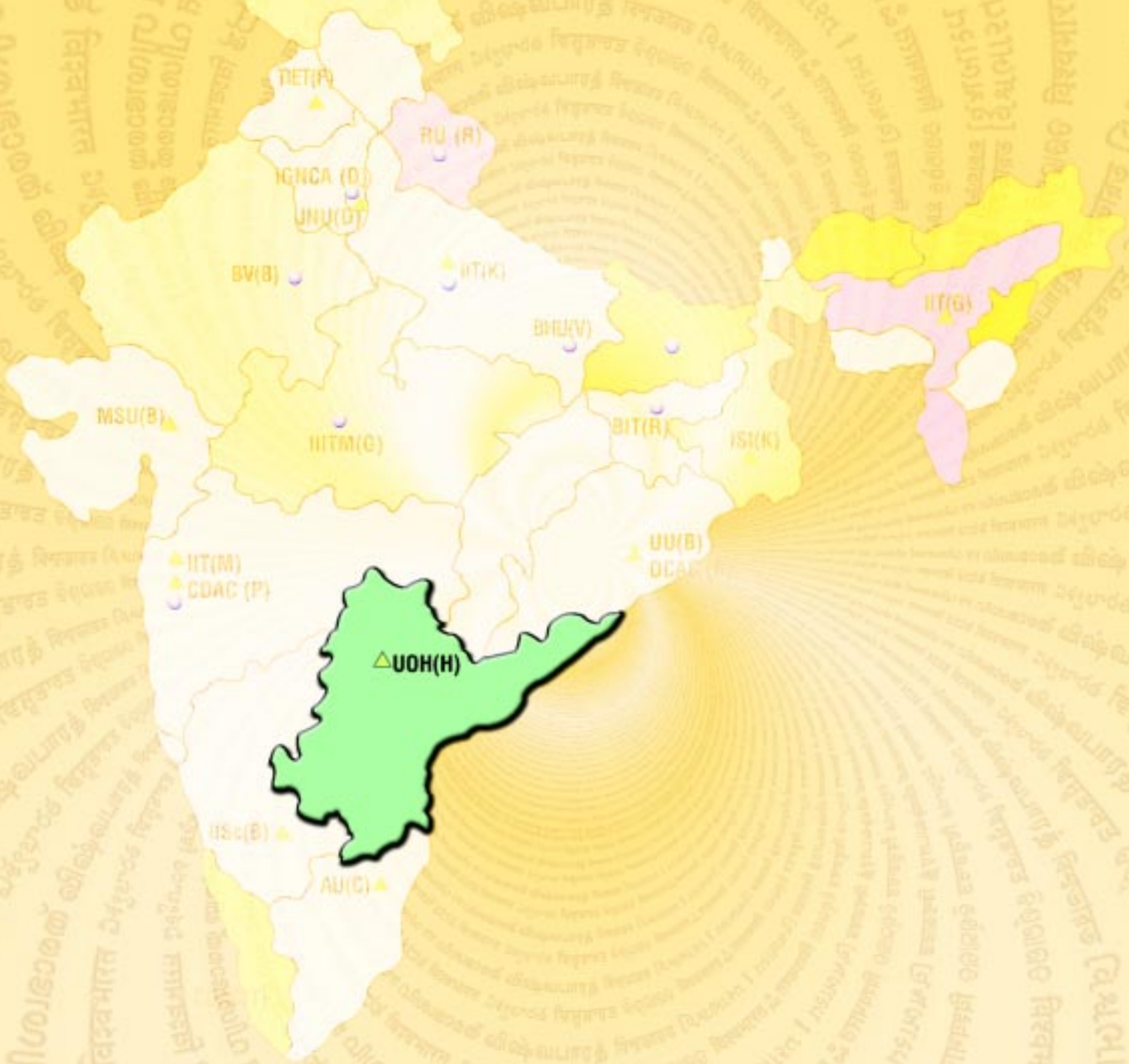
Resource Centre For
Indian Language Technology Solutions – Telugu
University of Hyderabad

Achievements



University of Hyderabad
Department of CIS, Hyderabad-500046
Tel. : 00-91-40-23100500 Extn. : 4017 E-mail : knmcs@uohyd.ernet.in
Website : http://www.languagetechnology.ac.in

RCILTS-Telugu
University of Hyderabad, Hyderabad

Introduction

University of Hyderabad is a premier institute of higher education and research in India. The University Grants Commission has selected University of Hyderabad, among four others in the country, as a "University with Potential for Excellence". The National Assessment and Accreditation Council (NAAC) has awarded the highest rating of five stars. University of Hyderabad is the only University in India to be included among the top 50 institutions in India under the "High Output - High Impact" category by The National Information System for Science and Technology (NISSAT) of the Department of Scientific and Industrial Research. University of Hyderabad has been rated the "number one" University in India in sciences by the Department of Scientific and Industrial Research.

1. Resource Centre for Indian Language Technology Solutions

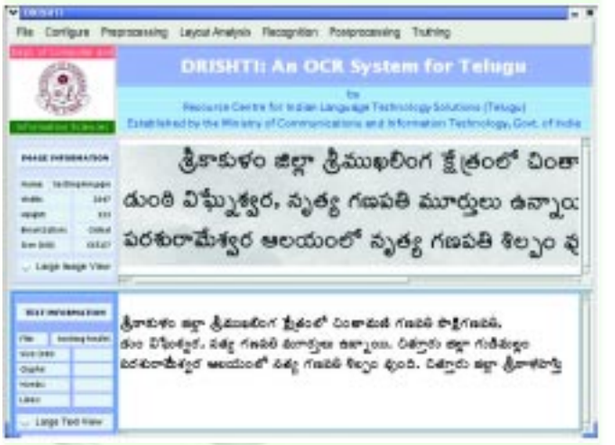
This Resource Centre for Indian Language Technology Solutions was established by the Ministry of Communications and Information Technology, Government of India, at the University of Hyderabad with a funding of nearly Rs. 100,00,000 spread over three years - April 2000 to March 2003. The project has since been extended till 30th September 2003 to enable thorough consolidation of all the work done. Two departments, eight members of the faculty, 20 to 30 students and research staff at any given point of time have put in their very best for the past three years and several products, services and knowledge bases have been developed. The core competencies, the data, tools and other resources developed here during this period will enable this team to scale new heights in future.

2. Products

2.1 DRISHTI: Optical Character Recognition (OCR)

An Optical Character Recognition (OCR) system converts a scanned image of a text document into electronic text just as if the text matter was typed-in by somebody. Scanned images are much larger in size compared to corresponding text files. The statement "A picture is worth one thousand words" is literally true here. Texts occupy less storage space and less network bandwidth when sent across a network. Converting images into texts makes it possible to edit and process the contents as normal text.

DRISHTI: An OCR System for Telugu and other Indian Languages



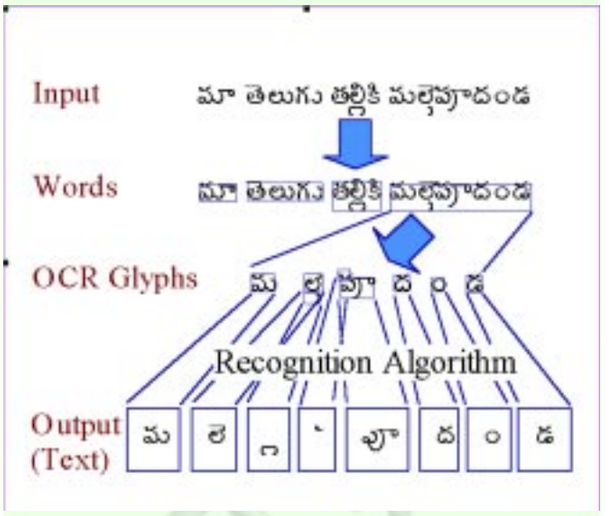
OCR systems can be used to convert available printed documents into electronic texts without typing. Since OCR engine can be run day and night on several computers parallely, we can generate large scale corpora with less time and effort. OCR engines can also be used for a variety of other applications. OCR systems have just started appearing for Indian scripts. Most of the current OCR systems for Indian languages are designed only for printed texts and perform well only on reasonably good quality documents. However, research work on hand-written document recognition is going on.

Drishti is a complete Optical Character Recognition system for Telugu language. Currently, it handles good quality documents scanned at 300 dpi with a recognition accuracy of approximately 97\%. The system is tested with a number of different fonts provided by C-DAC and Modular Infotech, and on several popular novels, laser and desktop printer generated pages and books. Preprocessing modules that separate textual and graphic blocks, handle multi-column text inputs, and skew correction are also implemented. **Drishti** is the **first comprehensive OCR system for Telugu**.

A **truthing tool** with facilities for creating ground-truth information, and to review the ground

DRISHTI: Truthing Tool

DRISHTI: How it works



truth against image data is also implemented. Such truthing tools are extremely important in objective and quick evaluation of OCR system performance. Benchmark standards were proposed in collaboration with Indian Statistical Institute (ISI) Kolkata, to enable uniform and objective evaluation and performance comparison of OCR systems and subsystems. In addition, several other useful modules and library functions that enhance or simplify adding new features are developed. Initial work is also done on touching characters that led to identifying major characteristics and issues in addressing the problem. Ours is the only major work in this area apart from that by ISI, Kolkata for Bangla characters [4]

Drishti, although designed for Telugu, has been tested with Kannada, Malayalam and Gujarati scripts, with recognition accuracies over 90\%. Our OCR technology was transferred to the resource centre for Gujarati. Work is underway in extending the system to Amharic script of Ethiopia.

2.1.1 System Overview

Drishti contains three stages: preprocessing stage, recognition stage and postprocessing stage. Binarization, separation of image regions into textual and graphical regions, multi-column detection and skew correction are the major tasks performed in preprocessing phase. Separation of text into glyphs, characters, words and lines, and recognition of individual glyphs are tasks of the recognition stage. Postprocessing comprises combining the recognized glyphs into valid characters and syllables, and spell-checking.

Preprocessing Stage

Binarization : refers to the conversion of a scanned 256-gray level image into a two-tone or binary (pure black and white) image. A binary image is appropriate for OCR work as the image document contains only two useful classes of data — the background, usually paper, and the foreground, the printed text. It is common to represent the background paper colour by white-coloured pixels and the text by black-coloured pixels. In image processing jargon, the background pixels have a value of 1 and the foreground pixels have a value of 0. Binarization has a significant impact because it provides input to every stage of an OCR system. **Drishti** provides three options — global (the default), percentile based and iterative method — to achieve desired performance on different types of scanned documents and scanners.

Skew Detection and Correction : deal with improper alignment of a document while it is scanned. The normal effect is that the lines of text or no longer horizontal but at an angle, called the skew angle. Documents with skew cause line, word and character breaking routines to fail. Skew also causes reduction in recognition accuracy. In **Drishti**, skew detection and correction are done by maximizing the variance in horizontal projection profile.

