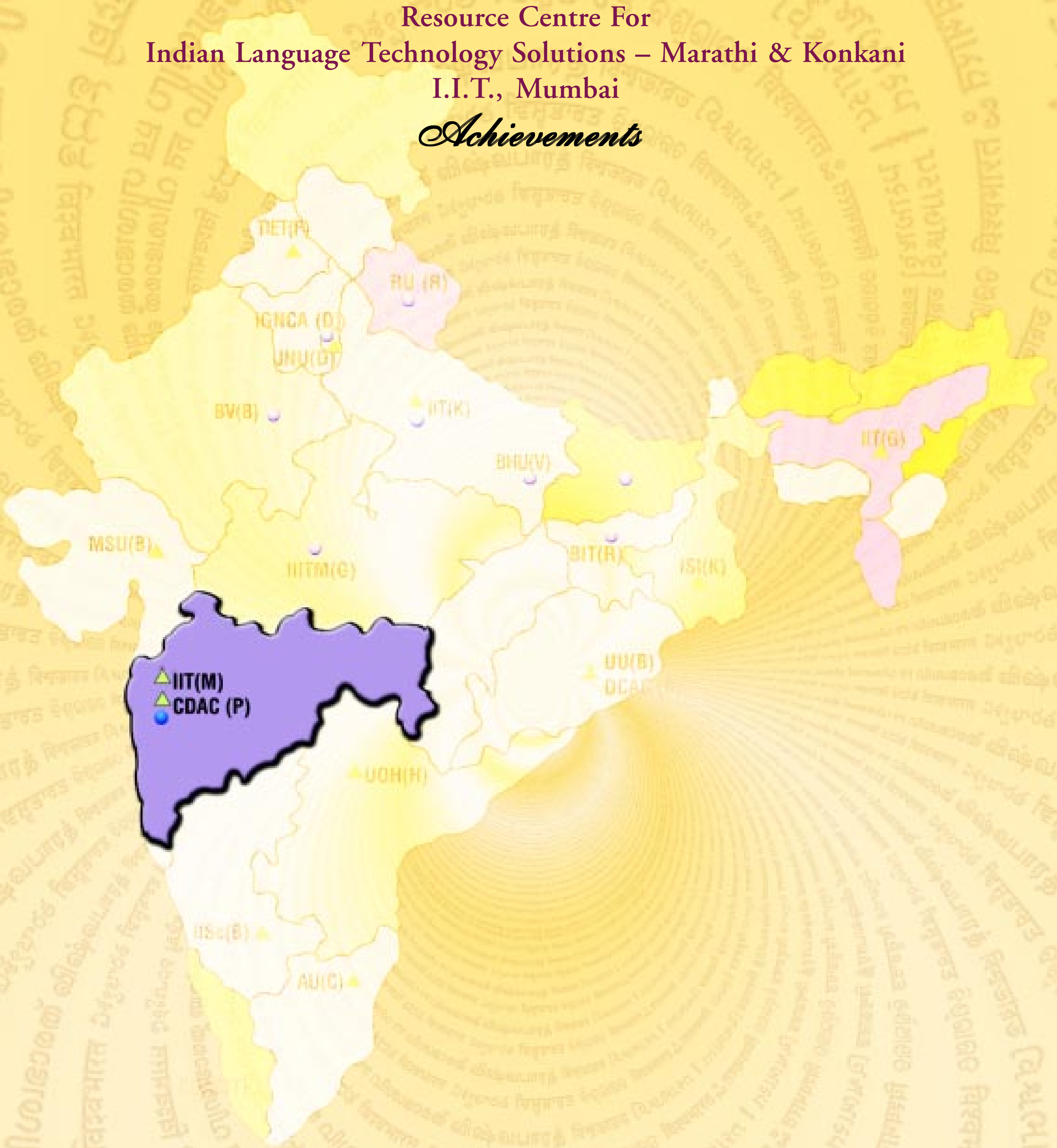
Resource Centre For Indian Language Technology Solutions – Marathi & Konkani I.I.T., Mumbai

Achievements



Computer Science and Engineering Department
Indian Institute of Technology, Mumbai-400076 India
Tel. : 00-91-22-25722545 Extn. 5879 E-mail : pb@cse.iitb.ernet.in
Website : http://www.cfilt.iitb.ac.in

RCILTS-Marathi & Konkani
Indian Institute of Technology, Mumbai

Introduction

The Resource Center for Technology Solutions in Indian Languages was set up at IIT Bombay in the year 2000. The center concentrated on developing niche technologies for Indian languages with special focus on the languages of Western India, in particular Marathi and Konkani. The main theme of the research has been development of (i) lexical resources like *wordnets, ontologies* and *semantics-rich machine readable dictionaries (MRD),* (ii) interlingua based machine translation softwares for *Hindi* and *Marathi* and (iii) Marathi search engine and spell checker. Over the years the center has built its strength and reputation in its chosen areas and has obtained national and international visibility. In the paragraphs that follow we describe the technologies already developed and in the process of being developed, and also the impact the center has made in the area of information technology localization.

