ENGLISH TO HINDI STATISTICAL MACHINE TRANSLATION SYSTEM

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Engineering
in
Software Engineering

Submitted By:
Nakul Sharma
(800931012)

Under the supervision of:
Parteek Bhatia
Assistant Professor

Varinderpal Singh
System Analyst

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2011

CERTIFICATE

I hereby certify that the work which is being presented in the thesis entitled, "*English to Hindi Statistical Machine Translation System*", in partial fulfillment of the requirements for the award of degree of Master of Engineering in *Software Engineering* submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of *Mr. Parteek Bhatia* and *Mr. Varinderpal Singh* and refers other researcher's work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

(Nakul Sharma)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

(Parteek Bhatia)
Assistant Professor
Computer Science and Engineering Department,
Thapar University, Patiala.

(Varinderpal Singh)
System Analyst
Computer Science and Engineering Department,
Thapar University, Patiala.

Countersigned by

(Dr. Maninder Singh)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

No volumes of words are sufficient to express my gratitude towards my guides Mr. Parteek Bhatia, Assistant Professor and Mr. Varinderpal Singh, System Analyst, Computer Science and Engineering Department of Thapar University. Their valuable advice, motivation, guidance, encouragement, moral support, sincere efforts were instrumental in completion of this thesis.

I am also thankful to Dr. Maninder Singh, Head of the Computer Science and Engineering Department and Mr. Karun Verma and Mr. Sumit Migani, P.G. coordinators, for the motivation and inspiration given. I am thankful for their invaluable suggestion and commitment. Their assistance helped a long way in finishing this thesis.

I would also like to thank the faculty and staff members of Computer Science and Engineering Department for their kind and valuable support which was indispensable towards this thesis completion.

I would also like to thank my friends, family members and relatives for all their love, guidance, encouragement and overall support. Last but not the least; I would like to thank God being there with me times of crises.

Abstract

Machine Translation is an important part of Natural Language Processing. It refers to using machine to convert one natural language to another. Statistical Machine Translation is a part of Machine Translation that strives to use machine learning paradigm towards translating text. Statistical Machine Translation consists of Language Model (LM), Translation Model (TM) and decoder.

In this thesis, English to Hindi Statistical Machine Translation system has been developed. The development of Language Model, Translation Model and decoder is done by making use of software's available in Linux environment. SR International's Language Model (SRILM) for Language Model, GIZA++ and mkcls for Translation Model, Moses for decoding, has been used in this system.

LM computes the probability of target language sentences. TM calculates the probability of target sentences given the source sentence and the decoder maximizes the probability of translated text of target language.

A parallel corpus of 5000 sentences in English and Hindi has been used in training of the system. The system was evaluated using manual evaluation method and a geometric average score of 2.693, 2.93 on the parameters of fluency and adequacy respectively, were found.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

Chapter 1
Introduction

The technology is reaching new heights, right from conception of ideas up to the practical implementation. It is important, that equal emphasis is put to remove the language divide which causes communication gap among different sections of societies. Natural Language Processing (NLP) is the field that strives to fill this gap. Machine Translation (MT) mainly deals with transformation of one language to another. Coming to the MT scenarios in India, it has enormous scope due to many regional languages of India. It is pertinent that majority of the population in India are fluent in regional languages such as Hindi, Punjabi *etc.* [9]. Given such a scenario, MT can be used to provide an interface of regional language.

1.1 Machine Translation

Machine Translation (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 [24]. At its basic level, MT performs simple substitution of words in one natural language for words in another. Current machine translation software often allows for customization by domain or profession (such as weather reports), improving output by limiting the scope of allowable substitutions. This technique is particularly effective in domains where formal or formulaic language is used. It follows that machine translation of government and legal documents more readily produces usable output than conversation or less standardized text [24].

1.1.1 Need for MT

Machine Translation system are needed to translate literary works which from any language into native languages. The literary work is fed to the MT system and translation is done. Such MT systems can break the language barriers by making available work rich sources of literature available to people across the world.

MT also overcomes the technological barriers. Most of the information available is in English which is understood by only 3% of the population [14]. This has let to digital divide in which only small section of society can understand the content presented in digital format. MT can help in this regard to overcome the digital divide.

1.1.2 Problems in MT

There are several structural and stylistic differences among languages, which make automatic translation a difficult task. Some of these issues are as follows.

Word order

Word order in languages differs. Some classification can be done by naming the typical order of subject (S), verb (V) and object (O) in a sentence [23]. Some languages have word orders as SOV. The target language may have a different word order. In such cases, word to word translation is difficult [15]. For example, English language has SVO and Hindi language has SOV sentence structure.

Word sense

The same word may have different senses when being translated to another language. The selection of right word specific to the context is important [15].

Pronoun Resolution

The problem of not resolving the pronominal references is important for machine translation. Unresolved references can lead to incorrect translation [15].

Idioms

An idiomatic expression may convey a different meaning, that what is evident from its words. For example, an idiom in English language *'Jack of all trades'*, would not convey the intend meaning when translated into Hindi language [15].

Ambiguity

In computational linguistics, Word Sense disambiguation (WSD) is an open problem of natural language processing, which governs the process of identifying which sense of a word (i.e. meaning) is used in a sentence, when the word has multiple meanings [15].

1.1.3 Types of Machine Translation Systems

The following are four types of Machine Translation (MT) systems:

MT for Watcher (MT-W)

MT for watchers is intended for readers who wanted to gain access to some information written in foreign language who are also prepared to accept possible bad 'rough' translation rather than nothing. This was the type of MT envisaged by the pioneers. This came in with the need to translate military technological documents [26].

MT for revisers (MT-R)

MT for revisers aims at producing raw translation automatically with a quality comparable to that of the first drafts produced by human. The translation output can be considered only as brush-up so that the professional translator can be freed from that boring and time consuming task [26].

MT for translators (MT-T)

MT for translator's aims at helping human translators do their job by providing on-line dictionaries, thesaurus and translation memory. This type of machine translation system is usually incorporated into the translation work stations and the PC based translation tools [26].

MT for Authors (MT-A)

MT for authors aims at authors wanting to have their texts translated into one or several languages and accepting to write under control of the system or to help the system disambiguate the utterance so that satisfactory translation can be obtained without any revision [26].

1.1.4 Approaches to MT

There are four approaches to Machine Translation. These are discussed as follows.

- Rule-based MT

A Rule-based MT system parses the source text and produces an intermediate representation, which may be a parse tree or some abstract representation [15].

- Direct-based MT

A direct-based MT system carries out word-by-word translation with the help of a bilingual dictionary, usually followed by some syntactic rearrangement [15].

- Corpus-based MT

Corpus based MT systems require sentence-aligned parallel text for each language pair. The corpus based approach is further classified into statistical and example-based machine translation approaches [15].

- Knowledge-based MT

Early MT systems are characterized by the syntax. Semantic features are attached to the syntactic structures and semantic processing occurs only after syntactic processing. Semantic-based approaches to language analysis have been introduced by AI researchers. The approached require a large knowledge-base that includes both ontological and lexical knowledge [15].

The Statistical Machine Translation (SMT) is part of corpus based Machine Translation. SMT requires less human effort to undertake translation. SMT is a machine translation paradigm where translations are generated on the basis of statistical models. These statistical models parameters are derived from the analysis of bilingual text corpora. In this thesis, proposed English to Hindi Statistical Machine Translation system has been presented. The translation system has three modules: Language Model, Translation model, and Decoder. For Language Model, SR International's Language Model toolkit (SRILM) is used to develop Language Model for Hindi. Hindi is the target language for the system. To develop Translation Model, open source GIZA++ software is used. To perform decoding, open source Moses software is used.

1.2 Organization of thesis

This thesis is organized into six chapters. Chapter 2 discusses the review of literature on Statistical Machine Translation. Chapter 3 discusses the details about problem

statement along with the gap analysis. Chapter 4 details of system design and implementation of the proposed English to Hindi Statistical Machine Translation system. The result and discussion, on the basis of the implementation of the proposed system, has been discussed in Chapter 5. The conclusion and future scope, of the system which has been developed, is discussed in Chapter 6.

Chapter 2

Literature Review

In this thesis, English to Hindi SMT system is developed. The literature review, for this work is discussed in next sections.

2.1 MT approaches

MT systems can be classified according to the means by which they perform translation. A classification of MT approaches is given in Figure 2.1 [14]. The MT systems can also be classified according to use of traditional or modern technology.

