# Integrating CAT and MT in AnglaBharti-II Architecture

**R. Mahesh K. Sinha**

Indian Institute of Technology, Kanpur 208016
India
rmk@iitk.ac.in

**Abstract** Machine translation (MT) is a complex and a difficult task. It is not possible to achieve human competing performance with the present state of technology. Automating the process of translation of natural languages requires a number of knowledge sources and their appropriate invocation in the translation engine. A practical machine translation system with limited resources cannot embody all the knowledge sources that the human beings use. However, performance of an MT system can be considerably improved if the automated translation system is integrated with supporting modules that provide synergy for arriving at correct translations. The computer assisted tools (CAT) that identify the limitations of the MT system and provide clues to cope up with them, constitute an important module for enhancing the MT system performance.

This paper presents details of AnglaBharti-II system architecture highlighting the role of CAT in the system. AnglaBharti-II is a system for translating English to Indian languages. AnglaBharti is primarily a rule-based system (RBMT). The input English sentence is transformed to a pseudo-interlingual structure called PLIL (Pseudo Lingua for Indian Languages) using a CFG like pattern directed rule-base. RBMT presents limitations of its own in dealing with real-life texts. We have tried to overcome some of these limitations in AnglaBharti-II architecture by integrating some additional modules. These additional modules are basically CAT tools incorporating translation memory, raw and generalized example-bases, interactive and automated pre-editing, paraphrasing, failure analysis and a number of heuristics that attempt to deal with a variety of constructs that are frequently encountered in a real life English text.

## 1. Introduction

Machine translation (MT) is a complex and a difficult task. The task of translating a natural language text requires an 'understanding' of the language (Allen 1987; Isabelle et al. 1993). Transplanting this 'understanding' on to the machine is a hard problem. A human translator uses a number of knowledge sources, a wide variety of context and background information to arrive at a target language text as close as possible to the original source language text.

Automating the process of translation of natural languages requires a number of knowledge sources and their appropriate invocation in the translation engine. A practical machine translation system with limited resources cannot embody all the knowledge sources that the human beings use. However, performance of an MT system can be considerably improved if the automated translation system is integrated with supporting modules that provide synergy for arriving at correct translations. Designing a practical machine translation system involves making several compromises to suit an application environment. The computer assisted tools (CAT) that identify the limitations of the MT system and provide clues to cope up with them, constitute an important module for enhancing the MT system performance.

This paper presents details of AnglaBharti-II (Sinha 2004) system architecture highlighting the role of CAT in the system. AnglaBharti-II is a system for translating English to Indian languages. AnglaBharti (Sinha 1995, 2003) is primarily a rule-based system (RBMT) (Sumita & Tsutsumi, 1988; Meyers et al 1998; Maruyama 1993). The input English sentence is transformed to a pseudo-interlingual structure called PLIL (Pseudo Lingua for Indian Languages) using a CFG like pattern directed rule-base. RBMT presents limitations of its own in dealing with real-life texts. It is difficult to produce hand-crafted transfer rules to cover a wide variety of input. Frequently, addition of rule may produce unpredictable side effects. Thus there is always a limit to which rule-base in RBMT can be grown. It is relevant to quote Becker (1975) here: *"Like all other scientists, linguists wish they were physicists. They dream (...) of having language behave in an orderly way so that they could discover the Universal Laws behind it all. Linguists have a problem because language just ain't like that."*

We have tried to overcome some of these limitations in AnglaBharti-II architecture by integrating some additional modules. These additional modules are basically CAT tools incorporating various functionalities. Some of these modules are: translation memory in the form of multi-word expressions; raw and generalized example-bases; interactive and automated pre-editing; paraphrasing; failure analysis; learning module; and a number of heuristics that attempt to deal with a variety of constructs that are frequently encountered in a real life English text.

In the following section we present details of these modules.

## 2. A look at assistance that an MT system needs

Having realized that a fully automatic quality machine translation is not feasible, let us examine the nature of assistance that can be provided to such a system to improve its performance. This in turn requires an analysis of the reasons for failures or deteriorated performance. Besides the sense, reference and structural ambiguities inherent in the natural language, there are a number of other sources of error and issues that need to be addressed while dealing with a real life text. Some of these are as under:

***i. Long sentences with too many nesting of embedded clauses:*** Parsing of long sentences with multiple & deep nesting is difficult and many a time are erroneous in their scope. Multiple coordinate conjunctions also pose a problem in identification of scope.

