

CREATING LANGUAGE RESOURCES FOR NLP IN INDIAN LANGUAGES

Rajeev Sangal
Dipti Misra Sharma
Language Technologies Research Centre
International Institute of Information Technology
Hyderabad, India
{sangal,dipti}@iiit.net

1. BACKGROUND

Non-availability of lexical resources in the electronic form is a major bottleneck for anyone working in the field of NLP on Indian languages. Some measures were taken to alleviate this bottleneck in a quick and efficient way. It was felt that if the development of these resources is linked with an example application then it can act as a test bed for the developing resources and provide constant feedback. Moreover, immediate results in terms of a performing system also enthruses the developers for such time consuming jobs. It was decided to take up the building of a machine translation system as an example application, which would also serve as a vehicle for building lexical resources.

2. DEVELOPING LEXICAL RESOURCES

The following lexical resources were built or are being built as part of a planned effort:

- a) Electronic dictionary (Shabdanjali English - Hindi dictionary)
- b) Transfer lexicon and grammar (TransLexGram)
- c) Part-of-Speech tagged corpora.

These are described below.

2.1 SHABDANJALI ELECTRONIC DICTIONARY:

As a first step in this direction a collaborative effort was undertaken to develop a bilingual electronic dictionary in the free software model. The interesting aspect of this effort was that the work was carried out by school children, teachers and others. People in about 8 cities were involved in the exercise. The school teachers participated, to some extent, in correcting and refining the work.

The development of the dictionary resource took advantage of the bilingual ability of the contributors. The contributors provided the basic data:

a) A number of Hindi equivalents required to cover various senses of the English lexical item in various contexts.

b) An English example sentence for every Hindi equivalent.

(The developed resource is now available as an "open resource" under General Public License. (<http://www.iiit.net/ltrc>))

2.2. TRANSLEXGRAM

The next level of work, based on the above resource, was to carry out the following tasks (TransLexGram) for the existing English-Hindi dictionary (called Shabdaanjali)

- Providing translation of the English sentences into Hindi.
- Creating a parallel grammar from English to Hindi.

2.2.1 AN EXAMPLE

a) Original entry from Shabdaanjali for English verb "charge".

The entry contains the English word (HEADWORD), its grammatical category (LEX_CAT), its Hindi sense in a given context (MEANING_H), an example sentence in English to exemplify the sense context (ENG_EXP) and a unique sense id (SID) for every sense of the given headword.

```
- SID::charge%v%1
- HEADWORD::charge
- LEX_CAT::V
- MEANING_H::1:: aaropa_lagaanaa
- ENG_EXP::The suspect was charged with
              murdering his friend.
- TR_NAT::
- FRAME_E::
- FRAME_I::
-----
- MEANING_H::2::uttejitaakaranaa / honaa
- ENG_EXP::The speaker charged up the
              crowd with his inflammatory remarks.
- TR_NAT::
- FRAME_E::
- FRAME_I::
-----
```

b) Output after translating every example sentence of English into Hindi (the first task mentioned above)

```
- SID::charge%v%1
- HEADWORD::charge
```

- LEX_CAT::V
 - MEANING_H::1:: aaropa_lagaanaa
 - ENG_EXP::The suspect was charged with
 murdering his friend.
 - TR_NAT:: saMdigdha vyakti para apane
 mitrako mArane kA *Aropa lagAyA gayA*
 - FRAME_E::
 - FRAME_I::

 - MEANING_H::2::uttejita_karanaa / honaa
 - ENG_EXP::The speaker charged up the
 crowd with his inflammatory remarks.
 - TR_NAT:: vaktA ne apane bhaDZakAnevAle
 kathanoM se bhIDZa ko uttejita
 kara diyA.
 - TR_NAT::2:: vaktA ke bhaDZakAnevAle
 kathanoM se bhIDZa uttejita Ho gaI.
 - FRAME_E::
 - FRAME_I::

This output becomes the input to the next task of providing parallel verb frames for every sense of the verb.

c) Output after the verb frames

- SID::charge%v%1
 - HEADWORD::charge
 - LEX_CAT::V
 - MEANING_H::1:: aaropa_lagaanaa
 - ENG_EXP::The suspect was charged with
 murdering his friend.
 - TR_NAT:: saMdigdha vyakti para apane
 mitrako mArane kA *Aropa lagAyA gayA*
 - FRAME_E:: A charged C with B
 - FRAME_I:: A ne C para B kA Aropa~lagAyA.

