English-Hindi and Hindi-English Machine Translation

GRADUATE PROJECT REPORT

Submitted to the Faculty of
the Department of Computing and Mathematical Sciences
Texas A&M University – Corpus Christi
Corpus Christi, Texas

In Partial Fulfillment of the Requirements for the Degree of
Master of Science in Computer Science

by

Sumit Agarwal
Fall 2007

Committee Members

Dr. Dulal C. Kar
Committee Chairperson

Dr. Mario Garcia
Committee Member

Dr. David R. Thomas
Committee Member
ABSTRACT

With the technological advancements of the world, the need for information exchange is paramount. One of the great challenges for information technology is to overcome language barriers across the whole of humanity so that anyone could talk and communicate to anyone else on the planet in real time. There are different living dialects being used in a predominant manner in different parts of the world. Machine Translation (MT) has carved a special niche for itself in such a world where fast and easy information dissemination is required in order to break the linguistic barriers. Currently manual translation has been limited to important official documents, news items, and award winning literary works.

This project focuses on development of an interlingua based MT system. The aim is to build a machine aid translator that can convert simple sentences from a source language such as English to a target language such as Hindi and vice versa. The system does not as such have to analyze the sentences for errors etc. Instead it has to merely extract out the semantic information and store it appropriately in a case frame format and then convert it later back to a sentence(s) of the target language while maintaining the same semantic meaning of the original sentences.

The system is implemented on a standard C++ platform (Qt Libraries used for the GUI), and incorporates inputs in fonts of respective languages. This project has wide applicability and shows a good insight into the complexities of natural languages. This
project is just a prototype of a large MT system and there is ample scope for further improvement and enlargement.

TABLE OF CONTENTS

Abstract..............................................................................................................................................i
Table of Contents..................................................................................................................................iii
List of Figures..........................................................................................................................................v
List of Tables...........................................................................................................................................vii
1. Background and Rationale.................................................................................................................1
   1.1 Introduction..................................................................................................................................1
   1.2 History of Machine Translation.................................................................................................1
   1.3 Machine Translation Strategies.................................................................................................3
   1.4 Need of Machine Translation.....................................................................................................6
   1.5 Project Objective.........................................................................................................................7
2. Narrative.............................................................................................................................................9
   2.1 Current Trend of MT System........................................................................................................9
   2.2 Scope of the Project.....................................................................................................................11
3. System Design and Implementation..................................................................................................13
   3.1 Overall System Design................................................................................................................13
   3.2 Bi-Lingual Dictionary................................................................................................................15
   3.3 Analyzer.....................................................................................................................................17
      3.3.1 Morphological Analyzer....................................................................................................17
      3.3.2 Syntax Analyzer and Semantic Analyzer..........................................................................18
3.4 Case Frame .....................................................................................18

3.4.1 Structure of Case Frame .............................................................19

3.5 Generator .......................................................................................26

3.5.1 Case Frame to English Rules ......................................................27

3.5.2 Case Frame to Hindi Rules ..........................................................30

4 Result and Discussion .......................................................................34

4.1 Results Obtained .............................................................................34

4.2 English – Hindi Translation Results ..............................................36

4.3 Hindi – English translation Results ................................................41

4.4 Hindi-English Random Sentence Translation .................................46

4.5 English-Hindi Random Sentence Translation .................................47

5 Testing and Evaluation .....................................................................48

5.1 System Performance .......................................................................48

5.2 Unit Testing .....................................................................................48

5.3 Regression Testing .........................................................................49

6 Conclusion .........................................................................................51

6.1 Further Scope .................................................................................51

6.2 Limitation .......................................................................................52

Bibliography and References ................................................................53

APPENDIX A .......................................................................................55

APPENDIX B .......................................................................................65

APPENDIX C .......................................................................................68
LIST OF FIGURES

Figure 1.1 MT Architecture ..........................................................4
Figure 1.2 Different approaches of MT...................................................5
Figure 2.1 Interlingua Model with English-Hindi language pair...............10
Figure 2.2 Proposed System Design....................................................14
Figure 4.1: English to Hindi translation..............................................34
Figure 4.2: Hindi to English translation...............................................35
Figure 4.3: Present Tense English-Hindi Translation............................36
Figure 4.4: Present Continuous Tense English-Hindi Translation............36
Figure 4.5: Past Tense English-Hindi Translation.................................37
Figure 4.6: Past Tense English-Hindi Translation.................................37
Figure 4.7: Past Tense English-Hindi Translation.................................38
Figure 4.8: Future Tense English-Hindi Translation..............................38
Figure 4.9: Future Tense English-Hindi Translation..............................39
Figure 4.10: Condition Tense English-Hindi Translation.......................39
Figure 4.11: Condition Tense English-Hindi Translation.......................40
Figure 4.12: Present Tense Hindi - English Translation.........................41
Figure 4.13: Present Tense Hindi-English Translation............................41
Figure 4.14: Past Tense Hindi - English Translation............................42
Figure 4.15: Past Tense Hindi - English Translation............................42
LIST OF TABLES

Table 4.1: Hindi- English Random Sentence .........................................................46

Table 4.1: English- Hindi Random Sentence .........................................................47

Table A.1: Different Case Ending in Hindi ..........................................................58

Table A.2: Suffixes and Morpho-Words for Hindi Verb Conjugations .................61

Table A.3: Verb Morphological Changes from English to Hindi Translation ........64
1. BACKGROUND AND RATIONALE

1.1 Introduction

This being an era of information explosion, information itself plays a vital role in the growth of science and technology. With the technological advancement of the world, the need for information exchange is paramount. MT system has carved a special niche for itself in such a world where fast and easy information dissemination is required in order to break the linguistic barriers. MT is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another. At its basic level, MT performs simple substitution of atomic words in one natural language for words in another natural language [John Hutchins 1992].

Though MT has not solved all the problems of translation, it has helped to solve the problem of translation in at least a restricted domain of knowledge. Today MT has become a major area for potential researchers all over the world. In view of this wide applicability of MT systems, it becomes essential to get an idea of the earlier achievements in the field and the approaches used by computer scientists of yesteryears [John Hutchins 1992].

1.2 History of Machine Translation

The first proposal of MT using computer were put forward by Warren Weaver, a researcher at the Rockefeller foundation. These proposals were based on information theory. Few years after these proposals, research began in earnest
at many universities in the United States. On 7 January 1954, the Georgetown-IBM experiment, the first public demonstration of a MT system, was held in New York at the head office of IBM. The demonstration was widely reported in the newspapers and received much public interest. During this time, operational systems were installed. The United States Air Force used a system produced by IBM and Washington University, while the Atomic Energy Commission in the USA and Euratom in Italy used a system developed at Georgetown University. While the quality of the output was poor, it nevertheless met many of the customer’s needs, chiefly in terms of speed [John Hutchins 1992].

Research in the 1960s in both the Soviet Union and the US concentrated mainly on Russian-English language pair. Chiefly the objects of translation were scientific and technical documents. A great boost comes to MT research in 1966 with the publication of the Automatic Language Processing Advisory Committee (ALPAC) report. The report recommends that tools be developed to aid translators, for example automatic dictionaries and that some research in computational linguistics should continue to be supported. While research in the 1960s concentrated on limited language pairs and input, demand in the 1970s was for low-cost systems that could translate a range of technical and commercial documents. This demand was spurred by the increase of globalization and the demand for translation in Canada, Europe, and Japan [John Hutchins 1992].

By the 1980s, both the diversity and the number of installed systems for machine translation had increased. Research relied on translation through some variety of intermediary linguistic representation involving morphological along
with syntactic and semantic analysis. At the end of 1980s there was a large increment in a number of methods for machine translation. One system was developed at IBM that was based on statistical methods. Other groups used methods based on large numbers of example translations; a technique which is now termed example based machine translation [John Hutchins 1992].