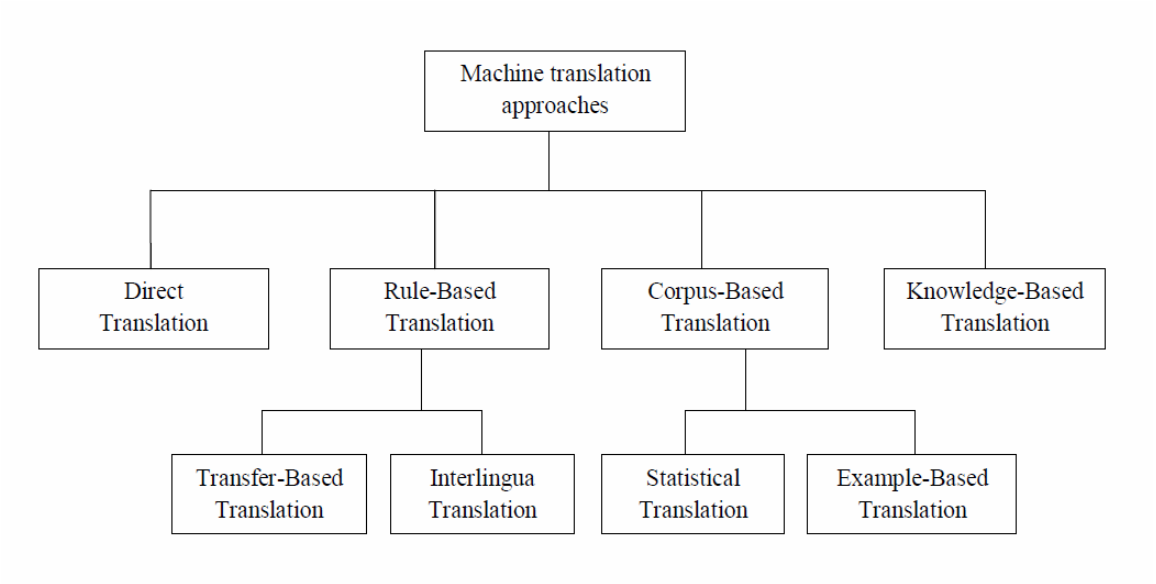

Figure 2.1: Machine Translation approaches [15]

2.1.1 Direct MT

Direct MT form of MT is the most basic one. It translates the individual words in a sentence from one language to another using a two-way dictionary. It makes use of very simple grammar rules [25]. These systems are based upon the principle that as MT system should do as little work as possible. Direct MT systems take a monolithic approach towards development, *i.e.*, they consider all the details of one language pair. Direct MT has following characteristics:

- Little analysis of source language
- No parsing
- Reliance on large two-way dictionary

The general procedure for direct translation systems can be summarized as shown in Figure 2.2. The direct MT system starts with morphological analysis. Morphological analysis removes morphological inflections from the words to get the root word from the source language words. The next step in direct MT system is bilingual dictionary lookup. A bilingual dictionary is looked up to get the target-language words corresponding to the source-language words. The last step in direct MT system is syntactic rearrangement. In syntactic rearrangement, the word order is changed to that which best matches the word order of the target language [15].

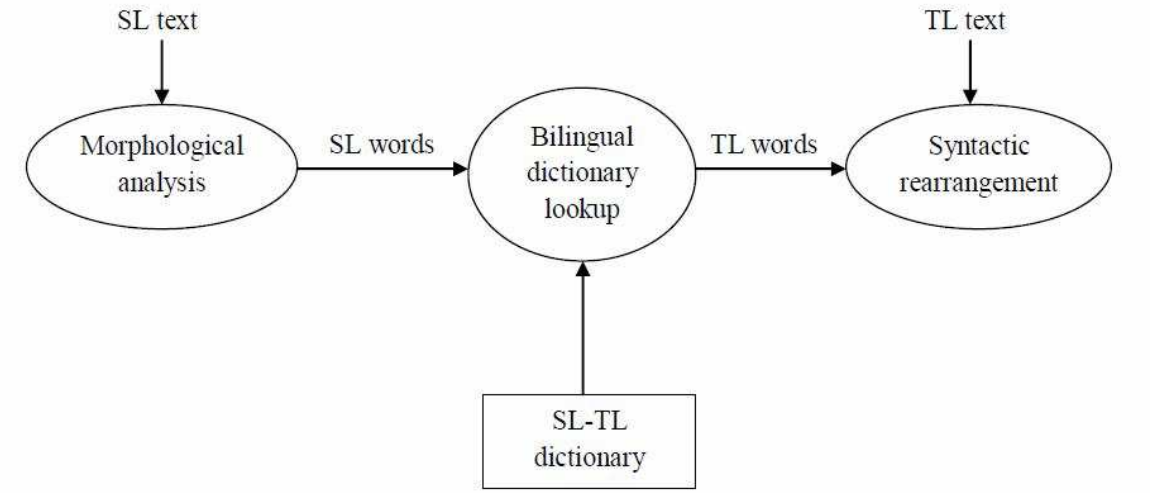

Figure 2.2: Direct Machine Translation [15]

Direct Machine Translation works well with languages which have same default sentence structure.

Advantages of Direct MT

The Direct MT systems have below mentioned advantages.

- Translation is usually comprehended by the reader with little effort [25].

Disadvantage of Direct MT

The Direct MT systems have following disadvantages.

- Direct MT involves only lexical analysis. It does not consider structure and relationships between words.
- Direct MT systems are developed for a specific language pair and cannot be adapted for different language pairs.
- Direct MT systems can be quite expensive, for multilingual scenarios [15].
- Some of the source text meaning can be lost in the translation [25].

2.1.2 Rule-based MT

In rule-based systems, the source text is parsed and an intermediate representation is produced. The target language text is generated from the intermediate representation. These systems rely on the specification of rules for morphology, syntax, lexical selection and transfer, semantic analysis and generation [14].

- **Transfer based MT**

In this translation system, a database of translation rules is used to translate text from source to target language. Whenever a sentence matches one of the rules, or examples, it is translated directly using a dictionary. It goes from the source language to a morphological and syntactic analysis to produce a sort of Interlingua on the base forms of the source language, from this it translates it to the base forms of the target language and from there a better translation is made to create the final step in the translation [25]. The steps which are performed are shown in Figure 2.3.

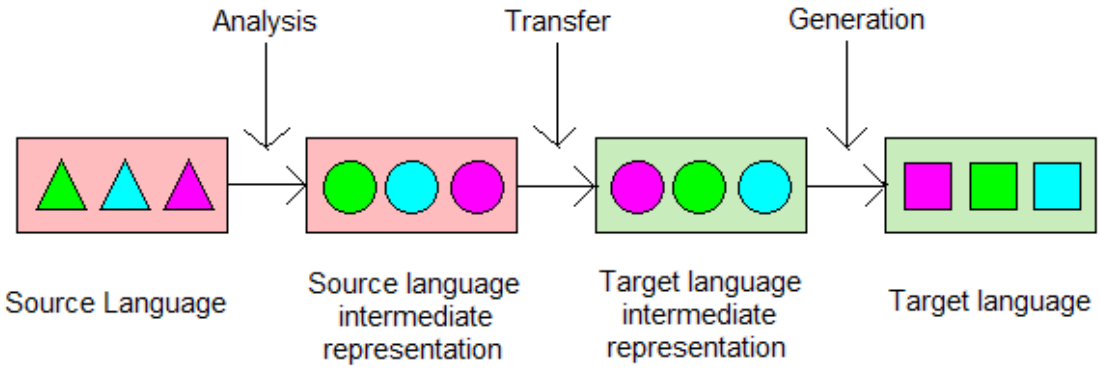

Figure 2.3: Description of Transfer-Based Machine Translation [25]

The major modules in transfer based MT is as follows.

Analysis: Analysis phase is used to produce source language structure [15].

Transfer: Transfer phase is used to transfer source language representation to a target level representation [15].

Generation: Generation phase is used to generate target language text using target level structure [15].

Advantages of Transfer-Based MT

Transfer-based approach has following advantages.

- It has a modular structure.
- The system easily handles ambiguities that carry over from one language to another [15].

Disadvantage of Transfer-Based MT

Transfer-based MT systems have following disadvantages.

- Some of the source text meaning can be lost in the translation [15].

- **Interlingua Machine Translation**

Inter is a sub version of Direct Machine Translation. The Interlingua Machine Translation converts words into a universal language that is created for the MT simply to translate it to more than one language. Figure 2.4 shows how different languages A, B, C, D can be translated through this system.

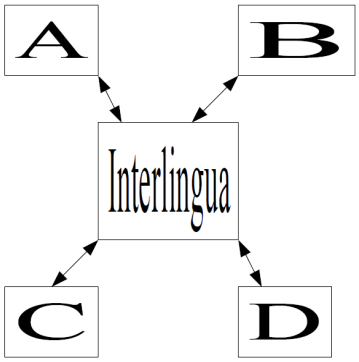

Figure 2.4: Interlingua language system [26]

Advantages of Interlingua Machine Translation

Interlingua MT systems have below mentioned advantages.

- It gives a meaning-based representation and can be used in applications like information retrieval.
- An Interlingua system has to resolve all the ambiguities so that translation to any language can take place from the Interlingua representation.
- The system is more practical when several languages are to be interpreted since it only needs to translate it from the source language. Figure 2.5 shows how language A can be translated into several languages [26].
- For specific domains, Interlingua approach can be used successfully [25].

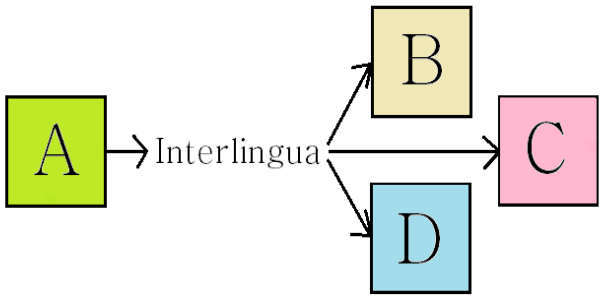

Figure 2.5: Multilingual MT system with Interlingua approach [25]

Disadvantage of Interlingua Machine Translation

Interlingua MT systems have following disadvantages.

- Time efficiency of this system is lower than the Direct Machine Translation system [25].
- Major problem lies in defining a universal abstract (Interlingua) representation which preserves the meaning of a sentence.
- Defining a vocabulary for a universal Interlingua is extremely difficult as different languages conceptualize the world in different ways.
- There may be many concepts in a language or culture which lack representation in another language [15].

2.1.3 Corpus Based Machine Translation

This is considered as a new approach of the era for machine translation. The corpus based systems are classified into statistical and example-based Machine Translation.

