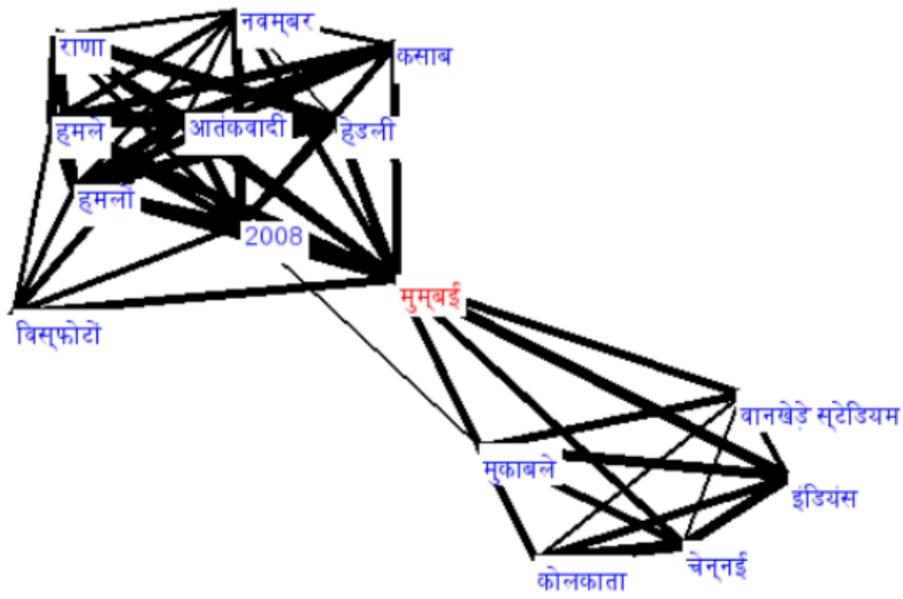


Figure 1: Word co-occurrence graph

Mapping Indian Languages onto the IMAGACT Visual Ontology of Action

M. Moneglia¹, S. W. Brown¹, A. Kar², A. Kumar², A. K. Ojha², H. Mello³, Niharika², G. N. Jha²,
B. Ray², A. Sharma²

¹UNIFI, Florence, ²J.N.U., New Delhi, ³UMFG, Belo Horizonte

E-mail: ¹moneglia@unifi.it, ²girishjha@gmail.com, ³heliana.mello@gmail.com

Abstract

Action verbs have many meanings, covering actions in different ontological types. Moreover, each language categorizes action in its own way. The range of variations within and across languages is largely unknown, causing trouble for natural language processing tasks and second language acquisition. IMAGACT is a corpus-based ontology of action concepts, derived from English and Italian spontaneous speech resources, which makes use of the universal language of images to identify action types. IMAGACT4ALL is an internet infrastructure for mapping languages onto the ontology. Because the action concepts are represented with videos, extension into new languages is done using competence-based judgments by mother-tongue informants without intense lexicographic work involving underdetermined semantic description. It has been already proved on Spanish and Chinese and it is now in the process of being extended to Hindi, Bengali, Sanskrit and Portuguese. The paper presents the infrastructure and the methodology for mapping languages onto the ontology focussing on the features that make it a promising infrastructure for processing Indian languages.

Keywords: Action Ontology, Cross-linguistic mapping, Visual representation

1. Introduction

Action verbs, which are highly frequent in speech, are often highly polysemous; that is, they often have many meanings, covering actions in different ontological types. Moreover, each language categorizes action in its own way, so accurate machine translation is difficult. IMAGACT is a cross-linguistic ontology of action concepts that are represented with prototypic 3D animations or brief films. This format makes use of the universal language of images to identify action types, avoiding the under-determinacy of semantic definitions. This ontology has been induced from the references to physical actions found in English and Italian spoken corpora (Moneglia et al. 2012a) and gives a picture of the variety of activities that are prominent in our everyday life, specifying the language used to express each one in ordinary communication.

A dedicated interface, IMAGACT4ALL, makes extension of the resource to languages other than English and Italian easy. Starting from this universe of prototypic scenes it is possible to extend the linguistic correlation to any language through competence-based judgments instead of corpus induction. Using this interface, the action concepts in IMAGACT have already been extended to Chinese and Spanish using competence-based judgments from native speakers. Once mapped onto the ontology, each language can be compared to every other. The largely language-independent extension interface and the image-based format of the ontology make IMAGACT a valuable resource for under-resourced languages.

The complex situation of the languages of India, where 22 languages belonging to four different families have 'national status', is a challenging opportunity for IMAGACT4ALL. Once each one each of these languages is mapped to the action ontology images, a language instantly has accurate translations to every other language implemented in IMAGACT and a means of comparing the semantic categorization of action with any other

language. These abilities can benefit both natural language processing and foreign-language learners, who need to acquire vehicular languages like Hindi, English and Sanskrit.

IMAGACT4ALL is currently being used for two main initiatives. The first one specifically concerns Indian languages. The first three Indian languages are in the processing phase: Hindi, Bengali, and Sanskrit. A second initiative concerns Romance languages, with Portuguese in both its Brazilian and European varieties as the current extension beyond the existing Italian and Spanish components. Polish and other Slavonic languages will soon follow.

In the rest of this paper, we will sketch the construction of the ontology and describe the extension interface.

2. IMAGACT

IMAGACT has been developed through corpus-based annotation by mother-tongue linguists. Working from English and Italian spoken corpora, we identified the variation of action-oriented lexicons across different action concepts. 521 Italian verbs and 550 English verbs (i.e., the high-frequency verbal lexicon most likely to be used when referring to action) have been processed.

The corpus-based strategy relied on an induction process that separated the metaphorical and phraseological usages from physical action occurrences and then classified the action occurrences into types, keeping granularity to its minimal level. The positive selection of occurrences in which verbs refer in their own meaning to physical actions preceded the annotation.

The annotation consisted of two stages, leading from occurrences of each verb in a language corpus to the identification of the action types in which the verb occurs, and then to the validation of the generated typology of actions productively referred to by the verb. The possible variation of each language verb found in the corpus has been made explicit by gathering occurrences under prototypes. Each verb can express one or more concepts,

while each concept can refer to one or more verbs. (Moneglia et al. 2012a).

The key innovation of IMAGACT is to provide a methodology that exploits the language-independent capacity to appreciate similarities among scenes, distinguishing the identification of action types from their definition. Crucially, only the identification (and not the active writing of a definition) is required to set up the cross-linguistic relations.

In IMAGACT the ontology makes use of the universal language of images, which allows the reconciliation in an ontology of the types derived from the annotation of different language corpora. 1010 distinct action concepts have been identified and visually represented with prototypical scenes, either animated or filmed (Frontini et al. 2012; Moneglia et al. 2012b). The cross-linguistic correspondences of those actions with the verbs that can refer to them in English and Italian have been established in a MYQL database.

different languages categorize the universe of action. For instance, Figure 1 shows the variation of the English verb to *attach* across different type of actions.

More specifically after the corpus-based induction, competence-based extension to other languages is possible through the interface IMAGACT4ALL.

Chinese and Spanish informants have mapped the action verbs of this language onto the ontology through this interface and the correspondences are now in the data base, allowing comparison between the action oriented lexicon of English, Italian, Spanish and Chinese. For instance, *attach* can be compared to its direct translation in Chinese (*tiē*) (Figure 2), and the different range of possible extensions becomes clear. *Tiē* extends to a larger set of 'sticking activities', but cannot be extended to the actions in which objects are 'connected'.

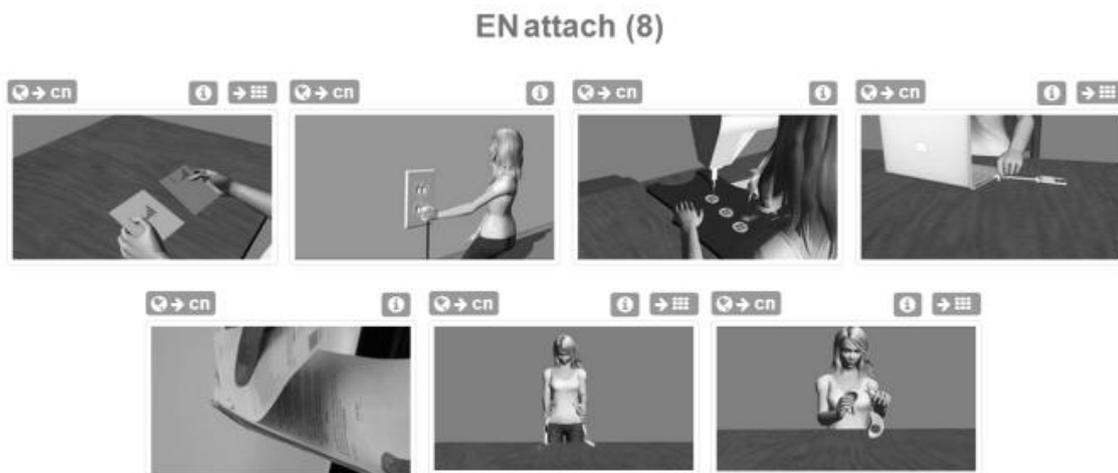


Figure 1 the variation of to *attach* in English

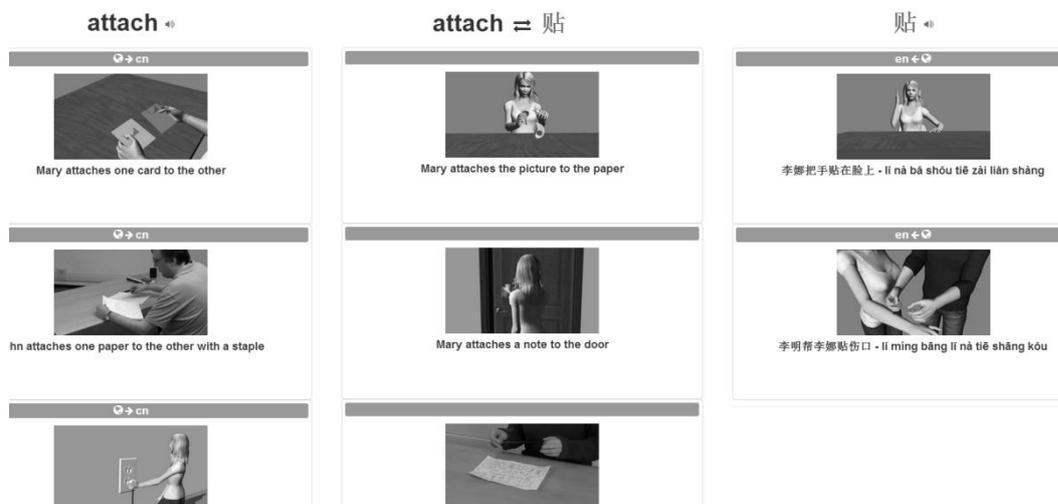


Figure 2 to *attach* vstiē

By comparing the variation of verbs within and across languages it is now possible to observe how verbs of

3. IMAGACT4ALL

We have created an infrastructure with a clear user interface, called IMAGACT4ALL that has already been successfully used to incorporate Spanish and Chinese into IMAGACT. This section describes the implementation procedure for mapping any language onto the IMAGACT ontology using this interface

3.1 Overview of the informant's work (CBE-Light)

The work of competence-based extension (CBE) of IMAGACT to a new language is performed by native-speaker informants. These informants receive a username and password that authorize them to process data for their language only. More than one informant can work simultaneously; however, the infrastructure in this release does not provide an explicit comparison of their work or an interface for conflict resolution. The work is in the responsibility of one supervisor (*superCBE*) who must give his OK before delivery.

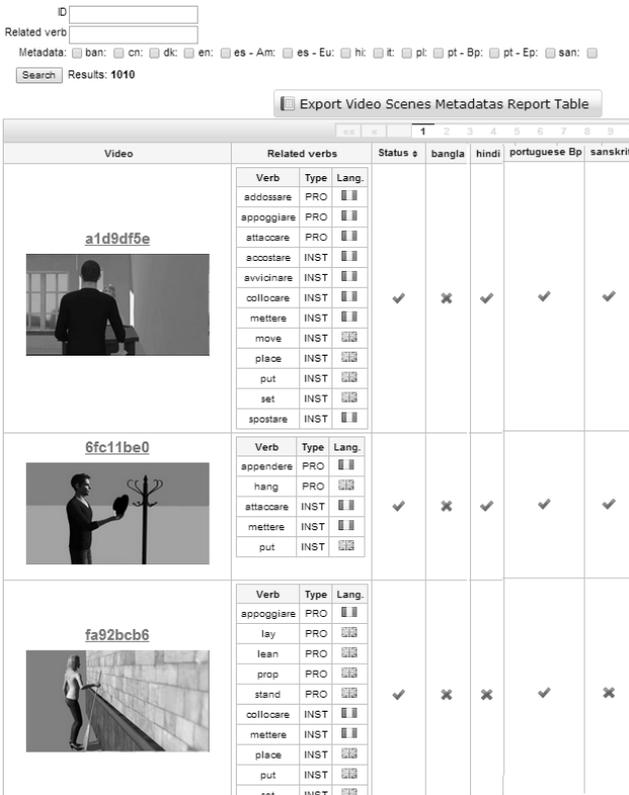


Figure 3. IMAGACT4ALL interface

The competence-based extension interface presents the IDs and thumbnail images of all the scenes, each one identifying a separate action concept in the ontology (Figure 3). On the right side of each scene the informant sees the English and Italian verbs mapped onto that scene in the corpus-based annotation. Moreover, he may see the processing status of each scene in the languages that have been opened for processing (here, those opened now).

In order to start processing, the informant clicks on the ID to access the full set of metadata for that concept (Figure 4). The language for each line is identified by a flag and a standard code. The informant can only modify information pertaining to his own language.

The first necessary operation is to properly appreciate the action represented in the video. The informant plays the video by clicking on it and observes the represented action.

Once he understands the intended action, the informant provides the verb or verbs in his language that can be used to refer to that specific action. The lemma should be annotated in its infinitival form, as it is commonly reported in dictionaries. The informant annotates this lemma (or lemmas) in the box corresponding to his language on the bottom of the page.

For each lemma he then writes in the caption box a simple sentence in the present tense, filling all the arguments of the verb that properly describe the action. This sentence will be used as the caption of the scene in the language of the informant. Both the verb and the caption should be written in the current writing system of the language of the informant. If this system does not use roman characters, after the annotation in the current writing system, the informant also provides the verb and its caption in roman characters, as can be seen for the Chinese Sanskrit and Hindilemmas in Figure 4.

Corpus verbs	Type	Script	Lang.	Caption
appendere	PRO	Marco appende la giacca all'attaccapanni	🇮🇹	Fabio appende il cappello all'attaccapanni
hang	PRO	John hangs the coat on the hook	🇺🇸	John hangs the hat on the hook
attaccare	INST	Fabio attacca il cappotto all'attaccapanni	🇮🇹	Fabio attacca il cappello all'attaccapanni
mettere	INST	Marco mette il quadro alla parete	🇮🇹	Fabio mette il cappello all'attaccapanni
put	INST	John puts a clamp on the door	🇺🇸	John puts a hat on the hook

ID: 6fc11be0



Assigned verbs					
Verb	Transliteration	Rejected	Lang.	Caption	Transliterated caption
挂	guà	<input type="checkbox"/>	🇨🇳	李明把帽子挂在衣帽架上	lǐ míng bǎ mào zi guà zài yī mào jiǎ shàng

Competence based verbs				
Verb	Transliteration	Lang.	Caption	Transliterated caption
meter		🇪🇸	Fabio mete el sombrero en el perchero	
colgar		🇪🇸	Fabio cuelga el sombrero en el perchero	
poner		🇪🇸	Fabio pone el sombrero en el perchero	
wieszać		🇵🇱	Jacek wiesza kapelusz na wieszaku	
powiesić		🇵🇱	Jacek powiesił kapelusz na wieszaku	
zawiesić		🇵🇱	Jacek zawiesił kapelusz na wieszaku	
pendurar		🇵🇹	Fabio pendura o chapéu no cabide.	

टांगना	TAMganA	🇮🇳	Hi	माँ कपड़े टांग रही है	mA.n kapa.De TAMga rahl hai
लटकाना	laTakAnA	🇮🇳	Hi	माँ कपड़े को खूँटी पे लटका रही हैं	mA.n kapa.De ko khU.nTI pe laTakA rahl haiM
सञ्ज	sa-nj	🇮🇳	Sa	रामः शिरस्त्रं प्रसजति	rAmAH shirastraM prasajati
क्षिप्	kShip	🇮🇳	Sa	रामः शिरस्त्रं क्षिपति	rAmAH shirastraM kShipati
संलग्	saMlag	🇮🇳	Sa	रामः शिरस्त्रं संलागयति	rAmAH shirastraM saMIAGayati
स्था	sthA	🇮🇳	Sa	रामः शिरस्त्रं स्थापयति	rAmAH shirastraM sthAPayati

Figure 4. IMAGACT4ALL interface for individual scenes

3.2 Ambiguity and supervision

Although the scenes in IMAGACT have been scripted and filmed in order to be as unambiguous as possible and to convey a single meaning, some ambiguity is still possible and can lead to misunderstanding or over-interpretation by the informant. To avoid this risk, the informant must reach a clear understanding of the specific action that we want to represent on each occasion. For instance, for the scene in Figure 5, an English informant could say that the represented action is not only *to lead*, but also *to help* or *to cross*.

In Figure 6, in addition to *taking* and *bringing*, we could also say that an act of *giving* occurred (*dar* in Spanish). However, these alternative categorizations fall outside the domain of IMAGACT. The informant should identify the specific physical action that is intended by the film. (For example, *to help* is not a physical action). Moreover, to identify the action, one needs to distinguish it from its consequences. For instance, *crossing* and *giving* are consequences of *bringing*, *taking* and *leading* and are beyond what is intended by this video.



Figure 5. The English verbs mapped to a scene



Figure 6. The English and Spanish verbs mapped to a scene

The possible alternatives must always be verbs that identify the physical action in a *simultaneous* manner (*giving* is not simultaneous to *bringing*). For this reason the lemmas chosen by informants must be evaluated by a supervisor to ensure the appropriate interpretation for the goals of IMAGACT. Given the need for informants to understand this strict categorization of physical action, we have chosen not to use crowd-sourcing as a means of gathering data.

3.3 Cross-linguistic aids for informants

To provide reliable competence-based judgments, the informant must be a native speaker of the language she is implementing. It is possible for the annotation to be completed without any reference to other languages; however, it is very helpful for the informant to also know (at least at a very basic level) English or one of the languages already implemented in IMAGACT (Italian, Spanish, or Chinese). As the number of implemented languages grows, the options for the second language of course increases as well.

This is necessary because verbs with different meanings can identify the same action. For instance, in English the action in Figure 6 can be identified with either *to take* or *to bring*. This is also true in Spanish, which picks up this action with the verbs *llevar* and *alcazar*. Therefore, the informant is asked to find multiple lemmas allowed by her language for each action. The informant is not required to provide the complete range of possible verbs in her language that can refer to that action, but only to consider this possibility.

However, simply viewing one film may be not sufficient to elicit all alternatives. The infrastructure provides one simple means to stimulate the thinking of the informant. Frequently, in the corpus-based annotation, more verbs have been found in English and Italian that can fit with the represented scene. Moreover, in the previous competence-based extensions, other alternatives in many languages have been provided. These alternatives should function as suggestions for figuring out alternatives in the language of the informant. This opportunity should be exploited in a systematic manner during the workflow after the first lemma has been determined.

For instance, the first verb for the action in Figure 7 that occurred to the Spanish annotator was the verb *colgar* (*hang*). However, looking at the English annotation, she could see that this event can be also identified by the verb *to put* in English. With this clue, the Spanish informant could figure out that in Spanish other verbs that are similar to *put* can be properly applied (*meter*, *poner*). The annotator is only requested to judge whether or not the alternatives suggested by other languages have translations in her language that can be used in referring to the event into question. If so, she will report a new verbal lemma and a new caption by adding a line to her language options.

4. Conclusion

IMAGACT offers a promising language-independent infrastructure for incorporating under-resourced languages into an extensive action ontology. Because the action concepts of the ontology are represented with videos rather than verbal definitions, extension into other languages is done without intense lexicographic work involving underdetermined semantic description. In addition, network effects promise that the more languages that are implemented, the greater the rewards. Once a language has mapped its verbs to the action images, it is instantly connected via those images to every other implemented language. Language pairings that normally would not have extensive and detailed verb correspondences in a resource (e.g., Bengali and Portuguese, or Polish and Mandarin) would suddenly have them.

5. Acknowledgements

The IMAGACT project is funded by the PAR/FAS program of the Tuscan region in Italy

6. References

- IMAGACT. <http://www.imagact.it>
- Moneglia, M. (2011). Natural language ontology of action: A gap with huge consequences for natural language understanding and machine translation. In Z. Vetulani (ed), *Human Language Technologies as a Challenge for Computer Science and Linguistics*. November 25-27, Poznań, Poland.
- Moneglia, M.; Monachini, M.; Calabrese, O.; Panunzi, A.; Frontini, F.; Gagliardi, G.; and Russo, I. (2012). The IMAGACT cross-linguistic ontology of action: A new infrastructure for natural language disambiguation. In N. Calzolari, K. Choukri, T. Declerck, M.U. Doğan, B. Maegaard, J. Mariani, J. Odičkand S. Piperidis (eds), *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*. European Language Resources Association (ELRA): Paris, pp. 2606-2613.
- Moneglia, M.; Gagliardi, G.; Panunzi, A.; Frontini, F.; Russo, I.; and Monachini, M. (2012). IMAGACT: Deriving an action ontology from spoken corpora. Paper presented at the *Eight Joint ACL - ISO Workshop on Interoperable Semantic Annotation (ISA-8)*. Pisa, October 3-5, 2012.
- Frontini, F.; De Felice, I.; Khan, F.; Russo, I.; Monachini, M.; Gagliardi, G.; and Panunzi, A. (2012). Verb interpretation for basic action types: Annotation, ontology induction and creation of prototypical scenes. Paper presented at the *CogAlex-III Workshop* as part of the COLING 2012 conference. Mumbai (India), December.

Handling Conflational Divergence in a pair of languages: the case of English and Sindhi

Pinkey Nainwani

Centre for Linguistics, Jawaharlal Nehru University, New Delhi, India

pinkeybhu39@gmail.com

Abstract

This paper discusses the nature of conflational divergence with reference to English and Sindhi. Dorr (1993) explained that a translation divergence arises when the natural translation of one language into another produce a very different form than that of the original. She demonstrated seven types of lexical-semantic divergences. One of them is conflational divergence which 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. Further, the paper describes the theoretical description of conflational divergence with reference to compound verbs, complex predicates, causative verbs, infinitival structures and many more. Due to the language complexities involved, I have adopted (S)tatistical (M)achine (T)ranslation approach to train English-Sindhi and Sindhi-English (parallel) corpora. In addition, it tries to illustrate to which extent SMT is able to capture sub-categorization of conflational divergence automatically.

Keywords: divergence, conflational divergence, SMT, machine handling, evaluation

1. Introduction

Machine Translation (MT) is the process of automated translation of text from source language (L_1) to target language (L_2). The outputs of MT systems have had known for their limitations in terms of translation quality, accuracy, coverage etc and have therefore been sparsely used for translating documents requiring quality. In order to achieve a desirable degree of accuracy, an MT system must capture language-independent information. Though, language understanding, language generation and mapping between language pairs are linguistic challenges in developing an MT system. Divergence is one such challenge which 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). However, there are instances where the structure of the translated sentences deviates from the normal structure and then these exceptions are called translation divergences. To obtain correct translation, one needs to examine the different grammatical as well as some of the extra grammatical characteristics to identify the types of divergences exhaustively.

Sindhi¹ belongs to an Indo-Aryan language family. The linguistic features that differentiate Sindhi from English can be divided into two broad categories: structural differences and stylistic differences. The major structural differences are English has SVO sentence pattern, known

as positional language, therefore fixed-order language, and the modifier of the object can precede or follow the object. On the other hand, Sindhi has SOV pattern, relatively free-order language, and the modifier of the object usually follows the object. There are minor stylistic differences too: many transitive verbs of English map to intransitive verbs of Sindhi and the modifiers like, *no*, *few*, *any*, or *little* followed by a noun in English which is not always the case in Sindhi.

There are various MT systems have been built for Indian languages with domain specific but a significant amount of work is still left as far as language divergences are concerned. The need of an hour is to build an MT system which can efficiently handle the divergences. The catering to divergences would eventually surmount the gap between machine assisted translation and machine translation.

2. Theoretical Background

Dorr ('90, '93, and '94) has discussed 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. Moreover, she has also 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: 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

¹ It is an official language of Sindhi (the Pakistan province). In India, it is one of the scheduled languages officially recognized by the federal government.

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).

Though, Gupta (2005) and Sinha et al. (2005), earlier remarked that the classifications of translation divergences as proposed by Dorr ('90, '93, and '94) are not sufficient to capture translation divergences between English and other Indian languages (specifically Hindi). They have observed the following list (though not exhaustive) of translation divergences which needs to be taken care of as far as English-Indian languages and vice-versa MT is concerned: 1) Reduplicative Words (repetition of root words to emphasize the context) 2) Determiner System (some Indian languages lack an overt article system whereas English has (in)definite articles that mark the (in)definiteness of the noun phrase exactly, 3) Conjunctions and Particles (Indian languages have different types of particles like “wAlA, na”, there is no exact counterpart in English), 4) Gerunds and participle clauses (The adjunct verbal clauses and complement verbal clauses in mostly Indian languages are realized by infinitival clauses in English), and 5) Honorific (Indian languages employ several linguistic markers such as plural pronouns and plural verbal inflections which are missing in English).

This paper is an attempt to expand the definition of conflation divergence which will encompass the other classification of verbs such as compound verb, complex predicate, causative verb, infinitival verb, exceptional case marking verb, passive verb, and phrasal verb. Verb is a key component in the sentence as it displays the action, tense, mood, and aspect. In most Indian languages, verb inflects for agreement feature with respect to gender, number, person, and honorificity.

It is essential to understand the minutiae of each classification of verb and their syntactic and semantic implications in English-Sindhi translation pair.

2.1 Compound verb

Compound verb is composed of two verbs i.e. V1+V2. The first component of the verb (V1) carries the semantic information and determines its arguments. It is usually in either root or conjunctive participle form. The verbs playing as V2 have been variously referred to as auxiliary, operator, explicator, vector, semantically weak verb, and linguistically known as Light Verb (LV). It carries inflections, indicating tense, mood, and aspect. Hook (1974) has reported that the list of light verbs ranges from 8 to 61.

These are some examples from Sindhi²:

- (1) E: John washed the clothes
S: jon laTA dhoi chaddayA
{John clothes wash leave}
LV= *chaddayA* {to leave}
- (2) E: He did the work
S: huna kamu kare varto
{He work did take}
LV= *varito* {to take}
- (3) S: hUa kitAbu khaNI AI
{She book brought come}
E: She brought the book.
LV= *AI* {to come}

These light verbs can also behave as a main verb but in compound verb constructions it entirely depends on the semantics of the preceding verb. In Sindhi compound verb constructions, V1 never occurs in root form and ends with vowel endings such as *i*, and *e* due to morpho-phonemic nature of the verb. In (1) and (2) constructions, explicator verb agrees with object whereas in the (3) construction, it agrees with the subject. The random nature of Sindhi verbs is difficult to capture via rules. On the contrary, the translations of compound verbs from Sindhi to English do not pose major challenges for MT provided that English to Sindhi verb dictionary has elaborated meaning.

2.2 Complex Predicate

A complex predicate consists of a semantically empty verb which expresses the grammatical meaning in a sentence and a noun which carries the lexical meaning of the entire phrase. The occurrence of complex predicates (also known as complex verbs) is high in South Asian languages, usually present in the form of noun+verb combinations. Conflation divergence subsumes the linguistic definition of complex predicate which says 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 below:

- (4) E: He stabbed me
S: huna mU-khe churI-sAM mArayo
{he me knife- kill}
- (5) E: He blessed me
S: huna mu-te AsIsa kaI
{he me bless did}
- (6) E: He cleaned the room
S: huna kamiyo sAf kayo
{he room clean did}

Linguistically, this is an interesting question for computational linguistics and natural language processing that in complex predicate construction, is the noun incorporated into the verb or is it an overt argument of the verb. The behaviour of complex predicates is not challenging as they can be easily translated given these words in the bilingual dictionaries having compositional meaning. The only problem exists for construction like (4)

² Sindhi has four implosive sounds; the doubling of letters in Sindhi examples reflects the same

where the postposition gets inserted between noun and verb.

2.3 Causative verb

Causative verb indicates an action where subject causes some other agent to perform the action or causes a change in state of a non-volitional event. Languages like Hindi, Sanskrit, Sindhi, and Finnish are rich in causative verbal form and employ morphological devices (affixation or inflection) that change verb into their causative verbs. In English, causatives are generally expressed by verbs like ‘make’, ‘get’ and ‘have’. Semantically, causative verbs also increase the valency (number of arguments required by the verb); the single causative should have three nouns and double causative should have four nouns.

For example

(7) E: She made me do her homework
S: huna mU-sAM pahiMjo kamu karAyo
{she me her work made do}

(8) E: I got him to repair my car
S: mU huna-sAM pahiMjI kAr thIk karAI
{I him my car repair got}

In above Sindhi examples, -A affix is attached on the left side of the root verb to get the causal meaning whereas –yo and –I are the markers of past tense for masculine and feminine gender respectively. Sindhi uses the same –A affix to express the single and double causative forms of the verb. These causative constructions pose big challenge for English-Sindhi and Sindhi-English MT pair as discussed in the subsequent section.