Text and Graphics Separation: refers to the process of identifying which regions of the document image contain text and which regions contain pictures and other non-text information that is not processed by

an OCR system. **Drishti** uses horizontal and vertical projection profiles for such separation as well as for many other preprocessing operations (see below). A horizontal profile is obtained by counting and plotting the number of text or black pixels in each row of the image. A vertical profile is obtained by counting the black pixels in each column of the image. Horizontal profiles show distinct peaks that correspond to lines of text and valleys that result from inter-line gaps. A line of text is revealed by a peak in the horizontal profile whose width is approximately the font size. A graphic object, in contrast, is much larger. The actual shape of the peak is also different because of higher density of black pixels in a graphics block. Thus, the profile shapes discriminate between text and graphical blocks.

Multi-column Text Detection: is done using Recursive X-Y Cuts technique proposed in [5]. It is based on recursively splitting a document into rectangular regions using vertical and horizontal projection profiles alternately. A different method that allows recognition of non-rectangular regions is also implemented but not yet included in Drishti.

The use of horizontal and vertical projection profiles for all the major preprocessing tasks minimizes system complexity and allows faster processing of documents. The preprocessing stages, except binarization, are not enabled in the basic version of Drishti, but are available as add-on options.

Recognition Stage

Line, Word, Character and Glyph Separation: is a very important task as the recognition engine processes only one glyph at a time. In **Drishti,** word and glyph separation are the key steps. Word segmentation is done using a combination of Run-Length Smearing Algorithm (RLSA) [8] and Connected-Component Labelling. Words are combined into lines using simple heuristics based on their locations. The performance of RLSA in accurately segmenting words is very high on good quality text but drops in the presence of complex layouts and tightly packed text that is sometimes seen in magazines. However, the difficulty in applying zoning techniques to Telugu because of the complex orthography requires further studies for improvement.

Words are decomposed into glyphs by running the connected component labelling algorithm again. The glyph separation is extremely accurate and very few segmentation errors were found in our experimentation.

Recognition: is based on template matching. A glyph database containing all the glyphs in the script was created from high-quality laser-printed text. Each glyph is scaled to a size of 32 X 32 pixels that forms a template for recognition and is stored in the database. When a document is scanned for OCR, the glyph obtained from the glyph separation step is scaled to the same size as the template and matched using fringe distance maps [3] against each of the templates in the database. The template with the best matching score is output as the recognized glyph.

Drishti provides several options and parameters that affect recognition. The default setup scales the glyphs using a linear scaling algorithm and matching is performed using a fringe distance map. Linear scaling is fast but suffers from problems with complex shaped glyphs at large font sizes and with small glyphs at small font sizes. Non-linear normalization was shown to improve performance[7] by selectively scaling regions of low curvature. Non-linear normalization, provided as a user option, gives better performance on Hemalata and Harshapriya fonts of C-DAC. Punctuation marks, which are easily distorted because of their small sizes, are handled separately without using template matching. Recognition accuracy is very high for punctuation marks using a location and stacking based heuristic developed for **Drishti**[2].

There are also several ways to modify the basic fringe distance measure to reflect the idiosyncrasies of the Telugu script. Experimentation was done on 18 distance measures for matching, including 6 new measures, and the best is chosen for recognition. Details on the overall recognition process and modifications for improving recognition accuracy may be found in [1,2,6,7].

Output: is written into a file. Information about the location of the glyph with respect to the text baseline, the type of the glyph, i.e., whether it is a base, maatra, vottu or punctuation glyph, and the recognized symbol code are written into the file. Also, there is a facility to output the k-best matches.

2.1.2 Postprocessing Stage

Assembling Glyphs into Syllables: is one of the most challenging tasks of **Drishti.** The complex orthography of Telugu permits glyph placement all around the base character and finding syllable boundaries is a non-trivial task. Currently, **Drishti** uses the relative positions and types output from recognition stage in conjunction with a stacking heuristic to identify syllables. The heuristic works correctly except in the case of certain large vottus with an error rate of about 2%.

Converting Glyph Codes into ISCII: is currently done by combining a simple table look-up method with type codes output by the recognition stage. Improvements are being done to the conversion code which currently does not work for a limited number of glyph/punctuation combinations and consonant clusters. Consequently, the accuracy of the ISCII output is approximately 5% - 7\% worse than that of the raw OCR output as on date. The improved conversion algorithm being developed is expected to mitigate this problem.

Spell-Checker: is recently added for detecting mistakes in the ISCII output. The current version recognizes nearly 98% of the misspelled words (false positive are 2%).

2.1.3 Salient Features

High Accuracy, Complete OCR System

Drishti is the first and currently the only complete OCR system for Telugu. Currently, several binarization algorithms (to select the best for given set of documents and print quality), text-graphics separation, multi-column layouts, skew detection and correction are available as optional plug-ins.

The raw recognition accuracy (i.e., considering the accuracy of the glyph codes) is currently 97%. The accuracy of the ISCII output generated from the Glyph Code-ISCII conversion process is currently lower because of the errors in identifying syllable boundaries and assigning ISCII byte codes. It is already improved in internal testing within the resource centre and the improved algorithms will be included in **Drishti** very shortly.

The basic system was tested by STQC unit of the Ministry of Communications and Information Technology, Government of India, and it performed with an accuracy of about 85% - 87% at the ISCII level implying a raw accuracy of 93% - 95% without using any of the preprocessing or postprocessing routines. On our scanner, **Drishti** performed with an accuracy of 96% - 97% on test documents provided at C-DAC, Noida in September 2002. On our tests on a number of documents after fine-tuning the input images for scanner contrast variations and other effects, **Drishti** consistently gives higher accuracies. Currently, preprocessing stage is being tuned to adapt for scanner differences.

A Unique Collection of Powerful Library Routines

Drishti is completely modular and implemented using a number of highly useful C-callable library functions for doing each of its tasks. The result is a main() routine that is only about 100 lines in length. The complete OCR system can be created by linking these 100 lines of code with the powerful, pre-compiled library routines. The design process permits changing the functionality of the system by calling different library function as and when needed.

Visual Truthing Tool

The visual truthing tool based on the proposed benchmark standards allows easy generation of ground truth data (including bounding boxes) from scanned documents. It is a very powerful addition to the OCR community to test and improve their systems.

2.1.4 First Workshop on Indian Language OCR Technology

The first OCR workshop for Indian scripts was organized by us. All the major groups in India working on OCR technology participated. The underlying technologies were discussed in great detail. The various systems under development were installed and tested during the workshop to identify the strengths and weaknesses of various approaches. The discussion and debate that followed helped all the centres do make further progress.

2.1.5 Conclusion

It is now possible, using the developed tools and library functions, to implement a working OCR system in less than a day. The development of a working Gujarati OCR system from an absolute scratch in under two days and subsequent transfer of technology is a testimony to the design of **Drishti.**

The result of the work on Telugu OCR system at RCILTS (Telugu) **is more than a product or a technology. It is a powerful set of research and technology tools and a platform that facilitates rapid development of OCR technologies and solutions for Telugu and other Indian languages in the future.**

2.2 Tel-Spell: Spell Checker

A Spell Checker consists of a spelling error detection system and a spelling error correction system. An ideal spell checker detects all spelling errors and does not raise false alarms for valid words. It also automatically corrects all misspelled words. Clearly, real spell checkers will not be able to match up to such as an ideal system. Real spell checkers may raise false alarms for some valid words and may also fail to catch some wrongly spelled words. Also, practical spell checkers rarely correct misspelled words automatically - they only offer a list of suggestions for the user to choose from. Some spell checkers can handle cases where an extra space has been typed in the middle of a word or when two or more words have been joined together into one. The performance of a spell checker may therefore be measured in terms of factors such as

- Percentage of False Alarm
- Percentage of Missed Detection
- Number of suggestions offered
- Whether the intended correct word is included in the list of suggestions or not
- The rank of the intended correct word in the list of suggestions
- Whether split and merged words are handled

Developing good spell checkers for Indian languages has been a challenge. No spell checkers were available at all for Telugu and a few other Indian languages. Tel-Spell is the first ever spell checker for Telugu and it includes both the spelling error detection and correction components.

2.2.1 How do spell checkers work?

Many spell checkers store a list of valid words in the language. A given word is assumed to be free of spelling errors if that word is found in this stored list. Otherwise it is presumed that the given word is wrongly spelled. Since no dictionary can be perfect, such a dictionary based approach is bound to produce some false alarms and some missed detections. A large dictionary may reduce false alarms but it is also more likely to increase missed detections since rarely used words may occur more because of errors in typing than by intention. The choice of words to be included in the dictionary is thus critical for the best overall performance of the spell checker.

When a misspelled word is detected, other words from the stored list that are similar to the given word in terms of spelling are given out as suggestions for correction. A quantitative measure of closeness of spelling such as the minimum edit distance can be used to select the words to be included in the suggestion list.

This description of spell checkers is obviously a highly over-simplified description. The number of techniques used for both detection and correction are large and varied. One may use statistical techniques such as n-grams. Sequences of characters that do not occur or occur with various frequencies are obtained from a training dataset to build a model of the language. This model can then be used to detect spelling errors. See Karen Kukich for a good survey of the various techniques.

2.2.2 Why is it difficult to build a spell checker for Telugu?

Indian languages in general and Dravidian languages in particular are characterized by an extremely rich system of morphology. Words in Dravidian languages like Telugu and Kannada are long and complex, built up from many affixes that combine with one another according to complex rules of saMdhi. For example, nilapeTTukooleekapootunnaaDaa?

which means something like “Is it true that he is finding it difficult to hold on to (his words/ something)?”

Telugu is both highly inflectional and agglutinative. Auxiliary verbs are used in various combinations to indicate complex aspects. Clitics, particles, vocatives are all part of the word. Telugu exhibits vowel harmony - vowels deep inside a verb may change due to changes at the boundaries of saMdhi. External saMdhi between whole words and compounds also occur in the language. See the references below for more on Telugu morphology. Suffice it to say that Telugu is one of

the most complex languages of the world as far as morphology is concerned.

It is therefore not practically feasible to store all forms of all words directly in a dictionary for purposes of spelling error detection and correction. At the same time building a robust morphological analyzer and generator is an extremely challenging task. Developing a good spell checker for languages such as Telugu is thus a very difficult task. No wonder no spell checkers are available for these languages to date.

2.2.3 Design of Tel-Spell

Perhaps the best and most thoroughly worked out morphological analyzer for Telugu is the one we have developed at University of Hyderabad over the past 10 years or so. The system has been tested on large scale corpora and enhancements and refinements have been going on for years. During this project, a thorough re-engineering work was taken up and a new version was developed. The new version is far simpler, more transparent, portable, well documented, and conforms to standards.

AKSHARA: Spelling Error Detection and Correction

Research work has also been taken up on developing stemming algorithms for Telugu. A pure corpus based statistical stemming algorithm has been developed. The performance of this stemmer for the spell checking application has been studied in various combinations with dictionary and morphology based approaches. See thesis by Ravi Mrutyunjaya below for reference.