1. Lexicon and Ontology

1.1 Lexicon

The lexicon is being prepared in the context of interlingua based machine translation. The interlingua is the *Universal Networking Language (UNL)* (http://ww.unl.ias.unu.edu) which is a recently proposed interlingua. The heart of the UNL based MT is the *universal word (UW)* dictionary. UWs are essentially disambiguated English words and stand for unique concepts. Our lexicon contains about 70,000 UWs and approximately 200 morphological, grammatical and semantic attributes. These entries are linked with Hindi headwords. The lexicon is available on the web (http://www.cfilt.iitb.ac.in/dictionary/index.html). UW-Hindi Lexicon is required for enconversion (analysis) and deconversion (generation) processes. The lexicon can also be used as a language reference for English and Hindi. We are now concentrating on completing the language coverage and standardizing the lexicon (standard restrictions and semantic attributes).

1.2 Ontology

An ontology is a hierarchical organization of concepts. Domain specific ontologies are critical for NLP systems. We have classified **Nouns** as *animate* and *inanimate*. Further sub-classification divides *animate* nouns as *flora* and *fauna* to cover plants and animals. The *inanimate* category is sub-classified as *object*, *place*, *event*, *abstract entity etc.* Concept nouns indicating *time*, *emotion*, *state*, *action* etc. have also been incorporated. **Verbs** have been divided into *do, be* and *occur* categories, with further sub-classifications *verbs of action*, *verbs of state*, *temporal verbs*, *verbs of volition*, *etc.* **Adjectives** have been classified as *descriptive* indicating *weight*, *colour*, *quality*, *quantity*, *etc.* and also as *demonstrative*, *interrogative*, *relational* and so on. **Adverbs** are categorized according to *time*, *place*, *manner*, *quantity*, *reason*, *etc.*

Statistics of Ontological Categories

We have 22 broad categories for Noun, Verb, Adjective and Adverb. Under these categories, we have 57 sub-categories for all word classes. The noun class has 55 sub-sub and 29 sub-sub-sub categories (*vide* the table below):

1.	No. of Categories of Noun, Verb, Adjective and Adverb	22
2.	No. of Sub-categories of Noun, Verb, Adjective and Verb	58
3.	No. of Sub-sub Categories of Noun	55
4.	No. of Sub-sub-sub Categories of Noun	29
	Total No. of Categories	164

2. The Hindi WordNet

The Hindi wordnet is an on-line lexical database. The design closely follows the English wordnet. The synonym set {घर, गृह} and {घर, परिवार}- for example- can serve as an unambiguous differentiator of the two meanings of {घर}. Such *synsets* are the basic entities in the wordnet. Each sense of a word is mapped to a separate synset in the wordnet. All word sense nodes are linked by a variety of semantic relationships, such as ***is-a*** (hypernymy/hyponymy) and ***part-of*** (meronymy/holonymy).

The lexical data entered by linguists and lexicographers are stored in a MySQL database. The web interface (www.cfilt.iitb.ac.in) gives the facility to query the data entered and browse the semantic relations for the search word. The interface also provides links to extract semantic relations that exist in the database for the search word. So far approximately 10000 synsets have been entered. This corresponds to about 30,000 words in Hindi. There is a morphological processing module as a front end to the system.