During the 1990s, encouraged by successes in speech recognition and speech synthesis, research began into speech translation. In early 1990s MT began to make the transition away from large mainframe computers toward personal computers and workstations and websites also become available on the Internet, such as Alta Vista and Google [John Hutchins 1992]

The field of MT has in the last few years seen major changes. Currently a large amount of research is being done into statistical machine translation and example based machine translation. With these developments, MT software is becoming a mass-market product, as familiar as word processing and desktop publishing. On the research front, the principal areas of growth are to be seen in example-based and statistical machine translation, in the development of speech translation for specific domains, and in the integration of translation with other language technologies [Taylor 1998].

1.3 Machine Translation Strategies

MT can use a method based on linguistic rules, which means that words will be translated in a linguistic way i.e., the most suitable words of the target language will replace the ones in the source language. A number of techniques
for information transfer using a computer have been developed. The general architecture for natural language translation is shown in Figure 1 [Surprise 1997].

![MT Architecture Diagram]

Figure 1.1: MT Architecture.

Most technique differs the way system was organized, the nature of problems encountered at different level of syntax in source language and the target language. But in spite of these differences, one can see three approaches to the problem of MT, viz:

a. Direct Translation or Word–for-Word Strategy

b. Transfer Strategy

c. Interlingua Strategy
The simplest and first used strategy is the word-for-word approach also called direct translation. As the name suggests, the translation is done word-for-word over some lexicon. Normally, this sort of translators is improved with phrase lexicons, programs for morphological control and generation, which are able to change the word order. For example Systran system, still used at commission of the European community is a direct translation system for Russian-English language translation. Word-for-word translation systems are necessarily bilingual and unidirectional [Hutchins, 1999].

The second type of translation is transfer strategy which has three stages, where translation proceeds in three stages, analyzing input sentences into a representation which still retains characteristics of the original, source language text. This is then input to a special component (called a transfer component) which produces a representation which has characteristics of the target (output) language, and from which a target sentence can be produced. Example of transfer strategy is Meteo system developed at TAUM, the University of Montreal’s machine translation project created by the Canadian
Government in 1965. It has been functioning to translate weather forecasts from English to French since 1977 and from French to English since 1989 [Hutchins, 1999].

The third type of strategy for MT system is interlingua approach, where the translation is done in two steps. Some well known systems are KBMT89 at Carnegie-Mellon University and ULTRA at New Mexico State University. The source language is first converted into an interlingua representation, said to be common for many languages. Then, from this interstate, a text is generated in the desired target language. The analysis and generation parts are quite complex and separate from each other. All ambiguities have to be taken care of in the source language text to enable the translation into target language in the next step. This interlingua approach is very economic when more languages are involved, but then the complexity of the interlingua is accordingly increased [Hutchins, 1999].

1.4 Need of Machine Translation

There are two primary needs for this type of research, first the technological motivation to build intelligent computer systems such as the MT system. The idea behind the work is to utilize the computer system in the various fields of artificial intelligence optimally. This motivation was perhaps the driving force behind the earlier computer scientists. They dabbled in the design of MT systems with the urge to see whether a system which can understand and effectively deal with a complex subject like natural language processing and translation can be implemented on a computer or not.

The second, the linguistic, or cognitive science, motivation was to gain a better understanding of how humans communicated by using natural language
and how the translations between natural language are effected by humans.
People with this motivation were not only interested in producing a structural
description of natural language but in the way the people actually produced and
comprehended natural languages. Their cognitive goal was to build a
computational analog of the human-language processing mechanism.

1.5 Project Objective

The objective of the proposed project to design a MT system is
accomplished. Designed system is capable of translating simple sentences from a
source language to a target language. System is suitably efficient in order to
resolve a reasonable number of semantic ambiguities of the source language and
the target language.

The efficient MT system must have certain modules such as a suitable
knowledge base, a syntax parser, a semantic analyzer, a translator, and a
generator. In the system design undertaken, an attempt has been made to
incorporate all these modules. A bottom up parser has been implemented which
incorporates both the syntax and semantic analyzers. The purpose of this parser
was to extract out all the semantic and syntactic information present within the
sentence.

Internal representation in MT system implementation is a frame based
representation. This frame is used for translating relevant meaning of sentences
with semantic and syntactic information of the sentences contained in it. A
translator module contains the rules for translation. The generator is used for
generation of the sentence to the target language. All this implementation
required a knowledge base, which in this case is in the form of dictionaries. The
dictionaries contained all relevant semantic information regarding the source
language and the target language. The primary task in this system is analysis of
sentence, parsing of sentence, analyzing the words lexically and morphologically,
and conceptualizing the source language sentence into a semantic case frame and
than translating the sentence into the target language using the bi-lingual
dictionary and linguistic information from the semantic case frame.
2. NARRATIVE

2.1 Current Trend of MT System

Translation plays an important role in information dissemination. The process of translation is not just a process of translating words but concepts, not a sequence of words in a particular syntactic relationship to one another but ideas. It is a thinking process and is not just matching of words nor a function of arithmetic or geometric progressions of a probable word combination occurrences. Translation can be briefly defined as follows “It is an art of handling and manipulating the lexical power of linguistic elements of a language. It is an intellectual exercise involving the expertise of a language and its grammar. It involves comprehension, analysis and conceptualization of thoughts and patterns in the source language and then expressing those grammatically and idiomatically in the target language”. A good human translator should normally possess sensitivity with intelligence, creativity with good organization and inventiveness with good discipline. He should also have the ability to handle and manipulate the lexical power of linguistic elements of a language. It involves the expertise and the ability to tackle even the most difficult situations pertaining to syntax, semantics and styles of a natural language during translation [Chellamuthu KC].

Although the goal in a MT system is to transfer the meaning and contents of a source language into a target language in a faithful way, it differs based on the techniques employed. These strategies depend on the syntactic and semantic nature of the source language and the target language, methodology and the tools adopted in modeling the MT etc.
Based on these factors currently MT through interlingua strategy is used for the translation. In this method, Artificial Intelligence tools such as rules for knowledge representation schemes involving a high level structure and appropriate inference mechanism to resolve syntactic and semantic ambiguities and pragmatics are adopted. Interlingua method employs a universal language which is independent of a natural language involved in the process. This intermediate language helps to resolve several problems of translation which otherwise could not be solved using the regular strategies. The various stages in an interlingua strategy are analyzing the text for conceptual representation, providing contextual world knowledge through inference mechanism, reproduction of the language free representation of the source language sentences into target language etc. This technique fully exploits the AI tools and the natural language processing analysis to achieve a meaningful translation [Chellamuthu KC].

**Figure 2.1: Interlingua model with English-Hindi language pair**

Due to above features interlingua strategy generates uniformity among various MT systems for different natural languages. Interlingua strategy has reduced the task of computer scientists and offered us a method for effective and
meaningful translation by resolving the various problems of natural language [James 1987].

2.2 Scope of the Project

The scope of the project is to design a MT system for translation of simple sentences from English language to Hindi language and vice-versa. To accomplish translation system should involve designing of language dictionary, grammar for the source language and target language, parser and generator. The search process for the dictionary has to meet the real time system requirements. For this purpose the dictionaries have been indexed by a trie (Appendix B) data structure. All the sentences of a source language and target language are in their respective fonts. It is mandatory that the translation of languages should be in there respective fonts. To display Hindi language sentences system uses Hindi ASCII fonts i.e., shusha fonts. These fonts have their key mapping with the roman keys and do not add any extra memory for the system.