 - MEANING_H::2::uttejita_karanaa / honaa
 - ENG_EXP::The speaker charged up the
 crowd with his inflammatory remarks.
 - TR_NAT:: vaktA ne apane bhaDZakAnevAle
 kathanoM se bhIDZa ko uttejita
 kara diyA.
 - TR_NAT::2:: vaktA ke bhaDZakAnevAle
 kathanoM se bhIDZa uttejita Ho gaI.
 - FRAME_E::A charges B with C

- FRAME_I::A B ko C se uttejita~karatA~HE

Thus a translation of lexicon and grammar (TransLexGram) is created from English to Hindi.

2.2. CREATING TRANSLEXGRAM FOR ENGLISH-INDIAN LANGUAGES (OTHER THAN HINDI)

The other task was to develop a TransLexGram for other Indian languages. For doing so, following tasks were undertaken

- Providing other language meanings to English words in correspondence with the Hindi senses of the given word.
- Translate every English example sentence of a given sense into the chosen Indian language.
- Prepare parallel frames for verbs

TransLexGram is now ready in - atleast four Indian languages - English-Hindi, English-Marathi, English-Telugu and English-Tamil

The work was done as a collaborative effort for which various Institutes came forward to participate.

3 DEVELOPING TAGGED CORPORA

Need for tagged corpora for any machine learning techniques is another requirement for Indian languages. Therefore, it was decided to develop following annotated corpora for Indian languages

- POS tagged corpora
- Morphosyntactically tagged corpora.

3.1. SOME ISSUES

3.1.1. GENERALISED TAGSETS FOR INDIAN LANGUAGES

Though most Indian languages are fairly close grammatically, still there are variations which need to be addressed while coming up with a common tagset for them.

3.1.2. A STANDARD TAGSET

The first issue which required addressing, before starting off a mammoth task of developing a sizable annotated corpora of Indian languages, was the issue of standards for various tagsets. The most commonly used Penn tagsets were studied for POS tagging. It was decided to keep the tags as close to Penn POS tags. However, modifications were to be made depending on Indian languages' requirements. A POS tagset(see Appendix) has now been developed and currently the work is on to develop the POS annotated corpora.

3.1.3. LINGUISTIC ANALYSIS - COARSE OR FINE

Another issue which required considerable deliberation was whether to keep the tagset coarse at the level of linguistic analysis or go to a finer level of analysis. For example whether to keep a tag N for all noun forms or to make the finer distinction of NO (noun oblique) and ND (noun direct). One can go to a still finer level of NOS (noun oblique singular), NOP(noun oblique plural) etc. The issue here was that the number of tags goes up in consonance with finer analysis. Consequently, in trying to achieve a finer linguistic analysis, one ends up with a higher number of tags. This

leads to the requirement of a larger annotated corpora for training any tool.

Therefore, a balance was drawn in arriving at the tagsets.

Work is also on for developing morphosyntactic tags for Indian languages. Initial effort in this direction is already made in AnnCorra Treebank (ref).

4. AN EXAMPLE APPLICATION - SHAKTI MACHINE TRANSLATION SYSTEM

As discussed at the beginning of this paper, it is important to develop an application which acts as a test bed and gives instant feedback for all the resources which are being developed. Shakti machine translation system is such an application.

4.1 INTRODUCTION:

Shakti machine translation system has been designed to produce machine translation systems for new languages rapidly. It is already producing output from English to five different languages (four Indian languages, and one African language). System is so designed that many of the benefits of improvement to the system flow automatically to outputs in all the languages.

The Shakti machine translation kit is also designed to take ready made sub-systems either as black boxes or as open source software and incorporate them into itself. The simplicity of the overall architecture makes it easy to do so. Available English analysis packages have been extensively adapted by the Shakti machine translation system for translating from English into any of the languages.

The Shakti machine translation system has a number of innovative design principles which are described below.