A lot of empirical work and experimental studies had to be conducted to come out with the best combination of dictionary and morphology to develop the first version of the spell checker for Telugu. A 10 Million word corpus developed by us has been used to build and test the performance. The performance of the system has been found to be satisfactory both in terms of detection and correction of spelling errors. See the references below for more details. Our Telugu spell checker technology has been transfered to M/S Modular Infotech. Ltd. on a non-exclusive and non-preferential basis for commercialization. The Telugu spell checker has also been integrated into our AKSHARA advanced multi-lingual text processing system.

2.2.4 Error Pattern Analysis

Large scale spelling error data has been obtained from our 10 Million word Telugu corpus. The raw corpus as it was typed has been compared with the final version after three levels of proof reading and certification by qualified and experienced proof readers. A number of tools have been developed to prepare such a data. Quantitative study of spelling error patterns in Telugu is being conducted. This will help us to build better spell checkers in future.

2.2.5 Syllable level statistics for spell checking

Since words are large and complex and hence too numerous in Telugu and proper morphological analysis is difficult, it would be useful to perform studies at lower levels of linguistic units. The syllable level is a natural choice since writing in Indian languages is primarily syllabic in nature. n-gram models have been built at syllable level. HMM models have also been built. These models can be used to detect spelling errors and to rank the suggestions for correcting a given word. See the references below for more technical details.

2.2.6 Further work

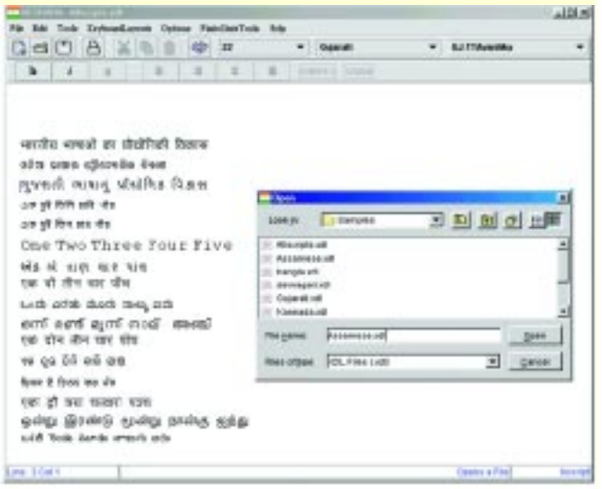
A agreement has been entered into with M/S Modular Infotech. Ltd. For further development and transfer of spell checkers for Telugu and Kannada.

2.3 AKSHARA: Advanced Multi-Lingual Text Processor

2.3.1 Why one more word processor?

A systematic study of various available word processors for Indian languages was performed in order to choose the best ones for our own use here.

AKSHARA: Advanced Multilingual Text Processor



The study indicated that none of the available software products were satisfactory. They were slow, fragile, unreliable, and they broke down when the data is large. Even very simple operations such as changing the font size cause the system to crash when the file is big. There are a number of problems and a detailed study convinced us that these are not merely implementation level bugs that can be hoped to be removed in the future versions. The basic design philosophies are faulty and short sighted. It appears that most of the commercial packages have been designed without thinking beyond type-compose-print paradigm for using computers as mere type-writers. Most commercial packages work only under Microsoft Windows platforms. The better ones are a bit too costly for most ordinary users. Adherence to standards is poor and compatibility across fonts, versions and different packages is a big problem. This motivated us to start developing our own advanced multi-lingual text processor named AKSHARA.

2.3.2 Encoding scheme

AKSHARA encodes texts in a standard character encoding scheme such as ISCII or UNICODE. Many commercial systems use font encoded pages to by-pass the character to font conversion process - itself a complex step for which there does not seem to be any fully satisfactory solution so far. These commercial systems also often use proprietary, non-standard fonts with secret encodings. The documents so created are not texts at all. There will be serious portability constraints and unfortunate as it is, often the only practical way to get back some text already in electronic form encoded in such fonts is to re-type the whole text! AKSHARA documents are always character encoded. Mapping to fonts is done only for the purposes of display and printing - all other operations are performed on character encodings. Commercial packages use proprietary and secret encodings using non-printable control characters for storing the attributes. In AKSHARA attributes are included in an open XML style markup language called Extensible Document Definition Language (XDL) developed by us. This makes it easy to convert to and from various other encoding schemes thereby ensuring highest levels of portability and platform independence.

2.3.3 Script Grammar

One of the unique features of Indian Language writing systems is a script grammar. While "akshara" or syllable is the basic unit of writing, these aksharas are actually composed of more basic elements such as vowels and consonants. Not all sequences of such basic elements are valid and the script grammar specifies legal combinations. Most commercial packages do not seem to respect the script grammar properly. A large percentage of errors in the corpora developed using other tools earlier has been found to be due to the inability of these tools to check and strictly apply the script grammar. What you see on the screen is not always what is stored in the file and hence there is no way to check and correct these mistakes by looking at the documents on the screen.

A unique feature of AKSHARA is that it understands the script grammar and warns you if you try to build ungrammatical syllables. AKSHARA has been successfully used to clean up all the corpora at CIIL, Mysore.

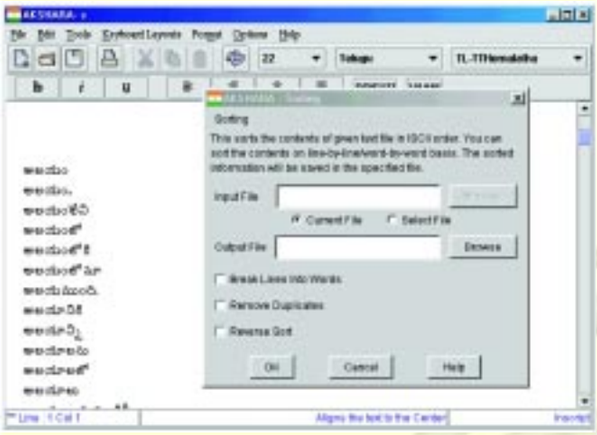
2.3.4 AKSHARA is Robust, Reliable and Platform Independent

AKSHARA is platform independent - you can use it on MS Windows, Linux and many other platforms. AKSHARA is also robust and reliable – you can comfortably work with large documents without worrying of silly restrictions such as line lengths. AKSHARA has been successfully used to develop a 10 Million word corpus of Telugu.

2.3.5 Advanced Text Processing Tools

AKSHARA is an advanced text processing tool - dictionaries, morphological analyzers, spell checkers, OCR systems, TTS systems, text processing tools including searching, sorting etc. are part of AKSHARA. Several text processing tools, Telugu spell checker and Telugu TTS have already been integrated. We would also be happy to integrate any of the other dictionaries, spell checkers, TTS etc. that other centres may have developed. Full support for Regular Expressions and Finite State Machines is being integrated.

AKSHARA: Text Processing Tools



2.3.6 AKSHARA as an email client

AKSHARA is unique in providing multilingual email sending as well as receiving facilities. All you need is a public email account somewhere. While many other systems allow you to send emails, receiving mails is not as easy. With AKSHARA there is no longer any need to depend on any third party sites on the Internet.

2.3.7 Developing Interactive web pages in Indian languages

AKSHARA also enables you to develop interactive web pages in Indian languages and English. Just use AKSHARA and any web browser to create, edit, modify and refine your web pages. These web pages will work across platforms and browsers. All your web pages will still be character encoded, not font encoded. Thus you will be really building a long lasting knowledge base. And what more, you can create interactive web pages - pages into which your users can directly type in in Indian languages. All this is made possible through our unique WILIO technology. You will find more on WILIO below. The web pages in our site www.LanguageTechnologies.ac.in have been developed using this technology. For example, you will be able to interactively search bilingual dictionaries from our website.

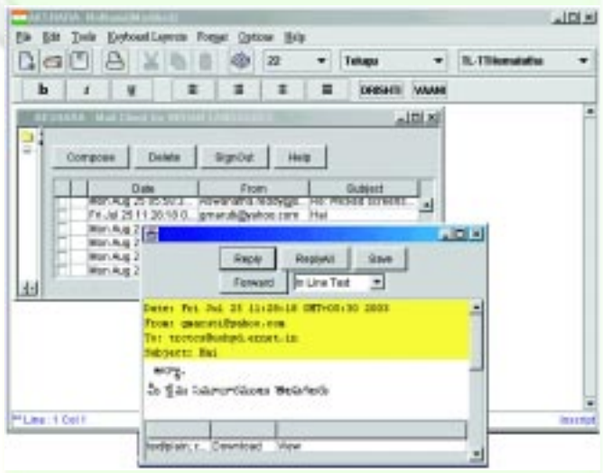
2.3.8 Availability

Wondering how much AKSHARA may cost? AKSHARA will be available for free and freely available. Let everybody have the basic Indian Language processing capabilities without restrictions.

2.4 email in Indian Languages

We have developed technology for composing, sending as well receiving emails in any combination of English and other Indian languages. Many other systems support only sending of mails, receiving mails would not be as straight forward. This technology has been integrated into our AKSHARA system. You will only need a public email account (that supports POP3 or IMAP protocols) somewhere. Unlike other technologies, here there will be no dependence on any other third party web sites. AKSHARA installed on your local machine will be your email client.

AKSHARA: email client



2.5 WILIO: Interactive Web Pages in Indian Languages

Developing web content in Indian Languages has been a challenge. None of the browsers understand the ISCII character encoding scheme. We may hope that UNICODE compatible browsers and free availability of UNICODE fonts will mitigate the situation to a

large extent in future. As on date, however, not all browsers support UNICODE, UNICODE fonts are not readily available for all Indian Languages and UNICODE scheme is itself still not completely satisfactory and revisions are going on. We explore here briefly various alternatives people have tried and present our own technology that we feel is far superior to the others.

2.5.1 Text as Pictures

The simplest way to ensure that every client sees exactly what you want him or her to see is to encode texts as images. This technology Will work irrespective of the platform and particular browser the user is using and whether he or she has the required fonts or not. However a picture is worth one thousand words (or a lot more) - both in terms of storage and network bandwidth required. Clearly, this cannot even be considered as a solution as there is no text at all.

2.5.2 Font encoded pages

We can have font encoded web pages and expect the users to have the fonts locally available on their machines. Unfortunately, fonts are not yet freely available for Indian languages and most computers in the country will have no Indian language fonts at all. Much more importantly, font encoded pages are not texts at all. There is no font encoding standard and encoding fonts in proprietary fonts is as good as encrypting them. Web sites must be viewed as knowledge bases – long lasting and easily maintainable. Unlike in the case of languages like English where character and glyph have a one to one correspondence, Indian scripts are complex and the mapping from characters to glyph sequences in a given font is a complex many to many mapping. Therefore font encoded web pages are no solution at all.

Dynamic Font Technology

One part of the problem with font encoded schemes, namely availability of fonts on the client machine, can be solved by using the so called dynamic font technology. The basic idea is to send the fonts also along with the requested documents to the client. The pages are still font encoded. Not much different from the previous method. Further,

dynamic font libraries are required - the usual fonts are not sufficient. One needs to buy tools to prepare dynamic font libraries. Otherwise you will have to depend upon some other service provider. Requests for a web pages will require connections to the service provider's web site too. Clearly, this is not a good solution.