The system addresses sentences only in the active form, and is capable of taking as input a sentence or a paragraph of simple sentences and converting it into the respective semantic case frame representation and similarly generating a sentence or a paragraph from the case frame representation.

MT systems based on interlingua strategy are ideal for the Hindi translation. The semantic structure used in the system is the case frame representation. Two modules are required, one to convert to case frame and one to generate back from the case frame to the language. The other modules of the system are based on the standard requirements of the MT system. The system has
a bilingual dictionary, syntax and lexical analyzer, a translator, and a generator.
The primary task in this system is the analysis of sentence parsing of sentence
analyzing the words lexically and morphologically, and conceptualizing the
source language sentence into a semantic frame, design and use of a bi-lingual
dictionary and translating the sentence into a target language using the linguistic
information from the semantic representation.
3. SYSTEM DESIGN AND IMPLEMENTATION

3.1 Overall System Design

The overall system design for the MT system is shown in Figure 4. The dictionary acts as a bridge between a source language and its target language. Each source and target language supports a dictionary and two modules, one for source language to case frame generation and the other for case frame to target language generation.

Both the language analyzer and generator use certain morphological modules to find the best word possible for the current context. The set of rules of analyzer, generator and the morphologies is language specific. The GUI layer of the system act over this architecture. The overall MT system consists of the following functional system:

1) Input-Output system: Enters data in one source natural language and system shows the converted version of source language into target language.

2) Translation support system: Supports correction of the source texts (pre-edit) and correction of translated texts (post-edit). This is necessary for efficient machine translation processing.

3) Sentence analysis system: Analyzes input sentences using morphological analysis, syntax analysis, and semantic analysis. Converts the analyzed data into case frame i.e., interlingua.
Figure 2.2: Configuration of MT System and Translation flow
4) Dictionary system: manages the information on each language necessary for machine translation, including grammar and semantic information.

5) Sentence generation system: user sentence generation, syntax generation, and morphological generation to process interlingua data into each corresponding language [CICC 2006].

3.2 Bi-Lingual Dictionary

The dictionary is the main link between all sets of languages. The dictionary takes us from the source language to a case frame and from a case frame to the target language. A well-defined structure for the dictionary is mandatory for the success of translation. The dictionary of a particular language contains the root words that are matched to find the exact meaning of a word. The dictionary is bi-directional i.e., it serves the purpose of evaluating the case word from a language and also evaluates the word that should be put in the final sentence given the case word. The search techniques applied in the dictionary should be fast enough to fit in a real time system.

The dictionary used in our system is a record base dictionary indexed by a trie data structure. Three separate records adjective, noun, and verb have to be maintained. Each of these records contains the relevant information regarding words. Each language L has its own set of three records termed as LAdjectiveRecord, LNounRecord and LVerbRecord. Each record set is of constant size depending upon the number of forms word it holds. The LAdjectiveRecord for English consist of only one word, the adjective itself. For
Hindi it supports the Hindi meaning of that word with English word. For example the adjective ‘good’ would be stored as “good-AcCa”.

The LNounRecord for English contain the noun word and information about what category that noun belongs to. The nouns divided into five categories. Root case word, it’s meaning in L and a character to represent the type of noun. Five type of nouns have been stored in the dictionary namely

1. Animate Male
2. Animate Female
3. Non Animate Male
4. Non Animate Female
5. Location

The Hindi Record set for the noun dictionary is very similar to the English record set except that it also supports the Hindi meaning of the record. For example the noun ‘car’ would be stored in the Hindi dictionary as “car-gaaD,I-4”. The LVerbRecord is the largest of all these records. It has to store a minimal form of forms of a verb from which the other forms can be generated using morphologies. For English all the five forms of the verb (Root Form, Present Form, Past Form, Present Continuous and Past Perfect) have been stored. So an English Verb record set for the verb “eat” would be “eat eats eating ate eaten”. Although storing only 5 of these forms and generating the rest from them is possible still the current record set for Hindi verb supports 11 forms of the verb. So in Hindi the verb “eat” would be “eat-2-Ka Kayaa Kayal Kata Katl Ka]Mgaal Kayaogal Ka ilayaa Ka laI”. The Hindi Verb
also contains information about the usage of the verb in Hindi language, e.g., some verbs in Hindi require the concatenation of the word “nao”‘. All such verbs have been marked ‘2’ and the rest are marked as ‘1’. As the different forms of a particular verb are very similar to each other the verb record uses these similarities to save valuable space. The root word for any verb is taken as base and the rest of the forms stored are generated from it. So the verb eat is stored as “eat-ate-3en-3s-3ing”.

The dictionary for a language L consists of three files Ladojectives.dry, Lnouns.dry and Lverbs.dry. These three binary files contain records mentioned above.

3.3 Analyzer

The analyzer has the task of analyzing a phrase and identifying the function of that phrase with regard to the entire sentence. Currently our system does not need to provide error checking on the input, as the input is guaranteed to be accurate. The analyzer of course does some basic checks of semantic and syntactic regularities. The analyzer consists of a morphological analyzer, syntax analyzer, and a semantic analyzer.

3.3.1 Morphological Analyzer

Morphological analyzer studies the internal structure of words. Words are accepted as being the smallest unit of syntax in many languages; words can be related to other words by rules as described in appendix A. Although a dictionary
alone can handle all kinds of forms of a particular word, maintaining them would be very cumbersome, e.g., a Hindi noun if stored with all form of its plurals will have at least 4 words into it. The morphological analyzer helps us to evaluate all these forms from the root word. The analyzer would be context based and would return the word that is best suited to fit in the sentence.

This system supports simple to complex morphologies to find out the exact use of a particular noun in a sentence. For English language, finding out a plural noun might sound a bit easy, but the regional languages in India support three to four forms of plurals for one word. Which one to be used depends on the context of word in the sentence.

### 3.3.2 Syntax Analyzer and Semantic Analyzer

The syntax analysis transforms linear sequences of words into syntactic structures that show how the words are related to each other. This transformation is done on the basis of a grammar defined in the system. The parser extracts multiple words from the sentence to judge the context of each word and draw a right semantic picture of the sentence. The parser makes several passes of the sentences for filling the case frame slots.

### 3.4 Case Frame

The case frame is the semantic representation of a sentence. The parser with the help of the dictionary, morphologies, syntax analyzer, and the missing words translations maps the syntax of a sentence to a case frame. The parsing
involves locating the main verb, the agent, object etc. of a sentence. All this is done on the basis of occurrence of different prepositions, different attributes of the noun phrase, and the location of the word in the sentence.

### 3.4.1 Structure of Case Frame

The case frame used in our system is as follows:

- Agent, Agent Determinant, Agent Possessor, Agent Adjective, Agent1, Agent Possessor Determinant, Agent Possessor Adjective, Agent Possessor Possessor, Agent Possessor1
- Object, Object Determinant, Object Possessor, Object Adjective, Object1, Object Adjective1, Object Adjective2, Object Possessor Determinant, Object Possessor Adjective, Object Possessor, Object Possessor1
- Dative
- Location
- Source
- Goal
- Co actor, Co actor Determinant, Co actor Possessor, Co actor Adjective
- Beneficiary
- Comparator, Cobject1, Cobject2
- Negative
Verb
Tense

The next sub-sections describe each of these terms with example. In every example cited in the next sub-sections, the particular case being discussed is underlined.

Agent
A noun phrase fills the Agent Case if it describes the instigator of the action that has been described in the sentence. The Agent is almost always specified with its characteristics such as possessor, determinant, or adjective.