- **Statistical Machine Translation (SMT)**

The general idea in SMT system is that the translation will be from the most likely translated word. The system consists of three different models. The Language Model (LM) computes the probability of the target language '*T*' as probability *P(T)*. The Translation Model (TM), helps to compute the conditional probability of target sentences given the source sentence, *P(T|S)*. Decoder maximizes the product of LM and TM probabilities. The basic sketch of SMT system is shown in Figure 2.6.

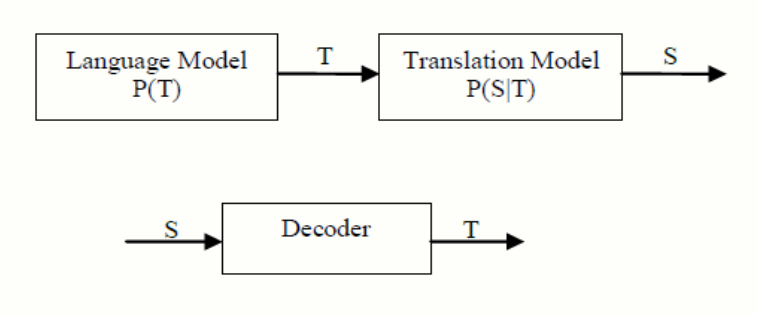

Figure 2.6: Statistical MT [15]

- **Example based Machine Translation**

Example based systems use previous translation examples to generate translations for an input provided. When an input sentence is presented to the system, it retrieves a similar source sentence from the example-base and its translation. The system then adapts the example translation to generate the translation of the input sentence.

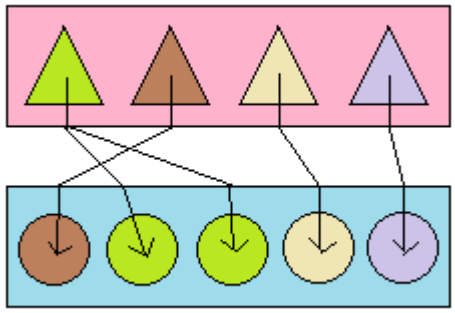

Figure 2.7: Translation Template of a phrase in two different languages [15]

Translation templates are a bilingual pair of sentences or phrases where words are coupled and replaced by variables. The goal is to have large corpus to be able to directly translate word after word in a sentence based on the *translation template* [15]. Figure 2.7 shows the *translation templates* of a phrase in two different languages. Differences between Example-based MT and statistical-based MT systems is given in Table 2.1

Table 2.1: Difference between example and statistical MT

Example-based MT	Statistical-based MT
Example-based MT systems use variety of linguistic resources such as dictionaries and thesauri, *etc.*, to translate text.	Statistical-based MT uses purely statistical based methods in aligning the words and generation of texts.

Number MT techniques are being combined to undertake translation. There is still a lot of research being done to couple a number of MT systems. There are some hybrid systems which are being proposed. One such system is Generation-Heavy MT (GHMT) [14].

2.1.4 Knowledge-based MT

Early MT systems are characterized by the use of syntax. There was little semantic analysis. The synthesis and analysis is restricted in early MT systems to sentence level. Semantic-based approaches to language analysis have been introduced AI researchers. The approaches require a large knowledge –base that includes both ontological and lexical knowledge [15].

2.2 Statistical Machine Translation

The SMT system is based on the view that every sentence in a language has a possible translation in another language. A sentence can be translated from one language to another in many possible ways. Statistical translation approaches take the view that every sentence in the target language is a possible translation of the input sentences [14]. Figure 2.8 gives the outline of Statistical Machine Translation system.

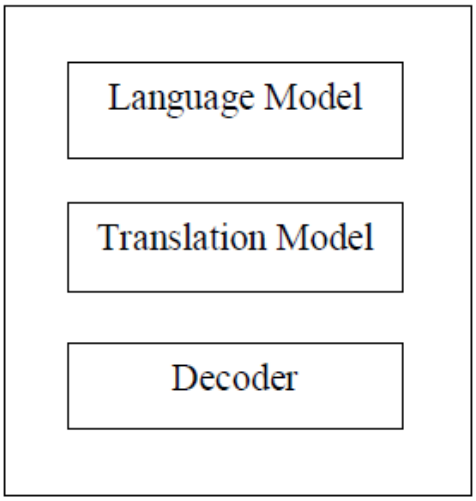

Figure 2.8: Outline Statistical Machine Translation system

2.2.1 Language Model

A language model gives the probability of a sentence. The probability is computed using *n-gram* model. Language Model can be considered as computation of the probability of single word given all of the words that precede it in a sentence [25]. The goal of Statistical Machine Translation is to estimate the probability (likelihood)

of a sentence. A sentence is decomposed into the product of conditional probability. By using chain rule, this is made possible as shown in 2.1. The probability of sentence P (S), is broken down as the probability of individual words P(w).

$P(s) = P(w_1, w_2, w_3, ..., w_n)$

$$=P(w_1)\ P(w_2|w_1)\ P(w_{3,}|w_1w_2)\ P\ (w_4|w_1w_2w_3)\ldots P(w_n|w_1w_{2\ldots}w_{n-1})) \quad \ldots (2.1)$$

In order to calculate sentence probability, it is required to calculate the probability of a word, given the sequence of word preceding it. An *n-gram* model simplifies the task by approximating the probability of a word given all the previous words.

An *n*-gram of size 1 is referred to as a *unigram*; size 2 is a *bigram* (or, less commonly, a *digram*); size 3 is a *trigram*; size 4 is a *four-gram* and size 5 or more is simply called a *n-gram*.

Consider the following training set of data given in Figure 2.9:

There was a King

He was a strong King.

King ruled most parts of the world.

Figure 2.9: Training set of data for LM

Probabilities for bigram model are as shown below:

P(there/<s>) = 0.67 P(was/there) = 0.4 P(king/a) = 1.0 P(a/<s>)=0.30 … (2.2)

P(was/he) = 1.0 P(a/was) = 0.5 P(strong/a) = 0.2 P(king/strong)=0.23 ... (2.3)

P(ruled/he) = 1.0 P(most/rules) = 1.0 P(the/of) = 1.0 ... (2.4)

P(world/the) =0.30 P(ruled|king)=0.30 ... (2.5)

The probability of a sentence: 'A strong king ruled the world', can be computed as follows:

P(a/<s>)* P(strong/a)* P(king|strong)*P(ruled|king)*P(the/ruled)*P(world|the)

=0.30*0.2*0.23*0.30*0.28*.0.30

=0.00071 ... (2.6)

2.2.2 Translation Model

The Translation Model helps to compute the conditional probability *P(T|S)*. It is trained from parallel corpus of target-source pairs. As no corpus is large enough to allow the computation translation model probabilities at sentence level, so the process is broken down into smaller units, *e.g.*, words or phrases and their probabilities learnt [14]. The target translation of source sentence is thought of as being generated from source word by word. For example, using the notation *(T/S)* to represent an input sentence *S* and its translation *T*. Using this notation, sentence is translated as given in 2.7.

(kutta bageche mae sooya | dog slept in the garden)

(कुत्ता बगीचे में सोया | dog slept in the garden) ... (2.7)

One possible alignment for the pair of sentences can be represented as given in 2.8:

(कुत्ता बगीचे में सोया | dog(1) slept(4) in(3) the(3) garden(2)) ... (2.8)

A number of alignments are possible. For simplicity, word by word alignment of translation model is considered. The above set of alignment is denoted as *A(S, T)*. If length of target is *l* and that of source is *m* than there are *lm* different alignments are possible and all connection for each target position are equally likely, therefore order of words in T and S does not affect *P (T|S)* and likelihood of *(T|S)* can be defined in terms of the conditional probability *P (T, a/S)* as shown in 2.9:

$$P(S|T) = \text{sum } P(S, a/T) \quad \text{... (2.9)}$$

The sum is over the elements of alignment set, *A(S, T).* English word has only exactly one connection. For the alignment,

P(कुत्ता बगीचे में सोया | dog slept in the garden), can be computed by multiplying the translation probabilities T(कुत्ता |dog(1)), T(बगीचे | garden(6)), T(में|in(4)), T(null| the(5)), and T(सोया | slept(2)).

To generate target sentence from source sentence, we have to follow the steps as given below:

i). Select the length of *S* with probability *L* where L=P[length (S)=*m*] is a constant *i.e.* All lengths are assumed to be equally likely with probability *L*.

ii). Select an alignment with probability *P(a|S)*. There are $(l+1)^m$ possible alignments [23]. Assuming all possible alignments are equally likely, the probability of alignment a, *P(a|S)*, is as shown in 2.10

$$P(a|S) = L \times 1/(l+1)^m \quad \text{... (2.10)}$$

iii). Select the j_{th} English word with a probability

The joint likelihood of Hindi string and an alignment given an English string isgiven in 2.11

$$P(S, a/T) = P(a/T)*P(S/a, T) \quad \text{... (2.11)}$$

T is the probability of seeing S_j in source sentence, given T_{aj} in target sentence. The alignment is determined by specifying the values of a_j for *j* from 1 to *m*, each of which can take value from 0 to l.

2.2.3 Decoder

This phase of SMT maximizes the probability of translated text. The words are chosen which have maximum like hood of being the translated translation [7].