2.5.3 Plug-in technology

Then there are plug-ins - add-on pieces of software that do the character to font conversion on the client machine. The plug-in will have to be downloaded and installed only once by a client. The pages will be character encoded. This sounds like a good solution but it has not worked well in practice. Plug-ins are add-ons to browser software and the browsers vary widely in terms of the support and the details of how they take on these add-ons. For each browser and in fact for each version of a browser, a suitable version of the plug-in will have to be developed. As new browsers keep appearing in the market, new versions of plug-ins will need to be developed too. Unless the browser developers themselves take on the responsibility of supporting Indian script standards, this technology is unlikely to be accepted as a good and permanent solution.

2.5.4 Forget the scripts, use Roman

Of course one may forget Indian scripts and encode Indian languages in the Roman script. This will provide complete immunity from platform and browser variations and font dependencies. Literate Indian language users who are not comfortable with the Indian scripts, say, non-resident Indians, will also be able to use this technology. However, this cannot be taken as a solution for Indian language support for web content!.

2.5.5 What is WILIO?

We have endeavored to develop a better technology which we call WILIO. WILIO permits standard character encoded web pages to be viewed on any browser and any operating system. The character encoded pages are received by the client and mapped to the required fonts before displaying them.

WILIO is unique in its ability to permit two-way communication. We can develop interactive web pages wherein the users can also type-in Indian language content directly into the browser. The required keyboard driver etc. are included in WILIO. Thus one may prepare lessons, ask questions, allow users to type in their responses, receive and validate the answerers and get back to the users accordingly. This will open up a whole new experience with Indian language web content. The web pages in

our site have been developed using WILIO technology. For example, look at our dictionary look up services at our website www.LangaugeTechnologies.ac.in

2.5.6 How WILIO works

WILIO works through a Java Applet. Browsers must support Java. Most browsers do. In case the Java plug-in is not installed, it will be automatically installed after getting user's confirmation. WILIO also requires that the fonts are locally available. A few fonts are available freely in any country and we hope we will not be far from such a day in India too. We are making all out efforts to make a few fonts freely available to everybody for non-commercial use. WILIO itself is fully integrated into AKSHARA and AKSHARA will be freely available and available for free.

2.5.7 Security

One of the very useful side-effects of this technology is document security - WILIO it more difficult for people to download and print the pages.

2.6 Telugu Corpus

A large, representative corpus is the first and most essential resource for language engineering research and development. A corpus is essential for building language models as well as for large scale testing and evaluation. Special emphasis was therefore laid on developing a fairly large corpus of Telugu language, the language of focus in the current project.

2.6.1 Status before the year 2000

Developing corpora for Indian languages has been more challenging than it may appear. These days most publishers use computers at some stage or the other. Why not simply compile such readily available material? While it is possible to get some material in electronic form directly from publishers, DTP centres and websites, it must be emphasized that there are no free fonts and the proprietary fonts used by various groups do not stick to any standard. While the ISCII national standard for character encoding has been around for a long time now, most of the documents continue to be developed using proprietary fonts embedded in proprietary commercial software. Thus it is not possible to simply download and add to the corpus.

Before the year 2000 corpora of only about 3 Million words were available for the major Indian languages. These corpora were developed with the support of the Ministry of Communications and Information Technology, then known as the Department of Electronics. Even these corpora were not released for many years for researchers because of legal and technical problems relating to copy rights.

2.6.2 How large is a large corpus?

Given the rich morphological nature of Indian languages, it was felt that a mere 3 Million word corpus would not be sufficient. In order to establish this fact, we conducted a growth rate analysis of the available corpus of Telugu. The corpus is split randomly into equal sized parts and type-token analysis is performed. A "type" is a particular word form and each occurrence of that word form constitutes a "token". For example, "word" is a type and there are two tokens of this type in the previous sentence. Each part of the corpus contributes a set of types and the cumulative number of types is plotted on the Y-axis against the size of the corpus (measured in terms of the cumulative number of tokens) on the X-axis. The resulting curve depicts the rate at which new types grow as the size of the corpus increases. If the curve shows signs of saturation and tends towards the horizontal, that means that most of the word forms have already been obtained and adding more corpus will not add too many new word forms. As long as the growth rate curve continues to show a high slope, it indicates that the corpus we have is insufficient and many of the possible word forms are yet to be seen even once in the corpus. Given below is the growth rate curves for all the major Indian languages for which corpora were available.

2.6.3 Dravidian Languages vs Indo-Aryan Languages

The distinction between Dravidian languages and Indo-Aryan languages is striking in this figure - there are many more word forms (types) in Dravidian languages than in the other Indian languages. While 150,000 to 200,000 word types should be giving an excellent coverage for northern languages, Dravidian languages such as Telugu spoken mainly in the southern parts of India require a much larger number of word forms. And, more importantly, the available corpus is not sufficient even to get a clear idea of how many words are there in the language. The morphology of these languages is so rich, no one so far has an idea how many different word forms are there in the language. One well known linguist has argued that each verb root in Telugu can give rise to as many as 200,000 different inflected/ derived word forms (Ref. Dr. G. Uma Maheshwara Rao, personal communication).

What this shows is that techniques which work well for Indo-Aryan languages may not be applicable to Dravidian languages. For example, it would be possible to simply list all forms of all words and use this for dictionary based spelling error detection and correction system for Hindi, Punjabi or Bengali but such an approach cannot not be expected to produce comparable performance results for say, Telugu or Kannada. It would thus is not proper to make out right comparisons of performance of language engineering products across these classes of languages. The inherent complexity of the languages must be factored in when making any comparative judgements of performance.

2.6.4 Copy Right Issues

Given the earlier experiences of groups developing corpora, we had to take every care to ensure that we did not get into copy right problems. At the same time, we realized that it is not going to be easy for us to take over the legal copy rights from the authors or publishers. Hence it was decided that we will only ask for right of electronic reproduction and rights for hosting selected works on our web-site, without asking for a legal transfer of copy rights. The original copy right holders would continue to hold their copy rights and would be free to sell, distribute or transfer their works to any other party at their will.

It was nevertheless an extraordinary convincing effort to get the best authors to part with their best works for our corpus without spending a single pie of money. It took a tremendous amount of time and effort to convince the copy right holders that what they and the country at large gains in the long run will be much more than the hypothetical loss incurred in giving us their works free of cost. A variety of strategies and tactics had to be used but at the end we have been able to obtain the rights for more than 250 of the best works of the best known writers. Add to this works for which copy rights have expired and we have a list of more than 500 books. (It will be interesting to note that the expectation of the funding body was 10 good books!).

2.6.5 The Status

A corpus of 225 books adding upto about 30,000 pages and 9.25 Million words has been completed. The corpus includes a variety of topics and categories - newspaper articles, short stories, novels, poetry, classical and modern writings etc. Each of these works has been typed in using our AKSHARA - advanced multi-lingual text processor and other such tools and subjected to two levels of thorough proof reading by qualified and experienced proof readers and finally certified free of errors. The entire corpus is encoded in the ISCII/ UNICODE character encoding standard and XML style annotation scheme is used for meta information.

The growth rate curve of types against tokens for the 12 Million word total corpus of Telugu available now was conducted recently. The curve shown below shows clearly that even this corpus is not sufficient - there is no sign of saturation and the growth rate has not reduced significantly. We still do not have even a single occurrence of most of the word forms although the number of types has already reached 20,000,000.

2.6.6 Tools

A number of tools have been developed to develop, analyze and manage large scale corpora. Some of these tools have also been given to other centres. A comprehensive tool kit is being developed.

We have also developed tools for semi-automatically decoding any unknown font with minimum effort. Using this technique, a mapping scheme can be developed to map the text strings encoded in the unknown font into an equivalent text string in a standard character encoding scheme such as ISCII or UNICODE. In fact several of the widely used fonts have been decoded and we can now add more free material such as newspaper articles at ease.

2.6.7 Plans

Now that our OCR system for Telugu and the Telugu spell checker have reached a level of performance that makes it suitable for use in content creation, we hope to be able to develop even larger corpora of Telugu very soon. A thorough investigation into the spread across genres and representativeness of the corpus is also being carried out so that further work can be fine tuned accordingly despite the non-availability per se of texts in some of the categories in Telugu language.

Plans for the future including various levels of annotation. English-Telugu parallel corpus development is also being considered.

2.7 Dictionaries, Thesauri and other Lexical Resources

Dictionaries are the most basic and essential data resource for any language. Accordingly, we have

developed a number of monolingual and bilingual dictionaries as detailed below. The dictionaries are available in the XML format for data exchange and indexed cleverly for efficient search. Look-up services are provided from our website using our unique WILIO technology for OS and browser independent deployment. All dictionaries are encoded in ISCII/ UNICODE standard character encoding schemes. Apart from dictionaries, we have also developed thesauri and more importantly, a tool by which we can develop a thesaurus of sorts for any language in just a few minutes from a suitable bilingual dictionary. Here is a summary of the dictionaries we have with us. Some of these are already being used by researchers in other centres.

2.7.1 C P Brown's English - Telugu Dictionary

Status : Completed; Size : 31,000 plus; Fields : POS, Meanings, Usage; XML? : Yes; Indexed? : Yes; Web-enabled? : Yes.

2.7.2 C P Brown's Telugu - English Dictionary

Status : Completed; Size : 31,000 plus; Fields : POS, Meanings, Usage, Etymology; XML? : WIP; Indexed? : WIP; Web-enabled? : WIP.

2.7.3 English - Telugu Dictionary suitable for Machine Aided Translation

Status : Completed; Size : 37,500 plus; Fields : POS, Meanings; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.4 Telugu - Hindi Dictionary suitable for Automatic Translation

Status : Completed; Size : 64,000 plus; Fields : POS, Paradigm Class, Meanings; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.5 English - Kannada Dictionary

Status : Completed; Size : 15,000 plus; Fields : POS, Meanings, XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.6 Basic Material for English Dictionary

Status : Completed; Size : 6,00,000 plus; Fields : POS, Frequency; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.7 English Dictionary

Status : Completed; Size : 80,000 plus; Fields : POS, Frequency; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.8 Telugu Dictionary

Status : Completed; Size : 64,000 plus; Fields : POS, Paradigm Class; XML? : Yes; Indexed? : Yes; Web-enabled? : WIP.

2.7.9 Kannada Dictionary

Status : Completed; Size : 12,000 plus; Fields : POS; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

2.7.10 Kannada Thesaurus

Status : Completed; Size : 12,000 plus; Fields : Synonyms, POS, Sense; XML? : Yes; Indexed? : Yes; Web- enabled? : WIP.

WIP: Work in progress

These dictionaries are all closely linked up with corpora, morphological analyzers and generators, spell checkers etc. Cross-validation and refinement continue on a regular basis.