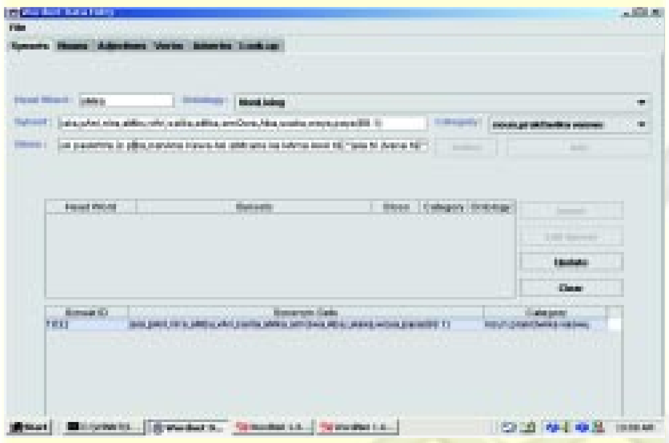


Figure 1: Snapshot of Data entry interface for entering synset data

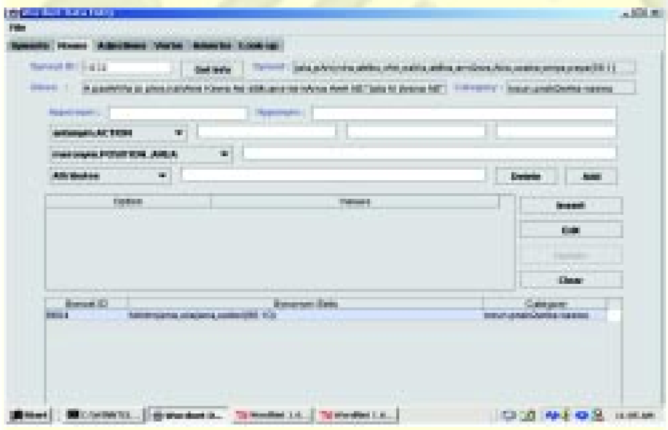
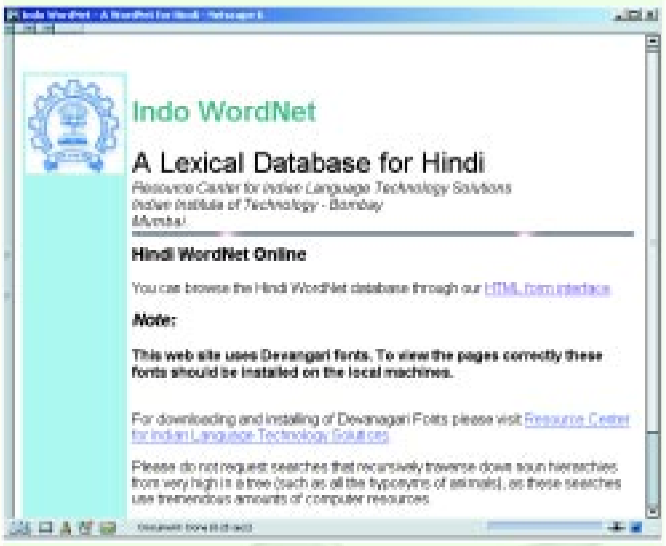


Figure 2: Snapshot of Data Entry interface for entering semantic relations for noun category



Indo WordNet

A Lexical Database for Hindi

Resource Center for Indian Language Technology Solutions
Indian Institute of Technology - Bombay
Mumbai

Hindi WordNet Online

You can browse the Hindi WordNet database through our HTML form interface

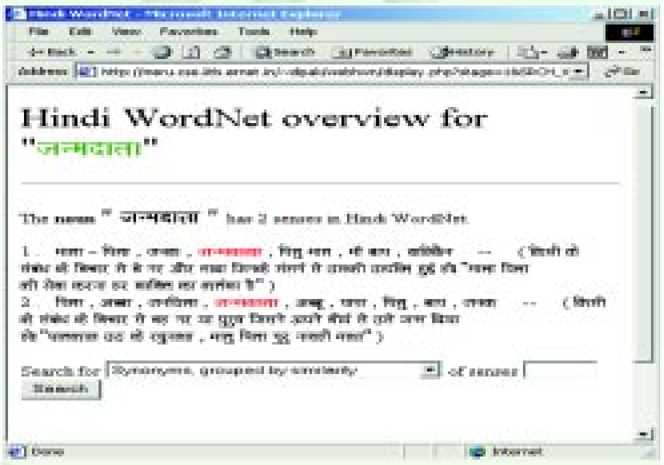
Note:

This web site uses Devangari fonts. To view the pages correctly these fonts should be installed on the local machines.

For downloading and installing of Devanagari Fonts please visit Resource Center for Indian Language Technology Solutions

Please do not request searches that recursively traverse down noun hierarchies from very high in a tree (such as all the hyponyms of animals), as these searches use tremendous amounts of computer resources

Figure 3: Online web interface of the Hindi WordNet



Hindi WordNet overview for "जन्मदाता"

The noun " जन्मदाता " has 2 senses in Hindi WordNet.

1. [illegible]

2. [illegible]

Search for Synonyms, grouped by similarity of senses

Search

Figure 4: Snapshot of web interface for the query result for the word "जन्मदाता"

3. The Marathi WordNet

Since Marathi and Hindi share many common features, it has been possible to adopt the Hindi wordnet as the basis for Marathi wordnet creation. Hindi and Marathi originate from *Sanskrit* and, therefore, the तत्सम (words taken directly from mother language, *e.g.* गति, कृति) and the तद्भव (a word which has evolved organically from the mother language, *e.g.* गृह → घर, कर्म → काम) words in both the languages are often same in meaning. Also, both the languages have the same script- *Devanagari*.

