Computer Processing of Kannada Language

K. Narayana Murthy
Department of Computer and Information Sciences,
University of Hyderabad, Hyderabad, 500 046,
email: knmcs@uohyd.ernet.in

Abstract

This paper is about language processing in general and about computer processing of Kannada language in particular. In the first part, a brief introduction to all the major aspects of language processing in general is given. Then we briefly sketch the work on Kannada being done in various organizations. We conclude with certain general remarks on a possible plan of action for further developing Kannada processing.

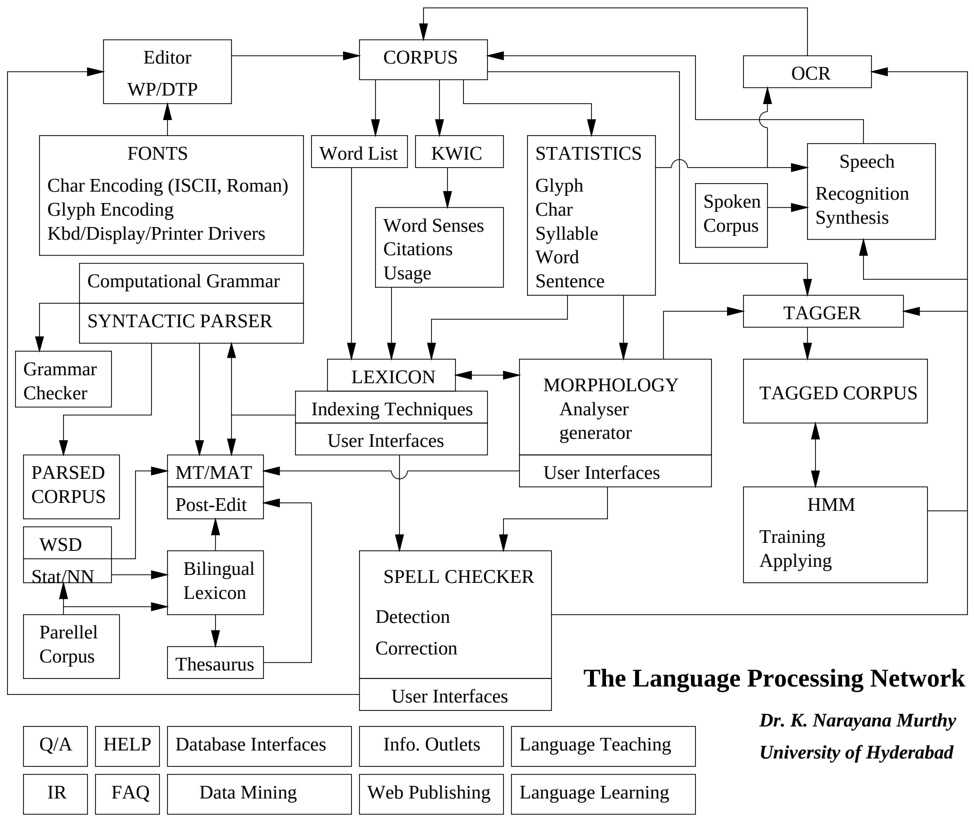
1 Introduction:

Human beings are the only animals capable of using language for communication. A language is a reflection of the world we live in, and as such, is a large and extremely complex object. People use their common sense and intelligence to use language effectively in real life. Computers have no common sense and attempts to make computers *intelligent* have not taken them anywhere close to human levels. Thus natural language communication with computers has remained a distant dream.

Nonetheless, computers have certain unique abilities which make them ideally suited for a variety of language processing. Computers can store large volumes of correlated or un-correlated data and retrieve the relevant parts with unmatched speed and accuracy. Try remembering the full text of all the books in your library! Computers can also carry out long and complex sequences of simple instructions mechanically with unfailing accuracy and great speed. They never get bored or tired. They don't make mistakes. It is simple for a computer to list down all the words used in all the books in your library, keep the counts of their occurrences, and give a frequency-wise sorted list of the distinct word forms.

It has been clearly demonstrated that any technical work in languages that is done entirely by manual methods, depending solely on one's intuitions or limited studies, is very likely to be incomplete and imperfect. Computers are essential tools required for almost any kind of language processing [3]. In the next section we briefly outline the whole gamut of language processing tasks, showing their inter-relationships and the role of computers in each task.