2.4 Infinitive verb

Infinitive verbs are mostly defined as non-finite verbs, with or without the particle to. These forms of verbs serve multiple functions:

(I) To-infinitive as a modifier of main verb:

(9) E: I would love to go to Bombay
S: mU bambai vanaNa pasand kaMdum
{I Bombay to go love would}

(II) To-infinitive related with adjective modified by “too”:

(10) E: Bombay is too expensive to stay
S: bambai rahaNa lAi ddadho mahANgo Ahe
{Bombay staying for very expensive be}

It can be observed that the infinitive forms of verbs in English get translated as a gerundial form (-Na) of the verb in Sindhi. The following examples show the semantic differences between gerundial and infinitive constructions:

(III) Gerundial Construction:

(11) E: I like singing
S: mU-khe ggAyaNa pasand Ahe
{I singing like be}

(IV) The infinitival construction:

(12) E: I love to sing
S: mU-khe ggAyaNa sutho laggado Ahe
{I singing like feel be}

The semantic difference between the above two sentences is in (11) the person appreciates singing whether (s)he knows it or not whereas in (12), the person knows singing and (s)he usually does it in spare time.

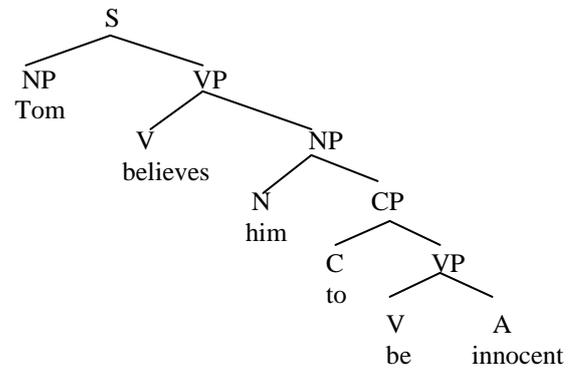
2.5 ECM verb

Exceptional Case Marking (ECM) denotes a phenomenon where the subject of the embedded infinitival verb gets its case from the main verb of the principal clause. The number of ECM verbs in English is small such as *want*, *prove*, *believe*, *judge*, and *appear*. The mapping of ECM verb from English to Sindhi differs; it takes two verbs [+tense] for both the clauses (principal and subordinate) in Sindhi. Moreover, the translation of above mentioned verbs in Sindhi doesn’t retain ECM construction anymore. Example:

(13) E: Tom believes him to be innocent
S: Tom vesAhu kaMdo ta hU begunAha Ahe
{Tom believes does that he innocent is}

“Tom” gets the case from its local verb “kaMdo” (does) and “hU” get the case from “Ahe” (is). Henceforth, all ECM constructions of English while translating into Sindhi change into complementizer constructions.

The underlying phrase structure construction of ECM verbs³:



The interesting phenomenon is Sindhi or various other Indian languages do not have these kinds of ECM verbs and therefore these exceptional case marking constructions do not arise and make the translations from Indian languages to English comparatively easier.

2.6 Passive Verb

The definitional property of passive verb has been assumed to be the inversion of the syntactic roles in

³ The object and subject NP as shown as the dependent on the matrix verb as the verb of the embedded clause is {-tense}; not able to provide the case to its subject.

actives and passives where the object gets promoted to the subject and the subject gets demotion to an adjunct. Unlike English, Sindhi uses passive verbs less frequently. In English, the passive uses specific form of the auxiliary ‘to be’ and the past participle form of the verb. The passive is important as it serves variety of functions including focusing on the object, demoting the subject, and suppressing the information of the doer.

Given are some examples:

(14)E: The elephants have been caught
 S: hAthI pakiDAIMjI vayA
 {elephant caught went}

(15)E: Blankets were taken out
 S: kambal nikArA vayA
 {blankets taken out went}

The verb, *vana*, ‘to go’ in its past perfective form *vayA*, ‘went’ is being consistently used and collectively marked for person, number, and gender. Sindhi also often uses the suffix *-IMjI* (also known as passive suffix) to the first verb to denote the action has taken by someone who is not known to speaker.

2.7 Phrasal Verb

A Phrasal verb contains a head verb followed by one or more particles and the meaning of the phrasal verb cannot be determined by combining the meaning of its constituents. Therefore, the meanings of phrasal verbs are semantically non-compositional, have fixed expressions with figurative rather than literal meaning and moreover they are morpho-syntactically irregular. Examples:

(16)E: You should stand by your father
 S: tokhe pahIMje pIu jo sAth ddiyoNo khape
 {you your father support to give should}

Here, the expression ‘*stand by your father*’, got translated into Sindhi as ‘should give support to your father’. These phrasal verbs are difficult to translate for any language pair and require lot of computational resources (such as Hybrid MT or multi-engine MT) to process them.

3. System Training

SMT has gained tremendous momentum in recent years. Generally, languages are so rich and complex that it becomes difficult to distil knowledge and frame rules which can be encoded through programs. SMT was first pioneered by IBM in late 1980s. The probability of SMT systems is calculated based on two components: (1) A language model that assigns a probability ($p(s)$) for any sentence in the source language and (2) A translation model that assigns a conditional probability ($p(t|s)$) to any source/target language pair of sentences. In order to develop an MT system a representative parallel corpus (20k) from English to Sindhi (English being a source language) has been built. Corpus is collected from sources like: a) High school grammar books, b) short stories, and children story books.

Table 1 contains the statistics about the size of parallel corpus. A word in this corpus is a white-space separated token and the sentence length is measured in words

	Sentences	Words	Avg. Sent. Length
English	20000	141797	7.08
Sindhi	20000	164797	8.24

Table 1: Parallel corpus size statistics

Table 2 displays the number of above mentioned verb groups in the corpus

Verb Groups	Counts
Compound verb	5359
Complex Predicate	6122
Causative	843
Infinitive	2962
ECM verb	2089
Passive	1013
Phrasal verb	1612

Table 2: Number of verb groups in corpus

To train these English to Sindhi and Sindhi to English systems, I have used Microsoft model (developed at Microsoft Research Lab, Redmond, under Microsoft Translator Hub)⁴. Their approach to MT is data-driven rather than writing explicit rules, they train their algorithms on human translated parallel texts which allow the algorithm to automatically learn how to translate. The language modelling is based on n-gram model which is key component in high-quality SMT. The goal is to create the simplest, most intuitively integrated and useful translation services available to end users while making ongoing improvements to translation quality.

Aligning Parallel corpora (Moore 2002) have been proved very crucial for applying machine learning to MT. Microsoft Model does sentence and word alignment in a three step process:

- Find the sentence pairs that align with the highest probability without the use of anchor points,
- Using the sentence pairs that have been assigned the highest probability of alignment to train an updated version of IBM Model,
- Finally, realigning the corpus, augmenting the initial alignment model (IBM Model) to produce an alignment based both on sentence length and word correspondences

This alignment method is similar to Wu (1994) which uses both sentence length and lexical correspondences to derive the final alignment; these lexical correspondences

⁴ <https://hub.microsofttranslator.com>

are derived automatically and therefore, no external lexicon is required.

English-Sindhi System Training

Total Manual Parallel Sentence count	20000
Extracted Sentence by Microsoft Model	20059
Aligned Sentence	19419
Actual in Training	18377
Testing	500
Tuning	500
Monolingual	25090
BLEU	14.30

Table 3: English-Sindhi training statistics

Sindhi-English System Training

Total Manual Parallel Sentence count	20000
Extracted Sentence by Microsoft Model	20160
Aligned Sentence	19426
Actual in Training	18147
Testing	637
Tuning	637
Monolingual	24614
BLEU	27.23

Table 4: Sindhi English training statistics

The above given training statistics shows the discrepancy between total manual parallel sentence count and the extracted sentence count by Microsoft Model due to different sentence terminators such as ., ;, ! which often brake single sentence into two if these terminators are used in between of the sentences for cases like *Mr. Ramesh and Very good! boys*, and therefore, reduces the number of aligned sentences. The bilingual data-set finally used for training removes duplicates if there is any.

a. Data Analysis

This subsection discusses the outputs given by machine after deployment and compares the human translations (HT) and machine translations (MT) with respect to each verb group classification:

3.1.1 English to Sindhi

Compound verb

E: John washed the clothes

S: jon laTA dhoi chaddayA [HT]
{John clothes wash leave}

jon laTA dhota [MT]
{John clothes wash}

Complex Predicate

E: He stabbed me

S: huna mU-khe churI-sAM mArayo [HT]
{he me knife-from killed}

huna mu-khe churI mArI [MT]
{he me knife kill}

Causative

E: They showed a Sindhi film after lunch

S: hunin maMjhidi ji mAnI poi sindhi film ddekhAi [HT]
{they lunch after Sindhi film showed}

uhe sindhi film jekA khA poi mAnI ddekhAyo [MT]
{theySindhi film who from after lunch showed}

Infinitive

E: I had no need to open the letter

S: mU-khe khatu kholaNa ji ko jarurata kona huI [HT]
{I letter to open of any need not had}

mU-khe khatu kholaNa ji jarurata kona huI [MT]
{I letter to open of need not had}

ECM

E: I want you to shut all doors and windows

S: mU cAhiMdo AhiyA ta tU sajja dara aim darIyUM [HT]
{I want am that you all doors and windows

baMdi kara [HT]
{shut do}

mU tU sajja dara aim darIyUM baMdi karaNa

{I you all doors and windows shut do

cAhiMdo AhiyA [MT]

{want am}

Passive verb

E: The elephants have been caught

S: hAthI pakiDAIMji vayA [HT]
{elephants caught have been}

hAthI pakiDAIMji vayA [MT]
{elephants caught have been}

Phrasal verb

E: I gave him no cause to break with me

S: mU huna-khe paNa sAM ladaNa jo ko mOko kona [HT]
{I him self from to fight to any chance not

ddino [HT]
{gave}

mU huna-khe acI ji ko vajaha kona ddini [MT]

{I him come to any reason not gave}

It is interesting to note that in the case of causative, machine could not translate adequately the subject NP into its oblique form and makes the translation somewhat ungrammatical, though little meaning is conveyed to native speakers. On the other hand, in case of special determiners such as ‘no’ (crucial aspect for MT) got correct translation in phrasal verb.

3.1.2 Sindhi to English

Compound verb

S: hUa kitAbu khaNI AI

{she book brought came}

E: She brought the book [HT]

She brought the book [MT]

Complex Predicate

S: rAM pahiMje paNa-khe dhaku haNo Ahe
 {Ram himself hurt has be}
 E: Ram has hurt himself [HT]
 Ram has hurt himself haNo [MT]

Causative

S: mU pahiMjA laTA pres karAiaNa cahiMdo AhiyAM
 {I our clothes iron to get want be}
 E: I want to get my clothes ironed [HT]
 I want to get my clothes pres them [MT]

Infinitive

S: huna mu-khe kamire sAM vanaNa lAi cayo
 {he me room from leave to said}
 E: He told me to leave the room [HT]
 He told me to go to the room [MT]

Passive verb

S: hAthI pakidAIMjI vayA
 {elephant caught have been}
 E: The elephants have been caught [HT]
 The elephant caught [MT]

Phrasal verb

S: hu paka I tAj poshIa lAi thlku kona Ahe
 {he certainly lead to fit not be}
 E: He is certainly not fit to lead [HT]
 He certainly is not well for the ceremony [MT]

Sindhi to English MT does not have much translation mapping issues. In the above examples, SMT has been resulted into an approximately a complete job; a smaller amount of human post-editing will get the qualitative results. Lexical variation according to context is a challenging task need to be handled by machines. In the case of infinitive, due to lexical variation, the English MT output (translated from Sindhi) cannot be considered perfectly adequate whereas the sentence is grammatically correct. Here, passive and phrasal verbs also require major human-editing on the grammatical aspect. In nutshell, Sindhi-English MT with respect to each verb group is comprehensible to a larger extent.

4. Discussion

A translation is true if it is faithful to the source language and fluent in target language and often it is impossible. There is no doubt that MT outputs require some human post-editing to achieve human-level quality. MT outputs have always been evaluated by humans on two parameters those are fluency and adequacy. I have also used these two parameters (fluency and adequacy) to analyze MT outputs with regard to each verb group⁵.

Fluency of the given translation is:

- (4) Perfect: Good grammar
- (3) Fair: Easy-to-understand but flawed grammar

⁵ Due to scarcity of Sindhi native speakers around, I myself have evaluated all MT outputs.

- (2) Acceptable: Broken - understandable with effort
- (1) Nonsense: Incomprehensible

Adequacy: meaning of the reference sentence is conveyed in the translation:

- (4) All: No loss of meaning
- (3) Most: Most of the meaning is conveyed
- (2) Some: Some of the meaning is conveyed
- (1) None: Hardly any meaning is conveyed

Table 5 displays the ranking of each verb group mapping from English to Sindhi MT on parameters of fluency and adequacy (based on 500 test set sentences)

	Fluency	Adequacy
Compound Verb	3	4
Complex Predicate	3	4
Causative	2	2
Infinitive	4	4
ECM	3	3
Passive	3	2
Phrasal Verb	1	1

Table 5: English-Sindhi MT Evaluation

As far as adequacy parameter is concerned for English-Sindhi SMT in relation to each verb group, the mappings of compound verb, complex predicate, and infinitive verb got the highest score of 4 (no loss of meaning) on scale (1-4). On the parameter of fluency, only infinitive verb mapping gets the highest score 4 (perfect: good grammar) on scale (1-4) and compound verb, complex predicate, ECM, and passive verb get the score of 3 (fair but flawed grammar). Causative and phrasal verb demand lot of attention from MT perspective. Moreover, causative is not a natural construction in English language and it is forcefully generated in a sentence with the help of light verbs like *make* and *get* and while translating them into any of Indian languages naturally would produce two verbs and make the structure grammatically unacceptable and henceforth meaning is broken. To deal with phrasal verb, a different kind of database should be built with all morphological variations and inflections.

Table 6 exhibits Sindhi to English MT each verb group mapping on parameters of fluency and adequacy (based on 637 test set sentences):

	Fluency	Adequacy
Compound Verb	4	4
Complex Predicate	3	4
Causative	3	3
Infinitive	4	3
Passive	3	3
Phrasal Verb	2	2

Table 6: Sindhi-English MT Evaluation

Interestingly, causative verb in Sindhi to English has given better results on parameters of fluency and adequacy, scored 3 and 3 respectively. Infinitive verb has scored highest (4) at Fluency and on the parameter of adequacy, compound verb and complex predicate have given better results. Complex predicate verb group from English to Sindhi and Sindhi to English has shown similar results on both the parameters. In Sindhi, the frequency of passive verb occurrences in corpus is highly low but surprisingly machine has produced better results for this verb group too. Doubtlessly, the phrasal verb group needs special care. To sum up, Sindhi to English MT outputs have retained meaning in translations but with some flawed grammar.

5. Conclusion and Future Work

Verb is a key aspect of any language as it is marked for tense, aspect, and modality features and each verb has certain sub-categorization features which allow the number and nature of other constituents that attach with the verb to form the clause. The aim behind the study of handling conflation divergence is to develop an efficient MT with little or no human intervention. There is not much accessible literature on translation between English and Sindhi and vice versa, particularly in the context of statistical MT.

Due to the differences in style and structure between English and Sindhi language pair; these translation divergence mappings become difficult to capture solely by SMT approach and requires a Hybrid MT approach to be plugged-in.

The promising avenue for further research could be building more and more parallel data for English-Sindhi; the bigger the corpus, the better the estimate!

6. Acknowledgements

I would like to thank Dr. Girish Nath Jha and Microsoft Translator Hub Team for providing me readymade MT models to conduct system training experiments for English-Sindhi language pair.

7. References

- Asamidinova, A. (2007). *Knowledge Base For Russian-English Machine Translation Divergences*. Doctoral dissertation. . Jawaharlal Nehru University, New Delhi.
- Choudhary, N. K. (2011). *Automatic Identification and Analysis of Verb Groups in Hindi*. Doctoral dissertation. Jawaharlal Nehru University, New Delhi.
- 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. (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*. Basil Blackwell, Oxford.
- Hook, P. E. (1974). The Compound Verb in Hindi. *The Michigan Series in South and South East Asian Languages and Linguistics: The University of Michigan*.
- Johannessen, J. B., Nordgård T. and Nygaard, L. (2008). Evaluation of Linguistics-Based Translation. In *Proceedings of the sixth conference on Language Resources and Evaluation (LREC'08)*. pp. 396-402.
- Koehn, P. (2010). *Statistical Machine Translation*. University Press, Cambridge.
- Moore, R. C. (2002) Fast and accurate sentence alignment of bilingual corpora. *Machine Translation: From Research to Real Users*, pp. 135-144.
- Sinha, R. M. K. and Thakur, A. (2005). Translation Divergence in English-Hindi MT. In *Proceedings of EAMT*. pp. 245-254.
- Smith, J. R., Quirk, C. and Toutanova, K. (2010). Extracting Parallel Sentences from Comparable Corpora using Document level Alignment. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pp. 403-411.

Wu, D. (1994). Aligning a Parallel English-Chinese Corpus Statistically with Lexical Criteria. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pp. 9-16.

Semi-automated annotated treebank construction for Hindi and Urdu

Jayendra Rakesh Yeka, Vishnu Ramagurumurthy, Dipti Misra Sharma

LTRC, IIIT-Hyderabad

Hyderabad, India

rakesh.yeka@research.iiit.ac.in, ramagurumurthy.vishnu@research.iiit.ac.in, dipti@iiit.ac.in

Abstract

In this paper, we speak about the structure and paradigms chosen for creation of the annotated corpora for Hindi and Urdu. We briefly talk about the Shakti Standard Format that was chosen to suit needs of Indian language dependency annotation. This paper aims to present a framework for the creation of annotated corpus. We proceed to discuss the methods of automation chosen to overcome the laborious and time-consuming process of corpora annotation. We present the methods chosen to overcome the errors and multiple analyses that result through the task of annotation. We also present various methods used, both manual and automated, to ensure the quality of the treebank. We finally report the current status of the annotated corpora.

Keywords: Annotated corpora, Annotation evaluation, Treebank, Hindi, Urdu

1. Introduction

Annotated corpora find their use in many NLP related applications like Parsing, Machine Translation etc. In the past, many annotated corpora such as Penn Treebank (Marcus et al., 1993) and Prague Treebank (Böhmová et al., 2003) have found prominence by aiding in many aforementioned NLP related applications. Indian languages inherently contain the property of free word order. This prevents from effectively expressing the analysis in a context free based annotation paradigm. To fully exploit the advantage of free word order in Indian languages, dependency based annotation paradigm has been preferred (Bharati et al., 1995).

After establishing a format like SSF to effectively represent both dependency annotation and context free grammar analysis for Indian languages and creating a fixed set of guidelines for annotation, focus has shifted to creation of annotated corpora. As part of multi-layered, multi representational treebank (Bhatt et al., 2009), the task of dependency annotation for Hindi is taken up at IIIT Hyderabad. Once the initial steps for creation of stable annotation framework for Hindi are completed, the task of creating an annotated corpora for Urdu has also been undertaken. In this paper, we talk about the pipeline framework chosen for creating the annotated corpora. We focus on its semi automated nature, where the data interacts with human side and machine side during its flow.

Keeping the focus of the paper on the structure and framework of the pipeline and its working, we only limit the scope of this paper to speak about system errors that are experienced during the functioning of the pipeline. However (Agarwal et al., 2012) gives a detailed explanation about different linguistic errors encountered because of the nature of modules in the pipeline, and also discusses methods of correction of errors of linguistic nature.

The rest of the paper is organized as follows. Section 2 gives an overview of SSF data representation format. Section 3 talks about the sources and the format of raw data. Section 4 gives a brief description of the tools used in the pipeline framework. Section 5,6 and 7 talks about the anno-

tation procedure, the pipeline framework and the different errors encountered and the steps taken to prevent them. In Section 8, we talk about several post-pipeline methods of quality assurance and corrections. Finally, we conclude by presenting the current status of the treebanks.

2. SSF

Shakti Standard Format - SSF in short is an annotated data representation format presented in (Bharati et al., 2007). It is based on XML syntax, but deviates from XML standards to facilitate better readability. SSF supports both phrase structured and dependency structured analyses and also facilitates representation of partial analysis of sentences. At text level SSF, has two major segments - header and body. The header contains information regarding the origin, creation, and distribution of the text enclosed in Creative Markup Language(CML) scheme.

The body contains uniquely identified blocks of text each representing a sentence and its analysis, enclosed in a *<Sentence>*XML tag. Every sentence block contains nodes separated by *newline* and each node contains 4 system properties separated by *tabs*, namely:

- **Address** - a unique value indicating the position of the node relative to chunk boundary
- **Token** - a value indicating any of chunk boundaries ((,)) or a word/numeric/symbol of the sentence
- **Category** - Category of the node which is either a POS tag or a Chunk label
- **Others** - Feature set corresponding to the specific node or corresponding to the chunk. It is absent for the end of chunk boundary -))).

SSF representation can be classified into two different types, depending on the values the four properties can take. They are:

2.1. Inter-Chunk dependency format

The chunk boundary “(” indicates the start of a new chunk and “)” indicates the end of the chunk. For the start of the chunk boundary “(”, the *Category* property is the chunk label and *Other* field contains information regarding the type of dependency relation with its parent and also a unique name formulated from the chunk label. In certain intermediate formats, the *Other* property also contains information about the head of the chunk. The end of chunk boundary “)” does not contain *Category* and *Other* properties. Additional information such as the type of voice in the sentence is also present.

For the remaining nodes, the *Category* property indicates the Part of Speech tag. The *Other* property contains several fields of analysis for each corresponding node. The *af* or abbreviated feature set contains the morphological analysis of the node, namely - root word/lemma, coarse POS tag, gender, number, person, TAM (tense-aspect-modality) and Vibhakti values in a comma separated format. The *Other* property also contains a unique node - *name* formulated from the token value and a relative position value - *posn* of the node in the sentence.

2.2. Intra-Chunk dependency format - Expanded

In the expanded format, the chunk boundaries are removed and information is added into the *Other* property of the node indicating the name of the chunk they are in. For each node, the *Other* property additionally contains information about dependency relation with the node’s parent and also information about the chunk label and the chunk type. Though the expanded format lacks the chunk boundaries, it contains all the information present in the inter-chunk format and more.

3. Raw Data

Conversation data is gathered from short stories written by Munshi Premchand, known as Premchand Kahani. The stories were extracted from the website <http://nandlalbharati.blog.co.in>¹, hosted by Nandalal Bharati.

Data corresponding to other domains is gathered from various sources like news paper articles and tourism websites. The data extracted from the sources then undergo a cleaning phase where the text is cleaned and filtered to ensure that only paragraphs and sentences are present in the text. Content corresponding to page formatting like headings, sub-headings, page numbers, chapter names, author names etc. are filtered out and stored in header section of SSF files as meta-tags. Due to varied sources and different writing formats, the text gathered undergoes necessary font conversions.

Data gathered from these sources not only is of varied fonts but also is in different encoding and formats. UTF-8 and UTF-16 are the most preferred encoding for Hindi and Urdu in text sources. Besides UTF many data sources also use WX format. WX format is an ASCII based transliteration scheme for Indian languages, where characters from Indian languages are mapped with Roman characters based

on their phonetic similarities, thus providing a unique representation of Indian Languages in Roman alphabet. The only non-intuitive mapping of characters is done with ‘W’ and ‘X’ characters, thus earning the format its name.

4. Tools

The following tools are the major modules used in pipeline:

- **Tokenizer:** Tokenizer was constructed as part of SETU project sanctioned by IIIT Hyderabad. It takes plain text file either in UTF or wx format as input and returns SSF sentences in respective format. It also adds meta-tags to the *header* section of SSF file based on the provided configuration file.
- **Morph:** Morphological analyzer was constructed as part of ILMT Project. It takes an SSF file in WX format as input and returns sentences tagged with morphological analyses in WX format. It outputs multiple possible analyses for each node as it works in a context-free manner.
- **POS:** POS tagger was constructed as part of ILMT Project. It takes an SSF file in UTF format as input and returns POS tagged sentences in UTF format. It takes context into account, hence gives a singular result for each node.
- **Chunker:** Chunker tool was constructed as part of ILMT Project. It takes an SSF file in WX format as input and returns chunked sentences in WX format. It gives a singular prediction for every chunk.
- **Pruner:** Morphological Pruner was constructed as part of ILMT Project. It takes an SSF file in WX format as input and returns sentences tagged with pruned morphological analysis in WX format. It only prunes those multiple analyses for which it is confident above a threshold of probability.
- **Converters:** Converters are constructed as part of ILMT Project. These tools are used as intermediate stages in the pipeline for data conversion from UTF format to WX format and vice versa as required by other modules.
- **Expander:** Expander was constructed by IIIT Hyderabad (Kosaraju et al., 2012) utilizing the Vibhakti Computation and Head computation modules developed as part of ILMT Project. This is a rule based tool that converts SSF sentences in Inter Chunk format to Intra Chunk format. It works on both UTF and WX formats.
- **Sanchay:** Sanchay is an open source platform for working on languages. Sanchay tool provides multiple features like syntactic annotation interface, JAVA SSF API, text editor, UTF to WX and WX to UTF conversions etc. . This tool is used by the annotators for annotation and validation of data during different stages of the pipeline framework. It works on both UTF and WX formats (Singh and Ambati, 2010). The

¹The link is currently down

tool is available for download from the official Sanchay website².

All the ILMT tools are made accessible to the members of ILMT consortia through internal services.

5. Annotation Process

The dependency annotation in treebank is divided into 2 stages. In the initial stage, the relations among the chunks, namely the Inter-Chunk dependencies are marked, and in the later stage the relations within the chunk, namely Intra-Chunk dependencies are marked.

Annotators are trained in schema of annotation (Begum et al., 2008) and working with the *Sanchay* tool. They are taught to disambiguate between multiple annotations. The dependency guideline sentences (Bharati et al., 2009) are used by the annotators as reference sentences for correctly annotating the sentences.

While the Inter-Chunk dependencies are marked manually by the annotators, the Intra-Chunk dependencies are marked automatically by a rule based tool called *Expander*. This division makes the annotation task more efficient as the human annotators are freed from the tedious work of annotating chunk internal dependencies which can be automatically marked with high degree of accuracy (Kosaraju et al., 2012).

6. Pipeline

Figure 1 shows the route map of how the raw data passes through different stages to reach the error validation stage. During the entire pipeline, there are some things that are automated and some things that are either validated or annotated manually and there will be constant interactions between both these sides.

6.1. Tokenization

Once the cleaning of the raw data is done, the data is passed to the machine side where the tokenizer tool is run. Each of the raw data files may have many paragraphs. The tool initially divides the file into paragraphs based on the space and *new line* marker. Within each paragraph, the sentences are divided by using the sentence end marker. The tokens/words in each sentence are divided by using space. This is a stage where plain text is converted into SSF format. Each sentence in a file will be assigned with a unique numeric ID. Also, for each token in the sentence, a unique integer is assigned as an address, which helps to access the token. The tags `<Sentence id=>` and `</Sentence>` indicate the start and end of the sentence.

Once the tokenizer tool is run, the output is passed to the human side where they validate the output of the tool and do necessary corrections, if any. This validation stage is required because, for some of the tokens like Mr. P. V. Narasimha Rao, the tokenizer tool fails to identify this as a single token, instead it identifies Mr, P, V and Narasimha Rao as 4 different sentences.

An SSF representation of the tokenization output can be seen in *Table 1*

6.2. Morph, POS and Chunking

Once we are done with the token validation stage, we now move to the machine side where Morph, POS and Chunker tools are run.

For each token in the sentence, the Morph Analyzer tool outputs all the possible morph analyses for that particular token, irrespective of the context. The Morphological information has 8 different categories. They are root, category, gender, number, person, case, tense/aspect, and suffix. All these categories are represented in the feature structure of the token/word together by using a special attribute called *af* or abbreviated features. The field for each attribute is at a fixed position and a comma is used as a separator. The field is left blank for undefined values.

POS tagger assigns a Part of Speech (POS) tag for each token.

The Chunker tool marks the boundaries of the chunks in the sentence and also assigns a chunk tag for each chunk. Chunk boundary is represented using “(” and “)” which indicate the initiation and closure of the chunk respectively. In this stage, the address of the chunk is given as an integer and the address of the token is given as a decimal. The whole part of the decimal indicates the chunk to which the token belongs to and the decimal part of the address indicates the relative position of the token in that particular chunk.

Once the Morph, POS and Chunker tools are run, the data is sent to the annotators for Validation of POS and Chunk information. After the validation of POS and chunk information by an annotator, the same information is cross validated by a different annotator to remove any erroneous cases, if present. This is done to ensure better quality of annotation. The Morph information is not validated here. It is done after the pruning stage.

An SSF representation of the Morph, POS and Chunked output can be seen in *Table 2*

6.3. Morph Pruning

After the POS and Chunk validation, the data is sent to the Machine side for Pruning. In the pruning stage, the morph analyses for which the value of the category attribute is not in mapping with the POS tag of the token are removed automatically by the tool. After this, the data is sent to the morph annotators for validation. Now, the morph annotators look at the pruned data and finally choose the morph analysis that perfectly fits into that context.

An SSF representation of the Pruner Validated output can be seen in *Table 3*

6.4. Position Marking

The data is again passed to Machine side to add a new attribute in the feature structure of every token, called *posn* which indicates the position of the word more explicitly in the sentence. The value for the *posn* attribute would be in multiples of 10, in order to facilitate the insertion of new tokens between the existing tokens.