For example the sentence “A charming boy attended my boring lecture” has the following case structure.

| AGENT: boy |
| AGENT ADJECTIVE: charming |
| AGENT DETERMINANT: a |

For processing some complex type of sentences certain additional attributes are provided with the agent case. The first one of these is Agent1. This case comes into picture when two nouns are combined to specify a particular agent. e.g., Haridwar city is a holy place.

| AGENT: city |
| AGENT1: Haridwar |

The other special attributes are actually the attributes of the agent possessor. Some times only the agent part of the sentence would span as many as 5-6 words.
An example for the same is given.

The daughter of Alok beautiful sister is playing.

<table>
<thead>
<tr>
<th>AGENT: sister</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT ADJECTIVE: beautiful</td>
</tr>
<tr>
<td>AGENT ADJECTIVE: alok</td>
</tr>
<tr>
<td>AGENT POSSESSOR DETERMINANT: The</td>
</tr>
<tr>
<td>AGENT POSSESSOR POSSESSOR: daughter</td>
</tr>
</tbody>
</table>

The Agent Possessor1 case comes into picture when two successive nouns describe the possessor, as was the case of agent1.

Girls of Delhi City is smarter than girls of Mumbai.

<table>
<thead>
<tr>
<th>AGENT: Girls</th>
</tr>
</thead>
<tbody>
<tr>
<td>AGENT POSSESSOR: City</td>
</tr>
<tr>
<td>AGENT POSSESSOR1: Delhi</td>
</tr>
</tbody>
</table>

**Object**

The object case describes the noun on which the agent is performing action. Like the agent it is also accompanied by extra information. The special attributes Object Determinant, Object Adjective, Object Possessor, Object Possessor, Object1, Object Possessor Adjective, and Object Possessor Determinant for the object are very much similar to that of an agent. An object is always an unanimated noun. An object appears in the latter part of the sentence in the active voice.
The object cases for the following sentence would be as shown.

I broke the beautiful ball of deepa sister.

<table>
<thead>
<tr>
<th>OBJECT</th>
<th>ball</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT ADJECTIVE</td>
<td>beautiful</td>
</tr>
<tr>
<td>OBJECT DETERMINANT</td>
<td>the</td>
</tr>
<tr>
<td>OBJECT POSSESSOR</td>
<td>sister</td>
</tr>
<tr>
<td>OBJECT POSSESSOR POSSESSOR</td>
<td>deepa</td>
</tr>
</tbody>
</table>

The object has been given few extra keywords that were not present with the Agent. An object can hold at most 3 adjectives at a time. This has been done to translate the sentences of following type.

There are five big trees in village.

<table>
<thead>
<tr>
<th>OBJECT</th>
<th>trees</th>
</tr>
</thead>
<tbody>
<tr>
<td>OBJECT ADJECTIVE</td>
<td>big</td>
</tr>
<tr>
<td>OBJECT ADJECTIVE1</td>
<td>five</td>
</tr>
</tbody>
</table>

**Dative**

The dative and the object only have a subtle difference between them. The dative is an animate noun on which the action is being performed. Only one among a dative or an object would exist so the dative informant is stored in the object’s information slots only. The sentence “I killed the lonely dog” will have the following representation.
**DATIV**E: dog

**OBJECT** ADJECTIVE: lonely

**OBJECT** DETERMINANT: the

**Instrument**

Instrument describes a tool, material using which a particular action has been performed by the Agent. The instrument is generally un-animate noun. The sentence “I cut the apple with her long knife” will have following instrument information.

**INSTRUMENT**: knife

**INSTRUMENT** ADJECTIVE: long

**INSTRUMENT** POSSESSOR: her

**Location**

The location indicates the place of the action.

The sentence “I live in Patna” has the following location information.

**LOCATION**: Patna

**Source**

The Source case signifies the place from which something moves. Note that for source no extra information is being stored, e.g., in the sentence “He came from the school” the case frame information would be as shown.
Goal

The Goal case is the place to which something moves. It is just opposite to the Source Case. In the sentence “I went to the school” the case frame would be as follows. Also note like source, no extra information about the Goal is stored.

| GOAL: school |

Coactor

The Coactor case is filled by an animate noun that acts together with the Agent in instigating some action. The choice of which noun becomes the Agent and which becomes the Coactor is merely by appearance which one appears first in the sentence. In the sentence “I played cricket with her daughter” the case frame information for the Co-Actor would be

| COACTOR: daughter |
| COACTOR POSSESSOR: her |

Beneficiary

An animate noun on whose behalf the action is carried out fills the Beneficiary case. For example in “I gave the book to Him”.

| BENEFICIARY: Him |
**Comparator**

The Comparator case is filled by those words that are used to compare two different objects. For example more, less, etc. It exists together with CObject1 and CObject2, the two objects it compares. Usually CObjec1 and CObject2 are nouns, but there are cases when the comparison is between the Agent and CObject2 in which case COobject1 takes the form of an adjective e.g. in “He saw more women than men”.

| COMPARATOR: more |
| COBJECT1 : women |
| COBJECT2 : men |

**Negative**

The negative case is just takes two values 0 or 1. A 0 represents that the sentence is said in the positive sense and 1 means the object has negative sense. The sentence “I am playing soccer” and “I am not playing soccer” have the same case frame structure except that the first has negative case as 0 and the second one a 1 negative case.

**Verb**

The verb Case contains the main verb of the sentence. The system only handles one verb per sentence. In the sentence “I played soccer” the verb case frame information would be

| VERB: play |
Tense

The Tense Case as the name suggests contains the tense of the sentence. As the sentences we are dealing are simple one sentence will have only one verb and one tense. The sentence “I have played soccer” has following case frame structure for the Tense.

**TENSE: present perfect**

The case frame used by the system tries to capture every single detail of a not so complex active voice sentence. There is no end to what all cases a case frame might possess. A third agent could also be added to the case frame. This case frame currently only supports one verb per sentence. A Verb2 case could also be added. The special attributes which have only been provided for the agent and the object could also be added to the coactor, beneficiary, instrument, source, goal or location.

3.5 Generator

Generator functionality is exactly opposite to the case frame generation. It decodes the case frames into the target language. It has its own set of rules and with the help of the dictionary, morphologies, syntax analyzer it converts the case frame into the target language. The case frame or intermediate representation contains all relevant syntactic and semantic information. Once the case frame is generated from any source language the task is to convert the case semantics into a proper English language sentence. The rules for doing the same are as follows.
For every case certain prepositions would be added to make the sentence grammatically correct.

3.5.1 Case Frame to English Rules

**Agent**

The whole agent is processed with its attributes. If the determinant is empty, a “The” has to be added in case it is required. Rules for converting the agent and its attribute cases to the sentence are as follows. The rules only get applied if that particular case exists.

Agent = Agent1 + Agent

Agent = AgentDet + AgentAdjective + Agent

If (AgentPos is a Noun)

If (AgentPosPos doesn’t exist)

Agent = Agent + AgentPosDet + AgentPosAdj + of + AgentPos

Else If (AgentPosPos is Noun)

Agent = Agent + of + AgentPosPos + 's + AgentPosPos

Else

Agent = Agent + of + AgentPosPos + AgentPos

Else

Agent = AgentPos + Agent
Object/Dative
The object / dative are handled like an agent. Depending upon the type of attribute present they are added to the object in that manner. The Object Adjective1 and Object Adjective2 if present are added in front of the Object Adjective.