Search for sentence T is performed that maximizes *P (S|T) i.e.*

$$Pr(S, T) = argmax\ P(T)\ P(S|T) \quad \text{… (2.12)}$$

Here problem is the infinite space to be searched. The use of stacked search is suggested, in which we maintain a list of partial alignment hypothesis [7]. Search starts with null hypothesis, which means that the target sentence is obtained from a sequence of source words that we do not know. We represent this entry sequence as

(कुत्ता बगीचे में सोया |*), where * is a place holder for an unknown sequence of source words. As the search proceeds, it extends entries in the list by adding one or more additional words to its hypothesis. For example, extend initial entry to one or more of the following entries:

(कुत्ता बगीचे में सोया | dog slept (2))

The search terminates when there is a complete alignment in the list that is more promising than any of the incomplete alignments.

2.3 Tools used for implementation of SMT System

Various tools are available for the development of Statistical Machine Translation. A SMT system for a pair of languages can be developed by using the combination of these tools. Figure 2.10, shows some open source tools that are available to use.

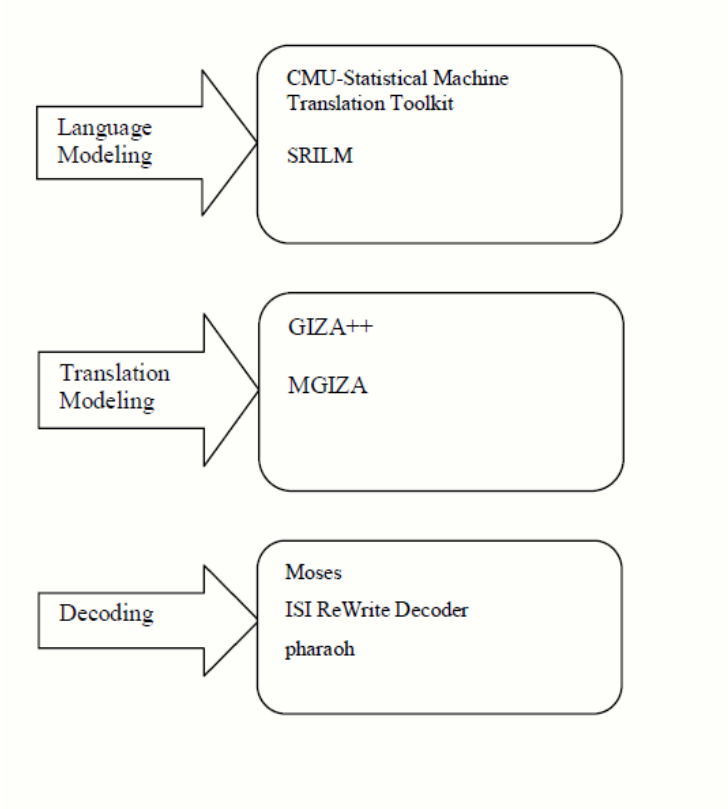

Figure 2.10: Statistical Machine Translation Tools

2.3.1 Language Model (LM) tools

There are many LM tools which are available. They are discussed as follows.

The CMU Statistical Language Modeling (SLM) Toolkit

The Carnegie Mellon University (CMU) Statistical Language Modeling Toolkit is a set of UNIX software tools designed to facilitate Language Modeling work for research purposes. It was written by Roni Rosenfeld, and released in 1994 [26].

SRILM

SRILM is a toolkit for building and applying statistical Language Models (LMs) developed by SRI Speech Technology and Research Laboratory. It has been under development since 1995 [1]. SRILM is freely available for download.

2.3.2 Translation Model Tools

There are many TM tools which are available to be used for SMT systems. They are discussed as follows.

GIZA++

GIZA++ is a tool developed by Franz Josef Och. and is an extension of GIZA developed by the Statistical Machine Translation team during the summer workshop in 1999 at the center for Language and Speech Processing at Johns-Hopkins University. This tool implements different models like HMM and also perform word alignment [5]. GIZA++ is freely available for download.

MGIZA

MGIZA++ is a multi-threaded word alignment tool based on GIZA++. It extends GIZA++ in multiple ways. It provides the concept of multi-threading, and memory optimization. It can resume training from any stage, and continue training from any stage. MGIZA is freely available for download [5].

2.3.3 Decoder Tools

There are many different tools for the decoding stage of SMT system. They are discussed as follows.

Moses

Moses is a Statistical Machine Translation system developed by Hieu Hoang and Philipp Koehn at the University of Edinburgh that allows the automatic training of translation models for any language pair. All that is required is a collection of translated texts (parallel corpus). Moses works with SRILM to develop Language Model, and GIZA++ to develop Translation Model [12]. Moses is freely available for download.

ISI ReWrite Decoder

ISI ReWrite Decoder is software that is used to perform decoding (searching) in development of Statistical Machine Translation systems. It works with CMU-Statistical Language Modeling toolkit and GIZA++ to perform translations from Source Language to Target Language [26]. It freely available for download and use at the link shown in 2.13:

http://www.isi.edu/publications/licensed-sw/rewrite-decoder/ ... (2.13)

Pharaoh

Pharaoh is a Machine Translation decoder developed by Philipp Koehn as part of his PhD thesis at the University of Southern California and the Information Sciences Institute to aid research in Statistical Machine Translation. The decoder works with the SRI Language Modeling Toolkit. It can be obtained from link shown in 2.14

http://www.isi.edu/licensed-sw/pharaoh/ ... (2.14)

2.4 Existing MT Systems

There are following MT systems that have been developed for various natural language pair.

Systran

Systran is a rule based Machine Translation System developed by the company named Systran. It was founded by Dr. Peter Toma in 1968. It offers translation in about 35 languages. It provides technology for Yahoo! Babel Fish and it was used by Google till 2007 [14].

Google Translate

Google Translate is service provided by Google Inc. to translate a section of text, or a webpage, into another language. The service limits the number of paragraphs, or range of technical terms, that will be translated [10]. Google translate is based on Statistical Machine Translation approach. It can translate text, documents, web pages *etc*.

Bing Translator

Bing Translator is a service provided by Microsoft, which was previously known as Live Search Translator and Windows Live Translator. It is based on Statistical Machine Translation approach.

Four bilingual views are available:

- Side by side
- Top and bottom
- Original with hover translation
- Translation with hover original

Hindi to Punjabi Machine Translation System

This is a Machine Translation system which translates sentence in Hindi to Punjabi. This system is based in Direct Translation approach [28]. It is developed by Punjabi University, Patiala. It can translate text file, web-pages from Punjabi to Hindi. It is available to use at link shown in 2.15:

http://h2p.learnpunjabi.org ... (2.15)

METAL

METAL is a translation system initiated in the late seventies by Siemens-Nixdorf, with the University of Texas [15]. It uses the concept of a controlled language to achieve high quality translation in various technical domains. It produces indicative translation for general texts, which needs to be post-edited for style. METAL is now called LANT-MARK, and marketed by LANT, a Belgian company. More information is available at link shown in 2.16:

http://www.lant.be/ ... (2.16)

2.5 Machine Translation Projects in India

India has 18 constitutional languages, which are written in 10 scripts. Hindi is the official language of the country and English is widely used in the media, commerce, science and technology and education. Many of the states have their own regional languages, which can be either Hindi or one of the other constitutional languages. Only about 5% of the population speaks English. There is big market for translation between English and various Indian languages, which task is currently preformed manually. Two specific examples of high volume manual translation are - translation of news from English into local languages, translation of annual reports of government departments and public sector units among, English, Hindi and the local language [14]. Various Machine Translation projects related to Indian languages are shown below.

Anglabharat (and Anubharati)

Anglabharti performs the Machine Translation from English to Indian languages, used mostly for Hindi and uses rule-based transfer approach for translation. The system handles ambiguity/complexity by post-editing—in case of ambiguity, the system retains all possible ambiguous constructs, and the user has to select the correct choices using a post-editing window to get the correct translation. This project is primarily

based at IITKanpur, in collaboration with ER&DCI, Noida, and has been funded by TDIL[15].

Anubharti is a recent project at IIT-Kanpur. It uses Example Based Machine Translation for dealing with translation from Hindi to English [15].

Anusaaraka

Anusaarka uses the principles of Paninian Grammar (PG) and exploits the close similarity of Indian languages. It maps local word groups between source and target languages. To deal with the differences in languages the system introduces extra notation to preserve the information of the source language. The project originated at IIT Kanpur, and later shifted mainly to the Centre for Applied Linguistics and Translation Studies (CALTS), Department of Humanities and Social Sciences, University of Hyderabad. It was funded by TDIL [15].

MaTra

It is a Human-Assisted translation project for English to Indian languages, currently Hindi, essentially based on a transfer approach. The system uses rule-bases and heuristics to resolve ambiguities to the extent possible – for example, a rule-base is used to map English prepositions into Hindi postpositions [12]. This system is meant for translators, editors and content providers. Currently, it works for simple sentences, and work is going on to extend the coverage to complex sentences. This system is mainly used in domain of news, annual reports and technical phrases, and has been funded by TDIL.

Mantra

The Mantra project is based on the TAG formalism from University of Pennsylvania. A sub-language English-Hindi MT system has been developed for the domain of gazette notifications pertaining to government appointments. Apart from translating the text, system is capable of preserving the format of input word documents across the translation. Recently, work has been initiated on other language pairs such as Hindi-English and Hindi-Bengali, as well as on extending to the domain of parliament proceeding summaries [15].

UCSG-based English-Kannada MT

The CS Department at the University of Hyderabad has worked on an English-Kannada MT system, using the Universal Clause Structure Grammar (UCSG) formalism, also invented there. This is essentially a transfer-based approach, and has

been applied to the domain of government circulars, and funded by the Karnataka government [15].