An SSF representation of the Position marked output can be seen in *Table 4*

²<http://sanchay.co.in/>

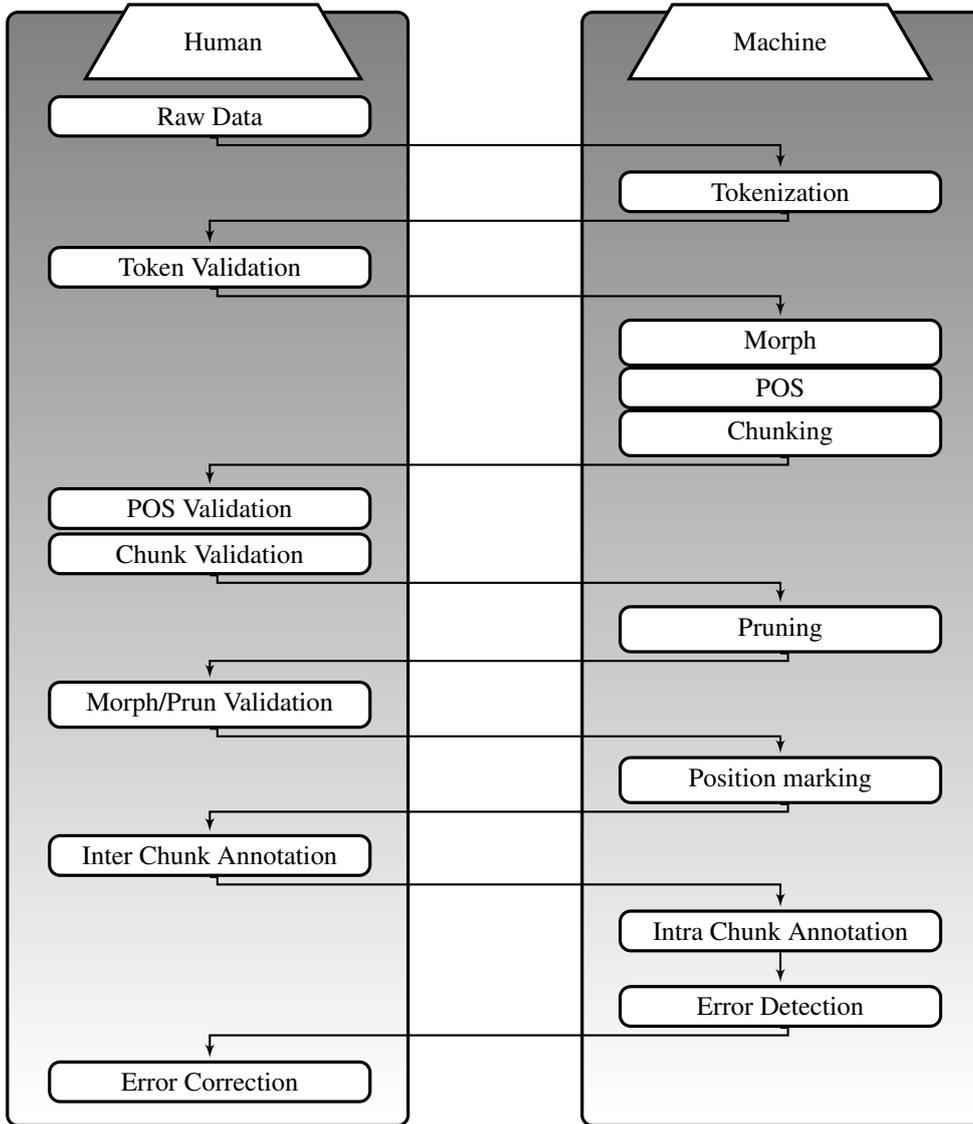


Figure 1: Treebanking Annotation Pipeline

```

<Sentence id='1'>
1      raama      unk
2      ne         unk
3      mohana    unk
4      ko         unk
5      niilii   unk
6      kitaaba   unk
7      dii       unk
</Sentence>

```

Table 1: SSF representation of Tokenized data

6.5. Inter-Chunk Dependency Annotation

After running the position marking tool, the data is passed to the dependency annotators. Their task is to create the dependencies between the chunks.

An SSF representation of the Inter-chunk marked output can be seen in *Table 5*

A graphical representation of dependency tree constructed after inter-chunk dependency annotation stage can be seen in *Figure 2*

6.6. Intra-Chunk Dependency Annotation

Once the manual annotation of Inter-Chunk dependencies are done, the Intra chunk dependencies are marked automatically using *Expander* tool. As part of the *Expander* tool (Kosaraju et al., 2012), the Inter-Chunk data is run through a head-computation module which identifies the head token for each chunk. The dependency relations that were initially present between the chunks at Inter-Chunk level will now be transferred between the head tokens of those

```

<Sentence id='1'>
1      ((      NP
1.1    raama    NNP  <fs af='raama,n,m,sg,3,d,0,0'>|<fs af='raama,n,m,pl,3,d,0,0'>|<fs
      af='raama,n,m,sg,3,o,0,0'>|<fs af='raama,n,m,pl,3,o,0,0'>
1.2    ne      PSP  <fs af='ne,psp,,,,,'>
      ))
2      ((      NP
2.1    mohana  NNP  <fs af='mohana,n,m,sg,3,d,0,0'>|<fs
      af='mohana,n,m,pl,3,d,0,0'>|<fs af='mohana,n,m,sg,3,o,0,0'>
2.2    ko      PSP  <fs af='ko,psp,,,,,'>
      ))
3      ((      NP
3.1    niilii  NNC  <fs af='niilii,n,f,sg,3,d,0,0'>|<fs af='niilii,n,f,sg,3,o,0,0'>
3.2    kitaaba NN   <fs af='kitaaba,n,f,sg,3,d,0,0'>|<fs af='kitaaba,n,f,sg,3,o,0,0'>
      ))
4      ((      VGF
4.1    dii     VM   <fs af='de,v,f,sg,any,,yaa,yaa'>
      ))
</Sentence>

```

Table 2: SSF representation of Morph, POS and Chunked data

```

<Sentence id='1'>
1      ((      NP
1.1    raama    NNP  <fs af='raama,n,m,sg,3,o,0,0'>
1.2    ne      PSP  <fs af='ne,psp,,,,,'>
      ))
2      ((      NP
2.1    mohana  NNP  <fs af='mohana,n,m,sg,3,o,0,0'>
2.2    ko      PSP  <fs af='ko,psp,,,,,'>
      ))
3      ((      NP
3.1    niilii  NNC  <fs af='niilii,adj,f,sg,,d,,'>
3.2    kitaaba NN   <fs af='kitaaba,n,f,sg,3,d,0,0'>
      ))
4      ((      VGF
4.1    dii     VM   <fs af='de,v,f,sg,any,,yaa,yaa'>
      ))
</Sentence>

```

Table 3: SSF representation of Pruned Data

```

<Sentence id='1'>
1      ((      NP
1.1    raama    NNP  <fs af='raama,n,m,sg,3,o,0,0' posn='10'>
1.2    ne      PSP  <fs af='ne,psp,,,,,' posn='20'>
      ))
2      ((      NP
2.1    mohana  NNP  <fs af='mohana,n,m,sg,3,o,0,0' posn='30'>
2.2    ko      PSP  <fs af='ko,psp,,,,,' posn='40'>
      ))
3      ((      NP
3.1    niilii  NNC  <fs af='niilii,adj,f,sg,,d,,' posn='50'>
3.2    kitaaba NN   <fs af='kitaaba,n,f,sg,3,d,0,0' posn='60'>
      ))
4      ((      VGF
4.1    dii     VM   <fs af='de,v,f,sg,any,,yaa,yaa' posn='70'>
      ))
</Sentence>

```

Table 4: SSF representation of Position Run data

```

<Sentence id='1'>
1      ((      NP      <fs drel='k1:VGF' name='NP'>
1.1    raama    NNP    <fs af='raama,n,m,sg,3,o,0,0' name='raama' posn='10'>
1.2    ne       PSP    <fs af='ne,psp,,,,,' name='ne' posn='20'>
      ))
2      ((      NP      <fs drel='k4:VGF' name='NP2'>
2.1    mohana  NNP    <fs af='mohana,n,m,sg,3,o,0,0' name='mohana' posn='30'>
2.2    ko       PSP    <fs af='ko,psp,,,,,' name='ko' posn='40'>
      ))
3      ((      NP      <fs drel='k2:VGF' name='NP3'>
3.1    niilii  NNC    <fs af='niilii,adj,f,sg,,d,' name='niilii' posn='50'>
3.2    kitaaba NN     <fs af='kitaaba,n,f,sg,3,d,0,0' name='kitaaba' posn='60'>
      ))
4      ((      VGF     <fs stype='declarative' name='VGF' voicetype='active'>
4.1    dii     VM     <fs af='de,v,f,sg,any,,yaa,yaa' name='dii' posn='70'>
      ))
</Sentence>

```

Table 5: SSF representation of Inter Chunk Dependency data

```

<Sentence id='1'>
1      raama    NNP    <fs af='raama,n,m,sg,3,o,0_ne,0' drel='k1:dii' vpos='vib_2'
      name='raama' chunkId='NP' chunkType='head:NP' posn='10'>
2      ne       PSP    <fs af='ne,psp,,,,,' drel='lwg_psp:raama' chunkType='child:NP'
      name='ne' posn='20'>
3      mohana  NNP    <fs af='mohana,n,m,sg,3,o,0_ko,0' drel='k4:dii' vpos='vib_2'
      name='mohana' chunkId='NP2' chunkType='head:NP2' posn='30'>
4      ko       PSP    <fs af='ko,psp,,,,,' drel='lwg_psp:mohana' chunkType='child:NP2'
      name='ko' posn='40'>
5      niilii  JJ     <fs af='niilii,adj,f,sg,,d,' drel='nmod_adj:kitaaba' chunk-
      Type='child:NP3' name='niilii' posn='50'>
6      kitaaba NN     <fs af='kitaaba,n,f,sg,3,d,0,0' drel='k2:dii' name='kitaaba'
      chunkId='NP3' chunkType='head:NP3' posn='60'>
7      dii     VM     <fs af='de,v,f,sg,any,,yA,yA' stype='declarative' voicetype='active'
      name='dii' chunkId='VGF' chunkType='head:VGF' posn='70'>
</Sentence>

```

Table 6: SSF representation of Intra Chunk Dependency data

Chunk Name	Parent Constraints	Child Constraints	Contextual Constraints	Dependency Relation
NP	POS==NN	POS==JJ	posn(parent)>posn(child)	nmod_adj

Table 7: A sample rule from Expander tool

chunks. To get the Intra-Chunk relations, a rule template has been created manually. A sample rule is shown in *Table 7*.

After running the Expander, the POS and Morph information of each token gets transferred from Inter-Chunk level to Intra-Chunk level without any modification. The bracketing which represent the chunks at Inter-Chunk level will not be present at Intra-Chunk level. But, the chunk information is still preserved in the Intra-Chunk format by using two additional attributes “ChunkId” and “ChunkType”, which are not present at the Inter-Chunk level. These attributes act as substitution for bracketing, as well as shows the chunk members in the role of head and child. The head node has “ChunkId” that gives it a unique chunk name which actually has the same value of “name” attribute for the original chunk. All the chunk members have “Chunk-

Type” that gives their membership type. The “ChunkType” takes one of the two values, either “head” or “child” of one of the chunks.

An SSF representation of the Intra-Chunk marked output can be seen in *Table 6*

A graphical representation of dependency tree constructed after Intra-Chunk dependency annotation stage can be seen in *Figure 3*

7. Errors and sanity

Due to limited training data and generic domain nature of the tools, most of the tools fail to predict proper analysis in several cases and result in crashes. This brings forth the need to ensure that these errors do not trickle along different stages of the pipeline and make sure that the crashes that have been detected are immediately localized.

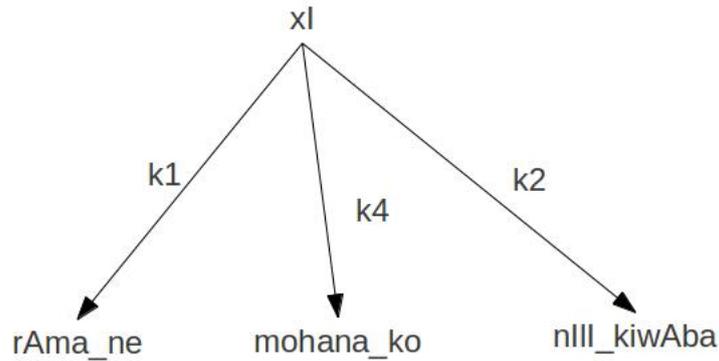


Figure 2: Inter-Chunk Dependency Tree

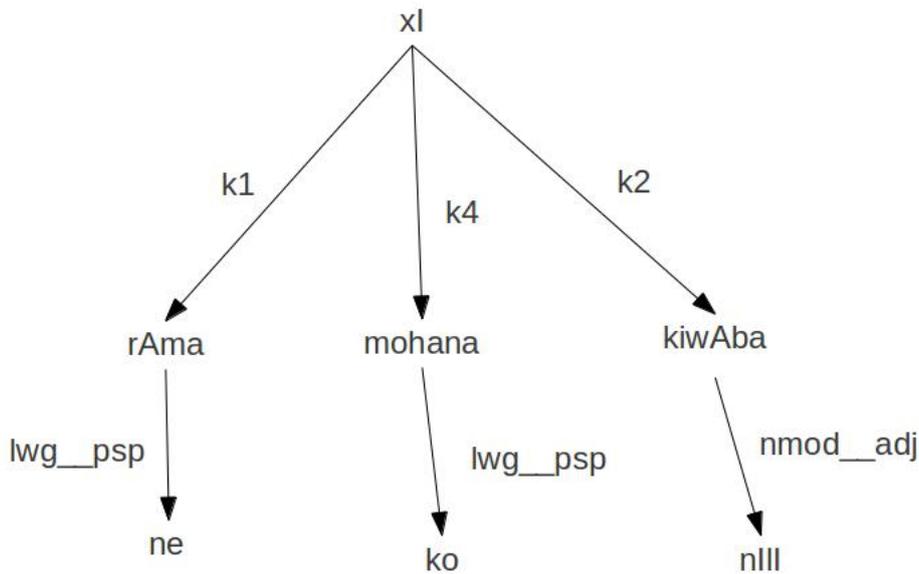


Figure 3: Intra-Chunk Dependency Tree

The Morph tool is prone to crashes due to limited entries in morphology table and in many cases the crashes result in partial sentences with missing nodes or even missing sentences. The chunker tool also is prone to crashes sometimes with similar errors.

In-pipeline sanity tools compare the input and output files after every run of module and check for discrepancies and errors. Only changes that are respective to the module are permitted and rest are reported by the tools. After every module-run, another script called the *difference-generator* checks and ensures that the word and sentence counts remain unchanged before and after every module-run. Every issue noticed is reported to manual validators to either modify corresponding word or sentence; in case of larger errors,

the sentence is removed. Thus, in-pipeline data sanity is ensured.

Another set of tools check for the correctness of annotations and aid in manual cross-validation of the dependencies. As previously mentioned, during the validation stage, the data passes through validations which cover Morph, POS, Chunking and Dependency stages individually. At these stages, automatic error detection tool (Agarwal et al., 2012) is applied to reduce the effort of the manual validators. The Error detection tool extracts the potential error cases from the data and these are manually verified by the annotators and do necessary corrections, wherever required.

8. Quality Assurance

Once the data completely goes through the pipeline, it undergoes several stages of quality checking where the complete data is scrutinized automatically for errors and reported errors are corrected manually .

Of the several stages of quality assurance, the initial stages concentrate on ensuring correctness of SSF format of the sentences.

- **Meta-data and CML correctness:** These checks ensure that the header section maintains consistency as per CML scheme. The tools check for the presence of specific meta-tags like source and date of creation.
- **Sentence repetition:** Due to good chance of crashes in the internal modules, sentences are prone to repeat both inside a file and sometimes across the files. These checks ensures that a proper sentence count is achieved.
- **Feature structure error detection:** These checks look for errors in formats of *Other* property and checks for the validity of the values in *Category field*.
- **Morphological error detection:** These checks ensure the validity of values present in the *af* field of *Other* property. Also, care is taken to ensure that the coarse POS tag and *Category* POS tag agree with each other.

Another set of stages check for the presence of errors in dependency annotation of sentences.

- **Dependency and forest checks:** The sentences are checked for errors in dependency annotations like validity of the type of dependency labels, validity of both parent and child. Also sentences are checked to ensure that the sentence is not divided into multiple dependency trees or *Forests* but a singular dependency tree exists.
- **Cycle Detection:** The sentences are checked for presence of looped dependencies or cycles. The algorithm checks whether all the chunks are connected finally to the root and reports if the root is not reached in fixed number of iterations. Thus presence of cycles is reported for correction.
- **Pattern based dependency checks:** The sentences are checked for presence of specific patterns in the dependency relations for individual verbs. Cross checking with a list of “Should exist together” and “Should not occur together” is done and violations are reported to the annotators for corrections.

Thus data is cleaned and quality of the data is ensured before the corpora is locked out as release-ready.

9. Status

Hindi Treebank contains data from three domains, general news articles, tourism and heritage, and conversations as represented in short stories, while Urdu Treebank has data only from newspaper articles. Table shows the sizes of the two treebanks.

	HTB		UTB	
	Sentences	Words	Sentences	Words
<i>News Articles</i>	17,882	395K	7,120	200K
<i>Tourism</i>	1,058	15K	-	-
<i>Conversation</i>	2,028	27K	-	-
Total	20,968	437K	7,120	200K

Table 8: Sizes of the two treebanks

10. Acknowledgements

The work done as part of the treebanking project has been supported by the funds provided by NSF. We also thank DEITY for providing the financial support for the construction of modules as part of ILMT consortium. We heartily thank Bharat Ram Ambati for this initial contribution and dedication towards the establishment of pipeline framework. We kindly thank Anil Kumar Singh for permitting us to use the Sanchay toolkit, without which the annotation of the treebank could not have taken place.

11. References

- Rahul Agarwal, Bharat Ram Ambati, and Anil Kumar Singh. 2012. A gui to detect and correct errors in hindi dependency treebank. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 1907–1911, Istanbul, Turkey, May. European Language Resources Association (ELRA). ACL Anthology Identifier: L12-1439.
- Rafiya Begum, Samar Husain, Arun Dhawaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*.
- Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. 1995. *Natural language processing: a Paninian perspective*. Prentice-Hall of India New Delhi.
- Akshara Bharati, Rajeev Sangal, and Dipti Misra Sharma. 2007. SSF: Shakti standard format guide.
- Akshara Bharati, Dipti Misra Sharma, Samar Husain, Lakshmi Bai, Rafiya Begam, and Rajeev Sangal. 2009. Anncorra: Treebanks for indian languages, guidelines for annotating hindi treebank.
- Rajesh Bhatt, Bhuvana Narasimhan, Martha Palmer, Owen Rambow, Dipti Sharma, and Fei Xia. 2009. A multi-representational and multi-layered treebank for hindi/urdu. In *Proceedings of the Third Linguistic Annotation Workshop*, pages 186–189, Suntec, Singapore, August. Association for Computational Linguistics.
- Alena Böhmová, Jan Hajič, Eva Hajičová, and Barbora Hladká. 2003. The prague dependency treebank. In *Treebanks*, pages 103–127. Springer.
- Prudhvi Kosaraju, Bharat Ram Ambati, Samar Husain, Dipti Misra Sharma, and Rajeev Sangal. 2012. Intra-

- chunk dependency annotation : Expanding hindi interchunk annotated treebank. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 49–56, Jeju, Republic of Korea, July. Association for Computational Linguistics.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.
- Anil Kumar Singh and Bharat Ram Ambati. 2010. An integrated digital tool for accessing language resources. In *LREC*.

Improved and Adaptable Statistical Morph Analyzer (SMA++)

Saikrishna Srirampur, Deepak Kumar Malladi, Radhika Mamidi

LTRC, IIIT-Hyderabad

Hyderabad, India

saikrishna.srirampur@research.iiit.ac.in, deepak.malladi@research.iiit.ac.in, radhika.mamidi@iiit.ac.in

Abstract

Morph analyzers play an important role in almost all the natural language applications. The morph analyzer (SMA++) we have developed is a data driven, statistical system. The system is a hybrid of the two best state of art statistical morph analyzers (SMA) viz. Morfette in Chrupała et al. (2008) and SMA in Malladi and Mannem (2013). We chose a robust feature set, which is a hybrid of the features used in the above SMAs. Our system predicts the gender, number, person, case (GNPC) and the lemma. The training and testing were done using the lib-linear classifier. Some rich features such as the *morph tag of the previous token* and the *Part of Speech* were used. Our goal is to come up with the best SMA which beats both the above SMAs. Our system is not language specific and can adapt to any language. Experimental results for the Indian language Hindi and sample results for Urdu have been shown, while results for other languages like Telugu etc. are in progress. The results for Hindi reflected higher accuracies than both of the above mentioned state of art SMAs.

Keywords: Morph Analyzer, Hybrid.

1. Introduction

Morphological analysis is the analysis of a word in terms of its lemma(L) and features such as gender(G), number(N), person(P), case(C), vibhakti¹, tense, aspect and modality. A tool used to predict the lemma and such features is called a Morph Analyzer (MA).

For Hindi, there are two major approaches to build morphological analyzers viz. Paradigm based and Statistical approach. Paradigm based MAs (PBA) are based on the concept of paradigm class, which contains those words which decline or conjugate in exactly the same way. Given a word, first its paradigm class is identified, then the corresponding suffix/prefix additions and deletions are done using the paradigm table (which contains the inflected forms of the paradigm class) to get the required morphological features of the given word. This approach is language specific, and is restricted to only the words present in the paradigm-word dictionaries. The second approach i.e. the statistical one, is a data-driven approach. Using the training data, statistical models are formed. These models help to predict the morph-analysis of the test data. This approach is language independent and works for all words, including out of vocabulary (OOV) words.

2. Related Work

Traditionally morphological analysis for Indian languages has been done using the paradigm based approach. There are two types of Paradigm Based MAs (PBA) viz. the Oracle PBA (O-PBA), which uses an oracle to pick the best analysis from the list of all analyses given by the PBA and the F-PBA, which picks the first analysis from the output, as the correct analysis. The PBA by Bharati et al. (1995) is one of the most widely used Hindi MA, among

¹Vibhakti is a Sanskrit grammatical term that encompasses post-positionals and case endings for nouns, as well as inflection and auxiliaries for verbs. It is also referred as case-marker

the NLP researchers in the Indian Community. Goyal and Lehal (2008) and Kanuparthi et al. (2012) analysers are the two advanced forms of the Bharati et al. (1995)'s analyzer. Kanuparthi et al. (2012) built a derivational morphological analyzer for Hindi by introducing a layer over the PBA. It identifies 22 derivational suffixes which help in providing derivational analysis for the word whose suffix matches with one of these 22 suffixes. There have not been major upgrades in PBAs in Hindi and the problem of not predicting OOV words is still a significant one.

Data-driven approaches are not as diverse as that of paradigm based. Very few attempts have been made for employing data-driven techniques on morphologically rich languages. Chrupała et al. (2008) and SMA in Malladi and Mannem (2013) are notable state of the art SMAs which employed the data-driven approach.

In this work, we present a statistical MA which is the hybrid of the above mentioned SMAs. We trained our system on the Hindi Tree Bank(HTB) and have predicted lemma, gender, number, person and case. The robust feature selection helped our system beat both the above mentioned SMAs. The paper has been divided as *our approach, experiments, results, conclusions and future work* related to the system.

3. Our Approach

Forming the class-labels for each of gender, number, person, case and lemma was the initial task. For gender, number, person and case, the class-labels were chosen from the training data itself.

For lemma, the class-labels were formed based on the edit-distance² operations required for the conversion of each token to its lemma. This idea was inspired by Chrupała

²Edit distance is a way of quantifying how dissimilar two strings (e.g., words) are to one another by counting the minimum number of operations required to transform one string into the other.

Feature	Description
Suffix Length	Length of the suffix.
morph_previous	Predicted Morph Tag of the previous tokens
word_form	Lower cased word form
character_types	Character types such as numbers, lower and upper cased alphabets, symbols etc. in the current token.
next_morph_tags	Set of morph tags of the next token, if it is found in the training corpus
word_form_previous	Lower cased word forms of the previous token.
word_form_next	Lower cased word forms of the next token.
length_token	String length of the token.
POS	The automatic Part Of Speech tag.

Table 1: Features.

(2006), who introduced the concept of edit-operations³ for lemmatization. A token and its lemma are taken from the corpus and are reversed. The *edit operations* required to convert the *reversed token to reversed lemma* represent a class-label. Doing this for each token forms the set of class-labels. The example shown below explains the process and the reason behind this approach.

Example:⁴ Take the token *crying*. The lemma for it is *cry*. Reverse the token and its lemma. *crying* becomes *gniyrC* and *cry* becomes *yrC*. To convert *gniyrC* to *yrC* we need to delete the characters at the 1st,2nd and 3rd indices. Hence the edit operations would be [d 1, d 2, d 3], where 'd' represents delete operation. [d 1, d 2, d 3] represents a class-label. Similarly, the class-label for the token *playing* and the lemma *play* would be [d 1, d 2, d 3].

If we did not reverse the token and lemma, the class-labels for the above two examples wouldn't be same. But, we intend the class labels for *playing* - *play* and *crying* - *cry* to be the same, because they have the common suffix *-ing*. Hence, we reverse the token and the lemma, and then construct the class-label. This approach suits suffix rich languages because, by reversing two tokens which have a common suffix, the relative position of the common suffix remains the same.

The feature set was formed by choosing the best mix of features from the two state of art MAs mentioned above. Table 1 shows the feature-set. The lib-linear classifier was used for predicting the outputs.

4. Experiments

The Hindi Treebank (HTB) released as part of the 2012 Hindi Parsing Shared Task (Sharma et al., 2012) was used to evaluate our models. All the models are tuned on development data and evaluated on test data. Table 2. represents the HTB statistics.

We also tested our models for Urdu language . The Urdu Treebank(UTB) released as a part of the 2012 Proceedings of TLT (Bhat and Sharma (2012)) was used for evaluation. Experiments conducted for Urdu were on a small scale. Table 3. represents the UTB statistics.

³The add, delete and replace operations required to convert one string to another

⁴Example is in English language. This is to give better clarity. The similar methodology even works for Hindi.

Data	#Sentences	#Words
Training	12,041	268,096
Development	1,233	26,416
Test	1,828	39,775

Table 2: HTB Statistics.

Data	#Sentences	#Words
Training	5,700	159,743
Test	1,453	39,803

Table 3: UTB Statistics.

5. Results

The *automatic POS* was the largest contributor to the accuracy, for all the attributes(lemma, gender, number, person and case). The *morph_previous* played an important role in almost all the attributes .This is because, Hindi has agreement⁵ which makes the attribute of a token percolate to the other related tokens. *Word form*, which is a lexical feature also had a notable effect.

The results are presented for each of the 5 attributes(L,G,N,P,C) individually, as well as in a combination.The outputs are compared with a baseline system. The baseline system takes the word form itself as the lemma and selects the most frequent value for the rest of the attributes. The outputs are compared with two versions of PBA, because PBA is a rule based analyzer, which gives more than one analysis for each word. The O-PBA and F-PBA were used for the analysis of results. The outputs are compared with two state of the art SMAs, namely, Chrupała et al. (2008)'s Morfette (M) and Malladi and Mannem (2013) SMA (SMA-M). There are two representations of our system, *SMA++** is the system without considering the automatic POS as a feature and the other *SMA++* is the one with the POS. We have two representations because, Morfette(M) does not use POS as a feature and we wanted to compare *SMA++** with the Morfette (M).

The results are placed in two sub-divisions. One, for the overall data(Table 4). The other for Out Of Vocabulary

⁵Agreement or Concord happens when a word changes form depending on the other words to which it relates

2*Analysis	Test Data - Overall (%)						
	Baseline	F-BPA	O-PBA	M	SMA++*	SMA-M	SMA++
L	71.12	83.10	86.69	94.14	98.08	95.84	98.43
G	37.43	72.98	79.59	95.05	95.05	96.19	96.21
N	52.87	72.22	80.50	94.09	94.24	95.37	95.47
P	45.59	74.33	84.13	94.88	95	96.38	96.28
C	29.31	58.24	81.20	93.91	94.20	95.32	95.43
L+C	16.46	48.84	72.06	88.56	92.63	91.39	94.01
G+N+P	23.05	61.10	73.81	88.36	88.85	91.11	90.36
G+N+P+C	9.72	45.73	70.87	84.43	86.95	87.78	88.51
L+G+N+P	20.27	53.29	66.28	83.44	87.71	87.51	89.26
L+G+N+P+c	8.57	38.25	63.41	79.73	84.21	84.25	85.87

Table 4: Hindi Results-Overall

2*Analysis	Test Data - Out Of Vocabulary (OOV) (%)						
	Baseline	F-BPA	O-PBA	M	SMA++*	SMA-M	SMA++
L	78.10	82.08	82.48	90.30	90.11	89.51	93.07
G	60.22	43.07	44.06	72.03	73.68	82.65	83.11
N	69.60	44.53	47.56	84.89	84.89	90.44	92.81
P	78.30	52.51	53.89	84.76	86.21	94.85	96.17
C	43.50	31.40	47.36	80.21	82.06	88.52	89.45
L+C	32.52	28.50	44.66	72.89	74.72	79.09	82.92
G+N+P	47.49	35.75	38.58	62.33	65.76	76.52	77.24
G+N+P+C	21.04	20.91	35.95	55.74	62.07	69.99	72.36
L+G+N+P	44.72	34.63	38.46	57.85	61.28	69.13	72.82
L+G+N+P+c	19.33	19.92	38.49	51.52	55.61	63.06	65.96

Table 5: Hindi Results-OOV

M : Chrupała et al. (2008)’s Morfette; SMA-M : SMA of Malladi and Mannem (2013); SMA++*: Our SMA without automatic POS as a feature; SMA++: Our SMA with automatic POS as a feature;

Analysis	SMA-M	SMA++
G	89.14	93.79
N	91.62	95.66
P	93.37	97.07
C	85.49	90.92

Table 6: Urdu-Results-Overall

(OOV) tokens (Table 5). The accuracies are the percentages of words in the test-data with the correct morph analysis. Our system out-performed PBA, Morfette, and SMA-M in almost all considered combinations and moreover our OOV section results are more pronounced. Table 4 and 5 present the accuracies of the above stated systems in comparison with our system.