In the construction of the Marathi wordnet, a Hindi synset is chosen and mapped to the corresponding Marathi synset. For instance, for the Hindi synset {**कागद, कागज**}- standing for paper- the Marathi synset is {**कागद**}. A gloss is added, which explains the meaning of the word, and an explanatory sentence is inserted. For the current example, the gloss is बांबू व गवत यांच्या लगदयापासून तयार केले जाणारे पातळ पान ज्यावर लिहीले वा छापले जाते and"त्याने को-या कागदावर माझी सही घेतली". So far, 3400 synsets have been entered. This corresponds to about 10,000 words in Marathi. Our aim is to cover all the common Marathi words. As the Marathi wordnet is based on the Hindi wordnet, all the semantic relations become automatically inherited. An additional benefit to accrue is the creation of a parallel corpus.

4. Automatic Generation of Concept Dictionary and Word Sense Disambiguation

A Concept (UW) Dictionary is a repository of language independent representation of concepts using special disambiguation constructs. A system has been developed for *automatically* generating document specific concept dictionaries in the context of language analysis and generation using the Universal Networking Language (UNL). The dictionary entries consist of mappings between head words of a natural language and the universal words of the UNL. The manual effort in constructing such dictionaries is enormous, because the lexicographer has to identify the disambiguation constructs and the semantic attributes of the entries. Our system, which constructs the UW dictionary (semi-) automatically, makes use the English wordnet and processes the part of speech tagged and partially sense tagged documents. The sense tagging step is a crucial one and uses the verb association information for disambiguating the nouns. The accuracy of the word sense disambiguation system is 70-75% and that of the DDG (document specific dictionary generator) is 90-95%.

The DDG makes use of a rule base for producing the dictionary. It is based on the principle of expert systems, doing inferencing and providing explanations for the choices made with respect to the *restrictions* and *semantic attributes.*

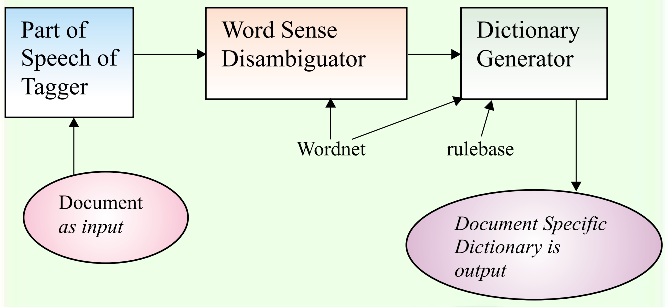


Figure 5: stages in automatic dictionary generation

5. Hindi Analysis and Generation

5.1 Hindi Analysis (Enconversion)

The analysis engine uses the *UW-Hindi dictionary* and the *analysis rule base*. The dictionary contains *headword, universal word* and its *grammatical and semantic attributes*. The process of analysis depends on these *semantic* and *grammatical attributes*. There are 41 relations defined in the UNL specification. The system is capable of handling all of them. The analysis system can deal with almost all the morphological phenomena in Hindi. At present, there are 5500 rules dealing with morphosyntactic and semantic phenomena. The system has been tested on corpora from the UN, the MICT and the agricultural domain.

5.2 Hindi Generation (Deconverion)

The generation process also uses the *UW-Hindi dictionary* and the *generation rule base*. The dictionary is the same for both analysis and generation purpose, only the rule base is different. The generation rule is formed from the grammatical and semantic attributes as well as the syntactic relations. Some of the features of the system are:

- *Matrix-based priority of Relation* has been designed to get the correct syntax.
- Extensive work has been done on Morphology.
- The system has gone through the UNL of corpora from various domains with satisfactory results.
- There are almost 6000 rules for syntax planning and morphology of all parts of speech.

6. Marathi Analysis

Marathi is a natural language of Indo-Aryan family.