2 The Language Processing Network:



The above figure is a summary of all the major tasks in language processing. The arrows indicate the primary inter-relationships. The following paragraphs explain the main points:

The basis for almost all aspects of language processing is a corpus - a large, carefully selected collection of representative texts in electronic form. A corpus can be created by three main methods - i) entering the texts through the computer keyboard using an editor, word processor or a Desk Top Publishing (DTP) software, ii) by scanning the printed or hand written texts using a scanner and then using an Optical Character Recognition (OCR) software to transform the scanned image into an editable text, and iii) by reading in texts into a microphone and using a Speech-to-Text software to transform the read sounds into editable text. In all cases, texts so created need to carefully checked manually for errors.

From the corpus we can do a Type-Token analysis and get all the distinct word forms and their frequency of occurrence. A careful study of these words can lead to the selection of entries for a Lexicon or a dictionary for a specified purpose. Note that the Type-Token analyser usually gives us complete word forms - inflected and/or derived, and thus morphological analysis has to be made to extract the roots. Using a Key-Word-In-Context (KWIC) tool, we can list down all the sentences in the corpus in which a given key word occurs. This helps us to study the different senses in which the particular word is used in different contexts. This is very useful for adding meanings or definitions for an entry in the dictionary. Human intuitions often miss out on some of the possibilities - a study based on a large and representative corpus is essential. By the same token citations, usage notes, examples, cross-references, etc. are best obtained from the corpus. Add indexing techniques for efficient retrieval [9] and appropriate user interfaces and you get an electronic dictionary designed for the specific purpose [1].

Electronic dictionaries are not merely electronic forms of conventional dictionaries. Apart from functioning as a fast page turner for us, an electronic dictionary provides additional functionality which if often very difficult to obtain from a conventional printed dictionary. For example, you may extract all verbs, all bisyllabic words, all words ending with a particular suffix, all words which have a noun as well as a verb sense, all the head words in the dictionary sorted in the length order and many more such classes. Also, an electronic dictionary may be required for use by the computer. The is a great deal of difference between dictionaries meant for people and dictionaries meant for computer systems. See [4] for more on electronic dictionaries.

Morphology, the study of internal structure of words, is an essential component in any language processing task. More so in the case of Indian Languages and even more so in the case of Dravidian languages like Kannada that exhibit a very rich and complex system of morphology. A morphological analyser takes a complete word form as input and produces an analysis of its structure in terms of the constituent morphemes. A generator or synthesizer, on the other hand, either generates a specified word form from its constituents or produces a complete paradigm - a systematic listing of all the inflected and/or derived forms of the given word. A morphological analyser is required to extract the roots from the word list obtained by a Type-Token analysis on the corpus. The morphology component in turn depends on the Lexicon for cross checking the roots for its own processing.

Statistical analysis of the corpus at character, glyph, syllable, word or sentence level is extremely useful for almost all aspects of language processing. It clearly shows us what is basic and frequent, and what is rare, idiosyncratic, or not really representative of the language. It helps to focus on the most important things and prune unimportant ones [12,13].

Once we have good dictionaries and morphological analysers and generators, we can integrate them into a spell checker. A spell checker has a Detection component and a Correction component. In the detection phase, misspelt words are identified. One way to do this is to cross check each word in the given text with the electronic dictionary. Words found in the dictionary will be taken as valid words (and so a 'note' typed as 'not' by mistake is not even detected) and any word that is not in the dictionary is taken to be due to a spelling mistake (so valid words not found in the electronic dictionary will be flagged off as errors). An alternative method is to rely on n-gram statistics to flag words with very low probability as errors. Combinations of linguistics and statistical techniques can also be used. In the correction phase, a spell checker tries to offer suggested alternatives for the user to choose from. In the dictionary based technique, all the valid words in the dictionary which are 'close' to the given word in terms of spelling are extracted and shown. A quantitative and precise definition of this notion of 'closeness' such as Minimum Edit Distance is employed. Of course the user may choose to ignore the suggestions and either ask for the word to be simply accepted or added to his personal dictionary. A good spell checker is a great value added to any editor or word processing software. Spell checking techniques are also useful for OCR and speech recognition [14].