Table 6 shows results for the Urdu language. Our system outperforms SMA-M (Malladi and Mannem, 2013) in all attributes of G, N, P and C.

6. Conclusions and Future Work:

SMA++ presents an improvised and a hybrid version of the two state of the art SMAs mentioned above. In Hindi, for the L+G+N+P+C our system achieved an accuracy of 85.87% on overall test-data and 65.96% on OOV test-data. It thereby out-performed those state of the art SMAs. Such

results depict SMA++ as a rich tool for morph analysis of Indian languages. Work on other Indian languages such as Urdu, Telugu etc. is in progress. We plan to replace the automatic POS feature, by clustering the tokens using Latent Dirichlet Allocation (LDA)⁶. In future, we would like to extend the statistical approach for Syntactic Parsers.

7. References

- Bharati, Akshar, Chaitanya, Vineet, Sangal, Rajeev, and Ramakrishnamacharyulu, KV. (1995). *Natural language processing: a Paninian perspective*. Prentice-Hall of India New Delhi.
- Bhat, Riyaz Ahmad and Sharma, Dipti Misra. (2012). A dependency treebank of urdu and its evaluation. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 157–165. Association for Computational Linguistics.
- Chrupała, Grzegorz, Dinu, Georgiana, and Van Genabith, Josef. (2008). Learning morphology with morfette.
- Chrupała, Grzegorz. (2006). Simple data-driven contextsensitive lemmatization. *Procesamiento del Lenguaje Natural*, 37:121–127.

⁶LDA is a generative probabilistic model for collections of discrete data such as text corpora. It can be used to model the tokens into various topics

- Goyal, Vishal and Lehal, Gurpreet Singh. (2008). Hindi morphological analyzer and generator. In *Emerging Trends in Engineering and Technology, 2008. ICETET'08. First International Conference on*, pages 1156–1159. IEEE.
- Kanuparthi, Nikhil, Inumella, Abhilash, and Sharma, Dipti Misra. (2012). Hindi derivational morphological analyzer. In *Proceedings of the Twelfth Meeting of the Special Interest Group on Computational Morphology and Phonology*, pages 10–16. Association for Computational Linguistics.
- Malladi, Deepak Kumar and Mannem, Prashanth. (2013). Context based statistical morphological analyzer and its effect on hindi dependency parsing. In *Fourth Workshop on Statistical Parsing of Morphologically Rich Languages*, volume 12, page 119.
- Sharma, Dipti Misra, Mannem, Prashanth, vanGenabith, Joseph, Devi, Sobha Lalitha, Mamidi, Radhika, and Parthasarathi, Ranjani, editors. (2012). *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages*. The COLING 2012 Organizing Committee, Mumbai, India, December.

Challenges in Indian Language Transliteration: a case of Devanagari, Bangla and Manipuri

K. Kabi Khanganba, Girish Nath Jha

Special Centre for Sanskrit Studies, JNU, New Delhi, 110067

khanganba.360@gmail.com, girishjha@gmail.com

Abstract

The paper presents a Meitei-Bangla-Devanagari transliteration system and the challenges therein. Manipuri is a scheduled Indian language and was recently added to the Indian Language Corpora Initiative (ILCI) project being run in a consortium mode at Jawaharlal Nehru University. The Manipuri group faced difficulty in keying in Manipuri data in the Meitei script as there are very few people who know typing in this script. Therefore, the project needed a transliteration system which could convert text written in Bengali script (which is known to most of the adult speaker) to Meitei. Automatic transliteration is a basic requirement in developing language technology in the diverse Indian language scenario. As most of the Indian scripts belong to the Brahmi family and have comparable sound systems, it is apparently not too difficult to create parallel arrays of utf charset encodings for comparing and substituting corresponding values between a pair of scripts. However, in reality they pose considerable challenges. Meitei presents special substitution challenges due to a slightly different representation scheme followed in Unicode for it. Another complication is due to the fact that in case of transliteration involving Meitei with another Indian language script (particularly from the north) we may be trying to substitute diverse phoneme sets leading to one-to-many and many-to-one matches.

Keywords: Puya, Meitei Mayek, Vowel tones, Consonant Clusters, Interrupted and Uninterrupted Sounds, Transliteration Tool, Nuances

1. Introduction

Manipuri is a Tibeto-Burman (a subdivision of Sino-Tibetan language family) language with a 1,466,705 strong speech community (census 2001) in the north eastern Indian state of Manipur. It is also a lingua franca in the state used as a language of convenience among other 33 dialects spoken in the state. It is spoken in the Indian states of Assam and Tripura as well as in the neighboring countries of Bangladesh and Myanmar. Manipuri was recognized as a scheduled language by the 71th amendment of the constitution in 1992. Now it is a medium of instruction up to undergraduate level in the state and taught up to postgraduate level by some major universities of India.

Manipuri was written in its own script presently known as Meitei Mayek (also known as Meitei/Meetei or Manipuri) till early 18th century. Some scholars have the opinion of the existence of the script since 1st century A.D. as per Chaitharol Kumbaba (the Royal Chronicle; see Figure 1). Its earliest inscription is dated back to 11th to 12th century (King Kiyamba inscription, Tengnoupal). During the reign of king Pamheiba (1709 A.D. to

1748 A.D.) with the adoption of Gaudiya Vaishnavism, the Meitei script was replaced by Bangla as the script of the darbar (the royal court).



Figure 1: A page from Chaitharol Kumbaba, N. Khelchandra Singh¹

Contacts with other literature and knowledge traditions, especially of Bengali and Sanskrit during 18th and 19th centuries brought major effects in the history of Meitei script. New consonant signs representing foreign sounds and some consonant clusters in the very Bangla pattern (style) are found

¹ <http://tabish.freeshell.org/eeyek/history.html>

in the manuscripts of these periods. Around 1000 Meitei Mayek written manuscripts have been discovered. (Tomchou: 1991). Some important texts like Chaitharol Kumbaba have been published. And some are available in different libraries and museums in the state. (Online catalogue: IGNCA).

The revival moment of the script started in 1940s and 1950s. On the basis of the scripts found in the manuscripts of ancient texts, Puyas, the Meitei Mayek Expert Committee reconstructed the present Meitei Scripts consisting of 27 alphabets (Iyek Ipi), 8 Lonsum Iyek, 8 Caitap Iyek, 3 Khudam Iyek and 10 Caishing Iyek (numerals) scripts. Most of the borrowed letters were not included and some new letters which are not found in the archaic Manipuri (language) were added to represent all the sounds in the present Manipuri. Later, on 16th April, 1980 the Govt. of Manipur approved it (Order No. 1/2/78-SS/E) to introduce it in the school education. From the academic year 2006, the Bengali script has been replaced by Meitei Mayek in the school syllabus with a view to upgrade every year. Now it has been upgraded up to the 10th grade in the school.

As the Bangla has been replaced, the demand and importance of Meitei script has been growing in many other fields. However, since Bengali is still a dominating script, there is a need to convert these in the native Meitei script for easy accessibility for the new generation of people.

Significantly, in its history, the Meitei script has been a part of Unicode Standard from 2009 onwards with the release of version 5.2. Now Unicode has two editions of Meitei scripts, i.e. Meetei Mayek (Range: ABC0-ABFF) and Meetei Mayek Extensions (AAE0-AAFF). 23 letters were added in the Meetei Mayek Extensions for research purpose of historical orthographies of the Meitei scripts. However, these are not specified in the Manipur Govt. Order No. 1/2/78-SS/E.

Devanagari is one of the major scripts being used by ten other Indian languages including Sanskrit which has a 6000 years of intellectual history and millions of texts in fundamental disciplines. Therefore converting Sanskrit-Devanagari texts to Meitei and Bangla will help extend their accessibility and research to newer language and script groups in the country.

2. Related Work

In the last decade, with the sudden remarkable development in computational technology, many transliteration tools of modern Indian languages have been developed using different approaches. Though there are a few works on Meitei scripts, (some) researchers at Manipur Institute of Technology, Imphal, Assam University, Silchar, CDAC, Pune and Mumbai have done timely efforts on Meitei script transliteration and editor for multi-lingual data development etc.

In the present paper, we present a phoneme based approach for a Meitei-Bangla-Devanagari transliteration tool which handles substitutions of diverse phoneme sets across these three Brahmi based scripts.

3. The Meitei Script

We, here refer to the Meitei scripts prescribed by the Govt. of Manipur (order No. 1/2/78-SS/E) which have been used as a standard in our tool.

Meitei Script fonts namely RATHA, RATHA99, rathayek, Meetei Mayek, Eeyek Unicode are available online free. We used Eeyek Unicode². Therefore, the Meitei characters in the paper will show correctly in the “Eeyek Unicode” font.

Like the Roman scripts, in Meitei scripts also there are differences between the letter names and the sounds they represent. They are named as Kok, Sham Laay etc. 24 of the 27 Iyek Ipi letters are consonants. Until a vowel is substituted or succeeded, each consonant letter represents two sounds, i.e., the letter itself and the vowel short ‘ə’ after it, like Devanagari (in context of Sanskrit). For example, the consonant letters , ,  etc. represent the syllables *ka*, *sa*, *la* respectively, while the same letters represent the sounds *k*, *s*, *l* when they are concatenated with other vowel modifiers as  ( + *f*),  ( + *u*),  ( + *e*) where they represent the syllables *ki*, *su*, *le*.

Remaining three letters of the 27 Iyek Ipi, i.e., , ,  are vowel letters. Iyek Ipi are given below in details in the table 1 below.

² <http://tabish.freeshell.org/eeyek/>

Iyek Ipi	Name of character	IPA	Correspondent Bangla/Devanagari	ITRANS
☐	Kok	/kə/	ক / ক	ka
☑	Sham	/sə/ or /eə/	শ, ষ, স, ছ / শ, ষ, স, ছ	sha, Sa, sa, Ca
☒	Laay	/lə/	ল / ল	la
☓	Meet	/mə/	ম / ম	ma
☔	Paa	/pə/	প / প	pa
☕	Naa	/nə/	ণ, ন / ণ, ন	Na, na
☖	Cin	/cə/	চ / চ	ca
☗	Til	/t̪ə/	ট, ত / ট, ত	Ta, ta
☘	Khau	/kʰə/	খ / খ	kha
☙	Ngau	/ŋə/	ঙ / ড	~Na
☚	Thau	/t̪ʰə/	ঠ, থ / ঠ, থ	Tha, tha
☛	Way	/ʋə/	ব / ব	va
☜	Yaang	/jə/	য় / য	ya
☝	Huk	/hə/	হ / হ	ha
☞	Un	/u/	উ, উ / উ	u
☟	Ee	/i/	ই	I
☠	Pham	/pʰə/	ফ / ফ	pha
☡	Atiyaa	/ə/	অ / অ	a
☢	Gok	/gə/	গ / গ	ga
☣	Jham	/jʰə/	ঝ / ঙ	Jha
☤	Raai	/rə/	র / র	ra
☥	Baa	/bə/	ব / ব	ba
☦	Jil	/j̪ə/	জ / জ	ja
☧	Dil	/d̪ə/	ড, দ / ড, দ	Da, da
☨	Ghau	/gʰə/	ঘ / ঘ	gha
☩	Dhau	/d̪ʰə/	ঢ, ধ / ঢ, ধ	Dha, dha
☪	Bham	/bʰə/	ভ / ঞ	bha

Table 1: Iyek Ipi Chart

The Caitap Iyek letters consist of 8 letters (Table 2). The seven of these are vowel signs, except the letter Nung (☛) which is a nasal sound. The first five (☛ , ☛ , ☛ , ☛ and ☛) do not have independent letters as the ☛ and ☛ have ☛ and ☛ respectively. Their independent forms are formed combining together with the letter ☛ (Atiyaa = ☛) as 1. ☛ + ☛ = ☛, 2. ☛ + ☛ = ☛, 3. ☛ + ☛ = ☛, 4. ☛ + ☛ = ☛, 5. ☛ + ☛ = ☛ (A, e, ai, o, au).

Caitap Iyek	Independent Form	IPA	ITRANS	Bangla/Devanagari
☛	☛	/a:/	A	া/া
☛	☛	/i/	i	ি/ি
☛	☛	/u/	u	ু/ু
☛	☛	/e/	e	ে/ে
☛	☛	/əi/	ai	ৈ/ৈ
☛	☛	/o:/	o	ো/ো
☛	☛	/əu/	au	ৌ/ৌ
☛	DE ³	~	M	ং/ং

Table 2: Caitap Iyek Chart

Khudam Iyek are given in the table (3):

Khudam Iyek	Name of Character	ITRANS
	Caikhai	. (Full-stop)
.	Lum	DE
—	Apun	DE

Table 3: Khudam Iyek Characters

³ Does not exist

Beyond the vowels discussed, there is tonal features of the vowel sounds. Each vowel has two tones--- ‘level’ and ‘falling’. (Singh: 2000). Three examples of both level and falling tones for each vowel are given below in italics (Table 4).

Though the letter . (Lum) represents the falling tone, it is not presently used. It is not found in the modern Manipuri language corpora. Hence, a vowel letter can have both the level and falling tones. The level tone of /a:/ in kAba / ka:bə (to be over fried) and falling of /a:/ in kAba / ka:bə (to climb) are presented by only ` (/a:/) (See Table 4, for examples of other vowels). Therefore, both the level and falling tones of a vowel letter must be decided by the context. There are many homographs in Manipuri which are written the same way (a common character representation in writing), but with different pronunciation and meaning.

Short level vowel	English	Short falling	English	ITRANS	Long level	English	Long falling	English	ITRANS
ᱠ	To be poor	ᱠᱢᱟ	To be thickened	naN^ba	ᱠᱟ	To be overfried	ᱠᱟ	To climb	kAba
ᱡ	To be bright	ᱡᱢᱟ	To offend	laN^ba	ᱡᱟ	To shrink	ᱡᱟ	To be broken	kAyba
ᱢ	To be dull	ᱢᱢᱟ	To be sharp	paN^ba	ᱢᱟ	To fly	ᱢᱟ	To hold	pAyba
ᱣ	To be wise	ᱣᱢᱟ	To revenge	ShiN^ba	ᱣᱟ	To dump	ᱣᱟ	To be thick	thonba
ᱤ	To search	ᱤᱢᱟ	To be ugly	thiba	ᱤᱟ	To boil	ᱤᱟ	To knit	lonba
ᱥ	A son	ᱥᱢᱟ	To give	pi^ba	ᱥᱟ	To hug	ᱥᱟ	To bend	konba
ᱦ	An owl	ᱦᱢᱟ	A cover	maku	ᱦᱟ	To be white	ᱦᱟ	To fry	N^auba
ᱧ	To be deep	ᱧᱢᱟ	To be clear	lu^ba	ᱧᱟ	To roast	ᱧᱟ	To take	lauba
ᱨ	To slip	ᱨᱢᱟ	To stretch	tu^ba	ᱨᱟ	To do	ᱨᱟ	To dig	tauba
					ᱨᱟ	To threaten	ᱨᱟ	To move	leN^ba
					ᱨᱟ	To drag	ᱨᱟ	To run	Cenba
					ᱨᱟ	To equalize	ᱨᱟ	Name of a person	Temba

Table 4: Examples of Level and Falling tones

The 8 Lonsum Iyeks are independent consonant letters which are not associated with the short a. For example, ‘ᱠ’ (Kok) means ‘k’ and ‘a’ while ‘ᱡ’ (Kok-Lonsum) means only ‘k’ which is an isolated pure consonant form of the ‘ᱠ’ (Table 5). They often occur in the final of a syllable or a word and do not used in forming a conjunct consonant. Except these consonants, other consonants do not have isolated forms. In order to write other consonants in isolation, the Khudam Iyek __ (Apun Khudam) is used. Amongst these Lonsum Iyeks the ᱠ (I-Lonsum=y) is a semi-vowel as it is derived from the vowel ᱠ (I / i).

Consonants with vowel ‘ə’	Consonants in Isolation	IPA	Bangla / Devanagari	ITRANS
ᱠ	ᱡ	/k/	क्/क	k
ᱢ	ᱣ	/l/	ल्/ल	l
ᱤ	ᱥ	/m/	म्/म	m
ᱦ	ᱧ	/p/	प्/प	p
ᱨ	ᱩ	/n/	ण, न्/ण, न्	N / n
ᱪ	ᱫ	/t/	ट, त्/ट, त्	T / t
ᱬ	ᱭ	/ŋ/	ङ्/ङ	~N
no	ᱮ	/j/	य्/य	y

Table 5: Lonsum Iyek Chart

A Lonsum Iyek comes at the end of a syllable/word and only these can occur at the end of a word in Manipuri. Unlike Bangla or Devanagari, it is the distinctive feature of Lonsum Iyeks that they represent only the isolated consonants which can be interrupted or paused. If there can be no such interruption in utterance, a consonant (isolated) is represent by the Apun Iyek (__).

Interrupted Utterance	Bangla	ITRANS	Un-interrupted Utterance	Bangla	ITRANS
ᱠᱢᱟᱦ	পাষাম	pAmbo	ᱠᱢᱟᱦ	লৈসাম	laimRm
ᱠᱢᱟᱦ	ঙফম	~NakSham	ᱠᱢᱟᱦ	হৈকু	haikrU
ᱠᱢᱟᱦ	লাকফম	lAkpham	ᱠᱢᱟᱦ	লৈক্রক	laikrak
ᱠᱢᱟᱦ	কপ্প	kappa	ᱠᱢᱟᱦ	প্রপ-প্রপ	prUp-prUp

ᱠᱟᱨᱢᱟ	লেঙজুম	le~Njum	ᱠᱟᱨᱢᱟ ᱠᱟᱨᱢᱟ (also ᱠᱟᱨᱢᱟ ᱠᱟᱨᱢᱟ)	ᱠᱟᱨᱢᱟ	~Nra~ N- ~Nra~ N
-------	--------	---------	--	-------	---------------------------

Table 6: Examples of Interrupted and Uninterrupted Isolated Consonants

In the examples ᱠᱟᱨᱢᱟ and ᱠᱟᱨᱢᱟ , both the ‘ ᱠ ’ and ‘ ᱢ ’ are the same consonants representing the same sound of the letter ‘m’. For Devanagari or Bangla it is only on process that “adding the ᱠ to the consonants ᱢ and ᱢ ” respectively, i.e., ᱢ and ᱢ . (mb and mr). In case of Meitei Mayek two different letters are being used to show the interrupted and uninterrupted sounds of the consonant ‘m’.

The Apun Iyek is the indicator of an isolated consonant. Wherever it comes, it means that the first letter is isolated; it isolates a dependent consonant from the vowel ‘a’.

Meitei Mayek	Bangla	ITRANS
ᱠᱟᱨᱢᱟ	খাঙ	Kwa~N
ᱠᱟᱨᱢᱟ	চৎকদ্র	catkadra
ᱠᱟᱨᱢᱟ	হাঈব্র	hAybra
ᱠᱟᱨᱢᱟ	তৌখ্রে	tauKre
ᱠᱟᱨᱢᱟ	ষ্টেত	stet
ᱠᱟᱨᱢᱟ	ব্লোকড	bloked

Table 7: Examples of Isolated Consonants Denoted the Apun Iyek

4. Meitei-Bengali-Devanagari to and from transliteration

The tool is developed on the Java platform. It is a web application running on Apache Tomcat web server. This is available online on our website <http://sanskrit.jnu.ac.in/ile/index.jsp>.



Figure 2: The tool on the server

With the thorough linguistic study of the Manipuri sound system, we tried to make relevant representations of Meitei in Bangla and Devanagari. Most of the Indian scripts have the same sound systems and parallel letters, as have derived from Brahmi. Manipuri uses lesser number of sounds in comparison to Devanagari and Bangla.

All the Meitei characters have been compared with Bangla and Devanagari and parallel arrays have been created. In case of unmatched letters, the closest letter in the target language has been substituted. Conditional statements are also used to control string processing for language specific nuances.

```

if(multilingual_word.equals("Manipuri")){
    new_word += MeiteiMayek [j];
    break;
}

String MeiteiMayek[j]
={ "\uABC0", "\uABC8", "\uABD2", "\uABD6", "\uABD7", "\uABD9", "\uABD0", "\uABC0", "\uABC8", "\uABD2", "\uABD6", "\uABD7", "\uABD9", "\uABD0", "\n", "\uABD1", "\uABD1\uABE5", "\uABCF", "\uABCF", "\uABE8", "\uABE8", "\uABD1\uABE6", "\uABD1\uABE9", "\uABD1\uABE7", "\uABD4\uABE4", "\uABD4\uABE4", "\uABC2\uABD4\uABED\uABE4", "\uABC2\uABD4\uABED\uABE4", "\uABD1\uABE3", "\uABD1\uABE6", "\uABCF", "\uABC0", "\uABC8", "\uABD2", "\uABD8", "\uABC9"

```

, "\uABC6", "\uABC1", "\uABD6", "\uABD3", "\uABC5", "\uABC7", "\uABCA", "\uABD7", "\uABD9", "\uABC5", "\uABC7", "\uABCA", "\uABD7", "\uABD9", "\uABC5", "\uABC4", "\uABD0", "\uABD5", "\uABDA", "\uABC3", "\uABCC", "\uABD4", "\uABC2", "\uABC B", "\uABC1", "\uABC1", "\uABC1", "\uABCD", "\uABE5", "\uABE4", "\uABE4", "\uABE8", "\uABE8", "\uABED\uABD4\uABE4", "\uABED\uABD4\uABE4", "\uABE4", "\uABE4", "\uABE6", "\uABE9", "\uABE3", "\uABE3", "\uABE3", "\uABE7", "", "\uABEA", "\uABEA", "\uABCD", "\uABEB", "\uABF0", "\uABF1", "\uABF2", "\uABF3", "\uABF4", "\uABF5", "\uABF6", "\uABF7", "\uABF8", "\uABF9", "\uABD1\uABE5\uABEA", "\uABE5\uABEA", "\uABE9\uABE9", "|"};

6. Challenges in mapping

The consonant finals or the consonants in isolation (the Lonsum Iyeks) are significant in Manipuri. The appropriate letter for a Lonsum letter is a consonant with the ̣ (Halanta), as shown above. However, as in most of the modern Indian languages, Halanta is not used to indicate a consonant final. It therefore creates problems in one-to-one mapping for consonants.

Another challenge comes from cluster conversion. The character Apun Iyek is used at the end of a cluster to identify the letters are in a cluster.

B/D cluster consonants	Values in Engine (Machine)	Corresponding Meitei Scripts
ꯀ	ꯀ+ꯃ+ꯀ	ꯀ+ ꯃ+ (= ꯀꯃ)
ꯁ	ꯁ+ꯃ+ꯀ	ꯁ+ ꯀ+ (= ꯁꯀ)
ꯂ	ꯂ+ꯃ+ꯀ	ꯂ+ ꯀ+ (= ꯂꯀ)

Table 8: Values for the Cluster Consonants

As shown here, for Bangla (and Devanagari), the cluster indicator is in the middle of the two consonants while it is at the end for the Manipuri. So, it cannot match either the phoneme or the conditional statements.

Indigenous Manipuri does not have retroflex sounds and the current scripts also do not have any retroflex character, though some scholars have tried to argue in support of having such sounds. They give the example of the word *ShaN* (a cow/bull).

Since the older Meitei Mayek standard in Unicode did not have the retroflex characters, the retroflex characters from the source text were replaced with

dentals. However, in the newer Unicode extension, the retroflex characters have been included.

7. Conclusion

A thorough study of Meitei, Bangla and Devanagari scripts helps us to identify the corresponding or nearest letter for substitution in automatic transliteration. Meitei scripts have fewer characters compared to Bangla and Devanagari thus creating a many-to-one map situation from Bangla/Devanagari to Manipuri and one-to-many for reverse case.

This feature of Meitei and the fact that it uses special conventions to identify clusters and lone consonants, creates challenges for mapping.

The Unicode Meitei Mayek Extensions mainly contains characters which resemble with diverse sounds in the Sanskrit/Bangla alphabet. This extension can be applied for developing an unambiguous transliteration system using them as internal coding values. However, since a majority of Manipuri adult speakers use Bangla script, we need to create a converter from it to Manipuri script so that corpora creation in the language continues.

8. References

- Alan Wood (2012). *Test for Unicode support in Web browsers: Meitei Mayekv*. Retrieved from <http://www.alanwood.net/unicode/meitei-mayek.html>
- Akanksha Bansal, Esha Banerjee, Girish Nath Jha (2013). Corpora creation for Indian Language Technologies – the ILCI project. Zygmunt Vetulani & Hans Uszkoreit (Eds), *Human Language Technologies as a Challenge for Computer Science and Linguistics (proceedings of the 6th LTC)* (pp.253-257). Poland: Fundacja, Uniwersytetu im. A. Mickiewicza, Poznan,.
- Chelliah, S.L. (1990). Level-ordered morphology and phonology in Manipuri, *Linguistics of the Tibeto-Burman Area*, vol. 13, no. 2 (pp. 27-72).
- Chungkham Yashwanta Singh (2000). *Manipuri Grammar*. New Delhi: Rajesh Publications.

- Grierson, G.A. and Konow (1967). Tibeto-Burman Family. *Linguistics Survey of India. Vol. III.* Delhi: Motilal Banarsidass.
- Inder Singh (1975). *Manipuri Phonetic Reader.* Mysore: Central Institute of Imphal.
- I. Allen, Julie D. (2009). *Unicode5.2.0.* Retrieved from <http://www.unicode.org/versions/Unicode5.2.0/ch10.pdf>.
- Kishorjit Nongmeikapam, Nignombam Herojit Singh, Bishworjit Salam, Sijaji Bandyopadhyay (2011). Transliteration of CRF Based Multiword Expression in Manipuri: From Bengali Script Manipuri to Meitei Mayek (Script) Manipuri. *International Journal of Computer Science and Information Technologies, Vol. 2 (4)* (pp. 1441-1447).
- Mutua Bahadur (2010). *Illustrated Manuscripts of Manipur.* Retrieved from E-Pao Picture Gallery: http://www.e-pao.net/epGallery.asp?id=1&src=Arts_Dances/Mutua_Bahadur/MBManuscript2010_3.
- Naorem Sanajaoba (1991). Minu Leima Lol, Meitei Alphabet (Meitei Mayek), the Outlines of the Constitutional History of Manipur. *Manipur: Past and Present, Vol. II.* New Delhi, Mittal Publications.
- N. Khelchandra Singh (1975). *Manipuri Language Status and Importance.* Imphal: N. Tombi Raj Singh publication.
- Pravabati Chingangbam and Tabish (2011). *Meetei Mayek.* Retrieved from <http://tabish.freeshell.org/Eeyek>.
- Radhabinod Aribam Sharma (2010). *History Of Meetei Mayak.* Retrieved from <http://manipurihistory.wordpress.com/2010/02/12/history-of-meetei-mayak>.
- Thoudam Doren Singh (2012). Bidirectional Bengali Script and Meetei Mayek Transliteration of Web Based Manipuri News Corpus. *South and Southeast Asian Natural Language Processing (SANLP)* (pp. 181–190).

Indian languages on the TypeCraft platform – the case of Hindi and Odia

**Girish Nath Jha, Lars Hellan, Dorothee Beermann, Srishti Singh, Pitambar Behera,
Esha Banerjee**

Jawaharlal Nehru University, New Delhi, India
Norwegian University of Science and Technology, Trondheim, Norway
(girishjha, singhsriss, pitambarbehera2, esha.jnu@gmail.com
(lars.hellan,dorothee.beermann)@ntnu.no

Abstract

This paper describes a plan for creating an aligned valence- and construction repository for Indian languages, starting with Hindi and Odia. The project will be a collaboration between the ILCI group at Jawaharlal Nehru University, India and the TypeCraft (TC) group at NTNU. In the first phase of the project, our task is to find a suitable theoretical framework for annotation at valence and construction level. In the second phase of the project (if the data download from the govt. of India data center site is opened to all), we will include a data portability and exchange module which will facilitate data import/export between the TC and the ILCI repositories.

Keywords: TypeCraft, valence, database, Indian languages, ILCI

1. The project

This paper describes a plan for creating an aligned valence- and construction repository for Indian languages, starting with Hindi and Odia. The project will be a cooperation between the ILCI group at Jawaharlal Nehru University, India, and the TypeCraft (TC) group at NTNU, Norway. In the first phase of the project, our task is to find a suitable theoretical framework for annotation at valence and construction level. In the second phase of the project (if the data download from the govt. of India data center site is opened to all), we will include a data portability and exchange module which will facilitate data import/export between the TC and the ILCI repositories.