UNL-based MT between English, Hindi and Marathi

The Universal Networking Language (UNL) is an international project of the United `Nations University, with an aim to create an Interlingua for all major human languages. IIT Bombay is the Indian participant in UNL, and is working on MT systems between English, Hindi and Marathi using the UNL formalism. This essentially uses an interlingual approach—the source language is converted into UNL using an enconverter‘, and then converted into the target language using a deconverter‘. Thapar University, Patiala and Punjabi University, Patiala are working on the development of UNL encoverter for Punjabi language [14].

Tamil-Hindi Anusaaraka and English-Tamil MT

The Anna University KB Chandrasekhar Research Centre at Chennai was established recently, and is active in the area of Tamil NLP. A Tamil-Hindi language accessor has been built using the Anusaaraka formalism described above. Recently, the group has begun work on an English-Tamil MT system.

English-Hindi MAT for news sentences

The Jadavpur University at Kolkata has recently worked on a rule-based English-Hindi MAT for news sentences using the transfer approach.

Tamil-Hindi machine aided translation system

The system Tamil-Hindi Machine-Aided Translation system has been developed by Prof. C.N. Krishnan at Anna University at KB Chandrashekhar (AU-KBC) research centre, Chennai. The translation system is based on Anusaaraka Machine Translation System, the input text is in Tamil and the output can be seen in a Hindi text [15].

Anuvadak English-Hindi software

Super Infosoft Pvt Ltd is one of the very few private sector efforts in MT in India. They have been working on the software called Anuvadak, which is a general-purpose English-Hindi translation tool that supports post-editing [15].

Sampark

It is an automated system for translating one Indian language to another. Sampark is a hybrid system consisting of traditional rules-based algorithms and dictionaries and newer statistical machine-learning techniques. It consists of three major parts and 13 modules arranged in a pipeline [16].

Anubaad hybrid machine translation system

Anubaad a hybrid MT system is developed in the year 2004 for translating English news headlines to Bengali, developed by Bandyopadhyay (2000) at Jadavpur University Kolkata. The current version of the system works at the sentence level [14].

Hindi to Punjabi machine translation system

Hindi to Punjabi Machine translation System developed by Goyal [13] at Punjabi University Patiala in the year 2009. This system is based on direct word-to-word translation approach. This system consists of modules like pre-processing, word-to-word translation using Hindi-Punjabi lexicon, morphological analysis, word sense disambiguation, transliteration and post processing. The system has reported 95% accuracy.

Chapter 3

Problem Statement

With each passing day the world is becoming a global village. There are hundred's of languages being spoken across the world. The official languages of different states and nations are also different according to their cultural and geographical differences.

3.1 Gap Analysis

Most of the content available in digital format is in English language. The content shown in English must be presented in a language which can be understood by the intended audience. There is large section of population at both national and state level who cannot comprehend English language. It has brought about language barrier in the side lines of digital age. Machine Translation (MT), can overcome this barrier. In this thesis, a proposed Statistical Based Machine Translation system for translating English text to Hindi language has been proposed. English is the source language and the Hindi is the target language.

3.2 Objectives

The objectives of thesis are as under:

- To understand the Language Model (LM), Translation Model (TM), and Decoding stages of SMT.
- To create a LM for Hindi with use of SRI's LM language model.
- To create a TM model with use of GIZA++ software.
- To generate Hindi sentences with use of Moses software.
- To evaluate and test the system.

3.3 Methodology

English to Hindi language translation is done by making use of Statistical Machine Translation (SMT). Main goal of this system is to undertake translation with minimum human efforts. There are many tools pertaining to LM, TM, decoder for undertaking SMT. SMT has three major parts of the system, Language Model, Translation Model and searching (decoder). The LM computes the probabilities with respect to the target language. The TM computes the probabilities regarding the substitution of target language word with source language word. For development of

LM, SRI international's SRILM Language Model toolkit is used. GIZA++ is used for creation of Translation Model. For decoding stage, Moses software has been used. The system is based upon Linux operating system. It will accept English sentence from the terminal and produce output in Hindi.

Chapter 4

Design and Implementation of Statistical Machine Translation System

In this chapter, the design and implementation of the system has been discussed. This includes development of corpus, data preparation, development of Language Model, Translation Model and training of decoder.

4.1 Development of Corpus

Statistical Machine Translation system makes use of a parallel corpus of source and target language pairs. This parallel corpus is necessary requirement before undertaking training in Statistical Machine Translation. The proposed system has used parallel corpus of English and Hindi sentences. A parallel corpus of more than 5000 sentences has been developed from which consist of small sentences and the life history of freedom fighters with reference to their trail in courts.

4.2 Architecture of English to Hindi Statistical Machine Translation System

The architecture forms the central role in making up SMT system. Language Model (LM), Translation Model (TM), decoder are used in undertaking SMT. Language Model is prepared from the target language. Decoder gives the probability of target sentence given the source sentences. The architecture of the system is shown in Figure 4.1.

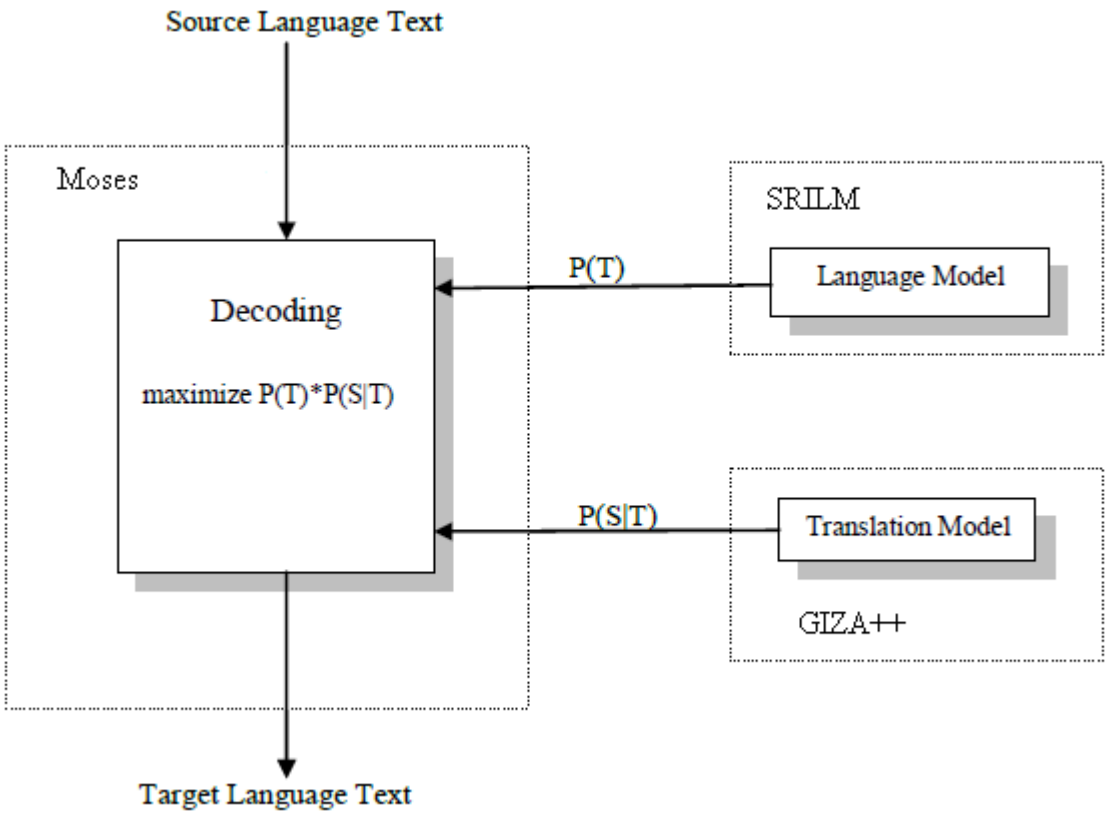

Figure 4.1: Architecture of Statistical Machine Translation system

4.2.1 Language Model

Language Model (LM) gives the probability of a sentence. The probability of a sentence depends upon the probability of individual words. *n-gram* is a sequence of words [17]. LM is developed for the target language. If '*T*' is the target language, LM computes '*P(|T)*' and feed this input to the decoder software.

SR International's Language Model (SRILM) for LM is used. SRILM is available freely for research purposes from their website http://www.speech.sri.com/projects/srilm/download.html.

4.2.2 Translation Model

The Translation Model (TM) computes the probability of source sentence '*S*', for a given target sentence '*T*'. Mathematically, the probability being computed by TM is given as, *P(S|T)*. Translations can be done word based or phrase based [5]. The output of TM is fed into Moses decoder. *GIZA++* along with *mkcls* is used to develop Translation model, which is developed.

4.2.3 Decoder

The decoder maximizes the probability of the generated sentence. It makes use of the a*rgmax ()* function to maximize the probability. *Moses* software which is freely available under open source licenses is used for decoder. *Moses* is compatible with SRILM and *GIZA++*. *Moses* decoder accepts as input the source language text and generates the target language text. The probability files are accepted from TM and LM. The decoder can be set in interactive mode to for doing translation [12].

4.3 Preparation of Data

Preparation of data involves tokenizing, cleaning, lowercasing the corpus. Before undertaking the training of the system the data must be pre-processed. The issues which need to be addressed in parallel corpus are as follows:

- To set the environment variable LC_ALL to C in Linux environment.
- The software needs one sentence per line. So there should be no empty lines in the corpus.
- The sentences having word limit more than 40 words are removed. The sentences having word limit from 1-40 are not removed.
- All sentences of parallel corpus need to be in lowercased. The uppercased sentences need to be changed to lower case [4].