4.2. SYSTEM ORGANIZATION PRINCIPLES

A number of system organization principles have been used which have led to the rapid development of the system. While the principles by themselves might not appear to be new, their application to machine translation in this manner is unique.

4.2.1 MODULARITY

The system consists of a large number of modules, each one of which typically does a small logical task. This allows the overall machine translation task to be broken up into a large number of small sub-tasks each of which can be accomplished separately. Currently the system has 69 different modules, about 9 of which are for analyzing the source language (English), 24 for bilingual tasks such as substituting target language roots and reordering etc., and the remaining for generating target language.

4.2.2 SIMPLICITY OF ORGANIZATION

The overall system architecture is kept extremely simple. All modules operate on a stream of data whose format is fixed. This format is called as Shakti standard format (SSF). All modules read data in Shakti Standard Format (SSF) and generate output in the same format.

If a module is successful in its task, it adds a new attribute or analysis to the output stream (in the same SSF). Thus, even though the format is fixed, it is extensible in terms of attributes or analyses.

This approach allows other ready made packages (such as, English POS tagger, chunker, and parser) to be used readily. In order to interface such packages to the Shakti kit, all that is required is to convert the output of such packages to SSF, and the rest of the modules continue to operate on the output seamlessly.

This approach follows the dictum: "Simplify globally, complicate locally." However, since the number of modules is large and each local module does a small job, the local complexity remains under tight control for most of the modules.

4.2.3 DESIGNED TO DEAL WITH FAILURE

NLP analysis modules are known to have limited coverage. They are not always able to produce an output. They fail to produce output either because of limits of the best known algorithms or incompleteness of data or rules. For example, a sentential parser might fail to produce a parse either because it does not know how to deal with a construction or because a dictionary entry is missing. Similarly, a chunker or part of speech tagger might fail, at times, to produce an analysis. The system must be designed to deal with failure at every level. This is facilitated by a common representation for the outputs of POS tagger, chunker and parser (all in SSF). The downstream modules continue to operate on the data stream, albeit less effectively, when a more detailed analysis is not available.

(If all modules were to fail, a default rule of no-reordering and dictionary lookup would still be applied.)

Similarly, if the word sense disambiguation (WSD) module fails to identify the sense of a word in the input sentence, it does not put in the feature ('tgt sense') for the word. This only means that the module which substitutes the target language root from the available equivalents from dictionary, will use a default rule because the detailed WSD was not successful (say, due to lack of training data).

The SSF is designed to represent partial information, routinely. Appropriate modules know what to do when their desired information is available and use defaults when it is not available. In fact, for many

modules, there are several levels at which they operate, depending on availability of information corresponding to that level. Each level represents a graceful degradation of output quality.

The above flexibility is obtained by using two kinds of representation: constituent level representation and feature-structure level representation. The former is used to store phrase level analysis (and partial parse etc) and the latter for outputs of all kinds of other tasks such as WSD, TAM computation, case computation, dependency relations, etc.

4.2.4 TRANSPARENCY FOR DEVELOPERS

An extremely important characteristic for the successful development of a complex software such as a machine translation system is to lay bare the input and output produced by every module. This transparency becomes even more important in a research environment where new ideas are constantly being tried with a high turn over of student developers.

In the Shakti system, unprecedented transparency is achieved by inventing a highly readable textual notation for the SSF, and requiring every module to produce output in it. In fact, the output is not just for human consumption, but is used by the next module in the data stream as its input. This ensures that no part of the resulting analysis is left hidden in some global variables; all analysis is represented in readable SSF (otherwise it is not processed at all by the subsequent modules).

Experience has shown that this methodology has made debugging as well as the development of the system convenient for programmers as well as linguists alike. In case, an output is not as expected, one can quickly find out what went wrong (that is, which module did not function as expected). In fact, linguists are using the system quite effectively to debug their linguistic data with ease.

4.3. TECHNICAL DESIGN PRINCIPLES

In the Shakti machine translation system we have taken a number of decisions regarding the nature of analyses or generation to be performed. The approach is summarised here.

4.3.1 HYBRID APPROACH

Shakti system combines rule-based approach with statistical approach. The representation (SSF) is designed to keep both kinds of information. The rules are mostly linguistic in nature, and the statistical approach tries to infer or use linguistic information. For example, statistical POS tagger tries to infer linguistic (part-of-speech) tags, whereas WSD module uses grammatical relation together with statistics to do word sense disambiguation.