1. In the following we first describe the two platforms involved, then we discuss the notion of valence and ways of representing this in a repository. We then exemplify the tasks involved through examples of annotations of the two languages.

2. The ILCI platform

The Indian Languages Corpora Initiative (ILCI) project was initiated by the Technology Development for Indian Languages (TDIL) program of the Department of Electronics and Information Technology (DeitY), Government of India in 2009 to facilitate parallel corpus building for the scheduled languages of India, including English (Jha, 2010). A consortium of major universities and institutions were formed from each state representing the state language. Urdu and English, being pan-Indian languages, were included with Hindi at Jawaharlal Nehru University, New Delhi, which also acted as the Consortium Leader. Phase 1 of the project, which ended in 2012, saw the development of 50,000 parallel, manually translated and Part Of Speech (POS) annotated sentences in 12

languages, including the source language Hindi, in the domains of Health and Tourism (Choudhary and Jha, 2011). The project is currently in Phase 2, which also saw the inclusion of 5 new languages, most notably from the Tibeto-Burman language family of north-east India. The aim of the current phase is to add 50,000 new sentences in each of the 17 languages in the domains of Agriculture and Entertainment (Bansal et al, 2013). The data is available for use for research purposes at tdil-dc.in.

The Bureau of Indian Standards (BIS) standardized a POS scheme tagset in 2011, based on the Indian Languages Machine Translation (ILMT) guidelines, which is used to annotate the ILCI corpus. This tagset employs layered hierarchical schemata, though it is built on a coarser level and uses categories of only two levels. This tagset has been developed keeping in mind the various characteristics of languages spread across different language families in India. A human tagger is free to choose from among them the POS tag which best fits his language. The tagset has 11 broad lexical categories, which contain sub-levels e.g. Personal, Deictic, Indefinite, Reflexive etc under the category Pronoun (Nainwani et al, 2011). The collection of corpus, translation and annotation tasks are carried out by the use of online tools available at Sanskrit.jnu.ac.in. The annotation tool (ILCIANN) uses prior information to facilitate semi-automated process of tagging (Kumar et al, 2012).



Figure 1: ILCI platform

3. The TypeCraft platform

TypeCraft (Beermann and Mihaylov 2013) is a linguistic service featuring a multi-lingual database and an online Interlinear Glosser which in addition to morpheme and word level annotations allows phrase level tagging.

Figure 2 shows a Ga IGT seen from inside of the TypeCraft (TC) linguistic editor. The Editor uses the standard tier format for interlinear glossing. In addition, Phrase level annotation, here called Construction Labeling, can be added through the use of an additional annotation matrix, shown below the IGT. The rightmost part of the screenshot, furthermore, shows drop-down menus for 8 named phrasal parameters.

In this way TC allows the harvesting of valence information in a linguistic environment designed for the manual annotation of data, especially from lesser described languages. TC 2.0 which is at present under development will allow the import of data from other linguistic platforms (Bouda & Beermann) directly into TypeCraft, which then can be used as a tool that allows the easy addition of valence annotation to already structured data.

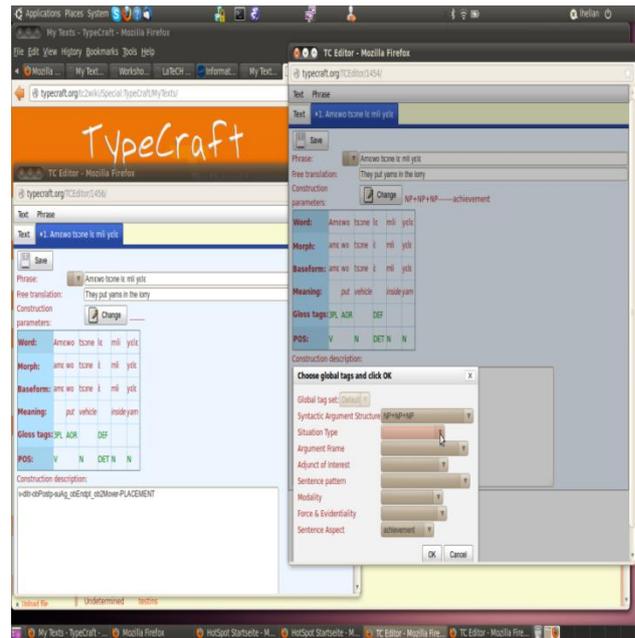


Figure 2: Valence-related annotation in TypeCraft

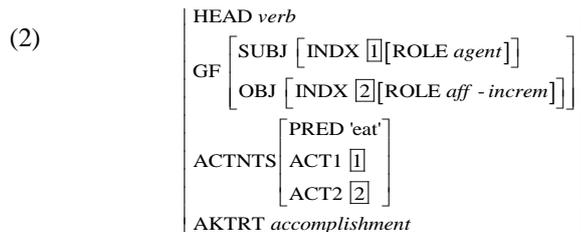
4. Annotating for valence

The notion of valence as such, originally proposed by Tesnière 1988, represents a multiplicity of factors which could be summarized as follows

- (1)
 - syntactic argument structure (or *valence*: whether there is a subject, an object, a second/indirect object, etc., referred to as *grammatical functions*);
 - semantic argument structure, that is, how many *participants* are present in the situation depicted, and which *roles* they play (such as ‘agent’, ‘patient’, etc.);
 - linkage between syntactic and semantic argument structure, i.e., which grammatical functions express which roles;
 - identity relations, part-whole relations, etc., between arguments;
 - aspect and Aktionsart, that is, properties of the situation expressed by a sentence in terms of whether it is dynamic/stative, continuous/instantaneous, completed/ongoing, etc.;
 - type of the situation expressed, in terms of some classificatory system of situation types.

The main state of the art analytic mechanism covering phenomena like those in (1) is that of *attribute-value matrices (AVMs)*, exemplified by (2) below for a sentence like *The boy eats the cake*, displaying grammatical functions (GF), Aktionsart (AKTRT), and semantic participants (as ‘ACTNTS’, with sub-attributes ‘ACT1’, ‘ACT2’, with semantic roles indicated), and with interlinking between the referential indices associated with the grammatical functions and those associated with the participants. The value of an attribute may open for other attributes recursively, by which the formalism is powerful enough to accommodate parameters like those

in (1).



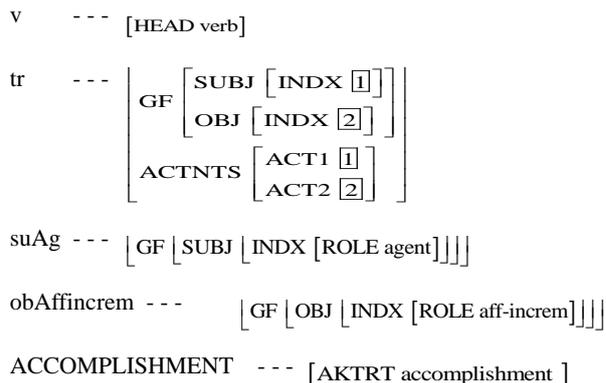
There are essentially two linguistic approaches using this general formalism which have been able to define grammar analysis systems amenable to automatic parsing and readily applied to a multiplicity of languages, viz. *Lexical Functional Grammar (LFG)*¹ and *Head-driven Phrase Structure Grammar (HPSG)*². Both approaches are rooted in unificational constraint/information-based grammar design, and both have decent-coverage grammars for around 10-12 languages in Europe and Asia (with 2/3 of the languages in common). LFG grammars in particular share a format of representation called *f-structure*, where grammatical functions and other syntactic factors in (1) are exhibited, while HPSG grammars share a semantic representation format called Minimal Recursion Semantics (MRS; Copestake et al. 2005), where every sentence is rendered on a form reminiscent of Predicate logic.

The goal of a valence annotation project is obviously not to build computational grammar fragments; nevertheless an understanding of how any given valence annotation relates (or ‘maps’) to the kind of structures defined in frameworks like those mentioned contributes to annotation consistency, both formally and in the application of valence-related notions. We illustrate these points relative to the two formats of valence annotation exemplified in fig. 2. The one called ‘Construction Labeling’ has a special format of hyphenated strings, an example being (3) below, which could be used for annotation of *The boy eats the apple*. Each hyphenated expression can be read as a property of the construction annotated, so that ‘v’ means verb-headed, ‘tr’ means ‘transitive’, ‘suAg’ means that the subject is an agent, etc.³

(3) v-tr-suAg_obAffinrem-ACCOMPLISHMENT

This hyphenation notation can be formally construed as unification, and if each constituent label receives a ‘local’ AVM definition like in (4), then the operation of unification applied across the string will yield the feature structure shown in (2).⁴

(4)



As for the codes available from the drop-down menus indicated in fig. 2, there are no such explicitly defined correspondences so far, but they can be provided in similar fashion. The most fine-grained labels in this format are those of ‘Syntactic argument structure’ (SAS), which is exemplified in Table 1 with some English sentences. In this system, arguments of the verb are identified, by linear order, in terms of POS-based syntactic category and some functional specifications such as ‘predicative’ and infinitival control properties:

EXPL	It rains
EXPL+APpred+S	It is nice that you came
EXPL+INF	It starts to dawn
EXPL+NP	There sits a mouse
EXPL+NP+Sdecl	It surprises me that he came
EXPL+NP+SquestWH	It eludes me who came
EXPL+NPpred+Sdecl	It is a success that he came
NP	John sits
NP+APpred	He is cute
NP+INF	He tries to run
NP+INF:equiSBJ	He tries to run
NP+INF:raisingSBJ	He seems to run
NP+NP	He sees the danger
NP+NP+APpred	She makes the city happy
NP+NP+INF:equiOBJ	She asks John to run
NP+NP+INF:equiSBJ	She promises John to run
NP+NP+INF:raisingOBJ	She expects John to come
NP+NP+NP	She gives the boy a house
NP+NP+NPpred	She calls him a hero
NP+NP+PP	She tells a story about him
NP+NP+PP+PP	She tells a story to Peter about Pi
NP+NP+PRTPpred:as	She regards him as a hero
NP+NP+Sdecl	She tells him that he has lost
NP+NP+SquestWH	She tells him who lost
NP+NP+SquestYN	She tells him whether he has lost
NP+NPpred	He is a fool
NP+NPrefl	He perjures himself
NP+NPrefl+NPpred	He considers himself a hero
Sdecl+NP	That he came surprises me

Table 1. Examples of SAS specifications for English

An alternative view on constructions can be presented in terms of notions like ‘transitive’, ‘intransitive’, and such, and through its 8 construction parameters with drop-down

¹ Cf. Bresnan 2001.

² Cf. Pollard and Sag 1994, Copestake 2002.

³ For extensive definitions of the code used, see (Hellan and Dakubu 2010).

⁴ See (Hellan 2010, Hellan and Dakubu 2009, Hellan and Beermann 2011).

menus, TC seeks to provide the mosaic of annotation specifications that can together give an adequate representation of the argument structure of a sentence.

In this respect, the Construction Labeling system illustrated in (3) and (4) may be seen as presenting as many constructional specifications as possible in a single string; which format one prefers may depend on the purpose of the specification.

Orthogonal to the issue of the formal definition of an annotation system is the question which are the valence/construction types relevant for Indian languages. It is not a given that all the types exemplified in table 1 are relevant for Indian languages, and it is well known that Indian languages contain types that go far beyond the inventory that one would establish for English. In parallel to the task of actually annotating a chosen assembly of sentences from Hindi and Odia, the project thus undertakes the task of identifying what may be called a ‘valence profile’⁵ of the two languages.

The next sections describe some salient properties of these languages, and also exemplify annotation of sentences using TC annotation code not only for valence properties but also for morphological annotation.

5. The Odia language

Odia, a recently declared classical language of India, earlier a Scheduled Language, is descended from the Indo-Aryan language family. Since it belongs to this family, it bears almost all the typical characteristics of its proto language such as- basic S+O+V sentence structure, morphologically rich with PN and TAM features, serial verb constructions, ECV and causative constructions, conjunct verbs, and enriched with collocations and metaphoric usage of language. We here exemplify five features of the language, illustrated with our TC annotations.

A. SOV word order

This constituent order is exemplified in (5), showing in the first line the ‘global’ properties, then the string, then a free translation, and then the table with morphological analysis:

(5)

imperfective-declarative ----directedMotion-

ସେ ମାଠିଆ ଭିତରକୁ ଚାହିଁଲା

“he peeped into the pitcher”

ସେ	ମାଠିଆ	ଭିତରକୁ	ଚାହିଁଲା
ସେ	ମାଠିଆ	ଭିତର	କୁ ଚାହିଁଲା
SBJ.NOM.AGT.3SG	pitcher.ACC	inside.LOC to peep.3SG.PAST	
PN	Ncomm	PPOST	V

Generated in TypeCraft.

⁵ See (Hellan et al. 2013, Hellan et al. 2014) for discussions of this and related notions in the context of building a multilingual lexical valence database.

This word order is generally followed, but there is the possibility and acceptability of SVO and VSO constructions like the above example, if the order is reversed accordingly:

(6)

- a. Se cahiMIA mAthiA bhitarku
SUBJECT VERB OBJECT
- b. cahiMIA Se mAthiA bhitara-ku
VERB SUBJECT OBJECT

B. Agreement

Verbs agree with their subjects on persons and numbers. So sentences can appear without subjects. In Oriya though the lexical gender is present, there is no any grammatical gender.

(7)

ସେ ପାଣି ପିଇବାକୁ ଚେଷ୍ଟା କଲା
“he tried to drink water”

ସେ	ପାଣି	ପିଇବାକୁ	ଚେଷ୍ଟା	କଲା
ସେ	ପାଣି	ପିଇବା	କୁ ଚେଷ୍ଟା	କଲା
he.3SG.NOM	water.3SG	drink.INF to try	do.3SG.NOM.PAST	
PN	N	V	COMP	AUX

Generated in TypeCraft.

C. Serial verb construction:

This phenomenon is exemplified in (8), again with global factors indicated first:

(8)

imperfective-visualEvidence--serialVerbConstruction-adverbPh
rase:manner-perception-NP+ADVppred

ଧୂରେ ଧୂରେ ପାଣିର ସ୍ତର ଉପରକୁ ଉଠିବାକୁ ଲାଗିଲା
“the level of water started rising slowly”

ଧୂରେ	ଧୂରେ	ପାଣିର	ସ୍ତର
ଧୂରେ	ଧୂରେ	ପାଣି	ର ସ୍ତର
slow.REDP	slow.REDP	water.SG	POSS level.SG.NOM
ADVm	ADVm	N	Nbare

ଉପରକୁ	ଉଠିବାକୁ	ଲାଗିଲା
ଉପର	କୁ ଉଠିବାକୁ	ଲାଗିଲା
up.LOC	to rise.INF	start.3SG.NOM.PAST
ADVplc	V1	V2

Generated in TypeCraft.

D. Causativization

In causative construction the root form of the verbs gets a derivational morpheme, reduction of the vowels in the root

forms, and the derived verb can have up to four arguments. Two word formations are shown in (9):

- (9)
 /khA/ - eat
 /khuA/-feed
 /nAc/- dance
 /nacA/-make someone dance

E. Conjunct verbs ('converbs'):

These are formed with the addition of markers like /ki/ and /kari/.

- (10)
 ସେ ପାଣି ପିଇ ଖୁସିରେ ଉଡ଼ିଗଲା

"he drank water and flew away happily"

ସେ	ପାଣି	ପିଇ	ଖୁସିରେ	ଉଡ଼ିଗଲା
ସେ	ପାଣି	ପିଇ	ଖୁସିରେ	ଉଡ଼ିଗଲା
3.SG.NOM	Water.SG	drink.PFV	happily	fly.3.SG.NOM
PN	N	V1	ADV _m	V2

Generated in TypeCraft.

6. The Hindi language

Hindi is an Indo Aryan Language with relatively free word-order which enables its components to scramble within the construction. It is an official language of 12 states in India. Hindi is a highly inflectional and morphologically rich language with gender, number, person (GNP) and tense aspect and mood (TAM) information embedded with the verbs; rich use idiomatic expressions like collocation and metaphor. Conjunctive Participle (Converbs), causativization, ECV, modifiers, passivization, scrambling, and PRO drop are commonly found constructions in Hindi. The particles like *_hi_*, *_bhi_* and *_to_* etc are free floating and can be placed anywhere in the sentence according to the focus of discourse.

We show some examples of data entered in TC, in the same format as in the above section.

- (11)
 state-declarative ---S:time-

एक कौआ था

"There was a crow"

एक	कौआ	था
एक	कौआ	था
one	crow.SBJ.NOM.3SG.MASC.ANI	be.PAST.1SG.MASC
CARD	Ncomm	V

Generated in TypeCraft.

- (12)
 imperfective-declarative
 -adverbPhrase:manner-directedMotion-NP+ADV_Ppred

उसने झाँक के मटके में देखा

"He peeped into the pitcher"

उसने	झाँक	के
उस	ने	झाँक
he.SBJ.3SG.MASC.NOM.ANIM	ERG	peep.PRTV
PN	V1	V2

मटके	में	देखा
मटके	में	देखा
pitcher.ACC.INANIM	in.LOC	see.PAST.3SG.MASC
N	PPOST	V

Generated in TypeCraft.

- (13)
 imperfective-visualEvidence--serialVerbConstruction-adverbPhrase:manner-perception-NP+ADV_Ppred

धीरे-धीरे पानी उपर आने लगा

"Gradually the water level rose"

धीरेधीरे	पानी	उपर	आने	लगा
धीरे	धीरे	पानी	उपर	आने
slow	slow	water.NOM	up.LOC	come start
ADV _m	Ncomm	ADV _m	V1	V2

Generated in TypeCraft.

7. TC and ILCI platforms and data binding

The goal is to use the TC interface for further annotation, primarily for verb argument structure, and also morphological properties, both of which cannot yet be accommodated in the ILCI interface at this point. Both the TC and the ILCI are cloud facilities and it would be easier to either let annotators benefit from additional linguistic information on the TC platform and be able to bind and search data by firing queries. The data ids from the ILCI corpora can point to additional information on the TC platform and can be assembled from ether platforms by firing suitable queries.

To summarize, the TC annotation interface covers the following aspects, described and illustrated above:

- Transcribing the text in the language(script of the language), for example Hindi in Devanagari

- Free translation of the text in English
- Words involved in the string
- Morphemic breaks in the words
- Possibility of transliteration for some languages into Latin script
- Base form of words/verbs in the string
- Meaning of the morphemes in the string
- Interlinear glossing of the morphemes
- Construction description, which involves the information about the type of construction feeded, it does not carry any sub-category and it is up to the annotator to give a suitable title to the construction. The Construction Labeling system can be used here (cf. Section 4).
- Global Tags, which is the syntactic and semantic description of the string with respect to its syntactic argument structure, sentence type, argument frame, modality, force and evidentiality, sentence aspect etc. Each contains a list of sub-categories to choose the tags from, presented in drop-down menus (cf. Section 4).

Given the possibility that the ILCI data would be open to everybody (not only to Indian researchers at this point), we can work out a model where data exchange between the two platforms can be automated. This will require a closer understanding of the data structures and backend processes in each platform so that data import/export can be worked out. The ILCI data (as of now) has translations from Hindi to 16 other Indian languages with POS (BIS standard) annotation which is hierarchical. The current phase of development is going to include chunk information as well. The TC format has richer linguistic information and various levels. The ILCI data can benefit by adding additional layers of information while the TC format can get additional languages.

8. Conclusion

The joint effort between NTNU and JNU research groups is in an initial development stage with significant variables. Some sample data for Hindi and Odia have been entered by research students. This work is expected to pick up in next couple of months. The overall goal and the philosophy of the project is to allow data integration from large data repositories across the continents with obvious benefits on either side. The project can be further developed under a joint S&T collaboration initiative by India and Norway.

9. References

- Bansal, A., Banerjee, E., & Jha, Girish Nath (2013) Corpora Creation for Indian Language Technologies – The ILCI Project. In the sixth Proceedings of Language Technology Conference (LTC '13).
- Beermann, D. and Mihaylov, P. (2013). Collaborative databasing and Resource sharing for Linguists. In: *Languages Resources and Evaluation*. Springer
- Choudhary, N., & Jha, Girish Nath (2011) Creating Multilingual Parallel Corpora in Indian Languages. In: Proceedings of 5th Language Technology Conference, Fundacja Uniwersytety im. A. Mickiewicza, Poznan. Pp. 85-89
- Hellan, L. and M.E.K. Dakubu (2009): A methodology for enhancing argument structure specification. In *Proceedings from the 4th Language Technology Conference (LTC 2009)*, Poznan.
- Hellan, L. and M. E. K. Dakubu, 2010: *Identifying Verb Constructions Cross-Linguistically. Studies in the Languages of the Volta Basin* 6.3. Legon: Linguistics Dept., University of Ghana. (http://www.typecraft.org/w/images/d/db/1_Introlabels_SLAVOB-final.pdf, http://www.typecraft.org/w/images/a/a0/2_Ga_appendix_SLAVOB-final.pdf, http://www.typecraft.org/w/images/b/bd/3_Norwegian_Appendix_plus_3_SLAVOB-final.pdf)
- Hellan, L., D. Beermann and T. Bruland.(2013) Towards a multilingual valence repository for less resourced languages. In *Proceedings from the 4th Language Technology Conference (LTC 2009)*, Poznan.
- Hellan, L., D. Beermann, T. Bruland, M.E.K. Dakubu, and M. Marimon (2014) MultiVal: Towards a multilingual valence lexicon. *LREC 2014*.
- Jha, Girish Nath (2010). The TDIL program and the Indian language corpora initiative (ILCI). In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010)*. *European Language Resources Association (ELRA)*.
- Kumar, R., Kaushik, S., Nainwani, P., Banerjee, E., Hadke, S., & Jha, Girish Nath (2012). Using the ILCI Annotation Tool for POS Annotation: A Case of Hindi. *IJCLA Vol. 3, NO. 2, Jul-Dec 2012*, pp. 93–104.
- Nainwani, P., Banerjee, E., Kaushik, S., & Jha, Girish Nath (2011). Issues in annotating less resourced languages—the case of Hindi from Indian Languages Corpora Initiative (ILCI). In the fifth Proceedings of Language Technology Conference (LTC '11).

Issues in Mapping of Sanskrit-Hindi Verb forms

Kumar Nripendra Pathak and Girish Nath Jha

Special Centre for Sanskrit Studies,
Jawaharlal Nehru University,
New Delhi, India -110067.
(nri.pathak, girishjha)@gmail.com

Abstract

Verb handling is the most important task for Machine Translation. A thorough syntacto-semantic study has been done in the Indian Grammatical Tradition which is highly appreciated by all the modern linguists worldwide. This paper deals with the syntactic patterns between Sanskrit-Hindi Verbs to formulate the possible algorithm to map the verbs for Sanskrit-Hindi Translator (SaHiT). This effort will help in producing linguistic rules for the required tool which can handle the verb forms in SaHiT.

Keywords: Sanskrit, Hindi , Verb, SaHiT, Machine Translation, rule

1. Introduction

Famous NASA Scientist Rick Briggs¹ (1985) in his research article “Sanskrit & Artificial Intelligence” drew great attention towards Sanskrit because of its grammatical technicalities for Natural language processing. Since 1991, NLP started in India, and many researchers explored the possibilities of developing “intelligent machines” to process the Natural Language. Panini, who had already processed the spoken language of his time, presents a model for computational algorithm. Nicholas Ostler’s², (a British scholar and author, who first studied Greek, Latin, philosophy, and economics and later studied under Noam Chomsky at the Massachusetts Institute of Technology, where he earned his Ph.D. in linguistics and Sanskrit) work “Empires of the Word: A Language History of the World” (2005) documents the spread of language throughout recorded human history. Nicholas Ostler (2001) in his research article *Sanskrit Studies as a Foundation for Computational Linguistics*, said “There is also evident competition in India both from Hindi and from English, as de facto and de jure languages of pan-Indian communication in the modern world. Nevertheless, Hindi has not had the benefit of 2,500 years of linguistic analysis on which to found its computer development. And English, despite its feverish development over the past 250 years, can never

offer the well-established cultural links with languages all over India that are inalienable from Sanskrit”....

“we can consider the potential role of Sanskrit in the future electronic notation, analysis and transmission of languages world-wide”.

At present, various efforts are being made towards the development of Indian Language to Indian Language Machine Translation systems. Sanskrit-Hindi Machine Translation System is one of them. In this particular task, verb handling has always been the challenging topic because of divergence and linguistic contrast between both language pairs. Some efforts have been made to deal with the verbs but much is needed.

Sanskrit Verb Argument Valence: A Computational Analysis by Subhash Chandra says that the verb argument valence analysis system and Knowledge database are the outcome of their research and development³. Sanskrit Wordnet discussed the divergence of verbs in Sanskrit and Hindi and its implication for constructing verbal synsets in SWN⁴. *Automatic Identification and Analysis of verb groups in Hindi* is the latest research work related to Hindi Verbs⁵. *Knowledge base for Karma karak* by Manji Bhadra, JNU and *Ontological Knowledge Base for selected verbs of Sanskrit and Bangla* by Subhash Chandra are dealing with the Sanskrit Verbs to

¹ <http://www.vedicsciences.net/articles/sanskrit-nasa.html>

² <http://www.emille.lanccs.ac.uk/lesal/ostler.pdf>

³

http://www.researchgate.net/publication/233341515_Sanskrit_Verb_Argument_Valence_A_Computational_Analysis

⁴ <http://www.cse.iitb.ac.in/~pb/papers/gwc12-swn-verb.pdf>.

⁵ Narayan Kumar Chaudhary, Centre for Linguistics, JNU.

represent the semantic processing. But the mapping of verb for Sanskrit-Hindi Translator and the possible rules for acceptable output has not been discussed. This paper is going to discuss some differences and present the issues and challenges in verb mapping for SaHiT. As the Sanskrit verbs are of two types: tingant and kridanta, this paper will be restricted only to the tingant part because kridant may also be used as noun (ram+ghany=raam), adverb (pachamaanah), adjective (pachantam), indeclinable (paayam paayam), gerund (gatvaa) etc which needs to be discussed separately.

2. Sanskrit-Hindi Verbs

According to Paninian Grammar, generally sentence has two main parts – *subanta* and *tinganta*. He says – ‘*sup-tingantam padam*’⁶. Here *subanta* is noun and the *tinganta* is verbs in a given Sanskrit syntax. *tinganta* forms are made of two kinds of roots, first (the original) one which are listed in the ten ganas of Paninian dhatupath, and the second one which are obtained by adding suffixes to the primary root to express some special meaning. These roots have different morphological forms in the ten lakaras and their forms further differ in their atmane-pada and Parasmai-pada divisions. The ten lakaras express the tense and mood. Present Tense takes lat-lakaara form, Future takes lut & lrut forms and Past tense takes lit, lang and lung lakaaras. Mode takes lot, vidhi-ling, aasirling and lrung lakaaras. The Sanskrit forms are inflectional while Hindi forms are periphrastic, the mapping of verb forms between two needs close observation.

2.1 Present Tense

In Sanskrit sentence, lat-lakara is used to convey the present. The lakaras are divided into three persons and three numbers so the verb has nine forms in parasmai-pada and nine forms in atmane-pada. The atmanepada type of morpho-syntactic difference has not been discussed in Hindi forms. In Sanskrit, the meaning of a ubhayapadi verb (having both the forms: atmanepada and parasmaipada respectively) does not change in active voice, thus we can take only parasmai-pada forms (for analysis) which are widely used in classical Sanskrit writings. The pattern of Lat-lakaara forms have roots followed by suffix tip, tas jhi.... and so on⁷. For

clear understanding, we can take root paTh (to read): paThati, paThatah, paThanti ... so on. Depending on the gender of agent, the form paThati can be translated as – paDhataa hai, paDhatI hai, PaDhate hain, paDha rahaa hai, PaDha rahI hai, PaDha rahe hain etc. A single Sanskrit verb form represents multiple Hindi verb forms. In this regard, it is important to note that Hindi forms are governed by the gender of agent as well as the honorific use. What is happening here is basically the root is giving the meaning and suffixes ‘tip, tas & jhi’ are giving the forms {taa hai, tI hai, te hain, rahaa hai, rahi hai, rahe hain} which is being added to the root form of Hindi. Here we should clearly understand that honorifics (bhavaan/bhavati and any respected person) have third person forms (tip, tas, jhi) in Sanskrit and it takes root form +{te hain/tI hain/rahe hain/rahi hain} in Hindi.

Morpho-syntactically, something different happens with the verbs used with second person ‘You (tvam)’ & first person ‘I/we’. Here we can see that the verb form with tvam, yuvaam, yooyam is the root + si, thah & tha. i.e. pathasi, pathathah & pathatha. Here we get the translation in Hindi: [tum/tum_dono/tum_sab {padha_te ho, padha_rahe ho}] So, ‘_te ho’ & ‘_rahe ho’ of Hindi form is representing masculine forms. When the agent is feminine, we will see ‘_tI ho’ & ‘_rahI ho’ with the verb root of Hindi.

First person, I & we (aham, aavaam, vayam) take [root+{_mi/vas/mas}] in Sanskrit, and in Hindi, it takes [root+ {_taa hoon/ tI hoon}/ {_rahaa hoon/rahI hoon}(singular form)] and dual as well as plural takes [root+ {_te hain/ rahe hain}].

In this way we find that the nine forms of Sanskrit verbs (in Lat lakaara) are presenting so many forms in Hindi. The Hindi forms are depending on the gender while Sanskrit (*tinganta*) does not. Hindi takes different forms to represent simple present and present continuous tense while Sanskrit shows both from one verb form.