The textual communication uses the *Devanagari* (देवनागरी) script. The analysis and generation of Marathi makes use of the Marathi-UW lexicon and the rules for the Marathi language grammar. There are about 2100 rules to handle Marathi verb phenomena which are listed below:

Sr. No.	Phenomenon	Description
1	Present tense	Time w.r.t. writer
2	Past tense	Time w.r.t. writer
3	Future tense	Time w.r.t. writer
4	Event in progress	Writer's view of aspect
5	Complement	Writer's view of reference
6	Passive voice	Writer's view of topic
7	Imperative mood	Writer's attitude
8	Ability of doing something	Writer's viewpoint
9	Intention	Intention about something or to do something
10	Should	To do something as a matter of course
11	Unreality	Unreality that something is true or happens

7. Speech Synthesis for Marathi Language

The aim of this project is to design speech synthesis system to speak out Marathi text written in Devanagari script. *Concatenative speech synthesis* model is used to achieve this goal. Ours is an unlimited vocabulary system. It employs basic units, *i.e.,* vowels and consonants as the basis for constructing words and sentences. The database of basic speech units, required in this approach, is few hundred kilobytes as against other approaches where the size is almost tens of megabytes. We are able to get intelligible speech with this approach. A graphical user interface for the system has been developed. It provides features for drawing waveforms and for amplifying the output speech signal. It also provides, help for displaying Marathi text through English keyboards.

Currently we are working in rendering newspapers as speech is taking shape. On-line reading of Marathi newspaper *Maharashtra Times* is being carried out as an illustration of the technology.

8. Project Tukaram

Tukaram Project provides the entire collection of Saint Tukaram's Abhangs in browsable and searchable format. Tukaram's abhangs are a household name in Maharashtra. It is possible to convert the existing version to a stand-alone system, which can be installed on a single machine. The abhangs are available for public viewing on http://www.cfilt.iitb.ac.in (also see the picture below). Browsability is provided at chapter and verse levels. Users can move from one chapter or verse to another by using links at the bottom of the webpage.

The search engine for Tukaram Project is made compatible with both Linux and Windows. Tukaram's Abhangas were keyed in Akruti font and were later converted to ISFOC font. ISCII encoding is used by the Tukaram search engine.

To enable the users to use the Tukaram search engine from both Windows and LINUX, we chose the XDVNG font for the input technology while keeping ISFOC for display purposes only. Clients can key in their queries through phonetic English designed by Sibal (1996). Clients' input queries typed in phonetic English are displayed at clients' terminal in Marathi using the *XDVNG* font. Phonetic English to *XDVNG* mapping is available as JavaScript from *http://www.sibal.com/sandeep/jtrans*. Since JavaScript is used for mapping, it is easy to integrate it in a page.

A query in XDVNG is sent to the server at IIT Bombay and is converted into ISCII at the server. For this conversion, an XDVNG to ISCII converter provided by IIIT Hyderabad is used after modifications. The features of Project Tukaram are summarized in table below:

Font and Package used for Original Script	Akruti in MS Word
Number of converted HTML chapters	165
Number of converted HTML verses	4614
Font used in HTML display pages	ISFOC

Total number of words excluding HTML tags	2,12,442
Number of distinct words used for the Indexing	34,773
Keyboard mapping at client	Phonetic English
Query Display Technology at Client	XDVNG
Result Display Technology at Client	ISFOC
Encoding used for Indexing	ISCII
Database Server	MySql
Languages used for Converters	Lex and C
Language used in Search Engine	JAVA
Language used in Client Side	JavaScript

9. Automatic Language Identification of Documents using Devanagari Script

The problem of language identification has been addressed in the past. There are existing methods like *unique letter combinations*, *common words*, and *N-grams* technique. A subtle issue here is the length of the text available for classification. Many methods require longer texts for language identification. Also, some methods require some a priori linguistic knowledge. Another issue arises when the test set for the classification program contains some semantic errors. The classification technique should be robust in such a way that these errors do not affect the accuracy of the classifier.

The N-grams method has been developed based on a rank order of short letter sequences. A rank ordered list of the most common short character sequences is derived from the training document set. A rank ordered list is prepared for every language under consideration. Each such list is called a profile. Language identification is done by measuring the closeness of the test document to each of these profiles. The profile, which is closest to the test document, gives the language of the test document. Figure 1 shows the data flow diagram of the classifier. We have extended this approach for Devanagari script, as the conventional approach was not giving desired accuracy.

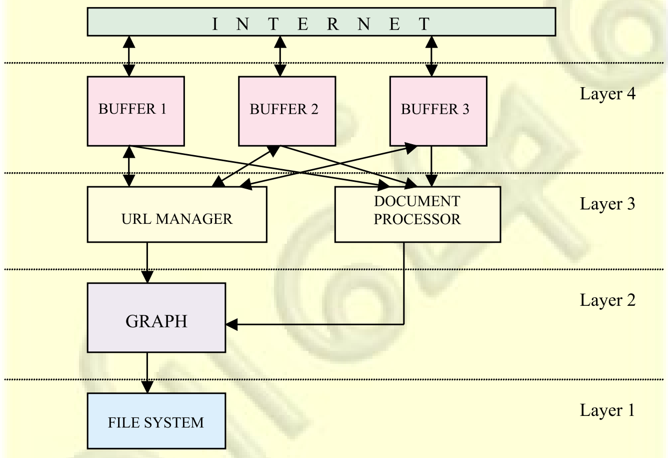
Letter Approach: In this approach, each letter in Devanagari is treated as a character in an N-gram. That means an N-gram may contain more than "N" characters. On the other hand, an N-gram will be exactly "N" (possibly half) letters in Devanagari script.