***ii. Failure in recognition of named entities:*** If the named entities are a also meaningful words in the source language, then the system has to decide whether to treat them as names or use the meaning given in the lexical data-base. It should be noted that the names are transliterated as per the target language. As an example the word, 'Bush' may become *'jhaaRii'* in Hindi if not treated as a name. Similarly, a building name 'White House' may get translated as *'safed ghar'* in Hindi if not recognized as a name.

***iii. Absence of expected punctuation marks:*** In a rule-based system, punctuation marks may also used in pattern matching and parsing. Usually, a human reader is able to identify the constituents easily without appropriate punctuation marks using contextual information. However, for a machine, this may be a source of problem.

***iv. Inaccuracy in identification of sentence and word boundaries:*** Treating '.' and '?' as sentence delimiters and blanks as word delimiters are too simplistic for a real life text. The sentences may have acronyms with a dot in it. If a sentence ends with such an acronyms, the sentence delimiter gets mixed with the acronym. There may not be blanks between words while writing currency or time. A word may have a single-quote attached to it at the end. This mark may be part of the word (apostrophe for plural nouns) or may be a matching single-quote mark.

**v.** ***Error in identification of nature of text to be translated:*** In a real-life text, the system must identify whether the text to be translated is a heading, running text with full sentences, partial sentences, an address part, a salutation in a letter, a letter reference number, a table etc. A translation system has to treat constituents of different types of texts differently. The translation of the same text under different categories may not be same. For this, different translation engines need to be invoked. For instance, in case of address translation, most of the nouns should be transliterated.

***vi. Dealing with acronyms, abbreviations, foreign words and unknown lexical items:*** It is not possible to enumerate these words/lexical items. We need to develop heuristics to identify them. A simple heuristic such as words with all upper-case be treated as an acronym does not work for a real-life text. The foreign words are morphologically transformed as per the grammar of the source language and these require identification before translation.

***vii. Identification of Arabic and Roman numerals, and numbers written in words:*** The Arabic numerals may have a dot, a comma, a slash, a colon etc as part of it and may also have certain affixes to denote nature of number such as currency, measurement, time etc. A Roman numeral such as 'I' is also a valid lexical item or may be an acronym. Further different languages may use different units for writing numbers in words which require identification and conversion.

***viii. Identification of currency and time:*** The styles of writing currency and time may not be same in the two languages. It is important to group and isolate them before performing syntactic analysis.

***ix. Dealing with parenthesized texts, texts within single and double quotes:*** The text appearing within parentheses has to be translated separately and substituted at appropriate position in the translated text. Further, there may be nesting within the parenthesized items. The texts within single or double quotes may not conform to the expected syntax of the entire sentence and we need to develop heuristics for dealing with them. Sometimes the double quotes cover multiple sentences as in direct speech or quoted quotes. A word may end with a single quote with no preceding matching quote symbol to signify relational parameter.

***x. Inaccuracies in GNP conformity:*** In Hindi, the gender as well as the number and person information gets reflected in the verb form. It may not solely depend upon the subject but may depend upon the object. It also depends upon whether the subject refers to an honorific status. As per Indian tradition, all elderly people are given an honorific status and are treated as plural as far as verb-form is concerned. Thus a sentence like, 'My father is visiting me' will be treated as 'My father are visiting me' for translation. Further, there may be a transfer of this honorific status based on relational construct. For instance, the sentence like, 'A friend of my father is visiting me' will be treated as 'A friend of my father are visiting me' but in case of the sentence like 'A friend of my son is visiting me' remains as it is for translation.

***xi. Inappropriate lexical choice and lexical gaps:*** The issue of lexical choice is not the same as that of lexical sense disambiguation. It is more to deal with pragmatics. Identification of lexical gaps and selection of an appropriate substitute word or phrase to fill in this lexical gap, is another task that needs to be performed by the translation system.