2.7.11 Tools

We have also developed a number of tools for developing electronic dictionaries, for efficient indexing, searching and other such operations on electronic dictionaries, for formatting in XML or other standards, for verification and validation, for web-enabling and offering web based services etc. We would be glad to host dictionaries developed by other centres using our platform independent and secure WILIO technology.

2.7.12 Automatic generation of Thesaurus from Bilingual Dictionary

We have developed a unique tool that can generate a thesaurus of sorts for any language in just a few minutes starting form a suitable bilingual dictionary. We would be glad to offer this service to any centre that has a suitable bilingual dictionary. Thesauri being extremely useful resources yet non existent for many Indian Languages, we believe the contribution of this tool is being appreciated very well from all quarters hope this tool would be very useful.

2.7.13 Technology for hosting dictionaries on the web

We have the unique capability to place Dictionaries on the web for efficient, secure, platform and browser independent services. We would be glad to host any other dictionary developed by any other centre through our technology from our site.

2.8 Morphology

2.8.1 What is Morphology?

Morphology deals with the internal structure of words. Morphology makes it possible to treat words such as *compute, computer, computers, computing, computed, computation, computerize, computerization, computerizable and computerizability* as variants of the same root rather than as different words unrelated to one another. Morphology makes it possible to store only the root words in the dictionary and derive other variants through the rules of the morphology. It helps us to understand the meaning of related words.

2.8.2 Indian Languages exhibit rich morphology

Morphology plays a much greater role in Indian languages because our languages are highly inflectional. While the English verb *eat* gives rise to only a few variants such as eats, ate, eaten and eating, the corresponding verb in Telugu can give rise to a very large number of variants. Words in Dravidian languages like Telugu and Kannada are long and complex, built up from many affixes that combine with one another according to complex rules of saMdhi. For example, nilapeTTukooleekapootunnaaDaa? which means something like "Is it true that he is finding it difficult to hold on to (his words/something)?"

Telugu is both highly inflectional and agglutinative. Auxiliary verbs are used in various combinations to indicate complex aspects. Clitics, particles, vocatives are all part of the word. Telugu exhibits vowel harmony - Vowels deep inside a verb may change due to changes at the boundaries of saMdhi. External saMdhi between whole words and compounds also occur in the language. See the references below for more on Telugu morphology. One linguist puts the number of variants for a single Telugu verb at nearly 200,000! [G. Uma Maheshwara Rao, Personal Communication.] The exact number of different forms that a verb can take in a language like Telugu is not yet clear. The growth rate analysis described in the section on corpora clearly shows that the 12 Million corpus available at present is not sufficient to give us even a single occurrence of many possible words in the language. While Indian languages in general are morphologically richer than languages like English, Dravidian languages are a lot more complex. The 12 Million word corpus of Telugu has nearly 20,000,000 different words and there will be many more as the growth rate curve indicates. In contrast, the Indo-Aryan languages have only about 1,50,000 to 2,00,000 words forms in all. Dravidian languages including Telugu, Kannada, Malayalam and Tamil are among the most complex languages of the world and can only be placed along with languages such as Finnish and Turkish. Clearly, there is no way we can hope to list all forms of all words in a dictionary. We cannot build a spell checker, for example, by simply listing all forms of all words. Morphology is not just useful but absolutely essential.

2.8.3 Design of Telugu Morphological analyzer

Building a morphological analyzer and generator for a language like Telugu is thus a very challenging task. Perhaps the only large scale system built for Telugu is ours. Our Telugu morphological analyzer has been built, tested against corpora and refined over the past 10 years. This system uses a root word dictionary of 64,000 entries and a suffix list categorized into a number of paradigm classes. The basic methodology is to look for suffixes, remove them taking care of saMdhi changes and then cross checking with the dictionary. Inflection, derivation, external saMdhi are all handled. See the references below for more technical details. There is also a separate morphological generator that can put together the roots and affixes to construct complete word forms.

2.8.4 Design of Kannada Morphological Analyzer

We have also developed a Kannada morphological analyzer and generator using our own *Network and Process Model*. A finite state network captures in a declarative and bidirectional fashion all the affixes, their ordering and the various combinations permitted. The process component takes care of saMdhi changes when affixes are added or removed. This model makes it possible to develop a morphological analyzer, test it against a corpus and then we get a generator of comparable performance with no extra effort since the same network is used both for analysis and generation. In this model, a complete and detailed analysis is made at the level of each affix.

2.8.5 Tool for developing Morph systems for other languages

Morphological analyzers and generators for several languages including Kannada, Tamil, Oriya etc. have been built using this Network and Process model. That a good Tamil morphological analyzer and generator could be built within a week using this system is a testimony to the quality of design and

implementation of the system. See references below for more details.

As we build larger and more representative corpora, further refinements to dictionaries as well as morphological analyzers and generators will continue.

2.9 Stemmer

As the above section on Morphology shows, it is very difficult to build a high performance analyzer or generator for Dravidian languages such as Telugu. An alternative short-cut approach that can be used in practice is stemming. Here a complete and detailed morphological analysis is not performed. Instead, the affixes are removed to obtain the root. For example, the common prefix in the words compute, computer and computing is *comput* and hence all these word forms are reduced by removing the affixes to the common stem *comput.* Note that *comput* is not a valid linguistic unit at all. Yet, such stemming techniques are useful and have been used in many areas including Information Retrieval. Stemming can also be used as the second line of defence when morphology fails.

A thorough study of various stemming techniques have been conducted. Ingenious corpus based statistical stemming techniques have been developed for stemming in Telugu. Vowel changes, gemination etc. need to be taken care of in building a stemmer for Telugu. The stemmer has been compared with the full morphological analyzer and various combinations have been tried out for the purpose of spelling error detection and correction in Telugu. See the references below for more technical details.

2.10 Part of Speech Tagging

2.10.1 What is POS tagging?

A dictionary lists all possible grammatical categories for a given word. The job of a Part of Speech (POS) tagger is to identify the correct POS for a given word in context. For example, the word *thought* is a verb in the following sentence and a noun in the sentence that comes next: I have thought about it from various angles. Suddenly this strange thought came to my mind.} POS tagging may be at the level of gross grammatical categories such as verbs and nouns or, more often, at a more fine grained level of sub-categorization.

2.10.2 POS tagging techniques for English and Indian languages

In English like positional languages, the category of a word can be determined in terms of the categories of the preceding words. As such Hidden Markov Models have been widely used. There are several other techniques too. However, as far as Indian languages are concerned, many of these sequence oriented techniques are not very much applicable. Our languages are characterized by free word order and hence it does make much sense to depend so much on previous or following few words. Instead, our languages are characterized by a very rich system of morphological inflection and it is here that we get maximum information about the correct part of speech of a word. The percentage of words that occur in some inflected form rather than in the bare stem form is far more for Indian Languages as compared to English. Morphology holds the key for POS tagging of Indian languages. In fact one may even go a step further and argue that a POS-tagged corpus does not make much sense. Whenever you process some text, you will need to perform morphological analysis and the job of POS tagger will be done there too.

However, developing robust morphological analyzers for Indian languages in general and Dravidian languages in particular have been difficult challenges. The performance of any POS tagger based on morphology would be limited by the performance of the morphological analyzer itself.

2.10.3 Degree and nature of lexical ambiguities

A systematic study of the degree and nature of lexical ambiguities at the dictionary and corpus levels is being conducted. Appropriate technologies for POS tagging based on morphology are being developed. The percentage of words that occur in some inflected form rather than in the bare stem form is far more for Indian Languages as compared to English. This has serious implications for the degree and nature of lexical ambiguities in running texts.

2.10.4 HMM system for POS tagging

In order to gain deeper understanding of POS tagging for various languages including English, a Tri-tag based HMM model has also been built and tested on the SUSANNE corpus.

2.11 VIDYA: Comprehensive Toolkit for Web-Based Education

2.11.1 eLearning

eLearning helps to overcome the barriers of distance and time in learning. Thrust is on learning, not

teaching. Students can thus learn whatever they like, in whatever order they please and at a pace that is best for them. Instead of teachers, there will only be facilitators.

There are many tools for eLearning. Only some of them are comprehensive tools that provide for the entire gamut of facilities from pre-registration counseling to maintenance of the alumni database. Good ones are very costly and most educational institutions in India will not be able to afford to buy such tools. Many developments are taking place in this area but the major usage so far has been limited to corporate training in Information Technology and related areas. The focus is on adult serious learners only. Profit seems to be the main motive in many cases.

2.11.2 Web Based Education - Technology for Quality Education

Our view point is very different. Traditional education is more teacher centric whereas eLearning is learner centric. The idea is not to choose one or the other but the right combination of both. Not all students are mature and serious enough to learn on their own. Teachers are required to guide, instill confidence and to inspire. The need is to look at education at all levels in a holistic sense.

The primary issue in question is quality of education. We produce very large number of BSc's and MSc's but very few scientists. We produce very large number of BE and BTech degree holders but very few engineers. Primary school level is worse. There are several problems and not all of them can be solved through technology. The question is what is it that technology can do to ensure quality education to everybody?

How do we ensure highest quality of education at all levels without barriers of distance and time? Good teachers are not always available. Distance and time are not the only barriers. Cost and language are bigger and more serious barriers. The ultimate objective should therefore be to reach out to all interested students and offer the highest quality of education without any kinds of barriers - distance, time, cost or language. Here is where technology can bring the services of best teachers, best course materials to every student in a cost effective manner. Our technologies must be Indian language enabled.

2.11.3 VIDYA - a comprehensive suite of tools

Given this scenario, we started developing a comprehensive suite of tools for web based education. Our suite is called VIDYA. Indian languages can be supported. Interactive web content can be created using our WILIO and AKSHARA technologies.

VIDYA supports inter-student and student-teacher interaction through email, chat, discussion rooms and white boards. It encourages collaborative problem solving and group activities.

VIDYA: Comprehensive suite of Tools for Web Based Education

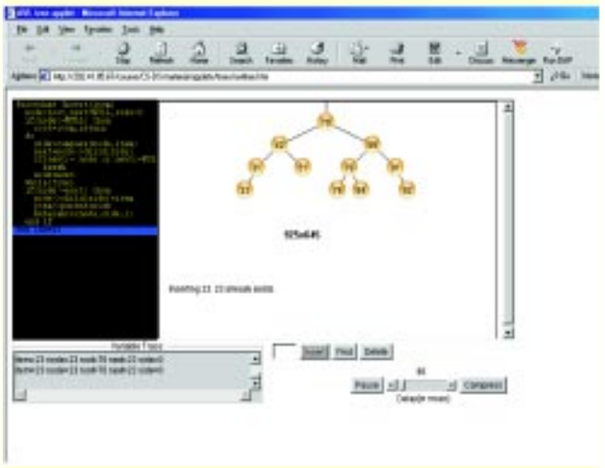
VIDYA has the unique facility to link with auxiliary servers for extra support such as for laboratories. For example, you may use VIDYA to do programming in Java, C++, C, and Perl languages without the need for these compilers on your machines. Coding, editing, compiling, executing, archieval are all supported. Similarly, science labs. And language labs. can be developed. Appropriate use of multi-media and learning by doing makes learning a pleasure and has much greater impact than reading text books and listening to class room lectures.