2.2 Past Tense

In Sanskrit, Past Tense takes three lakaras, lit lakara (past perfect), lan lakara (past imperfect) and lun lakara (Aorist). Lit or perfect is used in the sense that the past period was not seen by the speaker which refers to the historical past. While analyzing the past we find that the aorist would refer only to eventualities that have happened earlier during the present day, the imperfect

⁶ P-1.4.14

⁷ P-3.4.78

only to witnessed eventualities that have happened before the present day, and the perfect to non-witnessed eventualities that have happened long time ago. Paul Kiparsky explained this while referring that the perfect should block the imperfect and the imperfect in turn should block the aorist and noticed that the aorist is optionally used for remote past⁸. Dealing with past, V S Apte⁹ says that earlier the past forms were used in their exact senses but when Sanskrit became a less spoken language, the writers began to use these three tenses promiscuously. He differentiates the past tense by noticing the fact that the imperfect and the perfect are used in narrating events of remote past occurrence, and the aorist is used in dialogues and conversations which refer in recent past action but it is not used to denote past specified time, or to narrate events. Apte gives the examples of imperfect and perfect past from purushasukta for the events narrated referring to the non-witnessed past and presents the example of Aitareya Brahman for the recent past which is shown by the aorist. In post-Paninian Sanskrit, the aorist can refer to any past event, whereas the imperfect and perfect are restricted to the described events only.

If we compare the past forms of Sanskrit with the Hindi forms, we can find the differences of gender same as present (hota tha, hoti thi etc). But the time reference of the non-witnessed past is being shown as hua karte the, hue the, huyi thi (babhoova) etc with reference to the historical narratives. ho rahaa tha, huaa/hota tha, and ho chukaa etc are representing the past usage in Hindi for which abhavat/abhoot both are used. Here we can see that the use of imperfect and aorist is not clear in Hindi while perfect (past-perfect) is very clear in both the languages. We can find both the term 'ho chukaa' and 'ho chukaa tha' where we need to identify the remote past and recent past. For example, we can take root han to see the different form in both the languages:

Perfect: jaghaana = maara dala thaa, maaraa thaa, maarataa thaa, [(Raja choro ko) maaraa karataa thaa] etc may be the possible translation of the form given in the (past) perfect tense.

Imperfect: avadheet= maaraa thaa or maarataa thaa or maara rahaa thaa...

Aorist: ahan= maaraa...

Eventually the difference of these three Sanskrit forms is lost in Hindi. So we are finding these past forms – maaraa, maarataa thaa, maara rahaa thaa- frequently in Modern Hindi usage. This particular tense needs a thorough comparative research between both the languages so that the framework can be developed for Machine Translation.

2.3 Future Tense

Future takes two lakaaras lut & Irit respectively. Apte¹⁰ says that the lut denotes the remote future time not of today and the Irit denotes indefinite future time, today's future time, and recent and future continuous time. The periphrastic future (lut) is far less frequently used; and where it is used, it generally denotes a remote future action while the simple future is used to denote any indefinite future action. As –ramaH pathishyati= raama paDhegaa; raamaH paThitaa= raama paDhegaa. But both the tense has basic difference that the first verb-pathishyati denotes the indefinite future action while the next- paThitaa shows the remote future. If time is not mentioned in Sanskrit, the sense of the sentence will not be clearly shown in Machine Translation. When the close proximity of a future action is intended, the present or the future may be used. As- kada gamishyasi- esha gachchaami/gamiShyaami. When hope is expressed in a conditional form, the Aorist, the present, or simple future may be used in both the clause to denote a future time. As- devashchedavarShId varShati varshishyati vaa dhaanyamavaapsma vapaamo vapsyaamo vaa¹¹. Therefore we can come across these challenges in Sanskrit-Hindi Machine Translation mainly in the verb mapping.

3. Analysis and rules

On the basis of above discussed facts, we find that Sanskrit is inflectional while Hindi is periphrastic. Sanskrit verbs (tingant) are not agreeing with gender but Hindi verbs agree with the gender. Sanskrit verbs have dual number while the Hindi has dropped the dual, thus the Sanskrit dual becomes plural in Hindi. The regular and the progressive verbs are clearly expressed in Hindi but Sanskrit verbs have no such differentiation. Sanskrit has three different forms for past and two different forms for future, but Hindi doesn't show such morphological differences.

⁸ On the architecture of Panini's Grammar, p-38.

⁹ A students' guide to Sanskrit Composition, p-142.

¹⁰ The Students guide to Sanskrit Composition, p-148.

¹¹ The Students guide to Sanskrit Composition, p-147.

4. Conclusion

In this paper we tried to reflect the multiple possibilities of verbal translation in SaHiT which needs to be handled carefully to produce acceptable output into Hindi. This will be more challenging when the passive forms and the Kridant forms will be present in the input text. The compound and complex verbs also needs to be discussed in details to formulate the possible rules for verb mapping. Present work is ongoing research which will be providing the rules for the rule based SaHiT system. This can be used to understand the issues between all IL- Hindi MT system.

5. References

- Apte, V. S. (1885). *The Student's Guide to Sanskrit Composition*. Lokasamgraha Press, Poona, India,
- Bharati , A. (et al) (1995) *Language Processing. A Paninian Perspective*. Prentice-Hall of India, New Delhi,
- Kiparsky, P. (2002). *On the architecture of Panini's grammar*. In International Conference on the Architecture of Grammar, Hyderabad.
- Kiparsky. Paul, (2008) *On the Architecture of Panini's Grammar*, Sanskrit Computational Linguistics, volume 5402 of Lecture Notes in Computer Science, page 33-94. Springer
- Jha, Girish Nath (Ed.) (2010) *Sanskrit Computational Linguistics*, volume 6465 of Lecture Notes in Computer Science, Springer
- Dixit, Pushpa (2010) *Astadhyayisutrapath*, Sanskrit Bharti, New Delhi.
- http://www.researchgate.net/publication/233341515_Sanskrit_Verb_Argument_Valence_A_Computational_Analysis
- <http://www.cse.iitb.ac.in/~pb/papers/gwc12-swn-verb.pdf>.
- <http://www.emille.lancs.ac.uk/lesal/ostler.pdf>
- <http://www.vedicsciences.net/articles/sanskrit-nasa.html>

To handle the progressive forms in Hindi, we can add the scope of optional outputs. The gender agreement can be handled with the help of Anaphora resolution system so that machine can identify the gender of agent to give correct output for SaHiT system.

3.1 Rules for verb mapping

3.1.1 Present

- 1(a). root+ti=> root+ taa/ti +hai.
- 1(b) root+ti= root+ rahaa/rahi+hai.
- 2(a) root+ tah./anti=>root+te/ ti +hain,
- 2(b) root + rahe/rahi +hain.
- 3(a) root+si/thah./tha=>root+te/ti+ho
- 3(b) root+si/thah./tha=>root+rahe/rahi+ho
- 4(a) root+mi => root+ta/ti+hoon,
- 4(b) root+mi root+raha/rahi hoon
- 4(c) root+ vah./mah. => root+te/rahe+hain

Rules for past forms can be divided into remote and recent past so that the tentative rules can be written.

3.1.2 Remote past

- 1(a) root+a=> root+aa tha, root+i thi, root+e the [i.e. path+a (lit) - papaatha=> padh+aa tha, padh+i thi, path+e the]
- 1(b) root+atuh./uh. => root+e_the /i_thiAs remote past is generally used for third person, these rules may be sufficient to handle remote past.

3.1.3 Simple past

Third person simple past will take the rules of remote past as it is. Further the rule 1(b) will be repeated to handle the verb forms of simple past for second person. For 1st person (singular), the rule may be:

- 2(a) root+a => root+aa thaa, root+i thi
- 2(b) root+a => root+rahaa thaa, /rahi thi
- 2(c) root+iva/ima => root+e the/rahe the

3.1.4 Rules for Future

Hindi forms for both the lakaara are same so the rules can be written as:

Root+ future suffix=>root+gaa/gi, root+ge/gi, root+oge, root+ungaa, root+enge (depending on the agent of the verb).

Evaluation of Hindi-English MT Systems

Atul Kr. Ojha, Akanksha Bansal, Sumedh Hadke and Girish Nath Jha
Jawaharlal Nehru University, New Delhi
(shashwatup9k, akanksha.bansal15, sumedhkhadke & girishjha@gmail.com)

Abstract

Evaluation of any Machine Translation (MT) system is an important step towards improving its accuracy. In this paper, we are trying to evaluate Hindi-English module through two most widely used MT systems - Bing (Microsoft) and Google. These MT systems are Statistics-Based MT systems (SBMT) and are capable of providing translation in many languages across the globe other than Hindi-English. For the purpose of evaluation, we tested Health and General cooking data and evaluated the English output text. Human evaluation strategy has been used for the purpose of evaluation, on the basis of which problem areas in both the MT systems were identified and compared to reach a conclusive analysis in terms of the output's fluency and comprehensibility. The comparative analysis helps in understanding not only which system is better but also what works best for automatic translation and under what circumstances. The discrepancies found are discussed with some suggestions towards their solution.

Keywords: *Human Evaluation, Automatic Evaluation, Machine Translation, Natural Language Processing.*

1. Introduction

Low accuracy, fluency and acceptability of output of any machine translation system adversely affect the reliability and usage of that system. Evaluation task can ascertain how and in what ways are the results of these systems lacking. It is an unavoidable part in the process of development of a Machine Translation System (MT system), because without evaluating the final generation text on the parameter of accuracy, fluency and acceptability we cannot make any claim for the success of MT system. However, the need and demand for evaluating an MT system is always the same and at higher priority. Therefore in this paper, we undertake the task of evaluating the output of Hindi-English language pair through two MT systems – Bing and Google.

There are no universally accepted and reliable methods and measures for evaluation (Arnold et al., 1993., Falkedal, 1994. and AMTA, 1992) but there is a common assumption and agreement on the basic features of Machine Translation (MT) evaluation (Hutchins & Somers, 1992. and Arnold et al., 1994).

So, we have chosen this task to evaluate MT systems to point out the anomalies and possible linguistic basis for them. The feedback (i.e. provided by the evaluation tasks) may provide opportunity for improving the accuracy of the systems. For this purpose, we have examined the Bing (Microsoft) and Google MT (Translator) Hindi-English module and come up with some major problems in translation and target language generation.

1.1 Google Translator MT System:

Google MT/Translator is based on statistical and machine learning approaches based on parallel corpora. Google MT system is running for 73 languages pairs.¹

1.2 Bing (Microsoft Translator) :

Bing (Microsoft) MT is based on statistical and machine learning approaches based on parallel corpora. It also uses language specific rule-based components to decode and encode sentences from one language to another. One can say that it is a "Linguistically informed statistical machine translation". Bing MT is running for 44 parallel languages pairs.²

Both MT systems are easily accessible to users. Earlier (Jha, 2012) discussed some of the problems in both Google and Bing outputs and provided explanations the present study extends earlier study in a more comprehensive comparative study and analysis.

2. Error Analysis & Linguistic Descriptions

Simple sentences from the health and cuisine domains of the ILCI³ corpora were used for evaluating the MT systems. These sentences were entered in each of the systems in bulk and the output was crawled, and discrepancies were marked. In the resulting English output, several problems were noted particularly with respect to gender agreement,

¹ <http://translate.google.co.in/>

² <http://www.bing.com/translator>.

³ <http://sanskrit.jnu.ac.in/projects/ilci.jsp?proj=ilci>

structural mapping, Named Entity Recognition (NER) and plural marker morphemes.

While most theories of translation are skeptical about the possibility of exact translation s translation work is carried out nonetheless. The process of translation involves decoding of a text in one language followed by encoding it into another. The theories of translation talk about divergence between languages at several levels. When we switch to machine translation, various unexpected problems surface up. Several problems that are not generally noticed by human translators are encountered by the machine at decoding level, thus challenging the task of Machine Translation.

During the evaluation process the following kinds of errors were encountered.

(A)Tokenization:

(i) With Punctuation:

- (a) वह जाती है। (IS)⁴
 vaha jAtI hai. (Trans.)
 She go-IPFV is-3.PRS.M.SG (Glossing)
 She goes by. (BO)
 He is. (GO)
- (b) वह जाती है (IS w/o)
 vaha jAtI hai. (Trans.)
 She go-IPFV is-3.PRS.M.SG (Gloss)
 He is (BO)
 He is (GO)

Manual Translation: She goes.

Examples (a) and (b) above exhibit how the use of a punctuation mark can significantly affect translation. This variation in results is seen only in Bing. Google exhibits consistency. In Bing, input sentence with punctuation gave results of near-approximation but totally wrong translation for input sentence without punctuation mark. None of the results exhibited by Google in this case were accurate.

(ii) Error with input sentence or word:

- (a) थोड़ा सा पानी डालें और --
 Tho.DA sA pAnI DAleM aura --
 ADJ some water put-IMP and CONJ --
- आटा गूंद लें (IS)
 ATA gUMda leM (Trans.)
 flour grind take-IMP3.M/F.SG (Gloss)

⁴ IS= Input sentence, Trans. =transliteration in roman script, IS w/o= Input sentence without punctuation, BO = Bing output, GO= Google output.

Take a little bit of water and flour gund (BO)
 Please add a little water and flour Gund (GO)

After tokenization of a word, translation differs:

- (b) थोड़ा सा पानी डालें और ---
 Tho.DA sA pAnI DAleM aura ---
 ADJ some water put-IMP and CONJ ---

आटा गूथ लें (IS with token)
 ATA gUMtha leM (Trans.)
 flour grind take-IMP3.M/F.SG (Gloss)
 Take a little bit of water and flour interlock (BO)
 Please add a little water and flour interlock (GO)

Manual Translation: Put some water and knead the dough.

In both the scenarios, with and without a full stop, structural mapping goes wrong with the second clause.

(iii) Transliteration issue:

- (a) कढ़छी चलाते रहें और --
 ka.DhachI calAte raheM aura --
 kadchi ran-IPFV continue-IMP and CONJ --

एक उबाल आने पर आँच धीमी --
 eka ubAla Ane para A.nca dhImI --
 one simmer come-INF on-PP heat slow --

करें दो मिनट पकाएँ (IS)
 KareM do miniTa pakAe.n (Trans.)
 Do-IMP3PRS two minute fry-IMP3 (Gloss)
 The heat to a simmer and continue running
kadhchi slow, two minutes to Cook (BO)
Kdhci Stir and heat to a boil and cook about two minutes to slow (GO)

Manual Translation: Keep moving the spatula, lower the flame as it starts boiling. Cook for two minutes.

- (b) एक नौन-स्टिक तवा गरम करें (IS)
 eka nauna-sTika tavA garama kareiM (Trans.)
 One non-stick pan heat do-IMP3PRS.M/F.SG
 A naun-stick frying pan and heat (BO)
 A Non - stick frying pan and heat (GO)

Manual Translation: Heat the non-stick fry pan

(B) Morph Issue:

(i) Unknown words:

छुआरे	डालकर	मिलाएँ	और --
chuAre	DAlakara	milAe.n	aura --
date palm	put	mix-IMP.3PRSM.SG	and CONJ -
-			
एक	मिनिट	पकाएँ	(IS)
eka	miniTa	pakAe.n	(Trans.)
one	minute	cook-IMP3PRS.M.SG	(Glossing)
One	minute	into the match and put chuare	(BO)
Mix	and cook	one minute, add Cuare	(GO)

Manual Translation: Put date palm, stir and cook for a minute.

(ii) Error with Paradigm fixation:

कैंसर	1000	से अधिक	बीमारियों --
kaiMsara	1000	se adhika	bImAriyoM --
Cancer	thousand	than more	diseases --
का	एक	समूह	है (IS)
kA	eka	samUha	hai (Trans.)
of PSP	one	group	is-PRS.M.SG (Gloss)
Cancer	is a group	of more than 1000	berryman (BO)
Cancer	is a group	of more than 1000	illnesses (GO)

कैंसर	1000	से अधिक	बीमारी --
kaiMsara	1000	se adhika	bImArI --
Cancer	thousand	than more	disease--
का	एक	समूह	है (IS)
kA	eka	samUha	hai (Trans.)
of PSP	one	group	is-PRS.M.SG (Gloss)
Cancer	is a group	of more than 1,000	diseases (BO)
Cancer	is a group	of more than 1000	illnesses (GO)

Manual Translation: Cancer is a group of more than 1000 diseases.

(C) Structural/grammatical Differences:

एच.आई.वी. क्या है ?	(IS)
H.I.V kyA hai?	(Trans.)
H.I.V what is-PRS.M.SG	(Gloss)
What is the HIV?	(BO)
HIV what is it?	(GO)

Manual Translation: What is the HIV?

(D) Errors with Gender agreement:

वह जाती है।	(IS)
vaha jAtI hai.	(Trans.)
She go-IPFV is-3.PRS.M.SG	(Gloss)
She goes by .	(BO)

He is. (GO)
Manual Translation: She goes.

(E) Parser Issues:

आँख की मांसपेशियों की कमजोरी के कारण लेन्स अपना आकार नहीं बदल पाता पढ़ते या नजदीकी काम करते समय प्रकाश की किरणें रेटिना के पीछे पड़ती हैं यह 40 वर्ष और उससे ऊपर की उम्र में पाई जाती है (IS)

A.nkha kI mAmsapeshiyO kI kamajorI ke kAraNa lensa apanA AkAra nahI badala pAtA pa.Dhate yA najadkI kAma karate samaya prakAsha kI kiraNeM reTinA ke pIche pa.DatI haiM yaha 40 varSha aura usase Upara kI umra meM pAI jAtI hai. (Trans.)

Due to the weakness of the muscles of the eye lens cannot read or change their size does proximity to work while the light rays have it 40 years behind the retina and above in age (BO)

NO OUT PUT (GO)

3. Evaluation Strategies

Evaluation strategies are mainly divided into two sections: (a) Automatic evaluation and (b) Manual or Human evaluation. Automatic evaluation of any MT system is very difficult and is not as effective as human metrics are. There are several tested MT evaluation measures frequently used, for example: BLEU, mWER, mPER and NIST.

Apart from these automatic evaluation strategies, Human evaluation metrics also exist which are considered to be time taking and costly. But they are the best strategies to improve any MT system's accuracy, especially for Indian languages.

It is a common scenario where more than one translation of a sentence exists. At this level a human translator cum evaluator can judge the output correctly. Generally he/she use a quality scale of 1 to 5 i.e. 1 for Incomprehensible, 2 for Difficult, 3 for Non-native, 4 for Good, and 5 for Flawless. But here, the evaluating methodology & quality scale of 0-4 is used. This range has been found to be successful in effective evaluation strategy before (Pathak, Ahmad & Ojha, 2012).

On quality scale of Human evaluation metrics, adequacy and fluency are at higher priority during judgment. After fluency judgment, adequacy is assessed, and the evaluator has to be ready with a reference translation to assess how much information from the original translation is lost in the translation by marking on the quality scale.

Human evaluation strategy has been adopted to evaluate the Bing (Microsoft) and Google MT (Hindi-English) output.

Methodology of MT testing:

For testing to the MT systems, 1,000 sentences were used to test MT systems. Their outputs were then distributed into three different human evaluators who marked MT outputs based on comprehensibility and fluency approaches.

Instructions for Evaluators to Evaluate:

- (a) Read the target language translated output first.
- (b) Judge each sentence for its comprehensibility.
- (c) Rate it on the scale 0 to 4.
- (d) Read the original source sentence only to verify the faithfulness of the translation (only for reference).
- (e) Do not read the source language sentence first.
- (f) If the rating needs revision, change it to the new rating.

Guidelines of evaluation: on 5 point scale (over 0-4):

The following score is to be given to a sentence by looking at each output sentence:

(a) For Comprehensibility-

- 4= All meaning
- 3= most meaning
- 2 = much meaning
- 1= little meaning
- 0= none.

(b) For fluency -

- 4= for Flawless or Perfect: (like someone who knows the language)
- 3= for Good or Comprehensible but has quite a few errors: (like someone speaking Hindi getting all its genders wrong)
- 2 = for Non-native or Comprehensible but has quite a few errors: (like someone who can speak your language but would make lots of error. However, you can make sense out of what is being said.)
- 1= for Diffluent or Some parts make sense but is not comprehensible over all: (like listening to a language which has lot of borrowed words from your language- you understood those words but nothing more)
- 0=for Incomprehensible or Non-Sense: (If the sentence does not make any sense at all - It is like someone speaking to you in a language you do not know)

Evaluation Method:

If scoring is done for N sentences and each of the N sentences is given a score as above, the two parameters are as follows:

(a) Comprehensibility = (Number of sentences with the score of 2, 3, or 4) / N

(b) Fluency = $\frac{\sum_{i=1}^N S_i}{N}$

Where Si is the score of ith sentence, for instance, If N=10, and suppose the scores obtained for the each of the 10 sentences are: S1=3, S2=3, S3=2 S4=1, S5=4, S6=0, S7=0, S8=1, S9=0, S10=0 This gives the following histogram:

- Number of sentences with score 4 = 1
- Number of sentences with score 3 = 2
- Number of sentences with score 2 = 1
- Number of sentences with score 1 = 2
- Number of sentences with score 0 = 3
- Weighted sum =14, then this produces:

Comprehensibility = 40 %
(Because 4 out of 10 sentences gain with a score of 2, 3, or 4.)

Fluency = 14/10= 1.4 (on a scale of 0-4)
36% (on the max possible scale of 100)

Note: If we use the 1-5 point scale, the comprehensibility figure does not get affected. Fluency score would go down however.

4. Results

On the basis of methodology above Table 1 below shows the score of comprehensibility computed, Table 2 shows weighted score of fluency computed, Table 3 compared comprehensibility and fluency percentage of different human evaluator’s scores, and finally table 4 is averages of all human evaluator’s comprehensibility and fluency percentage Bing MT and Google MT systems.

Scale: 0-4	Bing MT			Google MT		
	User1	User2	User3	User1	User2	User3
0	338	350	10	110	115	10
1	152	175	135	167	155	55
2	244	220	510	310	295	465
3	147	105	300	245	225	365
4	119	150	45	168	210	105

Table 1: Score Table to Compute Comprehensibility

Scale: 0-4	Bing MT			Google MT		
	User1	User2	User3	User1	User2	User3
0	190	420	30	130	235	10
1	340	250	375	220	270	230
2	240	160	305	290	240	340
3	160	45	240	220	140	335
4	70	125	50	140	115	85

Table 2: Score Table to Compute Fluency

MT Systems	Comprehensibility (%)			Fluency (%)		
	User1	User2	User3	User1	User2	User3
Bing	51	47.5	85.5	39.5	30.1	47.6
Google	72.3	73	93.5	50.5	40.7	56.3

Table 3: Comprehensibility and Fluency percentage of evaluator's scores

MT Systems	Comprehensibility (%)	Fluency (%)
Bing	61.33	39.08
Google	79.6	49.20

Table 4: Average of total evaluators weighted score to compute Comprehensibility and Fluency

Error Rate:

The graph shown below highlights all the error categories comparing their error accuracy rate in both the MT systems, the category others is also highlighted in the graph shown below. Of the total 1000 sentences used as data the rate of short/simple sentences is 200 whereas that of long/complex sentences is 800.

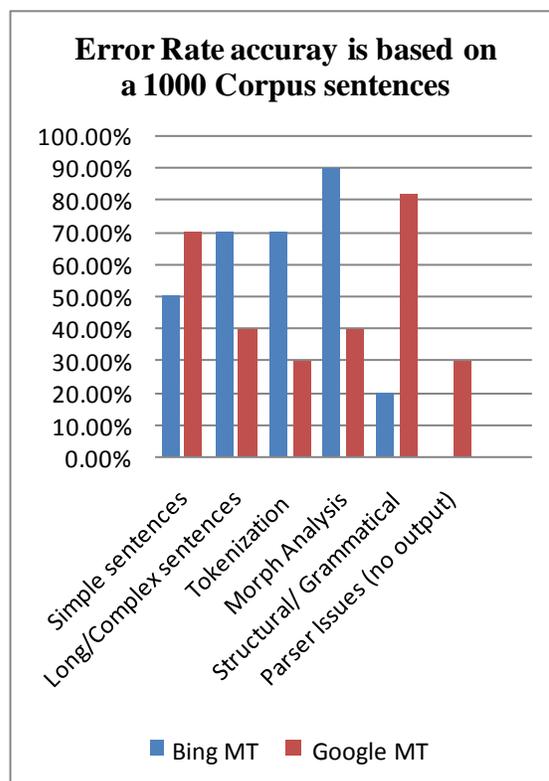


Table 5: Error Rate of Bing & Google MT Systems

5. Conclusion

In this paper, two major points have been presented: The first is the errors in MT output and second is the evaluation of Bing MT (Microsoft translator) & Google MT systems. When we examined and evaluated these systems, we found many errors. We have already mentioned above the discrepancies in Tokenization, Morph Error, Structural Errors and Parser issues etc. And when, we evaluated MT systems, the fluency was found to be very low but nonetheless it was almost comprehensible. On comparison, Google was found to be better than Bing MT in comprehensibility. When we compute the ratio of results achieved in Table 4 then Bing is found to be better than Google. If the problems mentioned for Bing MT system are resolved then it can outperform Google in future.

Finally, some suggestions are proposed on improving both MT systems on the basis of evaluation done above. As shown above while giving the input sentences tokenize them and avoid the use full stop marker in final place. Both MT systems should improve their morph dictionary through corpus data and make linguistics rules for paradigm fixation (how

to analyze inflectional and derivational category), and if MT systems are trained with large number of words and sentences then parsing issues might be resolved. Then, these systems will improve and the errors will decrease up to some extent. Following these steps, we can increase the Bing and Google MT systems in fluency as well as in comprehensibility.

6. References

- Ananthakrishnan, R., Bhattacharyya, Pushpak., Sasikumar, M. and Shah, Ritesh M. (2007). Some Issues in Automatic Evaluation of English-Hindi MT: More Blues for BLEU. *In the Proceedings of 5th International Conference on Natural Language Processing (ICON2007)*, 4-6 January, Hyderabad, India.
- Arnold, D., Balkan, L., Humphreys, R. Lee, Meijer, S. and Sadler, L. (1994Eds). *Machine Translation: An Introductory Guide*. Manchester/Oxford: NCC Blackwell.
- Baskaran, Sankaran., Kumaran, A., Bali, Kalika. (2008). A Dependency Treelet-based Phrasal SMT: Evaluation and Issues in English-Hindi Language pair. *In the Proceedings of 6th International Conference on Natural Language Processing (ICON2008)*, December 20- 22, 2008, CDAC Pune, India
- Bhattacharyya, Pushpak. (2013). SMT and Parallel Corpora. *Workshop on Indian Language Corpora*. Imphal, Manipur, August22, 2013.
- Bing Translator: <http://www.bing.com/translator>.
- Google Translator: <http://translate.google.com>.
- Hutchins, W. J. and H. L. Somers. (1992). *An Introduction to Machine Translation*. London: Academic Press.
- ILCI:
<http://sanskrit.jnu.ac.in/projects/ilci.jsp?proj=ilci>
- Jha, Girish N. (2012). Emerging trends in Language Technology Resources (LTR) in India. *Delivered at Microsoft Research, Redmond, USA, Apr 9, 2012*
- Jurafsky, Daniel. and Martin, James H. (2000). *Speech and Language Processing*. New Jersey: Prentice Hall.
- Kachru, Yamuna. (1996). *An Introduction to Hindi Syntax*. Urbana: Dept. of Linguistics, University of Illinois.
- Koehn, Philipp. (2010). *Statistical Machine Translation*. New York: Cambridge University Press.
- Mauser, Arne. Hasan, Sasa. and Ney, Hermann.(2008). Automatic Evaluation Measures for Statistical Machine Translation System Optimization. *Marrakech, Morocco, In the Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008)*, European Language Resources Association (ELRA).
- Mitakov, Ruslan. (2003Eds). *The Oxford Handbook of Computational Linguistics*. OUP.
- Papineni Kishore, Roukos Salim, Ward Todd and Zhu Wei-Jing (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. *In the Proceedings of the 40th Annual Meeting of the Report, prepared for the Commission of the European committees, Brussels*. EAGLES. Expert Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp.311-318.
- Pathak, Sanket. Ahmad, Rashid. and Ojha, Atul. (2012): A Language Engineering Approach to Enhance the Accuracy of Machine Translation Systems. Vol-2, Issue-1/201, *Shodh Prerak-Lucknow, ISSN: 2231 – 413X*, pp.205-214.
- Ramis, Adria de Gispert. (2006). *Introducing Linguistic Knowledge into Statistical Machine Translation*. Barcelona: Universitat Polit`ecnica de Catalunya PHD thesis.
- Singh, Pr. Surajbhan. (2010). *English-Hindi Translation Grammar*. Delhi: Prabhat Publication.
- Van Slype, G. (1996). Critical Methods for Evaluating the Quality of Machine Translation (*Final Report*), prepared for the Commission of the European committees, Brussels. EAGLES. Expert Advisory Group on Language Engineering, “Evaluation of Natural Language Processing Systems (Final Report)”, prepared for DG XIII of the European Commission.