Conjunct Approach: This is a variant of the previous approach. We draw motivation for this approach by describing *Conjuncts*. Indian scripts contain numerous such conjuncts, which essentially are clusters of up to four consonants without the intervening implicit vowels. The shape of these conjuncts can differ from those of constituting consonants. For example, इच्छा has a conjunct च्छा. It comprises of two consonants, च and छ. The consonant च is considered a half letter. In the Conjunct approach, a conjunct is considered as a single letter.

Results*:* From experimental results, it could be observed that although the common words technique is computationally cheaper, the N-gram techniques are more accurate. Also, it was clear that our extensions to N-grams work very well than the conventional N-grams approach. The N-gram approach was very much suitable for another reason. The documents used in experiments had textual errors. As N-grams method is quite robust to textual errors, the results were largely satisfactory. In our approach, although the developing the method itself required some linguistic knowledge, the method is completely automatic and does not require any finetuning as required in case of common words approach.

10. Object Oriented Parallel and Distributed Web Crawler

The vastness and dynamic nature of the WWW has led to the need for efficient information retrieval. A crawler's task is to fetch pages from the web. The crawler starts with an initial set of pages, retrieves them, extracts URLs in them and adds them to a queue. It then retrieves URLs from the queue in some specific order and repeats the process. We have designed and implemented an object oriented web crawler. Our crawler has 4 components: Graph component, UrlManager, Buffer and Document Processor. The crawler has a 4-tier architecture with the storage file system at the lowest layer and crawl-buffers at level 4. The architecture is shown in figure below. The Graph component is at level 2 and it interacts with the file system. The UrlManager and Document Processor are at level 3. This layer forms the business logic layer. The crawler architecture is parallel and distributed. It uses a CORBA environment and runs on a Linux Cluster.



4-tier Architecture of Web Crawler

User Level Storage System

A search engine requires large amount of storage space for storing the crawled pages. The existing databases and file systems are not specialized to handle storage of structures such as web graphs. We are building storage systems for efficient storage and retrieval to this requirement. In the storage system, webpages correspond to the nodes of the graph and the hyperlinks correspond to the edges. Graph traversal methods can be applied for crawling purposes.

In the current version of the storage system, Berkeley Database (BDB) over Linux has been used as the backend for storing webpages. An advantage of BDB is that the library runs in the same address space and no inter-process communication is required for the database operation. Secondly, because BDB uses a simple function-call interface for all operations, there is no query language to parse.

The storage system provides a set of API for application programs such as crawlers. The APIs help keeping users' applications independent of implementation of the storage system. The storage systems recognizes primitives such as *nodes* which represent web pages and *graphs* which represents a specific WWW graph. A storage API encapsulates read and *write* functions. It provides functions such as create graph, store and read a webpage, parse the page and get outlinks.

11. Designing Devanagari Fonts

The following three types of 'Arambh' family fonts are created:

- 8-bit true type font (.ttf)
- 16-bit Unicode bitmap ascii font (.bdf)
- 16-bit Unicode true type font (.ttf).

Though there are many Devanagari fonts available, no single font contains all glyphs required for displaying documents such as Damale's grammer and Saint Tukaram's abhangas. Hence a new font is being created to handle the additional requirements. The font called 8-bit *Arambh*, is a true type font, which covers all required glyphs. In this font we get the scalability feature of true type fonts. Two types of 16-bit Arambh Unicode fonts developed are *ArambhU (bdf)* font and *ArambhU (ttf)* font. Each glyph in BDF is represented as a matrix of dots. These fonts are rendered faster than TrueType fonts. Each glyph in the TrueType font is represented as outline curve thus facilitating scalability.

12. Low Level Auto Corrector

We have built a low level syntactic checker and autocorrector based on a classification of low-level syntactic errors. The tool uses a set of rules to detect and correct the errors. It was used in project Tukaram, and over 300 errors were detected and automatically corrected. The kinds of errors handled are visible and invisible errors due to multiplicity, ordering, mutual exclusion and association properties.

Types of Low Level Syntactic Errors

Below, a classification of low-level syntactic errors in a document written in an Indian Language is provided with a few examples. Reasons of these errors may be attributed to violations of the constraints of multiplicity, ordering and composition of ligatures, especially of the matra characters.