Coactor
A ‘and’ keyword has to be added for a coactor. The rules to generate the coactor’s sentence are a subset of Agent’s rules. That is
If a possessor is present
   Coactor= and + CoActorPos + Coactor
Else
   Coactor=and + CoactorDet + CoactorAdj + Coactor

Location
If the location is present an “in” is added to it. Therefore
Location=in + Location

Source
If the source is present a “from” is added to it. Therefore
Source=from + Source

Goal
If the goal is present an “in” is added to it. Therefore
Goal=in + Goal
Verb
The case frame holds the tense of the sentence that is being stored as a case frame. This very tense is passed to the dictionary with the verb for the Case to English information to get the exact word for that tense. For example, for the past tense of the case dictionary returns ‘ate’ for the eat verb. The negative case is also processed here. A ‘no’ or ‘not’ is added in front of the verb if negative is 1.

Beneficiary
A beneficiary is always an animate noun. We handle beneficiary by adding a ‘to’ in its front. Beneficiary = to + Beneficiary

Instrument
The instrument has its own set of attributes. The processing is very similar to that of a coactor. The exact formation of Instrument is as follows.

If ( Instrument Possessor exists ) then
Instrument = InsPos + Instrument

Else
Instrument = InsDet + InsAdj + Instrument

Instrument = by/with + Instrument.

Whether to choose by/with is decided by looking at the noun type of the instrument which we have stored in our dictionary.

Comparator
Comparator will only exist in place of an object / dative. Comparator along with the Cobj1 and Cobj2 (1st and 2nd compared objects do not have to be
changed and can be directly put in) by just adding than between Cobj1 and Cobj2.

\[
\text{Comp} = \text{Comp} + \text{Cobj1} + \text{than} + \text{Cobj2}
\]

After all these slots have been generated our final sentence is –

<table>
<thead>
<tr>
<th>Agent</th>
<th>CoActor</th>
<th>Verb</th>
<th>Object</th>
<th>Dative</th>
<th>Comp</th>
<th>Source</th>
<th>Goal</th>
<th>Location</th>
<th>Instrument</th>
<th>Beneficiary</th>
</tr>
</thead>
</table>

3.5.2 Case Frame to Hindi

In this case we have to first evaluate the meaning of the case word and then the morphologies evaluate how the word best fits in the sentence.

**Agent**

Agent, AgentDet, AgentAdj and AgentPos all need to be changed to their appropriate form in the Hindi natural language. If the agent is in plural then morphological processing comes into picture. The rules only get applied when that particular case exists. The key words \text{ka} / \text{ko} / \text{kl} are used for the possessors if they are noun. Which one to be used is evaluated on the basis of the Agent, e.g., if the agent is a singular male then \text{ka} is used. For plural male \text{ko} is used and for a Female \text{kl} is used.

\[
\text{Agent} = \text{Agent1} + \text{Agent}
\]

\[
\text{Agent} = \text{AgentDet} + \text{AgentAdjective} + \text{Agent}
\]

If (AgentPos is a Noun)
If(AgentPosPos doesn’t exist)

Agent=AgentPosDet + AgentPosAdj + AgentPos + ka / ko / 
kl + Agent

Else If (AgentPosPos is Noun)

Agent=AgentPosPos + ka / ko / kl + AgentPos + ka / ko / kl

+Agent

Else

Agent=AgentPosPos +AgentPos + ka / ko / kl + Agent

Else

Agent=AgentPos + Agent

Coactor

The proper Hindi equivalent of the word is retrieved from the dictionary
and appended with ‘ko saaqa’. Again morphology has to be applied in case of
plurals.

Coactor = Coactor + ko saaqa

Verb

According to the root verb and the tense, the ideal verb form is returned
from the dictionary, and after the required morphological process the suitable
verb form is procured. The morphology applied on the verb also applies on the
gender of the agent or the object depending on the form of the verb.
Object / Dative
   This is very similar to the Agent case except that the word kao is appended for the dative case.

Instrument
   After translating the instrument into Hindi the attributes of the instrument are added to it. The word saoo is appended to the instrument.

   If(InsPos doesn’t exist)
      Instrument = InsDet + InsAdj + Instrument + saoo

   Else
      Instrument = InsPos + Instrument + sao

Source
   Source is done by appending the word sao to it.

   Source = Source + sao

Goal
   Goal is done by appending the word tk to it.

   Goal = Goal + tk

Beneficiary
   Beneficiary is done by appending the word kao to it.

   Beneficiary = Beneficiary + kao

Location
   Location is done by appending the word maoM to it.

   Location = Location + maoM
**Comparator**

Comparator will only exist in place of an object/dative. Comparator along with the Cobj1 and Cobj2 (1st and 2nd compared objects). Again the equivalent Hindi words are retrieved and a ‘sao’ is appended after Cobj2, and a ‘kao’ after Cobj1 if it is an animate noun.

The final Hindi sentence after this entire slot filling is –


The above case frame used by the system tries to capture every single detail of a not so complex active voice sentence. This case frame currently only supports one verb per sentence. A Verb2 case could also be added. The special attributes provided for the agent and the object could also be added to the coactor, beneficiary, instrument, source, goal, or location.
4. RESULT AND DISCUSSION

4.1 Results Obtained

Here we have taken all the screen shots of result which shows the system results in translating the source language to the target language and vice-versa and sentences in different tenses form.

![Figure 4.1: English to Hindi translation](image)

There are five big trees in makanpur. His wife has a beautiful dog. His name is bruni. I am playing soccer. I went to the market in a bus. I came from the school in a bus. I broke the windows with big hammers. I will go to the market in a bus. I am reading Shyan's book. My son played soccer with her daughter. Her daughter is wearing my dress. Condition of roads is bad in makanpur. His father will not play chess with my beautiful daughter. Amitabh's sister killed the dog with a long stick.
Figure 4.2: Hindi to English translation

Hindi

कानपुर में पाँच घंटे पह थे। उसकी पत्नी के नाम एक पुनर्नाम कुल है। उसका नाम कुल है। वे फुटबॉल में खेल रहा है। वे दस में खेल रहा था। उसने फूटबॉल की खेल रहा है। उसने उसे फूटबॉल खेलने के लिए दिया था। उसका खेल फूटबॉल खेलने के लिए दिया था। उसका खेल फूटबॉल खेलने के लिए दिया था।

There are big trees in makanpur. Dog of his wife is beautiful. His name is baruni. I am playing football. I went to market in the bus. I came from the school in the bus. I broke windows with the big hammers. I shall go to market in the bus. I am studying book of shyam. My son played football of his daughter. His daughter is wearing my dress. Condition of road is bad in makanpur. His father will not play chess of beautiful daughter.
4.2 English - Hindi Translation Results

![Figure 4.3: Present Tense English-Hindi Translation](image1)

I eat food. He eats food.

मैं खाता हूँ। वह खाता है।

![Figure 4.4: Present Continuous Tense English-Hindi Translation](image2)

I am eating food. He is eating food.

मैं खाता रहा हूँ। वह खाता रहा है।
Figure 4.5: Past Tense English-Hindi Translation

Figure 4.6: Past Tense English-Hindi Translation
Figure 4.7: Past Tense English-Hindi Translation

I had eaten food. He had eaten food.

मे ने खाना खाया था। उसे ने खाना खाया था।

Figure 4.8: Future Tense English-Hindi Translation

I shall eat food. He will eat food.

मैं खाना खाऊँगा। उसे ने खाना खाएँगा।
Figure 4.9: Future Tense English-Hindi Translation

I shall be eating food. He will be eating food.

Figure 4.10: Condition Tense English-Hindi Translation

I should eat food if it were safe. He would eat food if it were safe.
I should be eating food if it were safe. He would be eating food if it were safe.