***xii. Inappropriate target text synthesis due to divergence in source and target languages:*** Due to certain kinds of divergence, the form of target Hindi text from English source may not be appropriate. For example, Hindi has causative verb forms that are derived from their root forms. English uses 'get', 'make' to represent the causative effect. A normal translation will yield an inappropriate translation if this is not recognized. This is also true for English sentences with 'have' forms. Similarly, modal verbs need to be treated differently because of their variation in forms in English and Hindi. For instance, the English sentences, 'He might come anytime', 'He may come anytime' and 'He can come anytime', all get translated to the same Hindi form 'vah kabhii bhii aa sakataa hai'. Similarly, for certain stative verbs like 'sit', 'stand', English has no difference in progressive state and indefinite state which leads to incorrect forms. Further details of English-Hindi divergence are given in Sinha and Thakur (2005).

***xiii. Inadequacies in dealing with certain types of source language constructs:*** The rule-base may not have adequate coverage to deal with all kinds of compound and complex constructs. However it may be possible to decompose the original construct into smaller constituent constructs that are covered in the rule-base. In certain cases, even the simple constructs can be re-phrased to yield more acceptable translation.

***xiv. Spelling and grammar errors in the source text:*** Spelling and grammar errors are very frequent in a real-life text. As long as such errors do not cause an ambiguity in the interpretation, the system should be made capable of translating them ignoring the errors in the input. This is akin to normal human interpretation.

***xv. Identification of e-mail addresses, URLs, reference numbers:*** The constituents such as e-mail, URLs, reference numbers of letters and documents must be treated as constants or must have a fixed unambiguous mapping. These must not be treated as names and inhibited from being transliterated.

***xvi. Domain/ topic identification:*** In case the domain and topic of the text for translation is known, it is possible to invoke domain/topic specific lexical data-base. For this, the lexical data-base has to be appropriately structured. This reduces the complexity of the lexical sense disambiguation task. A user may be asked to identify the domain/topic. This process can also be automated through training of the system with sample texts taken from different domains. The text domain categorization can be learnt statistically through this training.

# 3. Modules in AnglaBharti-II architecture assisting the translation engine

AnglaBharti-II is designed to translate English to Indian languages. The input English sentence is transformed to a pseudo-interlingual structure called PLIL (Pseudo Lingua for Indian Languages) using a CFG like pattern directed rule-base. PLIL is a structure that has the word-order of a group of Indian languages and carries all the syntactic and semantic information needed for synthesizing the target Indian language text belonging to that group. Detailed description of the system can be found in Sinha et. al. (1995, 2003, 2004). Here, we shall examine the modules that provide robustness to the system.

## 3.1. Hybridization strategy

The example based approach (EBMT) (Nagao, 1984; Sato & Nagao, 1990; Somers, 1999), making use of a set of previously translated sentences as examples provides another attractive paradigm where there are no hand-crafted rules as in case of RBMT.

However, the success of EBMT and other corpus based methods depends primarily upon availability of reliable parallel corpora with adequate coverage. There are other practical problems of dealing with size and matching in EBMT approach. An obvious answer is to hybridize the RBMT and EBMT (Sinha 2000). AnglaBharti-II uses a raw example-base (REB) and a generalized example-base (GEB) for hybridization with the basic RBMT paradigm. During the development phase, the example-base is grown interactively by augmenting it as illustrated in figure 1. When it is found that the modification in the rule-base is difficult and may result in unpredictable results, the user adds the input sentence as an example. At the time of actual usage, the system first attempts a match in the example-base before invoking the rule-base.

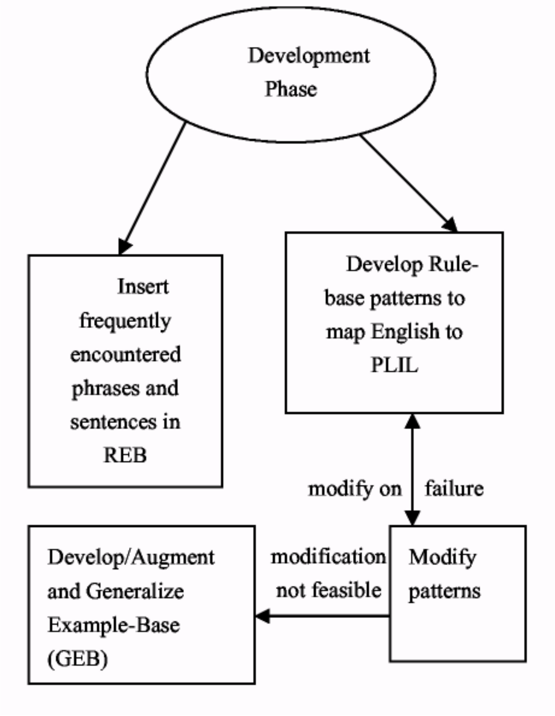
Development Phase
Insert frequently encountered phrases and sentences in REB
Develop Rule-base patterns to map English to PLIL
modify on failure
Develop/Augment and Generalize Example-Base (GEB)
modification not feasible
Modify patterns