A dictionary gives all possible Parts of Speech (POS) for a word, without worrying about the various contexts in which that word can be used. A statistical POS tagger is a software that relies on statistical techniques like Hidden Markov Models (HMM) for tagging words in a text with the correct part of speech and other grammatical features considering the context in which the word is used in that sentence [11]. Linguistic techniques based on morphology can also be used for tagging text. A tagged corpus is very useful for many purposes such as training an HMM. A good tagger can in turn aid in the creation of a tagged corpus in the first place. HMMs have also been used extensively for speech and OCR.

A spoken corpus is extremely useful in speech processing work. Apart from text-to-speech and speech-to-text systems, there are other tasks such as speaker identification or even language or dialect identification.

A computational grammar enables syntactic parsing systems to be built. A computational grammar needs to be a lot more detailed and a lot more precise than the traditional grammars meant for people. Traditional grammars assume that the users already know the language and focus more on exceptions, idiosyncrasies and special features. Any of a number grammar formalisms available can be employed for expressing the computational grammar [2]. A parse helps to create a parsed corpus which in turn can be used for context based spell checking, tagging, OCR, speech recognition, grammar checker etc.

A Machine Translation (MT) system, or more realistically, a Machine Aided Translation (MAT) system can be built using bilingual lexicons, parsers, morphological analysers and generators etc. It has been clearly established that fully automatic high quality translations in open domains is not feasible. High quality automatic translation is possible only in some highly limited domains such as weather forecasting reports. Human involvement in terms of pre-editing, post-editing or interactive translation is essential. However, a MAT system can produce crude translations at incredible speed and along with the other on-line tools, significantly reduce the time and costs involved in translation. A parallel corpus in two languages is very useful for building MAT systems.

With all this in place, we can think of a number of applications such as question answering systems, natural language based help systems, information retrieval systems, natural language query systems for databases, information kiosks, systems that store frequently asked questions and corresponding answers, text abstracting and summarization systems, data mining systems, tools for language learning and language teaching, and so on.

We thus see that the language processing is a closely inter-connected network of data, activities, tasks and software. Each component depends on several others for its own development and use. At the same time, it also contributes for the development and deployment of several other components. Boot-strapping is the only way to get started off and keep developing on all relevant fronts.

3 A brief sketch of work done in Kannada:

Kannada has a very rich and vibrant literature. The highest number of Jnanapeeth awards have been received by Kannada writers and we are all really proud of this.

However, when it comes to scientific and technical study of the language, the situation is pathetic. How many words are there in Kannada? Which of these are the most frequently used words? What is the vocabulary of a 1st standard student, 5th standard student or 10th standard student? How many different inflected and/or derived forms exist for a verb in Kannada? Which combinations occur in consonant clusters and with what frequency? Very little technical work has been done in Kannada compared to most other major languages of our country. Compared to what needs to be done, what is already done and what is currently being done is very small. The following few paragraphs attempt to briefly sketch the work being done in different institutions. This is not intended to be a comprehensive and thorough survey, but only an indicator of where we stand today.

The Central Institute of Indian Languages (CIIL), Mysore, has developed a corpus of about three million words under a project funded by the Department of Electronics (now named Ministry of Information Technology, MIT). Although the corpus itself was developed many years ago, it has been made available to selected institutions only recently. A preliminary tagged version is also available. Several related small tools have also been developed.

Academy of Sanskrit Research (ASR), Melukote, has been doing some work in Kannada, apart from Sanskrit. Dictionaries, word form generators etc. have been built.

The department of Computer and Information Sciences, University of Hyderabad, has developed a Machine Aided Translation (MAT) system for the Government of Karnataka [7]. This MAT system, meant for translating English texts into Kannada, includes a Kannada dictionary, an English-Kannada bilingual dictionary, a kind of thesaurus, morphological analyser and generator for Kannada [8] and a set of related tools for translation, post-editing, etc. A sample Kannada parser has also been developed. The MAT system is based on the Universal Clause Structure Grammar (UCSG) and associated parsing systems developed by the author [5, 6, 10]. A number of tools for text processing in general have been developed too. Under a new project funded by the Ministry of I.T., the University of Hyderabad is working on a number of topics with focus on Telugu language. Since Kannada and Telugu are very similar in orthographic, syntactic and semantic levels, a lot of this work could be carried over and applied to Kannada also.