Appendix

S. No.	Bing Translator						Google Translator					
	Comprehensibility			Fluency			Comprehensibility			Fluency		
	User 1	User 2	User 3	User 1	User2	User3	User1	User2	User3	User1	User 2	User3
1	1	0	1	0	0	1	2	2	3	3	3	3
2	1	0	2	1	2	3	2	3	2	1	3	3
3	2	3	2	1	0	1	1	3	2	4	3	2
4	4	4	1	1	0	1	1	2	1	2	1	1
5	1	2	2	1	2	1	3	3	2	2	3	1
6	0	1	4	4	4	4	3	4	4	3	4	4
7	0	1	2	3	3	2	3	4	4	2	3	3
8	0	0	3	2	3	2	1	2	3	2	1	1
9	1	0	2	0	1	2	3	4	4	4	4	3
10	2	3	2	1	0	2	2	4	3	1	0	1
11	1	0	1	0	1	3	1	1	1	0	0	1
12	0	1	1	1	2	1	3	1	3	1	1	2
13	0	0	1	0	0	1	3	2	2	2	2	1
14	1	0	1	1	1	1	1	3	2	2	1	2
15	1	1	2	0	0	2	2	4	4	4	3	1
16	3	4	3	0	0	3	1	1	2	0	1	1
17	2	2	2	1	1	1	1	4	3	4	3	3
18	1	2	2	1	4	2	1	3	2	3	3	4
19	1	2	3	1	2	3	0	2	2	1	3	2
20	1	1	2	2	2	1	1	1	2	0	0	1
21	3	4	0	1	3	2	1	1	2	2	3	3
22	1	0	1	0	1	1	1	1	2	0	1	3
23	1	0	3	4	4	4	2	3	4	4	2	1
24	0	0	2	1	1	1	2	2	2	3	3	3

25	1	1	2	3	1	3	4	3	2	3	1	1
26	0	0	2	1	1	1	2	2	4	2	1	1
27	0	0	3	2	1	2	4	2	3	4	4	4
28	1	0	2	0	0	2	2	4	3	0	1	1
29	1	0	2	0	0	1	1	4	4	4	3	3
30	1	0	1	0	0	1	1	1	3	0	1	1
31	2	2	2	0	0	2	2	2	1	3	1	1
32	1	0	2	3	2	3	2	0	2	2	2	1
33	0	1	1	2	0	1	2	1	2	1	1	2
34	4	4	1	0	1	1	3	4	2	2	1	1
35	3	4	2	1	3	3	4	4	3	3	2	1
36	4	4	2	2	3	2	2	3	3	3	2	3
37	2	2	3	2	3	3	1	2	2	0	3	3
38	1	2	3	2	2	3	1	4	4	2	3	2
39	3	4	3	4	4	3	2	4	3	4	3	3
40	2	1	2	1	2	2	2	1	2	2	2	2
41	2	0	4	4	4	4	1	1	2	4	3	1
42	3	4	1	0	1	1	2	1	2	1	3	4
43	2	1	2	1	1	1	2	4	4	2	3	2
44	1	1	3	3	4	3	3	4	3	3	3	2
45	1	0	2	2	1	1	4	2	3	3	2	2
46	1	0	2	2	0	1	3	4	2	2	2	2
47	1	0	2	2	1	2	3	2	2	3	2	1
48	1	0	2	2	0	1	0	1	2	3	2	1
49	2	2	2	2	1	2	1	1	2	2	3	1
50	4	4	2	2	0	1	1	1	2	3	2	1

Lexical Resources for Hindi Marathi MT

Sreelekha. S, Pushpak Bhattacharyya, Malathi.D

Department of Computer Science and Engineering

Indian Institute of Technology Bombay, SRM University

E-mail: {sreelekha, pb}@cse.iitb.ac.in, malathi.d@ktr.srmuniv.ac.in

Abstract

In this paper we describe ways of utilizing lexical resources to improve the quality of statistical machine translation. We have augmented the training corpus with various lexical resources such as IndoWordnet semantic relation set, function words, kridanta pairs and verb phrases. We augmented parallel corpora in two ways (a) additional vocabulary and (b) inflected word forms. We have described case studies, evaluations and have given detailed error analysis for both Marathi to Hindi and Hindi to Marathi machine translation systems. From the evaluations we observed an order of magnitude improvement in translation quality. Lexical resources do help uplift performance when parallel corpora is scanty.

Keywords: Statistical Machine Translation, IndoWordnet, Lexical Resources.

1. Introduction

Machine Translation (MT) is the process of translating text or speech from one natural language to another with the help of machines. There are many ongoing attempts to develop MT systems for various Indian languages using rule-based as well as statistical-based approaches. Since India is rich in linguistic divergence there are many morphologically rich languages which are quite different from English as well as from each other, there is a great need for machine translation between them (Nair et.al, 2012). It has 18 constitutional languages, which are written in 10 different scripts (Antony 2013). This paper discusses various approaches used in Indian language to Indian language MT systems especially in Marathi to Hindi Statistical MT system and vice versa to improve the quality of machine translation.

For example, consider the translated Hindi output from Marathi-Hindi SMT system for the Marathi sentence,

Marathi -
{hya karnaastav to naraz hota}

Hindi- |
{yah phalswaroop vah naraz ho}

Here the Marathi word “ ” {karnaastav} is wrongly mapped to Hindi word “ ” {phalswaroop} and the verb {hota} is wrongly mapped to हो गया {ho gaya}.

In order to learn various inflected and verb forms, lexical resources can play a major role. The detailed analysis about various linguistic phenomena and how the lexical resources can be used for improving the translation quality is explained in the following Sections.

In the case of Marathi, a single noun can have more than 200 forms that are either adjectives or adverbs. Similarly, a verb may exhibit over 450 forms. Also, the language is covering about 10,000 nouns and over 1,900 verbs. Moreover 175 postpositions can be attached to nominal and verbal entities. Some postpositions can occur in compound forms with most of the other postpositions. In addition, there are many kinds of derivable words such as causative verbs like ‘karavane’, i.e. ‘to make (someone) to do (something)’, which is derivable from root ‘karane’ i.e. ‘to do’, and abstract nouns like ‘gharpan’ i.e. ‘homeliness’, which is derivable from ‘ghar’ i.e. ‘home’ [Veena et.al., 2005].

Major difficulties in Machine Translation are handling the structural difference between the two languages and handling the ambiguities.

1.1 Challenge of Ambiguity

There are two types of ambiguities: structural ambiguity and lexical ambiguity.

1.1.1. Lexical Ambiguity

Words and phrases in one language often have multiple meaning in another language.

For example, in the sentence,

Marathi- आले होते
{aale hote mehnaan kaam jchale}
Hindi- अदरक आ |
{aadark tha isliye kaam hua}
English- Ginger was there so the work has done
or
Hindi- आ |
{maenne photo nikala}
English- came thatsy work was done

Here in the above sentence “आले” {aale}, has ambiguity in meaning. It is not clear that whether the word “आले” {aale}, is used in Hindi as the “ginger” (“अदरक” {aadark} sense or the “आए” {aayen}, sense. This kind of ambiguity will be clear from the context.

1.1.2. Structural Ambiguity

In this case, due to the structural order, there will be multiple meanings. For example,

Marathi - तिथे लड्ड मुली आणि मुले होती.

{tithe latth muli aani mulen hoti}

{There were fat girls and boys there}

Here from the words “लड्ड मुली आणि मुले” {lattha muli aani mulen} it is clear that, girls are fat but it is not clear that boys are fat, since in Marathi to represent tall girls and boys only one word “लड्ड” {lattha} {tall} is being used. It can have two interpretations in Hindi and English according to its structure.

Hindi - वहाँ मोटी लड़कियाँ और लडकें थे ।

{vahan moti ladkiyam our ladhkem the}

{There were fat girls and boys there}

or

Hindi - वहाँ मोटी लड़कियाँ और मोटे लडकें थे ।

{vahan moti ladkiyam our mote ladhkem the}

{There were fat girls and fat boys there}

To generate appropriate Machine Translations by handling this kind of structural ambiguity is one of the big problems in Machine Translation.

1.2 Structural Differences

In the case of Marathi – Hindi machine translation both languages follow the same structural ordering in sentences, such as Subject- Object-Verb (SOV). Even though there is ordering similarity, there are morphological and stylistic differences which have to be considered during translation. Marathi is morphologically more complex than Hindi, wherein there are a lot of post-modifiers in the former as compared to the later (Dabre et al, 2012, Bhosale, 2011).

For example, the word form “हस्तिनापुरच्या” {hustinapoorchya} {of / about the hustinapuri} is derived by attaching “च्या” {chyaa} as a suffix to the noun “हस्तिनापुर” {hustinapur} {a place in India} by undergoing an inflectional process. Marathi exhibits agglutination of suffixes which is not present in Hindi and therefore these suffixes has equivalents in the form of post positions. For the above example, the Hindi equivalent of the suffix “च्या” {chyaa} is the post position “के” {ke} which is separated from the noun “हस्तिनापुर” {hustinapur}. Hence the translation of “हस्तिनापुरच्या” {hustinapoorchya} will be “हस्तिनापुर के” {hustinapur ke}.

Consider an example for word ordering ,

Hindi- गढ़-मुक्तेश्वर हिंदुओं का पावन तीर्थ है ।

{gad-mukteshwar hinduom ka paavan teerth hai}

(S) (O) (V)

Marathi- गढमुक्तेश्वर हिंदुंचे पवित्र तीर्थ आहे .

{gadmukteshwar hindoonche pavitr teerth aahe}

(S) (O) (V)

English-Gadmuktheshwar is the holy place for Hindu.

(S) (V) (O)

Since the word order is same for both the languages it is an advantage for statistical machine translation system during alignment. And it will improve the quality of translation.

1.2.1 Participial Constructions

Constructions in Hindi having Participials in Marathi

Example: Hindi:

जो लडका गा रहा था वह चला गया

jo ladkaa gaa rahaa thaa wah chala gayaa

rel. boy sing stay+perf.+cont. be+past walk go+perf.

The boy who was singing, has left.

Marathi (Direct Translations):

जो मुलगा गात होता तो निघून गेला

jo mulgaa gaat hotaa to nighoon gelaa

rel. boy sing+imperf. be+past leave+CP go+perf.

The boy who was singing, has left.

Participial Constructions in Marathi(Actual Translations):

गाणारा मुलगा निघून गेला

gaaNaaraa mulgaa nighoon gelaa

sing+part. boy leave+CP go+perf.

The boy who was singing left

Note:-

Deletion/dropping of subordinate clause is common in Marathi as compared to Hindi. In the above sentence गाणारा {gaanara} { who was singing } is the contracted form of subordinate clause and the relative marker जो {jo} {who} is dropped.

1.3 Vocabulary Differences

Languages differ in the way they lexically divide the conceptual space and sometimes no direct equivalents can be found for a particular word or phrase of one language in another.

Consider the sentence,

काल मंगळागौरीची पूजा होती.

{kaal mangalgowreechi pooja hoti}

{Yesterday the pooja which happens in the month of sravan for married women got completed}

Here “मंगळागौरीची” {mangalagowrichi} {the pooja which happens in the month of sraven for married women} as a verb has no equivalent in Hindi, and this sentence has to be translated as,

“कल सुहागन की श्रावण मास में संपन्न होनेवाली पूजा थी

{“Kal sahan ki sravan maas mem sampannu honewali pooja thi.”}

{ Yesterday the pooja which happens in the month of sravan for married women got completed }

To determine translations of such language specific concepts pose additional challenges in machine translation.

2. Difficulties in SMT approach

As described in Section 1, there are many structural differences between languages. The statistical approach tries to generate translations based on the knowledge and statistical models extracted from parallel aligned bilingual text corpora. Statistical models take the assumption that every word in the target language is a translation of the source language words with some probability (Brown et al., 1993). The words which have the highest probability will give the best translation. Consistent patterns of divergence between the languages (Dorr et al., 1994, Dave et al., 2002, Ramanathan et al., 2011) when translating from one language to another, handling reordering divergence are one of the fundamental problems in MT (Kunchukuttan and Bhattacharyya 2012, Och and Ney, 2001, Koehn, 2007).

In the case of Marathi and Hindi, even though both the language follows same word order, there are structural difference between the language and in the generation of word forms due to the morphological complexity as described in Section 1. In order to overcome this difficulty and make the machine to learn different morphological word forms lexical resources can play a major role. Different word forms such as verb phrases, morphological forms etc can be used. Also the SMT system lacks in vocabulary due to the small amount of parallel corpus. Comparative performance studies conducted by Och and Ney have shown the significance of adding dictionary words into corpus and the improvement in the translation quality in their paper (Och, Ney 2003). In order to increase the coverage of vocabulary we have used IndoWordnet. IndoWordnet(Bhattacharyya, 2010) is a lexical database for various Indian languages, in which Hindi wordnet is the root and all other Indian language wordnets are linked through the expansion approach. Words and its concepts are stored in a structure called the Lexical Matrix, where rows represent *word meanings* and columns represents the *forms*. The extraction of bilingual mapped words and its usage in machine translation is described in the experimental Section 3. The comparative performance analysis with phrase based model and the phrase based model after augmenting various lexical resources is described in Section 4 & 5.

3. Experimental Discussion

We now describe the development of our Marathi- Hindi and Hindi- Marathi SMT system¹ (Sreelekha, Dabre and Bhattacharyya 2013), the experiments performed and the comparisons of results, in the form of an error analysis. For the purpose of constructing with statistical models we use Moses and Giza++². We have conducted various experiments to improve the quality of machine translation by utilizing various lexical resources.

¹ <http://www.cfilt.iitb.ac.in/SMTSystem>

² <http://www.statmt.org/>

Our experiments are listed below:

1. SMT system with an uncleaned corpus
2. SMT system with a cleaned corpus
3. Corpus with IndoWordnet extracted words
4. Corpus with Function words and Kridantha pairs
5. Corpus with verb phrases

3.1.1 SMT system with an uncleaned corpus

We have used corpus of Tourism and Health provided by ILCI consortium, DIT and EILMT. There were many misalignments, wrong and missing translations in the corpus. It affected the translation and the quality was not good.

For example, consider a sentence from the corpus where the translation is wrong,

Hindi :

मणिपुर के दूरस्थ उत्तर-पूर्वी राज्य में पोलो खेल का अस्तित्व कायम रहने के लिए, संसार आधुनिक पोलो के जन्म का ऋणी है, क्योंकि आज यह पूरी दुनिया में खेला जा रहा है ।

{manipur ke doorsth uththa-poorv raajy mem polo khel ka astivthv kaayam rahne keliye, samsar aadhunik polo ke janmu ka shruni hae, kyonki aaj yah poori duniya mem khela ja raha hae }

English : {To retain the existence of the game of Polo in the far north eastern state of Manipur, the world is indebted to the birth of modern polo as today it is being played all over the world. }

Equivalent Marathi Translation (wrong)

पोलो आज जगभर खेला जातो त्याचे श्रेय तो ईशान्य भारतातील मणिपूरच्या दुर्गम डोंगरांगांमध्ये जिवंत राहिला याकडे जाते.

{polo aaj jagbhar khelela jato tyache shrey to yieshany bharatateel manipoorchya durgm dongarangamadye jivant raahila yakade jaate }

English: Today Polo is being played all over the world and it's credit goes to retaining the existence of the game of Polo in the remote hills of Manipur of north eastern India.

The comparative performance results of cleaned corpus over uncleaned corpus were shown in the Table 4, 5, 6 and 7. From the error analysis, we came to a conclusion that in order to improve the translation quality, we need to provide a properly cleaned parallel corpus for training.

3.1.2 SMT system with cleaned corpus

We have corrected the misalignments between parallel sentences which improves the learning of word to word alignments. Sometimes we had to correct even the source sentences.

For example, for the above sentence,

Hindi :

मणिपुर के दूरस्थ उत्तर-पूर्वी राज्य में पोलो खेल का अस्तित्व कायम रहने के लिए, संसार आधुनिक पोलो के जन्म का ऋणी है, क्योंकि आज यह पूरी दुनिया में खेला जा रहा है ।

{manipur ke doorsth uththa-poorv raajy maem polo khel ka astivthv kaayam rahne keliye, samsar aadhunik polo ke janmu ka shruni hae, kyonki aaj yah poori duniya mem khela ja raha hae }

Correct Marathi Translation

मणिपूरच्या दूरस्थ ईशान्य राज्यांमध्ये पोलो खेळाचे अस्तित्व टिकून ठेवण्यासाठी, संसार आधुनिक पोलोच्या जन्माचे ऋणी आहे, कारण की आज हा संपूर्ण जगभर खेळला जात आहे.

{manipoorchyaa durasth eeshanya rajaymadye polo khelache asthivithv tikoona thevnyasati, samsar aadhunik polochya jenmache shruni aahe, kaaran ki aaj ha sampoom jagphar khetla jaat aahe.}

English : *{To retain the existence of the game of Polo in the far north eastern state of Manipur, the world is indebted to the birth of modern polo as today it is being played all over the world.}*

We have removed the stylistic constructions from the parallel corpus which prevent the learning of grammatical structures. Also we have removed the unwanted characters and wrong translations and corrected the missing translations and phrases. The resultant machine translation system's quality improved more than 40 %. The statistics of corpus used are shown in Table 1 and the results are shown in the Tables 4, 5, 6 and 7. During error analysis we came to know that, the machine lacks in sufficient amount of vocabulary and hence we investigated on the usage of IndoWordnet to improve the quality of machine translation.

3.1.3 Corpus with IndoWordnet extracted words

We have extracted a total of 437832 parallel Marathi-

Hindi words using bilingual mapping according to its semantic and lexical relations as described in Section 2. We have used an algorithm to extract the bilingual words from IndoWordnet. Bilingual mappings are generated using the concept-based approach across words and synsets (Mohanty et.al., 2008). For a single word considered it's all synset word mappings and generated that many entries of parallel words.

For example, the word अंतहीन{*antaheen*} has the following equivalent synset words in IndoWordnet.

अंतहीन: अनंत असमाप्य अंतहीन अनन्त अन्तहीन
अनवसान

{*antaheen: anantu asamapya antaheen anant antaheen anavasaan*}

{*endless: endless not-ending endless infinite endless not-ending*}

We have augmented the extracted parallel words into the training corpus. It helped in improving the translation quality to a great extent. The statistics of Wordnet synsets used are shown in Table 2 and the results are shown in the Tables 4, 5, 6 and 7. During the error analysis we observed that even though the machine translation system is able to give considerably good quality translation, it lacks in handling case markers and inflected forms. One of the advantages is that it helped the machine to handle word

Sl.No	Corpus Source	Training Corpus [Manually cleaned and aligned]	Corpus Size [Sentences]
1	ILCI	Tourism	25000
2	ILCI	Health	25000
3	DIT	Tourism	20000
4	DIT	Health	20000
Total			90000

Table 1: Statistics of Training Corpus

Sl.No	Lexical Resource Source	Lexical Resources in Corpus	Lexical Resource Size [Words]
1	CFILT, IIT Bombay	IndoWordnet Synset words	437832
2	CFILT IIT Bombay	Function Word, Kridanata Pairs	5000
3	CFILT IIT Bombay	Verb Phrases	4471
Total			447303

Table 2: Statistics of Lexical Resources Used

Sl. No	Corpus Source	Tuning corpus [Manually cleaned and aligned]	Corpus Size [Sentences]
1	EILMT	Tourism	100
Total			100

Table 3: Statistics of Tuning Corpus

Sl. No	Corpus Source	Testing corpus [Manually cleaned and aligned]	Corpus Size [Sentences]
1	EILMT	Tourism	100
Total			100

Table 4: Statistics of Testing Corpus

sense disambiguation well, since the synsets covers all common forms of a word.

3.1.4 Corpus with Function words and Kridantha pairs

Marathi and Hindi have 7 types of kridanta forms, its' post position, pre-position and inflected forms. We have augmented kridanta, akhyat, function words, suffix pairs etc into the training corpus. This helped in machine translation system to infer the grammatical structures and hence the quality of translation improved.

Sample Marathi-Hindi kridanta form pair is,

आयला_पाहिजे : _ना_चाहिए
 {aayla pahije : na chahiye}
 {need : need}

The statistics of function words used are shown in Table 2 and the results are shown in the Tables 4, 5, 6 and 7. From the error analysis we came to a conclusion that translation system is facing difficulties in handling verbal translations because of the morphological phenomena.

3.1.5 Corpus with verb phrases

In order to overcome the verbal translation difficulty we have extracted Marathi- Hindi parallel verbal forms and its translations which contains various phenomena. We have augmented the 4471 entries of verbal translations into the training corpus.

Sample Marathi-Hindi verb form pair is,

जरूर करवा लें : अवश्य करवून घ्या
 {zaroor karva lem : avasyu karvoon ghya}
 {should get it done : should get it done}

The statistics of function words used are shown in Table 2 and the results are shown in the Tables 4, 5, 6 and 7. The error analysis study has shown that the quality of the translation has been improved drastically.

4. Evaluation

We have tested the translation system with a corpus of 100 sentences taken from the 'EILMT tourism health' corpus as shown in Table 3. We have used various evaluation methods such as subjective evaluation, BLEU score (Papineni et al., 2002), METEOR and TER (Agarwal and Lavie 2008). We gave importance to subjective evaluation to determine the fluency (F) and adequacy (A) of the translation, since for morphologically rich languages subjective evaluations can give more accurate results comparing to other measures. Fluency is an indicator of correct grammatical constructions present in the translated sentence whereas adequacy is an indicator of the amount of meaning being carried over from the source to the target. Marathi Hindi Bilingual experts have assigned scores between 1 and 5 for each generated translation, on the basis of how much meaning conveyed by the generated translation and its grammatical correctness. The basis of scoring is given below:

- 5: The translations are perfect.
- 4: One or two incorrect translations and mistakes.

- 3: Translations are of average quality, barely making sense.
- 2: The sentence is barely translated.
- 1: The sentence is not translated or the translation is gibberish.

$S1, S2, S3, S4$ and $S5$ are the counts of the number of sentences with scores from 1 to 5 and N is the total number of sentences evaluated. The formula (Bhosale et al., 2011) used for computing the scores is:

$$A/F = 100 * \frac{(S5 + 0.8 * S4 + 0.6 * S3)}{N}$$

We consider only the sentences with scores above 3. We penalize the sentences with scores 4 and 3 by multiplying their count by 0.8 and 0.6 respectively so that the estimate of scores is much better. The estimate may vary from person to person as these scores are subjective, in which case an inter annotator agreement is required. We do not give these scores, since we had only one evaluator. The results of our evaluations are given in Table 4, 5, 6 and 7.

5. Error Analysis

We have evaluated the translated outputs of both Marathi to Hindi and Hindi to Marathi Statistical Machine Translation systems in all 5 categories as explained in Section 3. The detailed error analysis is shown in Table 8 and 9 for a sentence exhibiting a variety of linguistic phenomena and how the quality of Machine Translation system changes by augmenting various lexical resources. The results of BLEU score, METEOR and TER evaluations are displayed in Tables 6 and 7 and the results of subjective evaluations are displayed in Table 4 and 5. We have observed that, the quality of the translation is improving as the corpus is getting cleaned as well as more lexical resources are used. Hence, there is an incremental growth in adequacy, fluency, BLEU score, METEOR score and in TER score. The fluency of the translation is increased up to 83% in the case of Marathi to Hindi and up to 85% in the case of Hindi to Marathi.

Also we have observed that, the score of Hindi to Marathi translation quality is slightly higher than that of Marathi to Hindi translation. Since Marathi is morphologically richer than Hindi and Marathi have more agglutinative suffixes attached, while in Hindi it is not present, as explained in above Section 1.2. Therefore these Marathi suffixes have Hindi equivalents in the form of post positions. So during alignment from Hindi to Marathi, Hindi word can align to the words with agglutination in Marathi, since it is a single word. But on the other hand while aligning form Marathi-Hindi the agglutinative word can map to only root words, there is a chance to miss out the post position mapping, as it is separate words. So it will improve the translation quality of Hindi- Marathi SMT as compared to Marathi-Hindi SMT.

Marathi-Hindi Statistical MT System		Adequacy	Fluency
With Uncleaned Corpus	With Tuning	20.6%	30.8%
	Without Tuning	17.8%	24%
With Cleaned Corpus	With Tuning	58.6%	68.3%
	Without Tuning	54%	64%
With Wordnet	With Tuning	72.4%	80%
	Without Tuning	69.6%	78.2%
With Function Words, kridanta pairs	With Tuning	78%	81%
	Without Tuning	75%	80%
With verb Phrases	With Tuning	83%	88%
	Without Tuning	80%	85%

Table 4: Results of Marathi-Hindi SMT Subjective Evaluation

Hindi-Marathi Statistical MT System		Adequacy	Fluency
With Uncleaned Corpus	With Tuning	22.87%	31.3%
	Without Tuning	20.56%	28%
With Cleaned Corpus	With Tuning	59%	72.21%
	Without Tuning	53%	65%
Corpus with Wordnet	With Tuning	73%	83%
	Without Tuning	70.01%	80.04%
Corpus with Function words, kridanta pairs	With Tuning	79.36%	87.21%
	Without Tuning	76%	85.68%
Corpus with verb Phrases	With Tuning	85.01%	89.32%
	Without Tuning	82%	86.34%

Table 5: Results of Hindi-Marathi SMT Subjective Evaluation

Hindi-Marathi Statistical MT System		BLEU score	METEOR	TER
With Uncleaned Corpus	With Tuning	1.96	0.124	45.06
	Without Tuning	1.26	0.127	44.55
With Cleaned Corpus	With Tuning	7.76	0.193	84.05
	Without Tuning	3.97	0.190	83.92
Corpus with Wordnet	With Tuning	11.78	0.225	82.91
	Without Tuning	9.31	0.217	84.30
With Function Words, kridanta pairs	With Tuning	12.21	0.274	83.79
	Without Tuning	9.25	0.214	85.06
With verb Phrases	With Tuning	18.15	0.281	77.94
	Without Tuning	10.26	0.261	85.48

Table 6 : Results of Hindi-Marathi SMT BLEU score, METEOR, NER Evaluations

Marathi-Hindi Statistical MT System		BLEU score	METEOR	TER
With Uncleaned Corpus	With Tuning	1.86	0.119	50.32
	Without Tuning	1.10	0.105	49.52
Corpus with Cleaned Corpus	With Tuning	8.01	0.171	78.32
	Without Tuning	4.22	0.160	Infinity
Corpus with Wordnet	With Tuning	11.54	0.278	80.75
	Without Tuning	9.78	0.226	77.30
Corpus with Function Words, kridanta pairs	With Tuning	12.20	0.283	81.19
	Without Tuning	10.46	0.263	81.19
Corpus with verb Phrases	With Tuning	17.80	0.288	80.36
	Without Tuning	13.70	0.265	81.48

Table 7: Results of Marathi-Hindi SMT BLEU score, METEOR, NER Evaluations

Sr. No.	Sentence		Explanation of phenomena	
1	Source Sentence : Hindi			
	Meaning			
	With Uncleaned Corpus	With Tuning	केंद्रीय सरकारी संग्रहालय १८७६ में वेल्स के राजकुमार के भारत दर्शन के समय बनवाया गया था और 1886 में जनता के लिए खुला था।	1. Many words are wrongly translated. 2. Function words not translated: के 3. Verb translation has tense problem: था not translated. 5. Insertion case: के शेड डाला, गवताळ 6. Conjunction not translated.
		Without Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स ह्याच्या राजकुमार के भारत दर्शन वेळी घेतलेला होती 1886 मध्ये वह जनतेसाठी खुला था .	
	With Cleaned Corpus	With Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स करण्यासाठी राजकुमाराच्या भारतात दर्शन वेळ बनवले आणि 1886 मध्ये जनतेसाठी खुले होता.	1. Function word: के not translated 2. In the second verb part, there is an inflection problem खुले होता 3. Insertion verb case: करण्यासाठी 4. Inflection problem: "दर्शन के समय" not translated correct as: दर्शनाच्या वेळी 5. First verb not translated correctly केले होते
		Without Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स ह्याच्या राजकुमाराच्या भारत दर्शन करताना केले होते 1886 मध्ये जनतेसाठी खुला होता.	
	Corpus With Wordnet	With Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स येथील राजकुमाराच्या भारत दर्शन वेळ बनवले होते आणि 1886 मध्ये जनतेसाठी खुले.	"दर्शनच्या" suffix addition is missing. Verb "होते" is wrongly translated as "होती". "वेल्स येथील" is correctly translated.
		Without Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स येथील राजकुमाराच्या भारत दर्शन वेळी बनवले होते आणि 1886 मध्ये जनतेसाठी खुले होती.	
	Corpus with Function Words, kridanta pairs	With Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स येथील राजकुमाराच्या भारत दर्शनच्या वेळी बनवले होते आणि 1886 मध्ये जनतेसाठी खुले होता.	"दर्शनच्या" suffix addition is missing. "वेल्स येथील" got translated correctly. Verb "होता" didn't get translated.
		Without Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स येथील राजकुमाराच्या भारत दर्शनच्या वेळी बनवले होते आणि 1886 मध्ये जनतेसाठी खुले होते	
	Corpus with verb Phrases	With Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये प्रिंस औफ वेल्सच्या राजकुमार याच्या भारतभेटीच्या वेळी उभारण्यात आले होते व १८८६ मध्ये जनतेसाठी खुले होते .	"राजकुमार याच्या" is translated correctly. Verb "होते" didn't get translated.
		Without Tuning	केंद्रीय सरकारी संग्रहालय १८७६ मध्ये वेल्स येथील राजकुमार याच्या भारत दर्शनच्या वेळी बनवले होते आणि 1886 मध्ये जनतेसाठी खुले होते .	