**Figure l: Interactive growth of rule-base and example-base in AnglaBharti-II**

## 3.2. Pre-editing module

This is an interactive module that performs the following functions:

i. performs spell checking,

ii. identifies long and nested sentences that are likely to cause problem and prompts the user if (s)he would like to make a change,

iii. makes a guess on named entities, acronyms, abbreviations and prompts the user to confirm,

iv. marks various zones of the text based on inbuilt heuristics as a heading, addresses, part of the letter, reference number etc.

Although the user is encouraged to use pre-editing module, it is an optional stage at the discretion of the user. This module is designed to help the translation engine with additional information in those situations that are likely to yield erroneous translations.

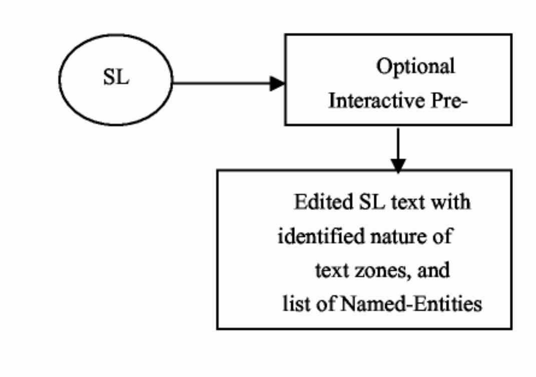
SL
Optional Interactive Pre-
Edited SL text with identified nature of text zones, and list of Named-Entities

**Figure 2: Optional Pre-editing in AnglaBharti-II**

## 3.3. Pre-processing module

This module performs multifarious tasks of preparing the text into a form that can be directly used by the translation engine. It uses a number of heuristics and patterns. The heuristics are developed through a detailed testing with a variety of sentences taken from corpus. The pre-processing module consists of following sub-modules:

### *3.3.1. Identifying text zones:*

In case no pre-editing is opted, this module marks different text zones based on a number of heuristics. The markings correspond to heading, partial sentence, running text, address, etc.

### *3.3.2. Identifying Sentence and Word boundaries:*

A number of heuristics are used to hypothesize a word, acronym and a sentence boundary. These are verified through shallow parsing of the hypothesized sentence.

### *3.3.3. Module identifying known acronyms and abbreviations:*

For the words that are not found in the lexical data-base, this module uses a number of heuristics to label the unknown as an acronym, a foreign word or a name. It transliterates (Sinha et. al. 1984) all unknown lexicons (Sinha 2001).

#### *3.3.4. Module identifying Arabic numerals based on numeric context:*

Roman numerals are identified using some heuristics. Numeric context is used to identify Arabic/Indian numerals. Roman numerals mixed with Arabic/Indian numerals are treated as constants for translation. The numbers written in the form of words are identified through pattern matching. The module stores these in a table with their corresponding Hindi equivalent derived through simple transformation. It should be noted that the forms in which numbers are represented are not uniform across the languages. There is a difference in the unit of measurement and also the place where commas are used to facilitate reading. While some of the units of measurements such as Centigrade/Fahrenheit or mile/kilometre are more widely acceptable and understood, the units like millions/trillions etc need to be converted to 'lakhs/crores' etc into Hindi to make it more acceptable. Further, the position of delimiting commas have got be accordingly modified in case of figures.

#### *3.3.5. Module identifying currency forms:*

A pattern matching is used to identify the currency forms and substitute them with dummy variables. A table stores these dummy variables with their corresponding Hindi forms derived through simple transformation.