However, the rule-based component is not always linguistic. Some modules also use semantic information. For example, in domain filters for named entity recognition relevant to a domain, such a markup is done based on rules which are not always linguistic. Similarly, there is a module which accepts translations of multi-word expressions given by bilinguals who are not linguists.

Although the system accommodates multiple approaches, it has a strong bias for linguistic analysis which serves as the backbone. Statistical and other approaches are interpreted in linguistic terms wherever possible. This allows generalizations to be performed which are not possible otherwise. Only when no linguistic interpretation is possible, is the information kept as it is. Thus, although the system has a strong linguistic bias, it is capable of dealing with non-linguistic information or patterns in their own right.

4.3.2 CONSTITUENT STRUCTURE WITH DEPENDENCY RELATIONS

The system takes advantage of phrase structure or constituency structure at the bottom most level. It represents the analysis in terms of chunks i.e., non-recursive noun phrases, prepositional phrases, verb groups, etc. Relations among these chunks are shown using dependency relations. This leads to two advantages.

- (a) Compactness and transparency of representation, and
- (b) Flexibility in approach or method to be used in identifying relations.

The latter follows because a module with "expertise" in identifying a particular kind of relation can be called without waiting to build the entire constituent structure. For example, a statistics based PP-attachment module can do its job given the chunks and some appropriate relations between them without waiting or trying to build the entire parse tree for the sentence first. It is much more difficult to modularize the building of constituent structure.

4.3.3 TRANSFER SYSTEM WITH INTER-LINGUAL PROPERTIES

A clear separation is made between the modules which analyse the source language and those which generate the target language. System is so designed that a module for generating a target language can be easily plugged in. As much analysis of the source language as possible is pulled into the source language analysis modules.

Still, it is a transfer system and not an interlingua-based system. Bilingual dictionaries and reordering rules etc., are still required for each pair of languages. However, for a group of closely related languages the bilingual lexical resources may be "aligned" or inter-linked. For example, there are sense-aligned dictionaries from English to Hindi, English to Telugu and into other Indian languages. Some of these dictionaries were "born aligned", in other words, at the time they were being created one or the other was used as a base. For example, a given English word and its senses enunciated in one dictionary, are used to create aligned entries in other languages. The initial task of creating the first dictionary is typically very detailed and hard. The subsequent tasks of giving equivalents in other languages is a lot easier. If there are already existing bilingual resources (e.g., dictionaries), they can be semi-automatically aligned later.

Aligned lexical resources have other advantages as well besides ease of their creation. For example, if the dictionaries are sense aligned, word sense disambiguation (WSD) needs to be done only once at the source language level. The generators for each of the target languages need only to pick out their corresponding equivalent words or expressions as given in their respective dictionaries. Thus, the major task of developing WSD module needs to be done only once common to the group of languages with aligned resources.

This is not to say that all the elements in language resources can be fully aligned. For example, there might be special senses of a word in a language not present in others. Such special cases may have to be processed specially (without the advantage of interlingual terms, but also without the overhead of interlingua).

What this approach suggests is that without waiting for a "universal" interlingua to be developed, one can get several advantages (though not all) of having an interlingua. The most important thing is that the pitfalls of interlingua are avoided (namely, there is no need to arrive at interlingual representation that is completely independent of the source language and thus avoiding the increase of errors in the machine analysis due to greater detail in analysis needed).

Thus, although it is a transfer based system, parts of it behave like an inter-lingua based system particularly for a group of related or similar languages and for a set of phenomena.

4.3.4 NAMED ENTITY RECOGNITION AND INFORMATION EXTRACTION

Named entities occur quite frequently in text. Domain specific modules for recognizing and marking up expressions that name entities in a text, may be run first. Such marked up expressions are then treated as a unit, and sentence analysis continued. All this is possible because of the modularity of the system.