Table 8: Hindi- Marathi SMT Error Analysis

For example, during Hindi-Marathi Translation, in Hindi-Marathi alignment the noun "हस्तिनापुर" {hustinapur} can easily align to "हस्तिनापुरच्या" {hustinapoorchya}, since it is a single word. But in Marathi-Hindi alignment, "हस्तिनापुरच्या" {hustinapoorchya} may be map to "हस्तिनापुर" {hustinapur} and there is a possibility to drop down the post position "के" {ke} which is a separate word from the noun "हस्तिनापुर" {hustinapur}. These features will have

impact on the translation quality and hence the inflected forms may not translate properly from Marathi to Hindi. On the other hand, Hindi to Marathi translation system will not face this difficulty because of alignments. Thus there is an improvement in quality of Hindi-Marathi MT comparing to Marathi-Hindi MT.

Sr. No.	Sentence	Explanation of phenomena		
1	Source Sentence : Marathi	एखादा भरतपूरमध्ये एका ठिकाणापासून दुसरीकडे जाण्यासाठी टॅक्सी, सायकलरिक्षा व ऑटोरिक्षा यासारखे वाहतुकीचे अनेक पर्याय घेऊ शकतो .		
	Meaning	In Bharatpur for going from one place to another can bring taxi, cycle rickshaw or auto rickshaw.		
	With Uncleaned Corpus	With Tuning	कोई भरतपूरमध्ये एक स्थान दूसरी ओर जाने टॅक्सी, सिसोदिया ऑटोरिक्षा यासारखे ह्यांच्या कार्य कई विकल्प सकता है ।	Here the function words “के लिए”, “से”, verb “घेऊ”, Conjunction “व”, Insertion cases: कार्य, ह्यांच्या etc are not translated. Also Modal Auxiliary verb “शकतो” is translated as “सकते” instead of “सकता है ।” Post-positions not translated for words मध्ये, पासून, कडे, साठी. Wrong translated word: सिसोदिया. Not translated words: यासारखे, वाहतुकीचे
		Without Tuning	कोई भरतपूरमध्ये एक स्थान दूसरी ओर जाने टॅक्सी, सिसोदिया ऑटोरिक्षा यासारखे ह्यांच्या कार्य यासारखे वाहतुकीचे कई विकल्प सकते	
	With Cleaned Corpus	With Tuning	कोई भरतपुर में एक सदर से दूसरी ओर के लिए टॅक्सी, सायकलरिक्षा और ऑटोरिक्षा जैसी यातायात पानी से कर सकता है ।	Here the पर्याय घेऊ is wrongly translated as Insertion case “पानी से कर”. The kridant “जाने” is missing in जाण्यासाठी translation, only post-position साठी translated. Missing word: जाण्या. Suffix चे not translated (वाहतुकीचे). Wrongly translated words-ठिकाण – सदर, अनेक पर्याय. Verb wrongly translated: घेऊ- कर
		Without Tuning	कोई भरतपुर में एक स्थान से दूसरी ओर के लिए टॅक्सी, सायकलरिक्षा और ऑटोरिक्षा जैसी यातायात पानी से कर सकता है ।	
	Corpus With Wordnet	With Tuning	कोई भरतपुर में एक स्थान दूसरी ओर टॅक्सी, रिक्शा व आटो_रिक्शा जैसी यातायात कई विकल्प ले	Krudant form “जाण्यासाठी” not transferred. Modal Auxiliary verb “शकतो” is translated as plural form “सकते” instead of the kridanta “सकता है”. Function word “से”, “केलिए”, “के”, are missing. Suffix “चे” not translated from वाहतुकीचे.
		Without Tuning	कोई भरतपुर में एक स्थान से दूसरी ओर जाने के लिए टॅक्सी, रिक्शा व आटो_रिक्शा जैसी का यातायात कई विकल्प ले सकता है ।	Here the translation is perfect but the function word “जैसी का” is wrongly instead of “जैसे के”. Suffix चे not translated from वाहतुकीचे.
	Corpus with Function Words, kridanta pairs	With Tuning	कोई भरतपुर में एक स्थान से कहीं और जाने के लिए टॅक्सी, रिक्शा व आटो_रिक्शा जैसी यातायात के कई विकल्प ले सकता है ।	The translation is good except the word “दूसरी” is misplaced as “कहीं” due to lexical choice.
		Without Tuning	कोई भरतपुर में एक स्थान से दूसरी ओर जाने के लिए टॅक्सी, रिक्शा व आटो_रिक्शा जैसे यातायात के कई विकल्प ले सकता है ।	
Corpus with verb Phrases	With Tuning	कोई भरतपुर में एक स्थान से कहीं और जाने के लिए टॅक्सी, रिक्शा व आटो_रिक्शा जैसी यातायात के कई विकल्प ले सकता है ।	The translation is good except the word “दूसरी” is misplaced as “कहीं” due to lexical choice.	
	Without Tuning	कोई भरतपुर में एक स्थान से दूसरी ओर जाने के लिए टॅक्सी, रिक्शा व आटो_रिक्शा जैसी यातायात के कई विकल्प ले सकता है ।		

Table 9: Marathi – Hindi SMT Error Analysis

6. Conclusion

In this paper we have mainly focused on the usage of various lexical resources for improving the quality of Machine Translation. We have discussed the comparative performance of phrase based Statistical Machine Translation with various lexical resources for both Marathi – Hindi and Hindi-Marathi. As discussed in the experimental Section, SMT, although lacks the ability to handle rich morphology, it can overcome by using various lexical resources, which will help the machine to improve the translation quality.

In our experiments we have used various measures to evaluate such as BLEU Score, METEOR, TER and Fluency and Adequacy using subjective evaluation. We can see that there is an incremental growth in both Marathi- Hindi and Hindi-Marathi systems in terms of BLEU-Score, METEOR, and TER evaluations. Also our subjective evaluation results show promising scores in terms of fluency and adequacy. This leads to the importance of utilizing various lexical resources for an efficient Machine Translation system. Thus, we can come to a conclusion that various lexical resources can play an important role in providing good machine translation system for morphologically rich languages.

Our future work will be focused on investigating more lexical resources for improving the quality of Statistical Machine Translation systems and there by develop an accurate MT system for both Marathi-Hindi and Hindi-Marathi Machine Translation.

7. Acknowledgements

We thank Almighty and truth for this work.

8. References

- Abhay Agarwal, Alon Lavie, 2008, *Meteor, M-Bleu, M-ter Evaluation matrices for high correlation with Human Ranking of Machine Translation output*, Proceedings of the Third Workshop on Statistical Machine Translation, pages 115–118, Columbus, Ohio, USA, June 2008. 2008 Association for Computational Linguistics.
- Ananthkrishnan Ramanathan, Pushpak Bhattacharyya, Karthik Visweswariah, Kushal Ladha, and Ankur Gandhe. 2011. *Clause-Based Reordering Constraints to Improve Statistical Machine Translation*. IJCNLP, 2011.
- Anoop Kunchukuttan and Pushpak Bhattacharyya. 2012. *Partially modelling word reordering as a sequence labeling problem*, COLING 2012.
- Antony P. J. 2013. *Machine Translation Approaches and Survey for Indian Languages*, The Association for Computational Linguistics and Chinese Language Processing, Vol. 18, No. 1, March 2013, pp. 47-78
- Arafat Ahsan, Prasanth Kolachina, Sudheer Kolachina, Dipti Misra Sharma and Rajeev Sangal. 2010. *Coupling Statistical Machine Translation with Rule-based Transfer and Generation*. amta2010.amtaweb.org
- Bonnie J. Dorr. 1994. *Machine Translation Divergences: A Formal Description and Proposed Solution*. Computational Linguistics, 1994.
- Franz Josef Och and Hermann Ney. 2003. *A Systematic Comparison of Various Statistical Alignment Models*. Computational Linguistics, 2003.
- Franz Josef Och and Hermann Ney. 2001. *Statistical Multi Source Translation*. MT Summit 2001.
- Ganesh Bhosale, Subodh Kembhavi, Archana Amberkar, Supriya Mhatre, Lata Popale and Pushpak Bhattacharyya. 2011. *Processing of Participle (Krudanta) in Marathi*. ICON 2011, Chennai, December, 2011.
- Kevin Knight. 1999. *Decoding complexity in word-replacement translation models*, Computational Linguistics, 1999.
- Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu. 2002. *BLEU: a Method for Automatic Evaluation of Machine Translation*, Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318.
- Latha R. Nair and David Peter S. 2012. *Machine Translation Systems for Indian Languages*, International Journal of Computer Applications (0975 – 8887), Volume 39– No.1, February 2012.
- Peter E Brown, Stephen A. Della Pietra. Vincent J. Della Pietra, and Robert L. Mercer*. 1993. *The Mathematics of Statistical Machine Translation: Parameter Estimation*. Association for Computational Linguistics.
- Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, Evan Herbst. 2007. *Moses: Open Source Toolkit for Statistical Machine Translation*, Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session, Prague, Czech Republic, June 2007.
- Pushpak Bhattacharyya, 2010. *IndoWordnet*, LREC 2010.
- Rajat Kumar Mohanty, Pushpak Bhattacharyya, Shraddha Kalele, Prabhakar Pandey, Aditya Sharma, Mitesh Kopra *Synset Based Multilingual Dictionary: Insights, Applications and Challenges*, Global Wordnet Conference 2008
- Sreelekha, Raj Dabre, Pushpak Bhattacharyya 2013. *Comparison of SMT and RBMT, The Requirement of Hybridization for Marathi – Hindi MT* ICON, 10th International conference on NLP, December 2013.
- Veena Dixit, Satish Dethe And Rushikesh K. Joshi, *Design and Implementation of a Morphology-based Spellchecker for Marathi, an Indian Language*, Archives of Control Sciences, Volume 15(LI), 2005, No. 3, pages 251–258

Issues in chunking parallel corpora: mapping Hindi-English verb group in ILCI

Esha Banerjee, Akanksha Bansal and Girish Nath Jha

Jawaharlal Nehru University

New Delhi, India

(esha.jnu, akanksha.bansal15, girishjha)@gmail.com

Abstract

A well annotated corpus is a treasure for Natural Language Processing (NLP) and can benefit NLP research activities like Machine Translation, Text Summarization and Information Retrieval. But since language is a dynamic and complex phenomenon, Part Of Speech (POS) annotation and Local Word Grouping or chunking prove to be challenging tasks mainly because of two reasons: first, maximum possible information about the structure of a sentence needs to be captured and second, the tags should be easy for the machine to map and facilitate desirable output resulting in an effective application. The present paper deals with issues faced in chunking verb groups in Hindi with respect to their mapping with English verb groups for machine translation. There are some verbal constructions in Hindi which are not present in English e.g. double causatives and serial constructions. Thus the task of mapping Hindi verbal groups with English for the purpose of translation can restrict the accuracy of the output attained. These divergences have been charted out with some relevant examples from both the languages. The purpose of describing these divergence issues is to find the most appropriate way of creating Chunk Annotation Tag-set standards which are currently under development for Indian languages.

Keywords: chunking, annotated, corpus, verb, machine translation, standards, Indian Languages Corpora Initiative (ILCI)

1. Introduction

The BIS, in 2011, standardized a POS tagset for Indian languages. The tagset is in layered hierarchical schema and contains 11 coarse-grained parent tags such as Noun, Verb, Pronoun, Adjective and Particle which are divided into second level finer tags, with up to 5 sub-layers (Nainwani et al, 2011). Human annotators have been trained in labeling their native languages with the present schema under the ILCI project and the BIS POS tagged corpus is presently one of the largest available annotated corpora for Indian languages. The project is currently in its second phase and 50,000 new sentences in 17 languages, including English, are expected to be annotated by the end of this phase. The standardization of chunking guidelines, based on the BIS POS scheme, is the next step as the second level annotation activity.

The ultimate goal of the ILCI project is creation of parallel annotated text so as to facilitate other NLP tasks such as Machine Translation. Chunking is therefore an extremely crucial step by which rules of context-free grammar would help break a sentence into phrases and subsequent mapping of the corresponding structures would take place. This would aid in handling translation divergences arising due to factors such as word order and encoding of morphological information. In this context, verb groups present one of the most complex problems in mapping sentence structures from different language families. Verb groups in morphologically rich languages embed a lot of information, which have direct relation to the morpho-syntactic structure of the rest of the sentence.

The focus of the paper is to chart out the challenges that arise in the verb group chunking between Hindi-English language pair. Hindi, belonging to the Indo Aryan language family, is morphologically rich and its verbal behaviour varies vastly from English. The paper will discuss these variations in verb groups between the two languages and attempt to highlight some issues that need

to be tackled in order to effectively chunk and map corresponding verb groups.

2. Chunking standards for Indian languages

A chunk has been defined as “a minimal (non recursive) phrase (partial structure) consisting of correlated, inseparable words/entities, such that the intra-chunk dependencies are not distorted” (Bharati et al, 2006). Chunking helps determine the syntactic structure of a sentence by bringing out the relationship between words of varied syntactic categories. It is currently a popular and relatively successful shallow parsing technique especially for statistical machine translation systems.

In Indian languages, little work has been done with regard to creating chunking standards. Indo-Aryan languages have a relatively free word-order and therefore pose a challenge to NLP researchers. While Ray et al (2003) see a sentence from these languages as a ‘bag’ of chunks moving amongst each other, Bharati et al (1995) sought to solve the problem of dividing Hindi sentences into phrases based on Paninian grammar. Villian and Day (2000) used transformation rules to capture chunk from POS tags. Bhat and Sharma (2011) have proposed a hybrid approach for chunking in Kashmiri. There have been attempts to extract complex predicates based on computational and statistical methods with some degree of success (Sriram and Joshi, 2005). An attempt was made to extract verb groups from parallel Hindi-English corpus using projection POS tags (Soni, Mukherjee and Raina, 2006) using word alignment techniques, which reported a success of 83% precision and 46% recall.

Full-fledged chunk tagsets have been developed by Bharati et al (2006) under the Indian Languages Machine Translation (ILMT) consortium project sponsored by the Technology Development in Indian Languages (TDIL)

program of Department of Electronics and Information Technology (DeitY) of GOI. ILMT project included manual annotation of chunks on seven Indian languages. According to the guidelines, chunking falls under the fourth level of annotation within the tagset, which follows a layered approach including following layers: Part Of Speech, Local Word Grouping, chunk, syntactic, thematic role/Predicate Argument Structure, Semantic properties of lexical items, Word Sense Disambiguation/Anaphora Resolution. The Chunking tagset contains the following tags:

Sl. No	Chunk Type	Tag Name	Example
1	Noun Chunk	NP	Hindi: ((merA ncyA ghara))_NP "my new house"
2.1	Finite Verb Chunk	VGf	Hindi: mEMne ghara para khAna ((khAyA_VM))_VGf
2.2	Non-finite Verb Chunk	VGNF	Hindi: mEMne ((khAte - khAte_VM))_VGNF ghode ko dekhA
2.3	Infinitival Verb Chunk	VGINF	Bangla : binu Borabela ((sAna karawe))_VGINF BAlobAse
2.4	Verb Chunk (Gerund)	VGNN	Hindi: mujhe rAte meM ((khAna_VM))_VGNN acchA lagata hai
3	Adjectival Chunk	JJP	Hindi: vaha laDaZki hE((suMlata_JJ sI_RP))_JJP
4	Adverb Chunk	RBP	Hindi : vaha ((dhire- dhire_RB))_RBP cala rAhA thA
5	Chunk for Negatives	NEGP	Hindi: ((binA))_NEGP ((kucha))_NP ((bole))_VG ((kAmA))_NP ((nahIM calata))_VG
6	Conjuncts	CCP	Hindi: ((rAmA))_NP ((OrA))_CCP ((SyAmA))_NP
7	Chunk Fragments	FRAGP	Hindi: rAmA (jo merA baDZA bhAI hE) ne kahA...
8	Miscellaneous	BLK	

Figure 1: ILMT Chunk tagset

These guidelines serve as a foundation on which the present ILCI project corpus will be chunked.

3. ILCI project

The Indian Language Corpora Initiative (Jha, 2010) is a project initiated by the TDIL program of DeitY, GOI, which has an aim to collect parallel translated annotated sentences in 22 scheduled languages of the Indian Constitution plus English. The project is currently in its second phase with 17 languages being translated from the source language Hindi and annotated with part-of-speech categories using the BIS POS scheme. The English translations in the ILCI project are tagged using the Penn Treebank tagset. The next level of annotation will involve manual chunking (shallow parsing), which will be based on chunking standards created at IIIT-H.

4. Motivation

Parallel corpus can be used to build statistical parsers by mapping manually parsed text from source and target languages. Hindi is a language with SOV word order

which is relatively free. Verbs are multiple word entities agreeing in gender and person with the noun. In the BIS tag scheme, the POS categories used to label verbs are main and the auxiliary. The main verb is divided into finite, non-finite, infinitive and gerund. However, for Hindi, these distinctions were not applied at the POS level as it was felt that the finiteness/non-finiteness of a verb would be a quality better expressed at the chunk level.

In South Asian languages a common phenomena is the occurrence of complex predicates in which a noun or adjective or particle or adverb co-occurs with light verbs, required to complete the meaning, to create complex verbal structures (Abbi, 1992, Butt, 1995, Alsina, 1996). More commonly divided into the compound and conjunct verb constructions, their behaviour is far more complex than English predicates. The IIIT-H tagset, on which the BIS tagset is based, chunks verb groups into four main categories – Finite (VGf), Non-finite (VGNF), Infinitive (VGINF) and Gerund (VGNN). It has been stated that constricting chunk categories to broad based ones would aid in simplifying machine learning. However, we argue that keeping in mind the syntactic dissimilarities between the two languages, it is imperative to map the information contained at this stage as not only does it reflect crucial information faithfully but with some additional linguistic information, statistical methods would display accentuated success rate.

5. Case study examples

The following examples serve to highlight the preliminary issues that were observed while examining the verb group patterns in Hindi and English by speakers fluent in both languages. Marking the verb groups with their inherent properties would serve to enhance automatic translation accuracy.

5.1 Double causative constructions

In double causatives, the verb ending *-vaayaa/-vaanaa* indicates a verb construction derived from ditransitive verbs. These are different from causative constructions in that whereas in causative construction, the subject makes the object do something, in double causative, the subject acts on an intermediary to make the object do something.

Example:

- (i) likh-vaanaa/likh-vaayaa
write-to make one/write-made one
to make someone write
- (ii) khilwaanaa/ khilwaayaa
eat-to make one/eat-made one
to make one eat
- (iii) bhijwaanaa/ bhijwaayaa
write-to make one/write-made one
to have thing sent

In such cases, the English equivalent is roughly ‘to make one do/ to have one do’.

(iv) maine khaanaa ghara [bhijwaayaa thaa]

I food home [send-gotten sent had]

I had got the food sent home

This translation structure can be confused with the verb construction

(v) maine khaanaa ghara [bhej diya tha]

I food home [send had]

I had sent the food home

Marking such phrases as causative constructions could enable translation as

‘made/had [NP] (do) verb root’.

5.2 Adverbial/Adjectival construction

The *-vaalaa* construction, when added to a noun or verb, lends an adjectival quality like *doodhvaalaa* (milkman), *mehanata karne vaalaa* (one who works hard) etc. (Sinha, 2009). Participial constructions such as *jaane vaalii gaa.Dii* (the train which is about to leave) are treated as part of the finite verb chunk (VGF) which leads to loss of adjectival quality.

Example:

(vi) subaha [jaane vaalii] gaa.Dii aba shaama
ko[jaane vaalii hai]

morning [scheduled to go] train now evening [go about to is]

The train [which was to leave] in the morning [will leave] in the evening now.

In the above case, *jaane vaalii* is being used both as a participial and main verb. In the absence of a participial verb chunk, important information about the sentence structure is lost.

Another case of adverbial/adjectival construct is the reduplicated verb where examples like the following appear:

(vii) baiThe-baiThe so gayaa

sitting-sitting sleep went

fell asleep while sitting

(viii) chalte-chalte ruka gayaa

walking-walking stop went

stopped while walking

(ix) ruka-ruka ke chalnaa

stop-stop to-walk

walk while stopping intermittently

Another type of example would be participial constructions like

(x) behta huaa jharnaa

flow –ING waterfall

flowing waterfall

(xi) rotaa huaa bachchaa

cry –ING child

crying child

In the example,

[rotaa huaa] bachchaa gira gayaa

cry –ING child fell went

The crying child fell down.

rotaa huaa (crying) is being marked as Non-Finite verb chunk whereas ‘crying’ takes the Adjective tag and is included within the noun phrase.

According to the IIIT-H tagset, adverbial/adjectival constructions are accorded the chunk tag of Verb Non-finite (VGNF). In all the above cases, marking the adjectival quality of the verb would disambiguate them from the general categories of finite and non-finite chunks.

5.3 Serial verb construction

In serial verb construction common to north Indo-Aryan languages, in examples like

(xii) fona uthaa kara bolaa

phone pick do said

picked up the phone and said

(xiii) botala uthaa kara dekhii

bottle pick do saw

picked up the bottle and saw

there are two separate actions being performed and neither of the verbs is subordinate to the other. In such cases, chunking the verbs as a single phrase would pose problems when mapping to English translations.

It would additionally create ambiguity with compound verb phrases like

(xiv) maara giraayaa

hit made-fall

defeat

which also contains Verb+Verb (V+V) construction. However, in compound verbs, the light verb (*giraayaa*, in this case) would not act as a separate action; rather, it would provide a composite meaning with the main verb (*maara*).

5.4 Conjunct verb

Conjunct verbs are traditionally defined as those verbal groups which have the structure Noun+Verb, such as

kShamaa karnaa (to forgive), *jamhaai lenaa* (to yawn), *maara khaanaa* (get hit), *bhaaga jaanaa* (to run). The guidelines state that due to the syntactic relation that exists between the Noun and Verb in such a construction, it would be marked as such at a later stage of annotation.

Example:

- (xv) [eka prashna] [karnaa]
 a question to-ask
 [to ask] [a question]

However, if Nouns and Verbs are chunked separately in such cases, the challenge that exists in these cases is the identification of those Noun+Verb combinations which create conjunct verbs. For example, there would have to be a diagnostic to disambiguate *choTa laganaa* (to get hurt) from conjunct verb construction *mana laganaa* (be interested). As can be observed, in English, the conjunct verb construction does not derive its semantics from the individual meaning of the verbs. Rather, the light verb in the conjunct verb construction loses semantic content and provides a distinct meaning to the verb phrase as a whole. Such Noun+Verb constructions are generally fixed in nature and if marked as such, can create a database of complex predicates and accurately map the corresponding verb from English.

5.5 Null Subject

Apart from the direct mapping difficulties that were observed, there were certain instances where the verb group chunking would affect the chunking of the Noun Phrase. Null Subject is one such example.

Some languages like Hindi exhibit the null subject phenomena wherein the number, person, gender attributes are expressed by the verb alone. The verb, in those cases, acts as an independent clause without an explicit subject.

Example: ghara gayaa/gayii/gaye

home went (+mas)/went (+fem)/went (pl)

He/she/they went home.

In the above case, the morphology of the verb gives the gender and number information about the subject. In such cases, the empty subject has to be mapped onto a corresponding noun phrase. Marking the verb chunk with relevant morphological cues helps this process.

5.6 Gender attribute

As mentioned above, verb groups in Hindi contain inflections for gender and number which determines the corresponding attributes of the subject or object. Without specifying these attributes, the chunk may conjoin to the noun phrase randomly while mapping, which gives rise to inaccurate results.

Examples:

- (xvi) raama ghara [jaataa hai]

Ram home goes

Ram goes home

- (xvii) siitaa ghara [jaatii hai]

Sita home goes

Sita goes home

6. Conclusion

The issues described in the paper highlight the dissimilar behaviour of verbs in Hindi and English and its effect on creating chunking standards. The behaviour of complex predicates in South Asian languages, although linguistically complex, is a phenomena well documented in literature. Various attempts have been made, both rule based and statistical, to map the linguistic structures of such languages to other languages which lack such structures, like English. With a view to facilitating machine learning, annotation of parallel corpora is deemed as one of the most effective techniques. This paper has dealt specifically with features of the Hindi language, most of which would be common in north Indo-Aryan languages like Punjabi. However, the research should be extended to other Indian languages which consist of five language families and which include phenomena like agglutination. Due to the intricate nature of Verb Phrase, this work has dealt with verb group mapping alone. However, due to the dependency of other POS features on the verb and the Gender, Number and Person agreement that Indian languages exhibit, it would have significant effects on the chunking of Noun Phrase as well. As shown by examples, if the chunking process can train machines with linguistically rich parallel corpus, the results would be expected to have far higher accuracy.

7. Acknowledgement

This research has received valuable input by the discussions between Principal Investigators (PI) of the language groups of ILCI and ILMT projects. We would also like to acknowledge TDIL, DeitY, GOI for the use of the ILCI corpus for study of translation divergences for this paper.

8. References

- Abbi, A., & Gopalakrishnan, D. (1991). Semantics of explicator compound verbs in south Asian languages. *Language Sciences*, 13(2), 161-180.
- Alsina, Alex, Joan Bresnan, and Peter Sells (1997) 'Complex predicates: Structure and theory'. Alex Alsina, Joan Bresnan, and Peter Sells (eds.) *Complex Predicates*, 1-12. Stanford, CA: CSLI Publications.
- Bali, K., Choudhury, M., Chatterjee, D., Maheswari, A., & Prasad, S. (2009). Correlates between Performance, Prosodic and Phrase Structures in Bangla and Hindi: Insights from a Psycholinguistic Experiment. *Proceeding of ICON*.

- Begum, R., Jindal, K., Jain, A., Husain, S., & Sharma, D. M. (2011). Identification of conjunct verbs in Hindi and its effect on parsing accuracy. In *Computational Linguistics and Intelligent Text Processing* (pp. 29-40). Springer Berlin Heidelberg.
- Bharati, A., Chaitanya, V., Sangal, R., & Ramakrishnamacharyulu, K. V. (1995). *Natural language processing: a Paninian perspective* (pp. 65-106). New Delhi: Prentice-Hall of India.
- Bharati, A., Sangal, R., Sharma, D. M., & Bai, L. (2006). Anncorra: Annotating corpora guidelines for pos and chunk annotation for indian languages. *LTRC-TR31*.
- Bhat, R. A., and Sharma, D. M. (2011). A Hybrid Approach to Kashmiri Shallow Parsing. In *LTC-2011: The 5th Language and Technology Conference: Human Language Technologies as a Challenge for Computer Science and Linguistics (LTC-2011)*.
- Chakrabarti, D., Mandalia, H., Priya, R., Sarma, V. M., & Bhattacharyya, P. (2008, August). Hindi Compound Verbs and their Automatic Extraction. In *COLING (Posters)* (pp. 27-30).
- Dave, S., Parikh, J., & Bhattacharyya, P. (2001). Interlingua-based English-Hindi Machine Translation and Language Divergence. *Machine Translation*, 16(4), 251-304.
- Dorr, B. J. (1993). *Machine translation: a view from the Lexicon*. MIT press.
- Gune, H., Bapat, M., Khapra, M. M., & Bhattacharyya, P. (2010, August). Verbs are where all the action lies: Experiences of Shallow Parsing of a Morphologically Rich Language. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters* (pp. 347-355). Association for Computational Linguistics.
- Jha, G. N. (2010). The TDIL program and the Indian language corpora initiative (ILCI). In *Proceedings of the Seventh Conference on International Language Resources and Evaluation (LREC 2010). European Language Resources Association (ELRA)*.
- Kachru, Y. (1980). *Aspects of Hindi grammar*. New Delhi: Manohar.
- Koul, O. N. (2008). *Modern Hindi Grammar*. Dunwoody Press.
- Mannem, P., & Dara, A. (2011, June). Partial parsing from bitext projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (pp. 1597-1606). Association for Computational Linguistics.
- Mukerjee, A., Soni, A., & Raina, A. M. (2006, July). Detecting complex predicates in Hindi using POS projection across parallel corpora. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties* (pp. 28-35). Association for Computational Linguistics.
- Mukerjee, A., Soni, A., & Raina, A. M. (2006, July). Detecting complex predicates in Hindi using POS projection across parallel corpora. In *Proceedings of the Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties* (pp. 28-35). Association for Computational Linguistics.
- Nainwani, P., Banerjee, E., Kaushik, S., & Jha, G. N. (2011) Issues in annotating less resourced languages—the case of Hindi from Indian Languages Corpora Initiative (ILCI). In the fifth Proceedings of Language Technology Conference (LTC '11).
- Sinha, R. M. K. (2009, December). Learning Disambiguation of Hindi Morpheme "vaalaa" with a Sparse Corpus. In *Machine Learning and Applications, 2009. ICMLA'09. International Conference on* (pp. 653-657). IEEE.
- Ray, P. R., Harish, V., Sarkar, S., & Basu, A. (2003). Part of speech tagging and local word grouping techniques for natural language parsing in Hindi. In *Proceedings of the 1st International Conference on Natural Language Processing (ICON 2003)*.
- Singh, S., Damani, O. P., & Sarma, V. M. (2012). Noun Group and Verb Group Identification for Hindi. In *COLING* (pp. 2491-2506).
- Sinha, R. M. K. (2009, August). Mining complex predicates in Hindi using a parallel Hindi-English corpus. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications* (pp. 40-46). Association for Computational Linguistics.
- Venkatapathy, S., Agrawal, P., & Joshi, A. K. (2005). Relative Compositionality of Noun+ Verb Multi-word Expressions in Hindi. In *Proceedings of International Conference on Natural Language Processing (ICON'05)*.
- Vilain, M., and Day, D. (2000). Phrase Parsing with Rule Sequence Processors: an Application to the Shared CoNLL Task. In *Proceedings of CoNLL-2000 and LLL-2000, Lisbon, Portugal*.