#### *3.3.6. Module identifying word groups that represent time and seasons:*

The time expressions (examples: 4pm, quarter to four, 4 o'clock etc) are identified using variable pattern matching and then translated into Hindi using matching with stored examples. It should be noted that the representation of time and season, may also be dependent upon the social, religious and geographical conditions besides being language specific. For example, the English abbreviations 'am'/'pm' when translated into Hindi dividing the entire twenty-four hours into two parts, simply leads to confusing and unacceptable appendage in Hindi. In Indian subcontinent, a day is divided into 4-8 different approximate time periods and 'am'/'pm' should be mapped on to appropriate Hindi names for a natural translation. This is not a case of lexical gap but a question of pragmatics and social acceptability. Similar things hold for seasons like 'fall', 'spring' etc.

#### *3.3.7. Module that extracts texts within parentheses, single and double quotes:*

The extracted text is sent for translation treating these either as a partial sentence or full sentence based on some heuristics. The location of the beginning of parenthesis/quote mark is flagged and this flag is used to substitute the translation of extracted text.

The module also identifies words with an apostrophe sign and substitutes it with a dummy symbol. The word with a closing single quote with no matching opening single quote is treated as an apostrophe.

### 3.4. Raw and generalized example-base module

The example-base consists of sentences or phrases that occur frequently or are difficult to handle by the rule-base. This ensures a more efficient and acceptable translation. There are two types of example bases used in the system. The raw example base (REB) consists of examples in their original form, whereas the generalized example-base (GEB) holds examples in generalized form by introducing variables with their properties. For example, City names like Kanpur, Delhi are substituted by a variable 'city'. Matching an input text with GEB is based on evaluating minimum 'distance' with the stored examples.

REB holds phrases and sentences that are likely to be encountered very frequently in the domain of application. For example, in a correspondence domain, phrases like 'Yours Sincerely', 'Best Regards' etc are stored in raw form. A user may also use coded raw example for his/her convenience for frequently used expressions. The input text is first matched with REB and in case of failure matched with GEB. In case of failure in match GEB, rule-base is invoked.

### 3.5. Multi-word expression module

This module identifies groups of words that when considered together has a distinct interpretation and cannot be directly derived from meanings of individual words during the translation process. These are referred to as multi-word expressions (MWE) (Reinhard 1996; Sinha et al 2004). MWE can be a noun-noun, adjective-noun, adverb-noun, verb-particle combination or a name consisting of multiple words. For instance, 'eye glasses', 'middle class people', 'bonded labourers', 'put on', 'Department of Atomic Energy', 'White House' are all MWEs. The verb-particle MWE called verb-phrasal can also be polysemous. For instance, 'put on' means 'wear', 'light up', 'gain (put on weight)' etc. Further, whether a group of words forms an MWE is context dependent and a mere listing of these is not sufficient. For instance, in the sentences 'The box is kept on the roof and 'He kept on talking', the verb-phrasal 'keep on' is applicable only for the second sentence. This module uses a number of syntactic and semantic rules to form MWEs and provide their disambiguation.

The unconditional MWEs (examples: high court, federal government, once upon a time etc.) are used to club the words in the input text and meanings obtained from the lexical database. The extraction of conditional MWEs requires at least a shallow grammatical analysis of the input text. Creation of MWE database (conditional and unconditional) is a time consuming and a tedious job. At present, in the absence of an extensive parallel corpora with wide coverage, this data is being created manually.

The MWE data is also generalized for efficiency purposes. Thus, the expressions like 'bonded labourers', 'bonded workers', 'bonded farmers' etc. are generalized to 'bonded <human>'. In contrast, expressions like 'bonded metal', 'bonded glass' etc. are generalized to 'bonded <solid>'. The meaning of 'bonded' in the two cases will come out to be different when matched the input text based on the semantics of the noun followed. Thus the size of the MWE database gets drastically reduced by such generalizations. It should be noted that each MWE has a predefined syntactic role and is not an arbitrary fragment of a sentence.