4.3.5 SUB-DIVIDING PROBLEMS ALONG STANDARD LINES

The extreme modularity of the system allows problems to be broken up along well understood lines when convenient, and in novel ways when necessary. Thus, PP-attachment can be processed independent of the rule-based parser as it is a well understood problem for which statistical techniques work well. On the other hand, identification of phrasal verbs (i.e., verb-particle pairs) is not handled adequately anywhere. Most existing parsers do not handle them well. Statistical techniques can be tried and the resulting phrasal verb groups can be passed over to the rule-based parser.

4.4. REPRESENTATION OF ANALYSIS

As discussed earlier, a common format (namely, Shakti Standard Format) has been used for representing the analyses by the modules. Each module reads its input analysis in this format, and adds its own analysis to the output in the same format.

Two properties are satisfied by SSF. First it is text based notation making it human readable. Its advantages have already been discussed: transparency, debugging and ease of development of modules. Second, it is extensible, in which new attributes can be added at will.

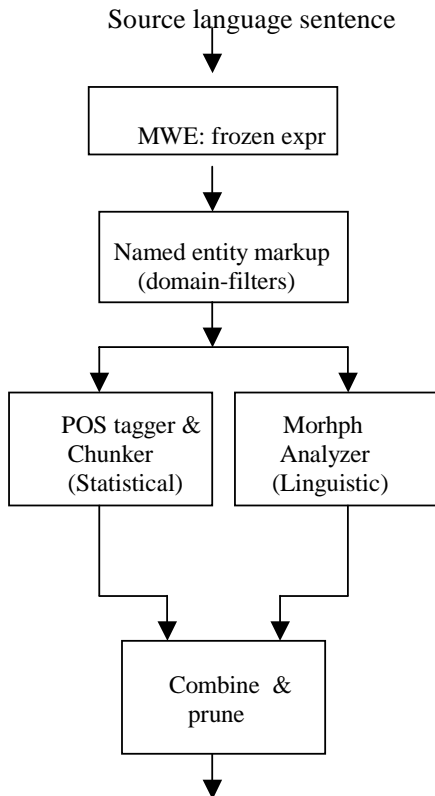
The representation is based on constituent structure with feature structures and statistical tags. A given sentence is represented as a series of chunks (elements of shallow parse). Each chunk has a category tag with words having part-of-speech category tag. Here is an example

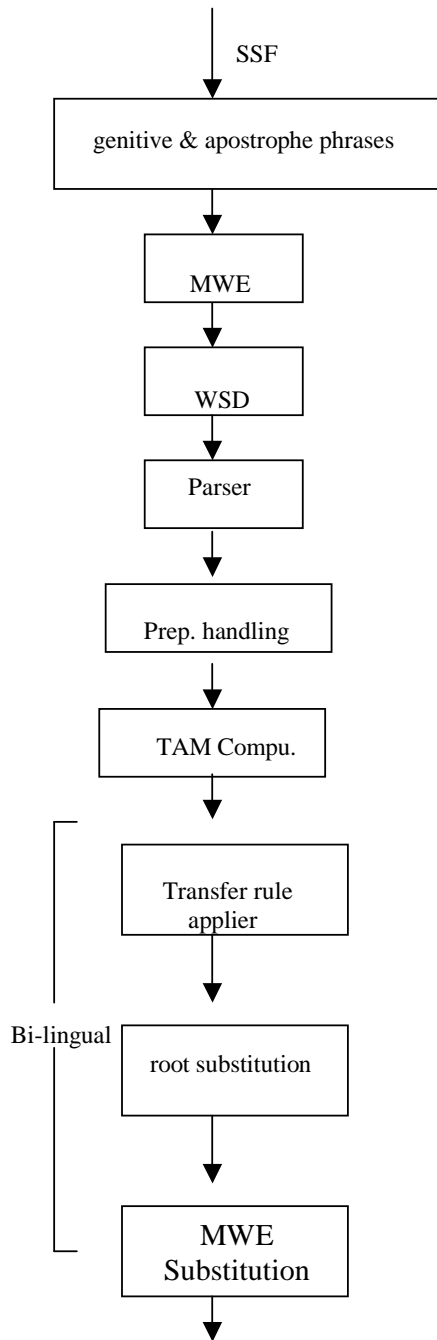
```

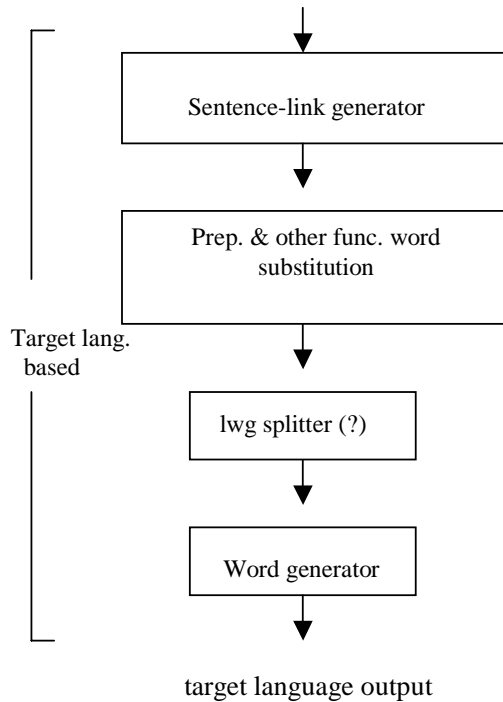
-----
SENTENCE
1  ((      NP
1.1 girls  NNS  < af=girl,n,m,p,3,0,,> )
1.2 )
2  (      VG
2.1 are    VBP  < af=are,n,m,s,3,0,,> |< af=be,v,m,p,3,0,,>
2.2 singing VBG  < af=sing,v,m,s,3,0,./aspect=PROG> ))
3  in     IN   < af=in,p,m,s,3,0,,> |< af=in,D,m,s,3,0,,> |< af=in,p,m,s,3,0,,>
4  ((      NP
4.1 their  PRP$ < af=he,det,m,p,3,0,./case=GEN>
4.2 rooms NNS  < af=room,n,m,p,3,0,,> )
4.3 }
END SENTENCE
-----