For the preparation of data, used in proposed system, PERL scripts have been used.

4.3.1 Tokenizing the corpus

Tokenizing of corpus makes use of a Perl script. The input to this script is the raw corpus and the output is tokenized corpus [4]. The script executed as given in 4.1.

```
zcat corpus_new4.en.gz |./tokenizer.perl –l en >
corpusforRP/corpus_new4.tok.en                    ... (4.1)
```

The screenshot for execution of script 4.1 is given in Figure 4.2.

Figure 4.2: Tokenizing corpus

As a result of successful script execution, *corpus_new4.tok.en* is created with tokenized content.

4.3.2 Filtering out long sentences

Filtering out long sentences makes use of PERL script, *clean-corpus-n.perl*. The output of *tokenizer.perl* is accepted as input for *clean-corpus-n-perl.* This script removes long sentences from the corpus. It also removes redundant space characters and empty lines. Long sentences, are those which exceed word limit of 40 words [4]. The system does not accept empty lines, hence they are removed. *GIZA++* takes very long time to train on long sentences. *Clean-corpus-n.perl* is used to reduce the length of sentences. The script is executed as given in 4.2.

./clean-corpus-n.perl corpusforRP/corpus_new4.tok en hicorpusforRP/corpus_ new4. clean 1 40 … (4.2)

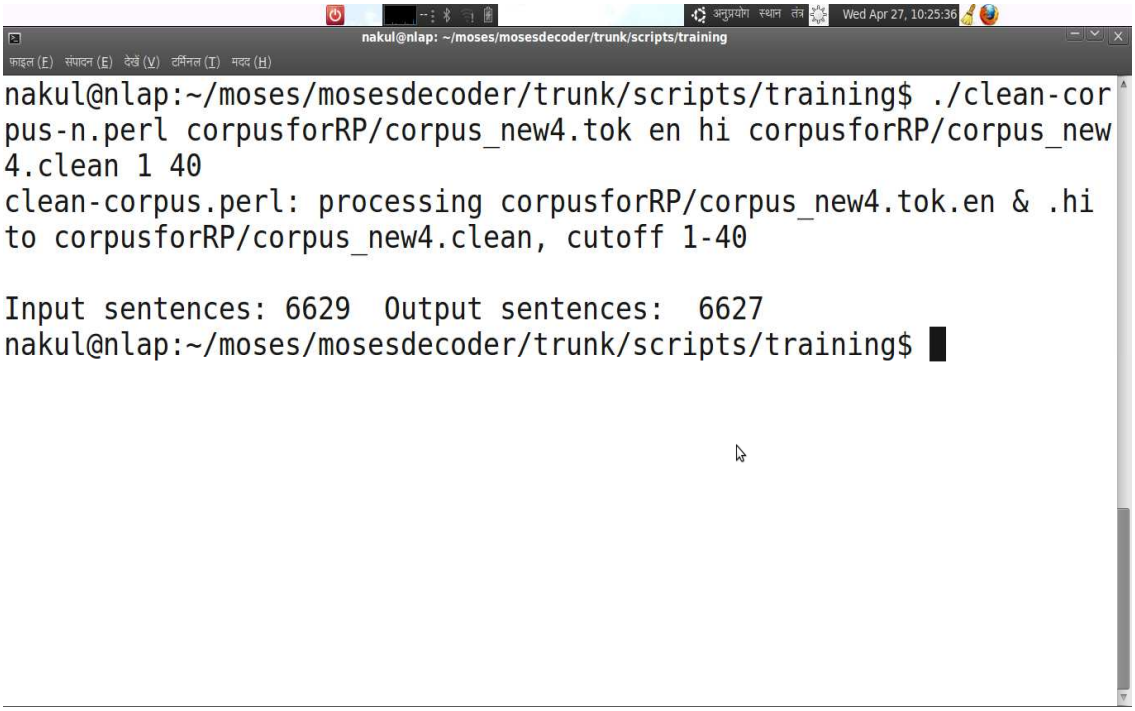

Figure 4.3: Filtering out long sentences

The successful execution of script 4.2, removes sentences having length of words more than 40. Figure 4.3 gives the number of input and output sentences. The output files created are *corpus_new4.clean.en* and *corpus_new4.clean.hi.*

4.3.3 Lowercasing data

The data which is fed in for training the Moses software must be in small case. This is accomplished using *lowercase.perl* [4]. Figure 4.4 shows a lowercased data. The script for lowercasing data is given in 4.4

./lowercase.perl <corpusforRP/corpus_new4.clean.en|more ... (4.4)

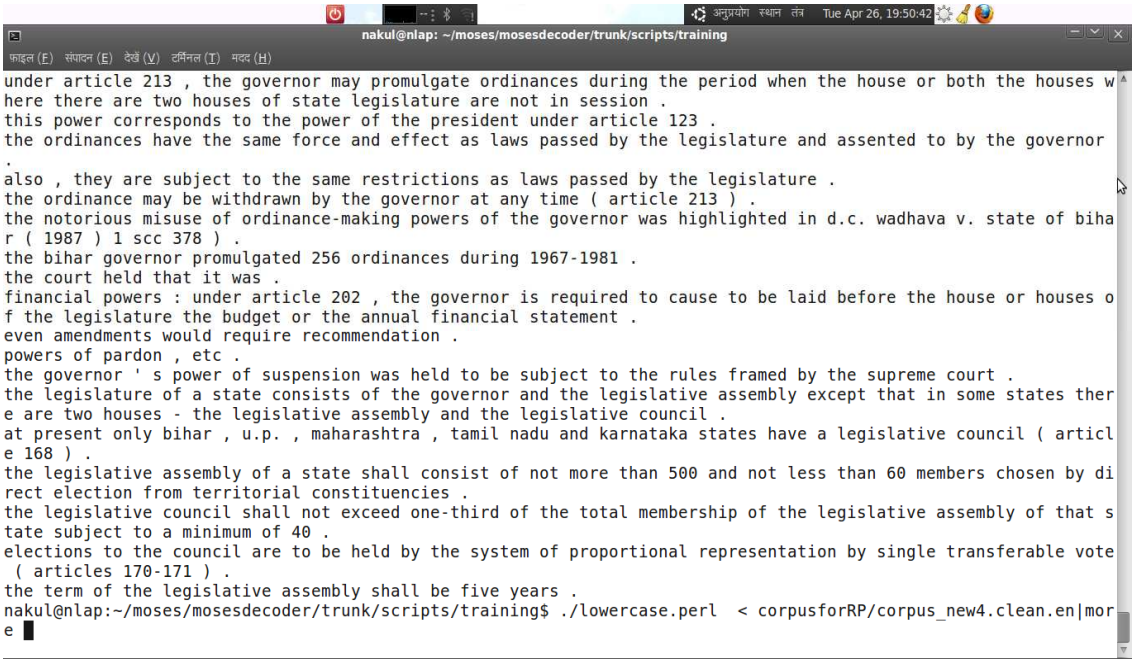

Figure 4.4: Lowercasing output

4.4 Language Model

For Language Model (LM), SR International's LM model (SRILM) is used. SRILM is compatible with Moses decoder and GIZA++ Translation Model (TM). Language Model (LM)'s directory structure along with its description is shown in Table 4.1:

Table 4.1: Directory Structure of LM Model

Directory Structure	Description
bin	Released programs
lib	Released libraries
include	Released header files
misc	Miscellaneous C and C++ convenience lib
destruct	C++ data structures
lm	Language Model libraries and tools

Ngram-count

Ngram-count counts the number of *n-gram* of the corpus. *Ngram-count* also builds the language model from the generated counts [2]. The format of LM is also shown by ngram-format file.

The command for generating language model is given in 4.5.

./ngram-count –order 3 –text corpus_new4.lowercased.hi –lm hindi.lm

–write count.cnt … (4.5)

The description of parameters for PERL script, *ngram-count* is given in Table 4.2.

Table 4.2: Parameters of ngram-count

Parameter	Description
order	This parameter sets the maximal order of N-grams to count and the order of estimated LM. Default value is 3.
text	Generate *n-gram* counts from text file. Text file should contain one sentence unit per line. Begin/end sentence tokens are added if not already present. Empty lines are ignored.
write	Write count into mentioned file

The initial contents of the *hindi.lm* file created by *ngram-count* are shown in Figure 4.5.