### 3.6. Automated Pre-editing module

The purpose of automatic pre-editing (Shimohata 2004; Yamamoto 2002, Yoshimi et al 1999) module is to transform/paraphrase the surface form of the input sentence to another 'equivalent' form that is more easily translatable. For example, a sentence like 'Had I gone there, this would have not happened' may be transformed to 'If I would have gone there, this would have not happened' as these mean the same thing. Some of the transformations may be dependent upon the language pair. For example, an English sentence 'He killed his own dog' may be transformed to 'He killed own dog' for translation to Hindi as it would avoid repeating the Hindi word 'apanaa' (Hindi meaning for both the English words 'his' and 'own') twice. Such transformations are transparent to the user.

Some of the problems of scope of multiple coordinate conjunctions also get handled by this module. The operands of the coordinate conjunctions are grouped if they belong to similar semantic category. This way the translation engine is relieved of disambiguating scopes in certain cases.

Automated pre-editing module may even fragment an input sentence if the fragments are easily translatable and the fragment translation is positioned appropriately in the final translation. For instance, a sentence like, 'State your income from agricultural sources, if any' could be easily broken into two parts for easy translation. Sentence fragmentation may also follow paraphrasing. For instance, the sentence, 'The box, I believe, contains explosives' is first converted to 'I believe that the box contains explosives'. Although, most of the systems will translate this sentence directly without a need for any fragmentation, this can be fragmented into simple sentences if needed. The sentence can be fragmented into the three parts: 'I believe', a conjunction 'that' and 'the box contains explosives'. The three parts are translated separately and translation juxtaposed in this case. The process is similar to that of performing chunk identification and chunk translation. However the chunking performed here is at a very shallow level and is mostly pattern driven. We have found this approach to be useful particularly in translation of forms etc. It should be noted that you cannot fragment every sentence at the occurrence of conjunction 'that'.

## 3.7. Failure analysis module

The failure analysis module is triggered when the system reports a failure of not finding a pattern matching with the input sentence. This module consists of a number of heuristics that speculate on what might have gone wrong with the pattern. For instance, a sentence of the type, 'If you go there he will accompany you' will fail without a comma in the sentence. The failure analysis module may insert punctuation marks based on a shallow analysis. This module may also trigger sentence fragmentation as in case of automated pre-editing. The system maintains a log of sentences on which it failed. These are manually analyzed and remedial measures are incorporated in the form of callable functions. Each one of these functions performs a transformation of the input sentence which may be introducing punctuation marks, revising the sentence boundary, decomposing the sentence, inserting new examples in the example-base etc. As the system gets more and more exposed to a variety of sentences, it acquires robustness through this module.

## 3.8. Ill-formed sentence corrector module

The module of ill-formed sentence corrector (Sinha 1993) is responsible for making corrections to the output generated by the target text-generator module of the translation system. These corrections are mostly of syntactic nature that also take pragmatics into account. In Hindi gender conformity and obliqueness are more prone to mistakes. We use Hindi grammar features and a number of heuristics to ensure gender, number and person conformity. Further, as per Indian tradition, all elderly people are given an honorific status and are treated as plural as far as verb-form is concerned. Thus a sentence like, 'My father is visiting me' will be treated as 'My father are visiting me' for translation. Further, there may be a transfer of this honorific status based on relational construct. For instance, the sentence like, 'A friend of my father is visiting me' will be treated as 'A friend of my father are visiting me' but in case of the sentence like 'A friend of my son is visiting me' remains as it is for translation. This module is responsible for making such corrections. Besides these, appropriate positioning of certain particles such as 'nahiiN', 'kabhii' etc. are also decided by this module.

Here one may argue that the functionalities of this module can very well be a part of the target text generator module. This is more a design decision. A separate module on ill-formed sentence corrector facilitates experimentation on form based on pragmatics. Here one concentrates merely on the target language text which is Hindi here. Well tested functionalities of this module can be transferred to the target text generator module.