Multiplicity

Typically, ligatures such as matra, halant and anuswar are used only once in a single letter composition. While typing, it may be possible that they get typed twice and the result is still not visible in the text in an Indian Language font. Such errors can be classified as invisible errors, while some remain visible in the display.

Invisible Errors

For example, in the case of an extra anuswar in letter अं the source in ITRANS is given by *aM*, whereas, *aMM* also produces अं. The Devanagari characters seen in the previous sentence were actually produced by two different encodings. While the result of this error is not visible in the display, this could be a cause of concern for an application such as a pattern matching algorithm, an index or a converter.

Invisible errors may also occur in the case of multiple ukar and matra (ु and `) characters. For example, in पू an extra ukar is present but it is invisible, or overlapped.

Visible errors

While some ligatures overlap in display and the errors remain undetected by mere human inspection of the display, some errors may be detectable by display inspection. This is possible due to positioning adjustments done on ligature placements.

For example, an extra kana in Marathi, such as in letter पा, is visibly detectable if it occurs more than once since the additional ligature is right shifted and either displayed as a kana, or taken as a separate character. An example of the later case is encoding *paaa* in ITRANS, which is displayed as पाअ

Ordering

In some cases, more than one vowel can occur in a single letter. For example, an anuswar and a kana in Marathi as in letter पां may be typed in Akruti_Priya_Expanded in two ways, as *hebe* (where h stands for प, e for ा and b for anuswar), or as *heeb*. While the first choice results in an unaesthetic placement of anuswar on top of the first vertical line, the second choice gives the desired result. With a smaller font size, the differences may get overlooked, but with an enlarged font size, they are easily detected.

Mutual Exclusion

In Indian Languages, the composition rules are also specified for mutually exclusive vowels. For examples, an ukar and a kana cannot occur in one character. Similarly, a velanti and an ukar or a kana cannot occur in one letter. Whereas, it is possible for a kana and a matra to occur together in one letter.

Association

Certain consonants do not accept rafar (i.e., र्) or specific conjuncts. For example, a combination of ळ and र् is invalid. Similarly, it is not possible to combine certain consonants such as ज्ञ with other consonants, or ह with प.

13. Font Converters

Following is the list font converters have been developed by IIT Bombay.

- IITK to ISCII
- ISCII to IITK
- Akruti_Priya_Expanded to ISCII
- SHUSHA to ISCII

- IITK to SHUSHA
- DV-TTYogesh to ISCII
- ISCII to DV-TTYogesh

14. Marathi Spell Checker

Under this ongoing project, a stand-alone spellchecker is being built for Marathi. The spellchecker will be available to spell check documents in a given Encoding.

From the CIIL (Central Institute of Indian Language) corpus, 12,886 distinct words have been listed. Similarly other Marathi texts at the center are being used to build a basic dictionary. A morphological analysis is being carried out on the collection of words. For example, an automatic grouping algorithm identified 3,975 groups out of 12,886 distinct words. First word is usually the root word. Thus, there are approximately 4000 root words from Marathi corpus. A manual proof reading will be done on these results. The morphology will be enriched.

A motivation behind the stand-alone spellchecker is that it can be used without an editor through a packaged interface, or it can be integrated with other compatible applications such as OCR.

Content creation tools

The Marathi content placed on the CFILT website, *viz.* Saint *Tukaram's Abhangas, Soundarya Meemansa* and *Damle's Marathi grammar* have been typed using Akruti Software along with MS Word. Akruti provides three keyboard layouts: *Typewriter, English Phonetic* and *Inscript*. MS Word has the facility to convert the Word files into web pages. We use this convert option. The web pages created by MS Word have a lot of Internet Explorer specific style sheet tags and attributes. Due to this the appearance of the Devanagari text on the web page is different in Internet Explorer on Windows and Netscape Navigator on Linux. Hence we remove those style sheet tags from the web pages by using a Java program. The files are then uploaded to the web server.

For Marathi search engine Marathi webpages were crawled from the web. The webpages crawled were in DV-TTYogesh, DV-TTSurekh font. These font codings of the webpage text was converted into ISCII coding. An inverted index is used to store the words and their attributes like whether the word appears in bold, italic, whether it appears in the title of the webpage and other attributes. MySql is used for storing the inverted index. The user interface uses J-Trans, which is a JavaScript code. The user types the query in phonetic English. When the user clicks the "search" button, the query words are looked up in the inverted index. The documents which contain all the query words are noted. Relevancy of a document is calculated according to the attributes of the query words in that document. For example: Bold or italic words are considered more important than non-bold or non-italic words. The documents are then sorted in descending order of their relevancy. The results are then displayed to the user. Thus the user gets the most relevant Marathi webpages.