VIDYA support a wide variety of testing, evaluation and reporting facilities. Adaptive testing, navigation control, timing etc. are supported. Full range of question types including multiple choice, short answer and essay type questions are permitted.

2.11.4 Status and Plans

VIDYA has been installed in several centres including CIIL, Mysore. VIDYA is being regularly used in University of Hyderabad for teaching courses at MCA and MTech levels. It has also been used for offering special courses to reputed industries. A recent study has shown that it is suitable for deployment in our distance education programme.

VIDYA: Interactive Multimedia Content



VIDYA could be used by schools, colleges, universities, research laboratories etc. for regular education, continuing education, part-time courses, in-house training etc. Suitable material can be developed and shared with others so as to maximize the impact. In particular, language teaching material already developed or being developed by various centres can be linked with with VIDYA to enable various classes of language learners to get maximum benefit. We would also be glad to enter into agreements for further collaborative development of the tool itself.

2.12 Grammars and Syntactic Parsers

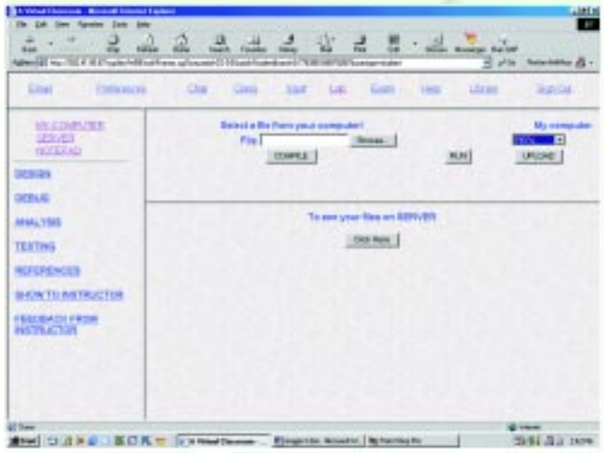
2.12.1 Computational Grammars for Indian Languages

There are no large scale computational grammars for any of the Indian languages. Computational grammars and syntactic parsers are very much required for taking Indian languages beyond the type-compose-print paradigm that is holding back the country from growing beyond using computers as some kind of type-writers. All language engineering applications including machine translation, information retrieval, information extraction, automatic categorization and automatic summarization would greatly benefit from syntactic parsers.

2.12.2 UCSG system of Syntax

The UCSG system of syntax was developed by us to place positional languages such as English on an equal footing with Indian languages that are characterized by relatively free word order. A careful study of both the Western grammar formalisms and the paaNinian approach to syntax showed that none of these would be equally suitable for positional and free word order languages and hence a new formalism had to be developed. Computational grammar and Parser has been developed for English

VIDYA: On-Line LABs



and demonstration level systems have also been developed for Telugu and Kannada. UCSG uses a combination of Finite State Machines, Context Free Grammars and Constraint Satisfaction to achieve the best overall performance. Grammars become simple and easy to write, parsers become computationally very efficient and the same basic framework works for English and other Indian languages. UCSG works from whole to part, rather than from left to right. See the references below for more details.

Further development of the UCSG English parser is going on. A much larger and more informative dictionary has been built based on the analysis of large scale corpora. Combination of linguistic and statistical models are being used to enhance the coverage and robustness of the system. Plans include the development of computational grammars for Telugu and other Indian languages.

2.12.3 Robust Partial Parsing

It is now well recognized that full syntactic parsers are difficult to build. Hence there is increased interest in robust but shallow or partial parsing. An extensive study of parsing technologies have been made and efforts are on to build a large scale robust partial parsing system for English. Efforts are also underway, in collaboration with linguists from CIIL to develop computational grammars and shallow parsing systems for Indian languages.

2.13 Machine Aided Translation

Automatic or Machine Translation is the one of the widely known applications in language engineering. It has been recognized very well that fully automatic high quality translation in open domains is difficult to achieve. Either restricted domains of applications with controlled language usage must be considered or the translation process has to be semi-automatic, the man and the machine doing what they are good at and seeking help from each other in other areas. Even with such restrictions, ensuring quality of translation is a very challenging task. Language is rich and varied in structure as well as meaning. Since the machine cannot be expected to "understand" the meaning of the given source language text in any real sense, the out of the machine can at best be good, calculated guesses. The situation is further complicated by the fact that expectations of users are very high when it comes to translation.

2.13.1 English to Kannada Machine Aided Translation System

A Machine Aided Translation system was developed here for the Government of Karnataka for translating budget speech texts from English to Kannada. English text is pre-processed and segmented into sentences. Each sentence is syntactically parsed using our UCSG English parser. The parsed sentences are translated to Kannada in a whole-to-part fashion using the bilingual English-Kannada dictionary and Kannada morphological generator developed by us. There is a powerful post processor that is tightly integrated with the dictionary, thesaurus, morphology and the translator. A full 150 page text is parsed and translated in just a couple of minutes on a desktop PC. The output is post-edited and then sent for final proof reading. Although this was a very short project with a very low budget, this project has demonstrated the merits of a well designed and well engineered product. There are several unique and powerful features in this system. This system has become a very good technology demonstrator and has inspired a lot of serious work in various directions by several groups across the country. See www.LanguageTechnologies.ac.in for more details.

The success of the MAT system for English to Kannada translation has inspired further work on dictionaries, morphology of Kannada, robust parsing and word sense disambiguation. Research work has also been taken up on a number of other specific topics such as corpus based machine learning techniques for sentence boundary identification.

The MAT2 system proposed purports to combine the best of linguistic theories, corpus based machine learning algorithms and human judgement based on world knowledge and commonsense to achieve high quality translations in a semi-automatic setup. A very useful by-product of this exercise will be a high quality POS and sense tagged, parsed, aligned, parallel corpus.

2.14 Tools

We have also developed a number of tools over the past many years for our own use and some of these tools could be useful to other groups as well. In fact some of these tools have already been given to other resource centres. Here we list some of the important tools developed by us. It may be noted that not all these tools were developed within the period of, or with the support of, this specific project.

2.14.1 Font Decoding

Unlike English, Indian scripts are syllabic in nature. The units of writing are akshara's or syllables. The total number of possible syllables is very large. Thus fonts are developed using shape units called glyphs which need to be composed to form complete syllables. The mapping from syllables to glyph sequences is complex. There is a proliferation of non-standard and proprietary fonts. In fact there is no font encoding standard as yet for Indian languages. Thus documents encoded in some unknown font is exactly like a coded message - one needs to do decode them before they start making any sense. Only documents encoded in a standard character encoding scheme such as ISCII or UNICODE can be considered as text. Font encoded documents re not texts at all. However a large number of documents are available only in font encoded forms. Some companies in fact use this as a means of achieving some degree of security for their documents.

We have developed a set of tools through which we can decode any unknown font and map it onto a standard character encoding scheme such as ISCII or UNICODE. This is a semi-automatic and iterative process. With this tool, now it possible to decode any unknown font and we hope this would encourage commercial companies to become more open and follow standards instead of pursuing myopic, proprietary, and restrictive practices.

Some of the other centres have developed direct mappings from one font to another. We believe that the best way to handle font-to-font variations is to go through a standard character encoding scheme. All documents must be encoded and processed in a character encoding scheme and mapping to fonts is to be used only for the purposes of display and printing.

2.14.2 Web Crawler for Search Engine

A web crawler searches the whole web and builds up an index of web pages that is structured and classified in a way that enables search engines to search the web efficiently. We have developed a basic web crawler through which such an index can be built. This tool can also be used to download whole web sites, for archival of web sites etc.

2.14.3 PSA: A Meta Search Engine

A search engine searches the web for the documents users request through a short query. There are several good search engines but there is no single search engine that is ideal in all cases. A Meta Search Engine accepts a user query, fires search engines, obtains the results and presents them to the user. Personal Search Assistant (PSA) is one such search engine designed and developed by us here.

PSA accepts user queries, formats them in the manner required for various search engines and fires the search engines accordingly. PSA can currently handle upto eight different search engines simultaneously. It is possible to work in the background mode so that users do not need to sit in front of the machine and wait for results. Status can be checked up at any given point of time. It is possible to monitor the network load and adjust accordingly. Results are collated, duplicates removed and stored in local database where required. Unlike a search engine a meta search engine can reside on local machines and can be customized to suit individual requirements. Some work has been done on personalization of PSA. PSA is being used within University of Hyderabad for several years now.

Plans include Indian language support for PSA. Queries can then be posed in Indian languages. Specialized versions of the web crawler can also be developed to locate web pages in Indian languages or web pages relating to India.

2.14.4 Corpus Analysis Tools

A number of tools have been developed for corpus analysis. Some of these tools are being used by other centres as well. It is planned to organize these tools in the form of tool kit so that it will become more convenient for others to use them.

2.14.5 Website development Tools

We have a whole range of tools including our AKSHARA and WILIO systems to develop interactive web sites in Indian languages. For example, our History-Society-Culture portal has nearly 500 pages of Indian language content developed and hosted through our technology. Interactive lookup services of our dictionaries also use these tools and technologies.

2.14.6 Character to Font mapping Tools

As has been noted elsewhere in this report, mapping between characters and fonts is a non-trivial process in Indian languages. Some have used table look-up method while others have used hand crafted rules. None of the systems seem to be satisfactory. Completeness, consistency, robustness, efficiency, transparency, extensibility, ease of development are some of the desirable features of such mapping systems. Given this scenario we have explored the possibility of developing good mapping systems using the Finite State Machine technology. Suitable extensions to the basic technology are proposed for the purpose.

2.14.7 Dictionary to Thesaurus Tool

Give us any bilingual dictionary and we can give you a kind of a thesaurus in a couple of minutes. Our tool can do a clever reverse indexing on any bilingual dictionary to identify closely related words for any given word. While this is not exactly a thesaurus in technical sense, the basic idea is the same - given one word, to identify other words in the language that are closely related to it in meaning. The Kannada thesaurus so developed has been demonstrated and has been judged to be very useful. Some researchers are already using this system and there are plans to do more work on Kannada and Telugu thesauri. We will be glad to develop thesauri for any other language given the suitable bilingual dictionaries.

2.14.8 Dictionary Indexing Tools

Dictionaries need to be indexed for efficient access. We have developed clever indexing schemes for efficient indexing of large dictionaries on any computer. Combinations of TRIE indexing, Hashing, B-trees, AVL Trees etc. are used.

2.14.9 Text Processing Tools

A number of text processing tools for working with word lists, dictionaries, etc. have been developed over the past many years. These tools have been found to be very useful for linguists and lexicographers too. Some of these tools have been integrated into AKSHARA.

2.14.10 Finite State Technologies Toolkit

Finite State technologies are increasingly being used in language engineering as they are simple yet very efficient. A full toolkit has been developed and tested on large scale data. Now it is possible to work with Regular Expressions, NFA, DFA etc. and perform all the usual operations without having to write any program code. It is planned to integrate these tools into AKSHARA.