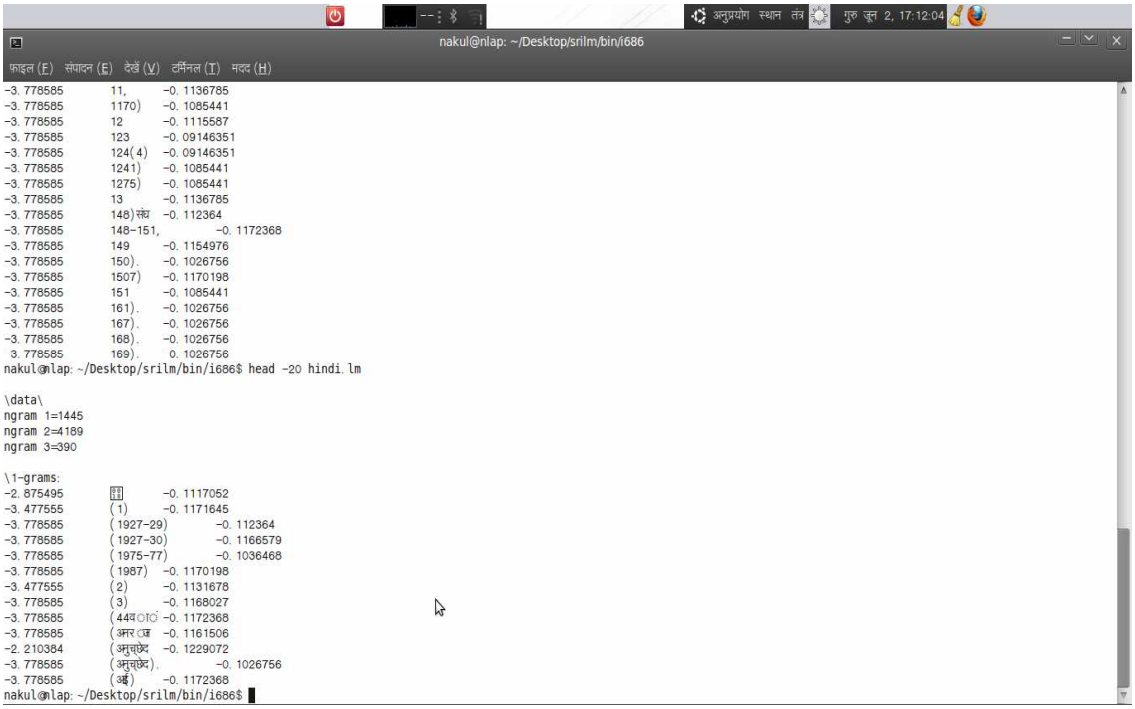

Figure 4.5: Contents of *hindi.lm* (in *ngram file format)*

The keyword *data*\ indicates the beginning of *lm* file. The total count of individual *n-grams*, found in the corpus is then mentioned after *data*\ keyword. For each n-gram (1-gram, 2-gram, *etc.*), there are individual sub-sections. Each sub-section starts with conditional probability of the *n-gram*. This probability is to the base of log 10. This is followed by the word which constitutes *n-gram* [2].

4.4.1 Installation of SRILM

The installation of SRILM involves following steps:

i). Unpack. It should give a top-level directory with the subdirectories listed in README, as well as a few documentation files and a Makefile.

ii). SRILM variable should then be set to the top-level Makefile. This path should be absolute starting from the root directory.

Specific to the architecture, the contents common/Makefile.machine.<platform> define the platform-dependent variables. The '*make*' command uses the dependencies in the Makefile to decide what parts of the program need to be compiled. The parameters are as shown in 4.6.

make MACHINE_TYPE=foo ... (4.6)

The variables in Makefile need to be changed are shown in Table 4.3.

Table 4.3: Variables in Makefile of SRILM to be changed

Variable	Changed value
CC,CXX	This variable should be set to the complier or compiler version.
PIC_FLAG	This variable should be set to indicate the position-independent code.
DEMANGLE_FILTER	If program "c++filt" is not installed, this variable is set to empty.
TCL_INCLUDE, TCL_LIBRARY	These variables point to the location of Tool Command Language's (TCL) header files.

4. Following free third-party software's are also required to build SRILM:
    - gcc version 3.4.3 or higher
    - GNU make
    - C shell (installed in /bin/csh)
    - John Ousterhout's Tcl toolkit.

In the top-level directory, command 4.7, 4.8 are run to build SRILM.

gnumake World …. (4.7)

make World …. (4.8)

This will create the directories:

bin/
lib/
include/

bin directory stores the executable files of SRILM software. The released library files are stored in lib directory. The released header files are present in include directory.

4.5 Translation Model

The software that aids in developing Translation Model is *GIZA++*. *GIZA++* is extension of *GIZA* software (http://www.clsp.jhu.edu/ws99/projects/mt/toolkit) which was developed at the Center for Language and Speech Processing at Johns-Hopkins University (CLSP/JHU). *GIZA++* includes a lot of additional features [5]. The extensions of *GIZA++* were designed and written by Franz Josef Och.

GIZA++ has following features:

- Implements full IBM-4 alignment model
- Implements IBM-5: dependency on word classes, smoothing,
- Implements HMM alignment model
- Smoothing for fertility, distortion/alignment parameters
- Improved perplexity calculation for models IBM-1, IBM-2 and HMM [5].

The latest version of *Moses* software embeds calls to *GIZA++* and *mkcls* software's, hence there no need to call them separately.

4.5.1 Installation of *GIZA++*

In order to compile *GIZA++*, g++ compiler version 3.3 or higher is needed. Some changes are required to be made in the Makefile of GIZA directory as follows:

The variables in Makefile of *GIZA++* directory need to changed, shown in Table 4.4.

Table 4.4: Variables in Makefile of *GIZA++* to be changed

Variable	Changed Value
CXX	This variable should indicate to version of g++ complier.
Opt	GIZA++ snt2plain.out plain2snt.out snt2cooc.out

GIZA++ is installed by issuing command given in 4.9.

$ make GIZA++ … (4.9)

4.6 Decoder

Moses software helps in decoding stage of SMT. It allows us to train translation models for any language pair. The pre-requisite for the translation is already translated, parallel corpus.

4.6.1 Installation of Moses

Moses can be got from any svn repository. Before installing Moses, which is the statistical decoder for SMT, corresponding LM and TM tools must be installed [4]. For LM, installation and compilation of SRILM must be done and for TM installation and compilation of GIZA++ must be done. Following compatible libraries are needed on UNIX system for running the SRILM software.

- A template-capable ANSI-C/C++ compiler, gcc version 3.4.3 or higher
- GNU make, to control compilation and installation.

- GNU gawk, required for many of the utility scripts.
- GNU gzip to unpack the distribution and to allow SRILM programs to handle compressed data files.
- The Tcl embeddable scripting language library [4].

These are installed by issuing the command as given in 4.10.

$> sudo apt-get install g++ make gawk gzip tcl8.4 tcl8.4-dev … (4.10)

The Makefile in the SRILM is changed as shown in Table 4.5.

Table 4.5: Variables to be changed in Makefile

Variable	Changed value
SRILM	This variable must point to the SRILM's home directory.
MACHINE_TYPE	This variable points to the architecture of the system (i686, i386).
CC	/usr/bin/gcc$(GCC_FLAGS)
CXX	/usr/bin/g++$(GCC_FLAGS)-DINSTANTIATE_TEMPLATES
TCL_LIBRARY	/usr/lib/libtcl8.4.so
TCL_INCLUDE	/usr/include/tcl8.4/

After changing the Makefile, compilation of *Moses* is done command given in 4.11:

$ sudo make …(4.11)

If no error come, then the command in 4.12 is run.

$sudo make World … (4.12)

Some of the extra packages which need to be installed are done by issuing command mentioned in 4.13.

$ sudo apt-get install autoconf automake texinfo zlib1g zlib1g-dev
zlib-bin zlibc ... (4.13)

The makefiles are regenerate as given in 4.12 to 4.16.

$ cd ~/mosesdecoder… … (4.12)

$ ./regenerate-makefiles… … (4.13)

Configuration for compilation is done as:

$ ln -s $SRILM …. … (4.14)

$ env LDFLAGS=-static && ./configure --with-srilm=$SRILM… … (4.16)

and compile:

$ make –j 4 ... (4.17)

4.6.2 Training Moses decoder

Moses toolkit embeds calls to Translation Model (GIZA++) software inside its training script. As a result, the phrase and reordering table get created. The script that does this is called train-factored-model.perl. Training of Moses decoder is done in nine steps. These are as follows.

- Prepare data
- Run GIZA++
- Align words
- Get lexical translation table
- Extract phrases
- Score phrases
- Build lexicalized reordering model
- Build generation models.
- Create configuration file

The preparation of data (corpus) for this is already discussed in the earlier sections. The executable of train-factored-model is called as given in 4.18. Table 4.6 gives explanation of the parameters of training Moses.

./train-factored-phrase-model.perl -scripts-root-dir /home/nakul/moses/mosesdecoder/trunk/scripts/training/moses-scripts/scripts-20110405-1055/ -root-dir . --corpus corpus_new5.loweredcased -f en -e hi -lm 0:3:/home/nakul/moses/mosesdecoder/trunk/scripts/training/moses-scripts/scripts-20110405-1055/training/hindi_lm5.lm>& training_new5.out & …(4.18)

Table 4.6: Parameters for training Moses

Arguments	Description
scripts-root-dir	The directory of Moses scripts which was created by doing make release.
corpus	Specifies the corpus files which are fed as input for undertaking training.
f	Source language corpus, from which translation will be done.
e	Target language corpus, into which

	translation will be done.
lm	Path to the Language Model file.

4.6.3 Tuning Moses decoder

The Moses software makes use of weights given in moses.ini to translate text. The default weights are generated by the system during its training. These weights are present in moses.ini, which is the configuration file of Moses. The most important part is tuning of model parameters set in Moses.ini file [3]. The quality of translation is improved, which is done by using PERL script (mert-moses.perl). The syntax of this command is given in 4.19.

./mert-moses.pl corpus_new5.loweredcased.en corpus_new5.loweredcased.hi model/moses.ini --working-dir /home/nakul/moses/mosesdecoder/trunk/mert/ --rootdir /home/nakul/moses/mosesdecoder/trunk/scripts/training/moses-scripts/scripts-20110405-1055/ --decoder-flags "-v 0" >& mert2.out& …(4.19)

Table 4.7: Parameters of mert-moses.pl

Arguments	Description
working-dir	The directory where all files will be created. This is the path to mert's directory
root-dir	This switch refers to the main directory in which system is working.
decoder-flags	This is a extra parameters for the decoder

The contents of mert2.out get updated as the script gets executed. Table 4.7 gives the explanation of parameters in tuning Moses.