## 3.9. Lexical data-base hierarchy

This module provides a hierarchy in the lexical data-base to facilitate domain/topic specific lexical choice. We build a hierarchy with user defined dictionary being at the top of the hierarchy and at the bottom lies the most general lexical data-base. In between there may be a number of lexical data-bases on various topics. The hierarchy in the topic lexical data-bases is structured in form of a directed acyclic graph (DAG). The topic hierarchy in the form of DAG allows multiple inheritance. The system first makes a search in the user defined dictionary and on failure enters into the topic hierarchy DAG. If it is not found in the hierarchy then finally, the search is directed to the general lexical database. For instance, if the user picks up a topic for translation as 'astronomy', the system will first search the user defined dictionary. If not found then it will search the lexical database on the topic of Astronomy. If it is not found here also, then the search is directed to the parents of the topic node in a breadth first fashion. In this case search will be directed to topics of Physics, Mathematics, Science etc, based on the built in hierarchy. Finally, if the word not found in any one of these lexical databases in the hierarchy, the general lexical database is searched. While building such an hierarchy, a subject taxonomy such as used in library may be used.

The lexical database hierarchy outlined here reduces the burden of sense disambiguation that the translation engine may be required to perform by limiting the choices of meanings. However, there is always a danger of forcing certain mappings that may not be intended. For instance, a word 'treatment' in medical domain would mean providing medical assistance ('upachaara' in Hindi) and in general domain as 'behaviour' (vyavahaar in Hindi). However, a sentence like 'On my visit to doctor, he treated me very badly' when translated in medical domain, may yield a wrong interpretation.

### 3.10. Learning user's lexical choice

AnglaBharti-II is a machine aided translation system architecture. Here the machine may yield multiple translations for ambiguous constructions that could not be resolved by the system. Similarly, it may give alternate meanings of the polysemic words. The user is expected to perform post-editing on the output generated by the machine. A user-friendly post-editing environment is incorporated into the system. The module on learning the user's lexical choice, performs statistical analysis on user's choices. The module constructs a data containing collocation preferences out of this analysis. This data is used in ranking the alternate translations generated by the system.

## 4. Conclusions

AnglaBharti architecture has been designed to translate from English to Indian languages. This primarily uses a rule-based paradigm. A rule-based architecture has inherent limitations of being fragile and not being scalable. All other translation strategies require reliable bi-lingual parallel corpora for them to be successful. Such bi-lingual parallel corpora are very scarce for Indian languages. An obvious way to alleviate limitations of a rule-based system, is to supplement the system with a number computer assisted tools that take care of situations where a rule-based system is likely to fail.

In this paper we have examined the nature of a real-life text that a typical machine translation system is required to handle. Keeping these in view, we present details of the additional modules that provide assistance to the base translation engine in AnglaBharti-II architecture in overcoming its limitation in dealing with real-life text. These additional modules are basically CAT tools incorporating various functionalities. Some of the major modules are: translation memory in terms of multi-word expressions; raw and generalized example-bases; interactive and automated pre-editing; paraphrasing; failure analysis; learning users' lexical preferences and lexical disambiguation; and a pre-processing module embodying a number of heuristics. The pre-processing module attempts to deal with a variety of constructs that are frequently encountered in a real life English text. Integration of these modules in AnglaBharti-II architecture provided it with greater efficiency, acceptability and robustness. A beta version of English to Hindi translation based on this methodology is under detailed evaluation. It is observed that the addition of these modules in AnglaBharti-II architecture has improved the performance of the system from an average of 40% to an average of 80% correctness.

Each of the modules of AnglaBharti-II architecture has been configured to enrich itself with additional knowledge contributing to greater overall accuracy of the system. As more data on parallel corpora becomes available, more reliable data on MWEs, learning lexical disambiguation and choice, named-entities, sentence boundary identification can be extracted using statistical techniques and incorporated into various modules of the system.

## 5. Acknowledgements

This work is supported by a grant from Ministry of Communication and Information Technology, Govt. of India.

## 6. References

ALLEN J. (1995). 'Natural Language Understanding', Benjamin/Cumming Publishing Company.

BECKER. Joseph D. (1975). 'The Phrasal Lexicon'. In Proceedings of Theoretical Issues in Natural Language Processing, (pp 70-73) Cambridge, Massachusetts.

HUDDLESTON, Rodney and Geoffrey K Pullum (2002). 'The Cambridge Grammar of the English Language'. Cambridge: Cambridge University Press.

ISABELLE, P. et al. (1993). 'Translation Analysis and Translation Automation'. In Proceedings of the Fifth International Conference on Theoretical and Methodological Issues in Machine Translation, Kyoto, Japan.