3. Services and Knowledge Bases

3.1 On-Line Literature

Telugu has a very rich literary tradition dating back to around 10th Century. In today's busy world where reading habits seem to be depleting, here is an attempt to bring some of the best works of Telugu at your door steps. Making literature available on-line means making it available anywhere, anytime. You will not need to goto a book store or otherwise order and buy a book. This service is absolutely free of charge and so you spend no money.

More than 550 of the best works in Telugu have been enlisted. We have obtained, by a great deal of convincing effort, the right of electronic reproduction and web enabling these works from the respective copy right holders. Not a single pie was spent to obtain the rights though. About 225 of these books have been converted into electronic form and checked by a three stage proof reading and certification process by qualified, experienced and professional proof readers. A panoramic selection of these works will be made available from our web site through our unique WILIO technology that guarantees platform and browser independence as well as some degree of security. It is planned to add Roman transliterated versions too for the benefit of those who know Telugu language but are not very comfortable with the Telugu script.

3.2 History-Society-Culture Portal

India is a uniquely pluralistic society that is the home of many religions, traditions and cultures. Here is an attempt to bring to you nearly 500 pages of

History, Society and Culture Portal: Temples

authentic material on the society and culture of the Telugu people and indirectly, on its history too. Get to know more about temples, music, dance, folk arts and many more items. Color photographs are included.

All the pages are available through our unique WILIO technology that works across operating systems and web browsers. Roman transliterations are also provided for those who would have difficulties in reading the Telugu script. Visit www.LanguageTechnologies .ac.in

3.3 On-Line Searchable Directory

Given the importance of networking of individuals and organizations with overlapping interests, it was decided to develop an on-line searchable directory of people and organizations interested in various aspects of language technology for Telugu. More than 1200 relevant entries have been developed and cross checked. The directory is available on line. A flexible search facility has been included. Kindly visit www.LanguageTechnologies.ac.in

3.4 Character encoding standards, Roman Transliteration Schemes, Tools

There is quite a bit of confusion in the country about the exact nature of character encoding schemes, fonts, rendering engines, character to font mapping schemes etc. Many hasty decisions are sometimes being taken without a full and in depth understanding of all the issues concerned. Hence a detailed article was written about the issues involved in character encoding schemes and related issues. A version of the article has been published in the Vidyullipi journal.

3.5 Research Portal

Research and development requires time, effort, money and other resources but as far as Language Engineering in India is concerned, the most important resource required is adequate trained manpower. Language Engineering is a highly multi-disciplinary field - it borrows from such diverse disciplines as Linguistics, Psychology, Philosophy,

Logic, Artificial Intelligence, Cognitive Science, Computer Science, Mathematics, Statistics and Physics. Clearly, there are no experts who know all these areas very well. This multi-disciplinary nature of the subject makes it so much more difficult to create quality training materials, books etc. that are understood by people across so many disciplines. Indian languages are also characterized by certain unique features compounding the difficulties in developing trained researchers and developers.

With this in mind, we have developed research portals in selected areas of Language Engineering. In one place, you will find basic and introductory material, tutorial and survey papers, classified and structured collection of large number of relevant research papers, pointers to people, departments, institutions, conferences and other regular events and so on. This would substantially reduce the time and effort needed by newcomers to these research areas. We could also develop these portals for news, discussion and debate, collaborative development etc.

3.6 VAANI: A Text to Speech system for Telugu

Text-to-Speech systems convert given text into speech form. Thanks to the maturity of the techniques and availability of required tools, it is now possible to develop minimal TTS systems in months. VAANI, our TTS system for Telugu, is one such attempt. Di-phone segmentation based approach has been used. Phonemes in the language are identified and for each ordered pair of phonemes, called di-phones, example words are recorded in speech form. From this raw data, di-phones are segmented using available tools. It would then be possible to process this raw data further and develop a database. To produce speech from any given text, the text is initially parsed. certain pre-processing steps are essential to handle numerals, homographs etc. Then the text is segmented into diphones and the corresponding speech units are concatenated to produce speech output. Segmenting at di-phone boundaries gives better continuity since the variations due to co-articulation effects are least in the middle of a phoneme as compared to its ends. Since the number of phonemes in a language is usually a small and closed set, this technology also leads to unlimited vocabulary TTS technology. VAANI is thus an unlimited vocabulary, open domain TTS system for Telugu. The system has been tested for intelligibility both directly and across telephone lines. Prosodic features such as duration, pitch and intonation can be added to make the sounds more natural. A substantial amount of research has also been done on other competing technologies and we would be able to deliver high quality unrestricted TTS systems in future. VAANI has already been into our AKSHARA - Advanced Multi-lingual Text Processing system.

3.7 Manpower Development

Trained manpower is a critical issue in language engineering in India. More than 100 students and staff have worked in the project for periods ranging from 3 months to three years on various research and development activities. In the process they have obtained significant theoretical knowledge as well as practical skills in language technologies.

The LEC-2002 International Conference on Language engineering included a full day of tutorials by distinguished experts from India and abroad. These tutorials were free for students. Several hundred students could benefit from these.

The first IL-OCR workshop organized by us here attracted many interested students and researchers from across the country. The detailed presentations, demos. and discussions were very helpful.

The research portals being set up here will be of value to beginners in language technology research.

A number of articles, technical reports and research papers have either been published or are being prepared for wider dissemination of the ideas, techniques and technologies. Our website is intended to serve a similar purpose and is being enhanced and updated accordingly.

A text book on Natural Language Processing with specific emphasis and examples from Indian languages is planned.

With this it should be possible now to organize training programmes on specific topics to identified target groups. Suitable course material can also be developed specifically for the purpose.

4. Epilogue

4.1 Strengths and Opportunities

Our strength lies in our core research competence. Our team has experts from linguistics, statistics, computer science, artificial intelligence and cognitive science. Each member of the team has a very rich and varied experience. What binds us all is the common research competence. The tools, techniques and algorithms used in bio-informatics, image processing, speech recognition and language technology have

many things in common. Our team emphasizes this common core.

We had a one semester long seminar cum discussion series on Markov Models. This semester we have semester long seminar cum discussion series on feature extraction and feature selection. See annexure for details. Thorough investigation of Word Sense Disambiguation and Shallow Parsing techniques are going on. This in depth understanding of the technologies will enable to do world class research in various areas and develop quality products and services.

We also have striven to develop large scale linguistic data resources that are essential for further research and development. Our centre is perhaps unique in having developed a 10 million word corpora and more than half a dozen dictionaries of significant size and quality. Large and representative data and the right kinds of tools will enable us to move much faster.

We have also striven to strike a good balance between the pure research and publications on the one hand and product development and technology transfer to meet the needs the society on the other hand. Many of our results are yet to be published and we hope to bring out several publications soon. We have also striven to strike a good balance between long term and short term goals. Two years ago when we started off almost from scratch on our OCR system, other experts felt that as far as content creation is concerned there is nothing better than simply typing in texts. Today our OCR is one of the successful ones in the country and combined with our spell checker, we will soon be able to develop much large corpora for Telugu and other Indian languages than would have been possible otherwise.

We have struggled in this first phase of development, to overcome the teething problems and look for long lasting and permanent solutions rather than hop onto short sighted and immediate solutions. This approach has payed off and with the data and tools we have with us now, we hope to be able to move much faster.

Our future efforts will be in more focused areas of language and speech engineering. We look forward to meaningful collaboration with other leaders in the world for research as well as technology development.

4.2 Outreach

The LEC-2002 Language Engineering conference organized by us was quite successful and we hope to be able to organize quality international conferences in future as well. We also organized the first ever OCR workshop for Indian scripts. This was a trend setter of sorts. We laid bare our OCR system in full detail and others followed. It was perhaps for the first time that different research groups got to know about each other's approach is such great detail. The systems developed by various groups were demonstrated and tested publicly. The discussion and debate that followed helped all the centres to make further progress. We have been the Indian coordinators for a Indo-French research network in computational linguistics and we plan to work more closely with other groups across the world.

Within the country, we have been maintaining close technical links with a number of organizations including the Society for Computer Applications in Indian Languages, Computer Literacy House, Computer Vignannam, AP Press Academy, CMC, Telugu University, Telugu Academy, etc. apart from commercial companies such as C-DAC and Modular Infotech.

A large number of very distinguished visitors have visited our labs. and offered their appreciation as well as very valuable suggestions.

We have been organizing seminars on a regular basis. So far more than 25 seminars have been organized. Some of the distinguished speakers include Prof. Gerard Huet, Dr. Mark Pedersen and Prof. Rajat Moona.

5. Publications

1. Chakravarthy Bhagvati, Atul Negi, and B.Chandrasekhar. Weighted Fringe Distances for Improving Accuracy of a Template Matching Telugu OCR System. In Proc. of IEEE TENCON2003, Bangalore, 2003.
2. Chakravarthy Bhagvati, T.Ravi, S.M.Kumar, and Atul Negi. Developing High Accuracy OCR Systems for Telugu and other Indian Scripts. In Proc. of Language Engineering Conference, Pages 18-23, Hyderabad, 2003. IEEE computer society Press.
3. R.L.Brown. The fringe distance measure: an easily calculated image distance measure with recognition results comparable to Gaussian blurring. IEEE Trans. System man and Cybernetics, 24(1): 111-116, 1994.
4. U. Garain and B.B. Chaudhuri. Segmentation of Touching Characters in Printed Devnagari

and Bangla Scripts Using Fuzzy Multifactorial analysis. In Proc. of Int. Conf. on Document Analysis and Recognition. IEEE Comp. Soc. Press, Los Alamitos (CA), USA, 2001.