15. IT Localization

IT awareness

There was a meet in October, 2001 with the media to apprise them of the technology development efforts going on in India and at the Resource Center in IIT Bombay on Indian Languages. Leading newspapers like Times of India, Maharastra Times, Loksatta, Lokamat, Sakaal sent their representatives to this meet. The systems developed in the IITB center were demonstrated and presented. The newspapers carried reports on this.

Technology solutions

IIT Bombay is providing its expertise for developing wordnets for Indian languages (please details below). We recently taught the fundamentals of wordnet building in the Indo-wordnet workshop at CIIL Mysore. Licensing agreement for transferring the lexical data, the user interface and the API for the Hindi wordnet is being worked out with advice from the Ministry.

Interaction with State Government

The interaction is mainly with the Marathi *Rajyabhasha Parishad* which is the authorized body of the Maharastra state government for developments related to the Marathi language.

Similarly interaction has been going on with the Government of Goa for the Konkani language. We conducted a very successful Language technology conference in Goa called the *International Conference on Knowledge and Language (ICUKL)* in Goa with generous support from the MICT-TDIL, in which the state of Goa participated very actively and supported the event wholeheartedly.

16. Publications

Dipali B. Choudhary, Sagar A. Tamhane, Rushikesh K. Joshi, *A Survey Of Fonts And Encodings For Indian Language Scripts*, International Conference On Multimedia And Design (ICMD 2002), Mumbai, September 2002.

Shachi Dave, Jignashu Parikh and Pushpak Bhattacharyya, *Interlingua Based English Hindi Machine Translation and Language Divergence,* to appear in Journal of Machine Translation, vol 17, 2002.

P. Bhattacharyya *Knowledge Extraction from Texts in Multilingual Contexts,* International Conference on Knowledge Engineering (IKE02), Las Vegas, USA, June 2002.

Hrishikesh Bokil and P. Bhattacharyya *Language Independent Natural Language Generation from Universal Networking Language*, Second International Symposium on Translation Support Systems, IIT Kanpur, India, March, 2002.

Dipak Narayan, Debasri Chakrabarty, Prabhakar Pande and P. Bhattacharyya *An Experience in Building the Indo WordNet- a WordNet for Hindi*, in First International Conference on Global WordNet, to be held in Mysore, India, January, 2002.

Shachi Dave and P. Bhattacharyya *Knowledge Extraction from Hindi Texts*, Journal of Institution of Electronic and Telecommunication Engineers, vol. 18, no. 4, July, 2001.

P. Bhattacharyya *Multilingual Information Processing Using Universal Networking Language*, in Indo UK Workshop on Language Engineering for South Asian Languages (LESAL, Mumbai, India, April, 2001.

17. The Team Members

Arti Sharma	arti_sharma80@yahoo.com
Ashish F. Almeida	ashishfa@cse.iitb.ac.in
Deepak Jagtap	deepak@cse.iitb.ac.in
Gajanan Krishna Ranegk	rane@cse.iitb.ac.in
Lata G Popale	lata@cse.iitb.ac.in
Jaya Saraswati	jayas@cse.iitb.ac.in
Laxmi Kashyap	laxmi_kashyap@hotmail.com
Madhura Bapat	madhurasbapat@rediffmail.com
Manish Sinha	manish@cse.iitb.ac.in
Prabhakar Pandey	pande@cse.iitb.ac.in
Roopali Nikam	roopali@cse.iitb.ac.in
Shraddha Kalele	s_mahapurush@yahoo.com
Shushant Devlekar	suSAMwa@yahoo.com
Sunil Kumar Dubey	dubey@cse.iitb.ac.in
Satish A Dethe	satishd@cse.iitb.ac.in
Vasant Zende	cukl2002@cse.iitb.ac.in
Veena Dixit	veena@cse.iitb.ac.in
M.Tech students	
Dipali B. Choudhary	dipali@cse.iitb.ac.in
Nitin Verma	nitinv@cse.iitb.ac.in
Sagar A. Tamhane	sagar@cse.iitb.ac.in
Ph.D. student	
Debasri Chakrabarti	debasri@cse.iitb.ac.in

Courtesy: Prof. Pushpak Bhattacharya
Indian Institute of Technology
Department of Computer Science & Engineering
Mumbai- 400 076
(RCILTS for Marathi & Konkani)
Tel: 00-91-22-25767718, 25722545
Extn. 5479, 25721955
E-mail : pb@cse.iitb.ernet.in