न काना खाना बंधक या। को काना सही या।
4.3 Hindi – English Machine Translation Results

Figure 4.12: Present Tense Hindi - English Translation

Figure 4.13: Present Tense Hindi-English Translation
Figure 4.14: Past Tense Hindi - English Translation

Figure 4.15: Past Tense Hindi - English Translation
Figure 4.16: Past Tense Hindi - English Translation

Figure 4.17: Future Tense Hindi - English Translation
Figure 4.18: Future Tense Hindi - English Translation

Figure 4.19: Hindi - English Translation
5. TESTING AND EVALUATION

5.1 System Performance

The objective of this system is language translation processing. This means the capability to translate between source language that is English into a target language that is Hindi and vice versa. The MT system employs the interlingua system of translation processing. Interlingua is language that replaces a language’s expressions, sentence semantics, and forms of expression with common forms of expression that are independent of any language. The performance of this system is measured by testing the meaning of sentences whether they remain same in the source language as well as the target language. The system has been coded in C++ and Qt libraries have been used for the GUI of the translator. The translator has two modes for translating the sentences and paragraph from a source language, i.e., English to a target language, i.e., Hindi.

5.2 Unit Testing

This type of testing is done to determine that individual program modules perform to specification. Each module is tested alone in an attempt to discover any errors in its code. It is ideal to develop the software applications components and be able to dynamically test each component individually by considering the input/outputs. The goal of unit testing is to isolate each part of the program and show that the individual parts are correct. Unit testing allows the programmer to re-factor code at a later date, and make sure the module still works correctly (regression testing). Unit testing helps eliminate uncertainty in the pieces themselves and can be used in a bottom-up testing style.
approach. By testing the parts of a program first and then testing the sum of its parts will make integration testing easier [Hunt, 2003].

5.3 Regression Testing

“Test everything tested before”- the purpose of regression testing is to detect unexpected faults especially those that occur because a developer did not fully understand the internal code correlations when modifying or extending code. Every time code is modified or used in a new environment, regression testing should be used to check the code's integrity. Regression testing can be used not only for testing the correctness of a program, but it is also often used to track the quality of its output. This is the only way to validate whether the new amendments to the software have introduced any flaws to the functionality working previously.

![Figure 5.1: English to Hindi translation](image-url)
Finally after the system was developed and tested with different sets of sentences it was assumed that the translation correctness of the system would be around 80%. This efficiency can increase with inclusion of more word in dictionary and also with the consideration of other attributes of the sentences.
6. CONCLUSION

In this system design, an approach based on artificial intelligence techniques for building the semantic representation for simple sentences is considered. It is based on a verb-centered, phase level approach, and it focused on the meaning of the input sentence for building a representational structure. The phrases identified in this approach were Noun phrase and Verb phrase. Of the conventional knowledge representation formalisms, the frame structure is used for representing the meaning of a sentence. This language free representation is used by the translator and generator modules to produce the target language sentence after appropriate transformations.

6.1 Further Scope

The system design is a prototype of an exhaustive MT system. The system can handle a paragraph of sentences which are independent. The system can be further improved to incorporate discourse analysis. The meaning of a sentence in the paragraph may depend on the sentences that precede it. For example, the word ‘it’ in the sentence, ‘John wanted it’ depends on prior discourse context. The system can be enhanced to integrate such an analysis. Moreover, there is tremendous scope of improvement in the grammar. The grammar can be expanded to handle complex sentences.

Currently there is no provision in the system to handle words with different meanings. Either the system can be made smart enough to incorporate
techniques to realize the exact word which is being used or the system can ask
the user to enter the word which is currently being used and learn.

6.2 Limitations

The system will not be able to handle complex sentences. Also no
discourse analysis can be performed as the system has been designed to translate
only a single sentence at a time. Apart from these limitations, the system
addresses only a very limited number of verb complement forms. All the other
verb complement forms are not covered. As soon as a word appears which has
two different meanings the system would only take the word which is first found
in the dictionary.

However these limitations should not be considered much of a
compromise. These features can be incorporated easily and the system design
improved upon, as the basic design has been done keeping such expansions in
mind.

[Chellamuthu KC] Dr. KC Chellamuthu. Russian to Tamil Machine Translation System. Tamil University. San Jose, CA, USA


[Quirk 1976], A University Grammar of English, English Language Book Society, Longman.


APPENDIX A

1. English and Hindi Language Variations

English and Hindi languages are of two different origins, so study of their general structural properties is necessary. In this discussion, some of the basic concepts of the translation from English to Hindi are briefly outlined. Some of the general structural properties of English and Hindi (Kachru, 1980) (Kellogg and Bailey, 1965) (Singh, 2003) (Qurick and Greenbaum, 1976) are described below. For example,

- **Sentence Pattern:** The basic sentence pattern in English is Subject (S) Verb (V) Object (O), whereas it is SOV in Hindi. Consider for example “Radha eats mango” here “Radha” is subject; “eats” is the verb while “mango” is the object. So the words occur in the order SVO. But in Hindi it becomes

  radha (S) aama (O) khaatii hai (V)

  Radha mango eats

- **Order of Words in a Sentence:** English is a positional language and is therefore has (relatively) fixed order. Relations between various components of the sentence are mainly shown by the relative position of the components. Consider this example as:

  Radha watches the sparrows.

  is very different from

  The sparrows watch Radha.

Hindi is (relatively) free-order. Relations between various components of the
sentence are mainly shown by inflecting the components. Changes of position of components normally change the emphasis of an utterance, and not the basic meaning.

2. Hindi and English Language Variations

For Example: *radha chidiyaan dekhatii hai*

(Radha) (sparrows) (watches)

has the same meaning

*chidiyaan radha dekhatii hai*

(sparrows) (Radha) (watches)

Above mentioned differences are structural differences between English and Hindi. Some differences are in the part of speech properties of English and Hindi languages. These discrepancies are as follows:

- **Noun**: Hindi nouns are affected by gender, number and case ending (Kellogg and Bailey, 1965). These are as follows:

  1. **Gender**: English has four genders-*masculine* (MASC), *feminine* (FEM), *common* and *neuter*, whereas Hindi has only two-*masculine* and *feminine*. The neuter gender of Sanskrit (Origin of Indian languages), Hindi as well as the closely related languages, has vanished.

  2. **Number**: As English, Hindi also has two numbers- Singular and Plural.

There are some possible suffixes for singular to plural conversion in Hindi, which are as
follows (Kellogg and Bailey, 1965):

For example:

<table>
<thead>
<tr>
<th>Singular</th>
<th>Plural</th>
</tr>
</thead>
<tbody>
<tr>
<td>ladkaa - boy (MASC)</td>
<td>ladke - boys</td>
</tr>
<tr>
<td>ghar - house (MASC)</td>
<td>ghar - houses (No change)</td>
</tr>
<tr>
<td>kapadaa - cloth (MASC)</td>
<td>kapade - clothes</td>
</tr>
<tr>
<td>ladkii - girl (FEM)</td>
<td>ladkiyaan - girls</td>
</tr>
<tr>
<td>kakshaa-class(FEM)</td>
<td>kakshayen- classes</td>
</tr>
</tbody>
</table>

3. **Case ending**: There are eight case endings in Hindi, which are given below in Table A.1. All these are appended to the oblique form of the noun, where such a form exists.

There are some rules for making oblique nouns.

Some of them are as follows:

(a) Masculine singular nouns ending in “aa” change into “e” when some case ending is added: e.g. ladkaa + ne ~ “ladke ne”. Nouns ending in other vowels do not undergo such changes e.g. “ghar ko”, “daaku kaa”.

(b) If a noun (masculine or feminine) ends in “a”, it is changed into “aon” in plural, when a case ending is added. For example: “in the house” ~ “ghar mein” while “in the houses” ~ “gharon main”. Note that, normally the plural of “ghar” is “ghar”, but it changes to “gharon” in the above example because
The addition to the oblique form of noun of certain particles is commonly called postposition.