5. G.Nagy, S.Seth, and M.Vishwanathan.A prototype document image analysis system for technical journals. Computer, 25(7), 1992.
6. Atul Negi, Chakravarthy Bhagvati and B.Krishna. an OCR system for Telugu. In Proc. Int. Conf. on Document Analysis and Recognition. IEEE Comp. Soc. Press, Los Alamitos (CA), USA, 2001.
7. Atul Negi, Chakravarthy Bhagvati, and V.V. Suresh Kumar. Non-linear Normalization to Improve Telugu OCR. In Proc. of Indo-European Conf. on Multilingual Communication Technologies, pages 45-57, Tata McGraw Hill Book Co., New Delhi, 2002.
8. K.Wong, R. Casey, and F.Wahl. Document analysis system. IBM J. Research and Development, 26(6), 1982.
9. K. Narayana Murthy, B B Chaudhuri (Eds), LEC-2002: Language Engineering Conference, IEEE Computer Society Press, 2003
10. K V K Kalpana Reddy and Jahnavi A, "Text to Speech system for Telugu", MCA thesis
11. K. Narayana Murthy, "UNICODE: Issues in Standardization of Character Encoding Schemes", Vidyullipi, April 2002
12. K. Narayana Murthy, Nandakumar Hegde, "Some Issues relating to a Common Script for Indian Languages", International Conference on Indian Writing Systems and nagari Script, 6-7 february 1999, Delhi University, Delhi
13. K Narayana Murthy, "An Indexing Technique for Efficient Retrieval fro Large Dictionaries", National Conference on Information Technology NCIT-97, 21-23 December 1997, Bhubaneswar
14. P R Kaushik, K Narayana Murthy, "Personal Search Assistant: A Configurable Meta Search Engine", Proceedings of the AusWeb99 – The Fifth Australian World Wide Web Conference, 17-20 April 1999, Southern Cross University, Australia
15. P R Kaushik, "PSA - A Meta Search Engine for World Wide Web Searching", M. Tech. thesis, Department of Computer and Information Sciences, University of Hyderabad, 1998
16. P Naga Samba Siva Rao, "Enhancements to the PSA model for Meta Search Engines", M.Tech. thesis, Department of Computer and Information Sciences, University of Hyderabad, 1999
17. K. Narayana Murthy, "MAT2: Enhanced Machine Aided Translation System", STRANSS 2002 Symposiu on Translation Support Systems, 15-17 March 2002, Indian Institute of Technology, Kanpur
18. K. Narayana Murthy, "MAT" A Machine Assisted Translation System", Fifth Natural Language Pacific Rim Symposium, NLPRS-99, 5-7 November 1999, Beijing, China
19. K. Narayana Murthy, "UCSG and Machine Aided Translation from English to Kannada", Indo-French Symposium on Natural Language Processing, University of Hyderabad, 21-26 March 1997
20. Kasina Vamsi Krishna, "Word Sense Disambiguation: A Study", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
21. D Madhusudhana Rao, V V Raghuram, "A Generic Approach to Sentence Segmentation", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
22. K. Narayana Murthy, "Universal Clause Structure Grammar", PhD thesis, Department of Computer and Information Sciences, University of Hyderabad 1996
23. K. Narayana Murthy, A Sivasankara Reddy, "Universal Clause Structure Grammar", Special issue on Natural Language Processing and Machine Learning, Computer Science and Informatics, Vol. 27, No. 1 March 1997 pp 26-38

24. K. Narayana Murthy, "Universal Clause Structure Grammar and the Syntax of Relatively Free Word Order Languages", South Asia Language Review, Vol VII, No. 1 Jan. 1997 pp 47-64

25. K. Narayana Murthy, "Parsing Telugu in the UCSG Formalism", Indian Congress on Knowledge and Language, January 1996, Mysore, Vol 2 pp 1-6

26. Sreekanth D, "A Statistical Syntactic Disambiguation Tool", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 1998.

27. Ashish Gupta, "Improvements to the UCSG English Parser", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001.

28. D. Srinivasa Rao and M. Suresh Babu, "Design of a Web Based Education Tool", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2000

29. B. P. V. Prasad and A. Rambabu, "Design of VIRAT: A ProtoPUBLISHER Virtual Authoring Tool for Web-Based Education", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003

30. P. Uday Bhaskar and R. Krishna Kishore, "A Tool for Web Based Education", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003

31. Rajesh V Patankar and Tammana Dilip, "VIRAT: An Authoring Tool for Publishing on the Web", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001

32. T. S. Vivek and J. Reddeppa Reddy, "Web Based Education Tool – Online Testing & Evaluation Module, MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001

33. P. Siva Rama Krishna and G. Nagachandra Sekhar, "Web Based Education Tool (Instructor Module)", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001

34. T. Ramesh and T. Phani Raju, "Web Based Education Tool (Software Lab Module)", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001

35. G. Murali Krishna and P. Ravi Kumar, "Generic Framework for Developing Web-Lab Experiments", M. Tech.thesis, Department of Computer and Information Sciences, University of Hyderabad, 2002

36. G. Vamsidhar, "White Board Application - A GUI Tool for Web Based Education Tool", M.Tech.thesis, Department of Computer and Information Sciences, University of Hyderabad, 2002

37. M. Narsimhulu, "An Enhanced Architecture for Web Based Education", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 2002

38. K Vasuprada, K. Narayana Murthy, "Part-of-Speech Tagging using a Tri-Tag HMM Model", Second National Symposium on Quantitative Linguistics, 28-29 Feb. 2000, Indian Statistical Institute, Kolkata

39. K. Vasuprada, "Part of Speech Tagging and Syntactic Disambiguation using stochastic parsing techniques", M.Tech. thesis, Department of Computer and Information Sciences, University of Hyderabad, 1999

40. Ravi Mrutyunjaya, "Corpus-Based Stemming Algorithm for Indian Languages", M.Tech.thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003

41. K. Narayana Murthy, "Theories and Techniques in Computational Morphology", Proc. of the National Seminar on Word Structure of Dravidian Languages, 26-28 November 2001, Dravidian University, Kuppam pp 365-375

42. K. Narayana Murthy, "A Network and Process Model for Morphological Analysis/ Generation", Second International Conference

on South Asian Languages ICOSAL-2, 9-11 Jan. 1999, Punjabi University, Patiala

43. Anil Kumar, "Morphological Analysis of Telugu Words", M.Tech.thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
44. K. Narayana Murthy, "Electronic Dictionaries and Computational Tools", Linguistics Today, Vol 1, No. 1, July 1997 pp 34-50
45. A Sivasankara Reddy, K. Narayana Murthy, Vasudev Varma, "Object Oriented Multipurpose lexicon", International Journal of Communication, Vol. 6, No 1 and 2, Jan-Dec 1996 pp 69-84
46. K. Narayana Murthy, "An Indexing Technique for Efficient retrieval from Large Dictionaries", National Conference on Information Technology NCIT-97, 21-23 December 1997, Bhubaneshwar
47. Vasudev Varma, K. Narayana Murthy, A Sivasankara Reddy, "electronic Dictionaries: A Model Architecture", National workshop on Electronic Dictionaries for second language learners, Bharatiar University, Coimbatore, 29 Sept. to 7 Oct. 1993
48. Surya Kiran Mamidi, "Developing keyboard and Display drivers for Indian Language Editors, MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001
49. . Maruti Kumar and Y. N. V. Ganesh Kumar, "Editor for Indian Languages", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2001
50. V S S Neelima, "Enhancements to the AKSHARA Text Processing System", M. Sc. Computer Science thesis, K R R Vignan degree and PG College for Women, Hyderabad, 2003
51. K. Narayana Murthy, "UNICODE: Issues in Standardization of Character Encoding Schemes", Vidyullipi, April 2002
52. Karen Kukich, "Techniques for Automatically Correcting Words in Text", ACM Computing Reviews, Vol 24. No. 4 December 1992
53. Narayana Murthy, "Issues in the Design of a Spell Checker for Morphologically Rich Languages", 3rd International Conference on South Asian Languages - ICOSAL-3", 4-6 January 2001, University of Hyderabad.
54. K. S. RajyaShree and K. Narayana Murthy, "Statistical Spell Checking", Indo-UK Workshop on LESAL - Language Engineering in South Asian Languages", 23-24 April 2001, National Centre for Software Technology, Mumbai
55. K. Narayana Murthy, "Markov Models of Syllables for Spell Checking", Statistical Technqiues in Language Processing, 11-12 June 2001, central Institute of Indian Languages, Mysore
56. Naresh Pidatal and Dhanjay Kumar Singh, "Spelling error Detection and Correction", MCA thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
57. Anil Kumar, "Morphological Analysis of Telugu Words", M.Tech.thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
58. Ravi Mrutyunjaya, "Corpus-Based Stemming Algorithm for Indian Languages", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, 2003
59. K. Narayana Murthy, "Design of a Spelling Error Detection and Correction System for Telugu", National Seminar on Language Technology Tools: Implementation for Telugu, September 17-19, University of Hyderabad
60. Thesis in the area of Indian Language Technologies Thejavath Ramdas Naik and Manoj Kumar Pradhan, "Off-Line printed Devanagari page Layout Analysis and Script Recognition", MCA Thesis, Department of Computer and Information Sciences, University of Hyderabad, June 2000.
61. Buska Krishna, "Design and Implementation of a Telugu Script Recognition System", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, December 2000.

62. L. K. Prashant Kumar, "A Study of Binarization methods for Optical Character Recognition", M.Sc.(Tech) Thesis, School of Physics, University of Hyderabad, December 2000.

63. D. Vijaya Bhaskar, "Preliminary Study of GOCR to implement OCR for Telugu", MCA Thesis, Department of Computer and Information Sciences, University of Hyderabad, 1998-2001.

64. Ajay Koul, "A Study of Gradient Based Feature Extraction for Telugu OCR", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, December 2001.

65. G. Raghava Kumar, "A Study of Nearest Neighbour Computation Techniques and their applications", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, December 2001.

66. Y. Pradeep Kumar and G. Koteswar Rao, " A Tool-kit for Document Image Processing", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, January 2002.

67. V. V. Suresh Kumar, "Non-Linear Normalization Techniques to improve OCR", M.Tech. Thesis, Department of Computer and Information Sciences, University of Hyderabad, January 2002.

68. Tanuk Ravi and S. Mahesh Kumar, "Enhancements and Redesign of DRISHTI, An OCR System for Telugu", MCA Thesis, Department of Computer and Information Sciences, University of Hyderabad, June 2002.

69. Anand Kumar and Arafath Pasha, "Software for OCR", BE Thesis, Department of Computer Science, 2001-2002.

70. D. Bhavani, C. Madhuri and S. Shesha Phani, "GUI for an OCR System", B. Tech. Thesis, Department of Computer Science, March 2002,

71. I. N. Lekha, " An Experimental Evaluation of Fringe Distance Measures for Malayalam Printed Text OCR", M.Tech.(CT) Thesis, School of Physics, University of Hyderabad, May 2003.

6. The Team Members

Dr. K. Narayana Murthy, Dr. Arun Agarwal, Dr. B. Chakravarthy, Dr. S. Bapi Raju, Dr. Atul Negi, Dr. G. Uma Maheshwara Rao, Dr. P. Mohanthy, Dr. P. R. Dadegoankar, Raman Pillai Rajesh, K. Rajini Reddy, P. Muttaiah, K. Naga Sabita, Ganesh Raju, Maruthi, Naresh, Suresh Kumar, Anand, Kwaja Sirajuddin, B.Navatha, P. RamaKrishna Prasad, O. Bhaskar, Vaishnavi, K. Balaji Rambabu, Suresh Kumar, P. Siva Ramakrishna, M. Surya Kiran, T. Ramesh, Ravi Raj Singh, Venu Gopal

Courtesy: Prof. K. Narayan Murthy
University of Hyderabad, Dept. of CIS,
Hyderabad -500046 (RCILTS for Telugu)
Tel: 00-91-40-23100500, 23100518
Extn. 4017, 23010374
E-mail: knmcs@uohyd.ernet.in

Editorial Comment : Because of very large number of publications by the Resource Centre & constraint of space we could not include the publication details here. For getting the publications please contact Prof. K. Narayan Murthy