KACHRU, Yamuna (1980) 'Aspects of Hindi Syntax'. Delhi: Manohar.

MEYERS, A., Yangarber, R., Grishman, R., Macleod,
C. and Moreno-Sandeval, A. (1998). 'Deriving Transfer Rules from Dominance-Preserving Alignments', COLING-ACL, (pp 843-847).

MARUYAMA, H. (1993). 'Pattern-Based Translation: Context Free Transducer and its Application to Practical NLP', In Proceedings of National Language Processing Pacific Rim Symposium (NLPRS'93), Fukuoka, Japan.

NAGAO, M. (1984). 'A framework of a mechanical translation between Japanese and English by analogy principle'. In A. Elithorn and R. Banerji (Editors), Artificial and Human Intelligence, (pp 173-180), North Holland, Amsterdam.

SATO, S. and Nagao, M. (1990). 'Towards Memory-Based Translation', Proceedings of COLING-90.

SCHÄLER, Reinhard (1996). 'Machine translation, translation memories and the phrasal lexicon: the localisation perspective', In TKE 96, EAMT Machine Translation Workshop, (pp 21-33) Vienna, Austria.

SHIMOHATA, Mitsuo (2004). 'Acquiring paraphrases from corpora and its application to machine translation', Ph.D. Thesis, Nara Institute of Science & Technology, Sept. 2004. (http://cl.aist-nara.ac.jp/thesis/dthesis-shimohata.pdf)

SINHA, R.M.K, and Thakur, Anil (2005). 'Translation Divergence in English-Hindi MT', Proceedings EAMT 2005 Conference, May 30-31, 2005, Budapest, Hungary.

SINHA, R.M.K (2004). 'An Engineering Perspective of Machine Translation: AnglaBharti-II and AnuBharti-II Architectures', Proceedings of International Symposium on Machine Translation, NLP and Translation Support System (iSTRANS-2004), Tata McGraw Hill, New Delhi.

SINHA, R.M.K. et. al. (2003). 'AnglaHindi: An English to Hindi Machine Translation System', MT Summit IX, New Orleans, USA.

SINHA, R.M.K (2001). 'Dealing with Unknown Lexicons in Machine Translation from English to Hindi', Proc. of IASTED International Conference on Artificial Intelligence and Soft Computing, (pp 333-336), Cancun, Mexico.

SINHA, R.M.K (2000). 'Hybridizing Rule-Based and Example-Based Approaches in Machine Aided Translation System', 2000 International Conference on Artificial Intelligence (IC-AI'2000), Las Vegas.

SINHA, R.M.K et. al. (1995). 'ANGLABHARTI: A Multi-lingual Machine Aided Translation Project on Translation from English to Hindi', 1995 IEEE International Conf. on Systems, Man and Cybernetics, (pp 1609-1614) Vancouver, Canada.

SINHA, R.M.K (1993). 'Correcting ill-formed Hindi sentences in machine translated output', in Proceedings of Natural Language Processing, Pacific Rim Symposium (NLPRS'93), (pp 109-119), Fukuoka, Japan, 1993.

SINHA R.M.K and Srinivasan, B. (1984). 'Machine transliteration from Roman to Devanagari and Devanagari to Roman', Jour. of Institution of Electronics. & Telecomm. Engineers.30(6), 243-45.
SINHA, R.M.K, and Thakur, Anil (2004). 'Multi-word Expressions in English and Hindi: Problems in Contextualization', International Symposium on MT, NLP and TSS (iSTRANS), Tata McGraw Hill, New Delhi.

SOMERS, H. (1999). 'Review Article : Example-based Machine Translation', Machine Translation, 14(2), 113-157.

SUMITA, E. and Tsutsumi, Y. (1988). 'A Translation Aid System Using Flexible Text Retrieval Based on Syntax-Matching', TMI, 1988.

YAMAMOTO, Kazuhide (2002). 'Machine Translation by interaction between paraphraser and transfer', http://acl.ldc.upenn.edu/C/C02/C02-1163.pdf

YOSHIMI, Takehiko and Sata, Ichiko (1999). 'Improvement of Translation Quality of English Newspaper Headlines by Automatic Preediting', MT Summit VII, 1999.