**Table A.1: Different Case Ending in Hindi**

<table>
<thead>
<tr>
<th>Case</th>
<th>Case-endings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nominative case</td>
<td>ne</td>
</tr>
<tr>
<td>Accusative case</td>
<td>ko</td>
</tr>
<tr>
<td>Agent case</td>
<td>se(by, with and through)</td>
</tr>
<tr>
<td>Dative case</td>
<td>ko(to), ke liye (for), ke waste</td>
</tr>
<tr>
<td>Ablative case</td>
<td>se (from, since)</td>
</tr>
<tr>
<td>Possessive case</td>
<td>kaa, ke, kii</td>
</tr>
<tr>
<td>Locative case</td>
<td>mein, par (in, on )</td>
</tr>
<tr>
<td>Vocative case</td>
<td>he, ajii, are</td>
</tr>
</tbody>
</table>

No postposition is used with the nominative and Vocative. Here we will discuss three cases nominative, accusative and possessive case. Other cases work same as English case ending. These cases as follows:

1. **Nominative case:** The subject of a sentence takes the nominative sign “ne” only when its predicative is a transitive verb in the past tense (past indefinite, present perfect and past perfect). The use of this case is to make a noun or pronoun act as subject of a verb. In that case, verb agrees with the object in gender and number. For example,
Ram narrated a story. ~ ram ne kahaanii sunaayii

(Ram) (story) (narrated)

The farmer has sowed the seeds. ~ kisaan ne biij boyee hain

(farmer) (seeds) (sowed has)

Here in these two examples the objects of translated sentence are “kahaanii”, “biij”. The number and gender of these nouns are singular feminine and plural masculine, respectively.

2. **Accusative case**: “ko” is the sign of this case and it is generally added only to animate objects. Sometimes it is also added to inanimate objects, either to intensify its effect or to express a special significance. For example:

**The boy beats the dog:** ~ ladkaa kutte ko martaa hai

(boy) (dog) (beats)

3. **Possessive case**: The signs of this case are “kaa”, “ke” and “kii”. These words are used with noun according to gender, number and case-ending of the following noun.

- **Preposition-Postposition**: In English, preposition occurs before the noun, e.g. “on the table”, “in the box”. But in Hindi it occurs after the noun (e.g., “meja (table) par (on)” –
“on the table”), and hence this may be called postposition instead of preposition.

However, for ease of understanding we shall call them “preposition” only.

- **Article**: As Hindi has no article, the distinction indicated in English by the definite and indefinite articles cannot always be expressed in Hindi. As “ghodha” may be either “a horse” or “the horse”; “istriyaan” may be “women” or “the women”. The indefinite article may sometimes be rendered by the numeral “aka”, “one”, or the indefinite pronoun, “any” ~ “koyii”, “some ~ kuchh”.

### 3. Verb Morphological and Structure Variations

Every language has its own grammar rules. In other words, we can find same sentence following different grammatical aspects corresponding to the language concerned. For example, consider an English sentence “He will be sleeping at the moment”. Its translation in Hindi is “wah iss samay so rahaa hogaa”. As per English grammar rules, verb phrase follows future tense and progressive aspect (or continuous aspect) but at the same time Hindi sentence verb phrase comes under definite potential type of mood according to Hindi grammar. For the translation work, we have followed English grammar categorization for verb phrase structure (Quirk and Greenburm, 1976) which involves different combination of tense, aspect and mood. To understand English to Hindi verb structure, conjugation of root verb in Hindi has been presented in the following subsection.

#### 3.1 Conjugation of Root Verb

Verb morphological variations in Hindi depend on four aspects: **tense** and **form** of
the sentence, *gender* of the subject, *person* of the subject and *number* of the subject. All these variations affect the root verb of a sentence. Since there are three tenses (i.e. Present, Past and Future) and four forms (i.e., Indefinite, Continuous, Perfect, and Perfect Continuous), in all one can have 12 different conjugations. In Hindi, these conjugations are realized using suffixes attached to the root verbs, and/or adding some auxiliary verbs, which we call “Morpho-Words” (MW). Table A.2 gives the total number of morphological words and suffixes in Hindi, for all the tenses and their forms.

**Table A.2: Suffixes and Morpho-Words for Hindi Verb Conjugations**

<table>
<thead>
<tr>
<th>Tense form</th>
<th>Present Tense</th>
<th>Past Tense</th>
<th>Future Tense</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indefinite</td>
<td>Suffix: taa, tii, te MW: hoon, hai ho, hain</td>
<td>Suffix: taa, tii, te MW: thaa, thii the</td>
<td>Suffix: oongaa, Oongii, oge, ogii, egaa, egii, enge, engii</td>
</tr>
<tr>
<td>Continuous</td>
<td>MW: rahaa, rahe, Rahii, hoon, hai, ho, hain</td>
<td>MW: rahaa, rahe, Rahii, thaa, the, thii</td>
<td>MW: rahaa, rahe, rahii, hoongaa, hoongii, hoonge, hogaa, hogii, hoge</td>
</tr>
<tr>
<td>Prefect</td>
<td>MW: hoon, hai, hain, Ho, chukaa, chukii, chuke</td>
<td>MW: thaa, the, thii, chukaa, chukii, chuke</td>
<td>MW: hoongaa, hoongii, hoonge, hogaa, hogii, hoge, hongee, chukaa,</td>
</tr>
<tr>
<td>Perfect Continuous</td>
<td>Same as Continuous</td>
<td>Same as Continuous</td>
<td>Same as Continuous</td>
</tr>
</tbody>
</table>

Above suffixes and morphological words in present prefect, past indefinite and past prefect are used for literal translation of a sentence. Actually conjugation in root verb
is “aa”, “e” and “ii”. It has been observed that according to Table A.2 suffixes \{taa, te, tii\} are added in the root form of past indefinite tense form. According to the tense forms, the morpho-words \{thaa, the, thiig\}, \{chukaa, chukii, chuke\} and \{hoon, hai, ho, hain\} are added after the main verb of the sentence.

Another possible way of expressing these three tenses and forms in Hindi is that, in place of above mentioned suffixes different conjugations of verbs is used that is different from the verb of tenses and forms discussed earlier. The morpho words \{thaa, the, thii\} or \{hoon, hai, ho, hain\} is added depending upon the tense towards the end of the sentence.

Some rules of these conjugations of verbs are as follows (Sastri and Apte, 1968):

- If the root of the verb ends in “a” (silent) lengthen it to “aa” in masculine singular and change it into “e” for masculine plural; in feminine singular it becomes “ii” and in feminine plural “iin”. For example the verb “play” – “khelaa” is in Hindi khelaa (masculine singular), khelii (feminine singular), khele (masculine plural) and kheliin (feminine plural).

- If the root ends in “aa” or “oo”, “yaa” is added, which changes according to the “aa”, “ai” and “ii” rule 1. Sometimes “e” is used in place of “ye”; and “ii” and “iin” in the place of “yii” and “yiin”, respectively. For example, the verb is “come” – “aa”, in masculine “aayaa” (singular) and “aaye” or “aae” (plural), and in feminine “aayii” or “aaii” (singular) and “aayiin” or “aaiin” (plural).
If the verb-root ends in “uu”, change it into “u” and add “aa” and “e” in masculine and “ii” and “iin” in feminine. For example the verb is “touch”, in masculine “chhuuaa” and “chhue”, and in feminine “chhuii” and “chhuiin”.