```

4.5. OVERVIEW OF THE SYSTEM







CONCLUSION:

The transparent architecture of Shakti MT system allows for rapid testing of many of the lexical resources being built for Indian languages currently. Such a testing gives rapid (even immediate) feedback to linguists and other developers. It also permits a large number of programmers and developers of algorithms to build the system and conduct experiments in machine learning. All this happens in the context of an application which enthralls the teams with visible progress because of their efforts.

ACKNOWLEDGEMENT:

Satyam Computers has financially supported the activity. Ministry of Communications and Information Technology, New Delhi has also provided funds to carry out various tasks for developing language resources in Indian languages.

REFERENCES:

1. Penn Tree Bank <http://www.cis.upenn.edu/~treebank/home.html>
2. Shabdaanjali: English - Hindi e-Dictionary ver.0.2, 2000 <http://www.iiit.net>. (click on 'Resources'.)
3. Bharati, Akshar, Dipti M Sharma, Rajeev Sangal, TransLexGram : An Introduction, Technical Report no: TR-LTRC-011, LTRC, IIIT Hyderabad, Feb 2001, <http://www.iiit.net/ltrc/Publications/Techreports/TR-LTRC-11>
4. Bharati, Akshar, Dipti M Sharma, Rajeev Sangal, Guidelines for TransLexGram: Translation, Technical Report no: TR-LTRC-012, LTRC, IIIT Hyderabad, Feb 2001, <http://www.iiit.net/ltrc/Publications/Techreports/TR-LTRC-12>
5. Bharati, Akshar, Dipti M Sharma, Rajeev Sangal, TransLexGram : Guidelines for Verb Frames, Technical Report no: TR-LTRC-013, LTRC, IIIT Hyderabad, Feb 2001, <http://www.iiit.net/ltrc/Publications/Techreports/TR-LTRC-13>
6. Bharati, Akshar, Dipti M Sharma, Rajeev Sangal, AnnCorra : An Introduction, Technical Report no: TR-LTRC-014, LTRC, IIIT Hyderabad, Mar 2001, <http://www.iiit.net/ltrc/Publications/Techreports/TR-LTRC-14>
7. Bharati, Akshar, Rajni Moona, Prashant Reddy, Bhavani Sankar, Dipti Misra Sharma, Rajeev Sangal, Machine Translation: The Shakti Approach, Pre-Conference Tutorial at ICON-2003: International Conference on Natural Language Processing is to be conducted on 19th December 2003.