Indian Institute of Technology (Kanpur) has developed Kannada-Hindi dictionary for use in their Anusaraka system for translating from one Indian Language to another.

Several other institutions and individuals have been working and developing a variety of tools such as word processors, fonts, etc.

Thus it can seen that the work already done or being done in Kannada is very small compared to what all needs to be done.

4 Plan of Action:

It is clear from the above that no single individual or organization will be able to take up and complete all the required work in a reasonable amount of time. The aim should be not just to catch up with others but to excel and lead the way. Needless to say team work is the only solution. A concerted and well coordinated effort involving several organizations, departments and individuals is required. Thus the very first task is to identify such team members.

The language processing network shows the need for experts from many and varied disciplines to come together and work with a common mission. We need expertise in languages, linguistics (lexicography, phonology, morphology, syntax, etc.), acoustics, phonetics, computer scientists and engineers, software engineers, Internet and Web specialists, mathematicians and statisticians, neural network experts, etc.

The main focus should be on the basic data - plain, tagged, parsed, and spoken corpora, text compression tools, efficient storage and indexing schemes for working with large corpora, KWIC indexing, statistical analyses, etc. These are difficult, time consuming and challenging tasks where the best available resources must be deployed.

Work on lexicon, morphology, spell checking, parsing, translation etc. can be taken up by organizations that have strong linguistic as well as computational expertise.

Specialized topics such as OCR and Speech processing can be taken up by centres with expertise and skill in these relevant areas.

Application areas can be taken up by any individual or organization with special interest and background in those areas.

Collaboration and cooperation can take several forms starting from student projects, PhD programmes, individual collaborative research work or institutional collaboration at project level. University of Hyderabad is willing and eager to collaborate, cooperate, participate or coordinate with all interested individuals and institutions at all levels.

References

1. Bran Boguraev, Ted Briscoe (Eds), Computational Lexicography for Natural Language Processing, Longman, 1989.

2. Eugene Charniak, "Statistical Language Learning", MIT Press, 1993

3. Gerald Salton, "Automatic Text Processing", Addison-Wesley, 1989.

4. K. Narayana Murthy, "Electronic Dictionaries and Computational Tools", Linguistics Today, Vol 1, No 1, July 1997 pp 34-50

5. K. Narayana Murthy, A. Sivasankara Reddy "Universal Clause Structure Grammar" Computer Science and Informatics, Vol 27, No 1, March 1997 Special Issue on Natural Language Processing and Machine Learning, pp 26-38.

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

7. K. Narayana Murthy, "MAT: A Machine Assisted Translation System", Proceedings of the Fifth Natural Language Pacific Rim Symposium, NLPRS-99, 5-7 November, Beijing, China

8. K. Narayana Murthy, "A Network and Process Model for Morphological Analysis/Generation", To appear in the Proceedings of the Second International Conference on South Asian Languages, 9-11 January 1999, Punjabi University, Patiala, India

9. K. Narayana Murthy "An Indexing Technique for Efficient Retrieval from Large Dictionaries" To appear in the Proceedings of National Conference on Information Technology NCIT-97, 21-23 December 1997, Bhubaneswar

10. K. Narayana Murthy "Parsing Telugu in the UCSG Formalism" Proceedings of the Indian Congress on Knowledge and Language, vol 2, pp 1-16 11-17 January 1996, Central Institute of Indian Languages, Mysore

11. K. Vasuprada, K. Narayana Murthy, ”Part-of-Speech Tagging using a Tritag Model”, To appear in the Proceedings of the Second National Symposium on Quantitative Linguistics, 28-29 February 2000, Indian Statistical Institute, Calcutta

12. Marie-Francine Moens, “Automatic Indexing and Abstracting of Document Texts”, Kluwer Academic Publishers, 2000.

13. Michael P. Oakes, ”Statistics for Corpus Linguistics”, Edinburgh University Press, 1998

14. S N Srihari, ”Computer Text Recognition and Error Correction”, IEEE Computer Society Press, 1984.