These rules are defined as PCP verb form rules. Of the English verb group, above mentioned morpho word and suffixes are called the morphological transformations in Hindi. Table A.3 provides some conjugation of verb “write”, in a view of the systematize knowledge which has been given in Table A.2.

Similar discussion can be done for the passive verb form also. Passive form can be formulated for transitive verbs only. The morphological variation depends on the gender and number of the object of the active form of the sentence that is basically the subject in the passive form (Sastri and Apte, 1968). The subject of the active form occurs in the passive forms as the instrumental case followed by “by” and its Hindi is either “se, “ke duwaraa” or “duwaraa”. In passive form, the changes in the main verb are according to the rules of PCP form of verb as discussed in the section A.1. Moreover an extra verb “jaa” is introduced after the main verb and the suffixes that are given in Table A.3 are added in this additional verb instead of the main verb of the sentence. The morpho words are added after the conjugation of verb “jaa”.

Suppose the set of examples are:

we add sugar to milk.

\[ \sim ham dudh mein shakkar daalte hain \]

(we) (milk) (in) (sugar) (add)

Sugar is added to milk by us
The first example is in the active form and the second example in the passive form. The verb morphological changes are according the above discussion.

**Table A.3: Verb Morphological Changes from English to Hindi Translation**

<table>
<thead>
<tr>
<th>English Sentence</th>
<th>Gender</th>
<th>Tense</th>
<th>Hindi Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>I am writing a letter</td>
<td>M/F</td>
<td>Present continuous</td>
<td>Mai part likh rahaa hoo&lt;br&gt;Mai part likh rahee hoon</td>
</tr>
<tr>
<td>You write a letter</td>
<td>M/F</td>
<td>Present indefinite</td>
<td>Tum part likhte ho&lt;br&gt;Tum part likhtii ho</td>
</tr>
<tr>
<td>I write a letter</td>
<td>M/F</td>
<td>Present indefinite</td>
<td>Main part likhtii hoon&lt;br&gt;Main part likhtaa hoon</td>
</tr>
<tr>
<td>He (she) was writing a letter</td>
<td>M/F</td>
<td>Past continuous</td>
<td>Wah part likh rahaa thaa&lt;br&gt;Wah part likh rahii thii</td>
</tr>
<tr>
<td>We will write a letter</td>
<td>M/F</td>
<td>Future indefinite</td>
<td>Hum part likhenge&lt;br&gt;Hum part likhengii</td>
</tr>
<tr>
<td>Site wrote a letter</td>
<td>F</td>
<td>Past indefinite</td>
<td>Sita ne patra likhaa</td>
</tr>
</tbody>
</table>
APPENDIX B

1. **Overview of TRIE**

   In computer science, a trie, or prefix tree, is an ordered tree data structure that is used to store an associative array where the keys are strings. Trie is a multi way tree structure useful for storing strings over an alphabet. It has been used to store large dictionaries of English (say) words in spelling-checking programs and in natural language programs. Unlike binary search tree no node in the tree stores the key associated with that node; instead, its position in the tree shows what key is associated with the node. All the descendants of any one node have a common prefix of the string associated with that node, and the root is associated with the empty string. Values are normally not associated with every node, only with leaves and some inner nodes that happen to correspond to keys of interest. [Knuth 1997]

2. **Dictionary representation**

   Some common application of a trie is storing a dictionary, such as one found on a mobile telephone. Such applications take advantage of a trie’s ability to quickly search for, insert, and delete entries. The following pseudo-code represents the general algorithm for determining whether a given string is in trie. Note that children are an array of a node’s children and a terminal node is one which contains a valid word.
Example:

Given the data “an, ant, all, allot, alloy, aloe, are, ate, be”. The corresponding trie would be:

```
function find(node, key) {
    if (key is an empty string) {  // base case
        return is node terminal?
    } else {  // recursive case
        c = first character in key  // this works because key is not empty
        tail = key minus the first character
        child = node.children[c]
        if (child is null) {  // unable to recurse, although key is non-empty
            return false
        } else {
            return find(child, tail)
        }
    }
}
```

Figure 7.1: Trie Example [Allison 2006]
The idea is that all strings sharing a common stem or prefix hang off a common node. When the strings are words over \{a…z\}, a node has at most 27 children, one for each letter plus a terminator. The element in a string can be recovered in a scan from the root to the leaf that ends a string. All strings in the trie can be recovered by a depth-first scan of the tree [Allison 2006]
# APPENDIX C

## A. Source Code

<table>
<thead>
<tr>
<th>File Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnglishRecord.cpp</td>
<td>Basic English Record contains English Adjective, Noun and Verb.</td>
</tr>
<tr>
<td>HindiRecord.cpp</td>
<td>Basic Hindi Record contains Hindi Adjective, Noun and Verb.</td>
</tr>
<tr>
<td>TrieNode.cpp</td>
<td>The file contains a Trie Node used in the main Trie</td>
</tr>
<tr>
<td>FileNode.cpp</td>
<td>The file contains a basic record that is written in the index.</td>
</tr>
<tr>
<td>Trie.cpp</td>
<td>The implementation of the Trie</td>
</tr>
<tr>
<td>WriteEnglishFile.cpp</td>
<td>Module to write the English dictionary</td>
</tr>
<tr>
<td>WriteHindiFile.cpp</td>
<td>Module to write the Hindi dictionary</td>
</tr>
<tr>
<td>WriteEnglishTrie.cpp</td>
<td>Module to put down the index for the English Dictionary</td>
</tr>
<tr>
<td>WriteHindiTrie.cpp</td>
<td>Module to put down the index for the Hindi Dictionary</td>
</tr>
<tr>
<td>CaseToEnglish.cpp</td>
<td>Case To English Translation</td>
</tr>
<tr>
<td>CaseToHindi.cpp</td>
<td>Case To Hindi Translation</td>
</tr>
<tr>
<td>EnglishToCase.cpp</td>
<td>English To Case Translation</td>
</tr>
<tr>
<td>HindiToCase.cpp</td>
<td>Hindi To Case Translation</td>
</tr>
<tr>
<td>findec.cpp</td>
<td>The bridge between English To Case analyzer and the dictionary</td>
</tr>
<tr>
<td>findce.cpp</td>
<td>The bridge between Case To English analyzer and the dictionary</td>
</tr>
<tr>
<td>findhc.cpp</td>
<td>The bridge between Hindi To Case analyzer and the dictionary</td>
</tr>
<tr>
<td>findch.cpp</td>
<td>The bridge between Case To Hindi analyzer and the dictionary</td>
</tr>
<tr>
<td>toshusha.cpp</td>
<td>English to Shusha Font Translator</td>
</tr>
<tr>
<td>File</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>toenglish.cpp</td>
<td>Shusha to English Font Translator</td>
</tr>
<tr>
<td>ec.cpp</td>
<td>English To Case Analyzer</td>
</tr>
<tr>
<td>ce.cpp</td>
<td>Case to English Generator</td>
</tr>
<tr>
<td>hc.cpp</td>
<td>Hindi To Case Analyzer</td>
</tr>
<tr>
<td>ch.cpp</td>
<td>Case to Hindi Generator</td>
</tr>
<tr>
<td>stoeform.cpp</td>
<td>Shusha To English Translator GUI</td>
</tr>
<tr>
<td>etosform.cpp</td>
<td>English To Shusha Translator GUI</td>
</tr>
<tr>
<td>mainform.cpp</td>
<td>The Main GUI</td>
</tr>
<tr>
<td>etableform.cpp</td>
<td>The GUI to display English tabular files</td>
</tr>
<tr>
<td>htableform.cpp</td>
<td>The GUI to display Hindi tabular files</td>
</tr>
</tbody>
</table>