Running Moses decoder

The Moses decoder's executable file is present in directory '*/home/nakul/mosesdecoder/trunk/moses-cmd/src/moses*'. The essential parameter required to run Moses, is the path to configuration file of Moses (Moses.ini).

The script 4.20 allows Moses decoder to run in interactive mode. The English language sentence is given as input and corresponding result in Hindi is produced.

./moses -f ~/moses/mosesdecoder/trunk/scripts/training/moses-scripts/scripts-20110405-1055/training/model/moses.ini ... (4.20)

Figure 4.6 shows Moses decoder running in an interactive mode.

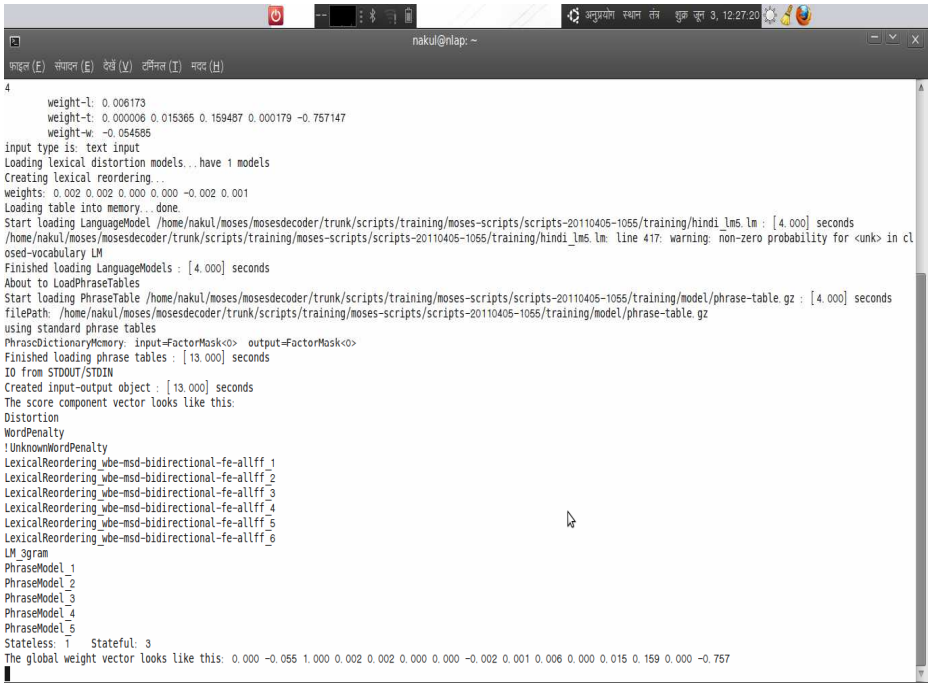

Figure 4.6: Interactive mode of Moses

Consider an English sentence ‘how are you?’ Moses decoder accepted this input in the interactive mode. The result of this translation is shown in Figure 4.7

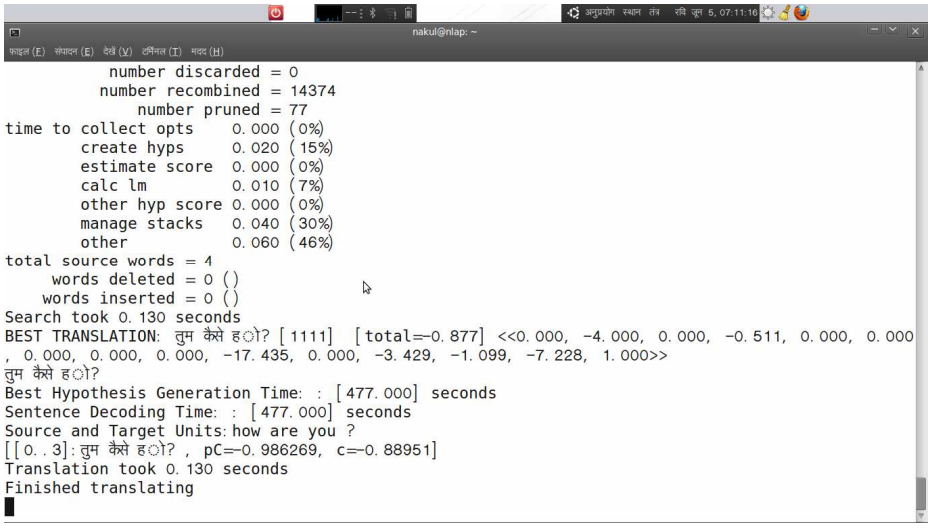

Figure 4.7 Result of English sentence ‘how are you?’

By executing Moses in interactive mode, 90 sentences were translated to Hindi language. Table 4.8 gives the English sentences along with the corresponding translation done by Moses into Hindi language.

Table 4.8: English to Hindi sentences generated by Moses

Sr. No.	Input English Sentence	Output Hindi Sentence generated by the system
1	how is your health now?	कैसे क्या तुम्हारा स्वास्थ अब ?
2	how is your business going on?	कैसे क्या तुम्हारा व्यापार जा रहा है ?
3	I will remember your name	मैन क्या याद रखोगे आपका नाम
4	how is this man's nature?	यह कैसा आदमी प्रकृति ?
5	he is a thief	वह तो वह चोर
6	water is coming from river	पानी से नदी है
7	sea water is coming from rivers.	सागर जल से नदियों है
8	a change is possible.	एक परिवर्तन संभव
9	you should go to school.	तुम्हेन ' जाते विद्यालय
10	goal should be one.	लक्ष्य चाहिए एक
11	This is part of life.	यह भाग का जीवन
12	you should not accept this.	तुम्हेन ' अस्वीकार कर यह
13	what do you want to say?	तुम क्या कहना चाहता हून ' ?
14	what was the aim of discussion?	क्या को मुकदमे के बातचित ?
15	what is your state of mind?	क्या तुम्हारा के चित्त ?
16	time is most important.	समय सबसे महत्वपूर्ण है
17	what is sacred in your life?	तुम्हारे जीवन पवित्र क्या है

		?
18	your future is not good.	आपका भविष्य नहीं अच्छा
19	future is not in our hands.	भविष्य में नहीं है हमारे हाथ
20	what are rights of the citizens?	क्या अधिकारों के नागरिक ?
21	he wants to eat apple.	वह इसके सेब को ' खाएन ' गे
22	this is city's fort.	यह शहर के किल्ला
23	there are many problems in this route	अनेक हैं रास्ता इस समस्यायें
24	the time has come to take control.	आने के समय के नियंत्रण ले रहा हून '
25	we are living a life of standards.	हम एक स्तर के जीवित रही है
26	women and children need protection	निजों की आवश्यकता तो महिलाओं और बच्चों की रक्षा
27	did you find any solution?	तुमने कोई नतीजा प्रमाणित किया गया ?
28	this journey is very difficult	यह यात्रा कत ^ हिन लगा है
29	world is like a global village	एक ही के अति महत्वपूर्ण विश्वव्यापी विश्व गांव समान है
30	the apple is sweet to eat	को मीत ^ हा है है सेब के ' खाएन ' गे
31	change must be brought	परिवर्तन लाया गया बताना जरूरी है

32	you do not have the right to go inside.	तुमने दोबारा अधिकार नहीं , बल्कि उस के भीतर जाना
33	the officer has no values	अधिकारी के में कोई मूल्यों में
34	the officer is not duty	अधिकारी कर्तव्य के पर नहीं है
35	everyone must follow the law of state	उसके प्रत्येक व्यक्ति ' कि मैन कानून का राज्य
36	children are very simple	साधारण बहुत बच्चे
37	we will wait for you	तुम के लिए हम इन ' तज़ार करो
38	this child like to go to zoo	यह बच्चे की चाहता हूं जाते चिड़ियाघर में
39	i am a boy	मैं एक लडका
40	i am a girl	मैं एक लडकी
41	children are innocent	बच्चे हैं निर्दोश
42	children go to school	बच्चे जाते विद्यालय
43	he is going to school	वह करने जा रहे विद्यालय
44	this is a temple	यह एक मंदिर
45	my name is john	मेरा नाम है जोन
46	he is not well	वह नहीं अच्छा
47	He reached on time.	वह समय पर पहुंचा
48	this is not the solution	यह नहीं है समाधान
49	he is safe.	वह सुरक्षित
50	he is the shelter	वह शरण
51	he is a good speaker	वह एक अच्छे वक्ता

52	he was a good writer	वह एक अच्छी लेखक थे
53	building is tall	भबन ऊंचा है
54	there is a hole	एक छेद देखा गया है
55	cow eat grass	गाय घांस ' खाएन ' गे
56	men is a social animal	सामाजिक है पशु आदमी
57	he is not in india	वह भारत में नहीं है
58	he is on official duty	वह कर्तव्य औपचारिक पर है
59	Please help him	क्पया उन्हें मदद
60	those good days	अच्छा दिन वे
61	the purpose is important	इस प्रयोजन महत्वपूर्ण है
62	how much money?	कितना धन ?
63	source is fast	तेजी से स्रोत है
64	this is not required	यह ज़रूरत पदे नहीं है
65	the place is clean	स्वच्छ जगह है
66	sun is bright	सूर्य चमिकिलॅ है
67	you had arrested my attention	मेरा क्या ध्यान गिरफ्तार कर लिया गया
68	day show the morning	दिन सुबह दिखाते थे
69	can we go to home now?	अब हम घर जा सकता जाते ?
70	what is the new name?	नये नाम क्या है ?
71	what is the new method?	नये क्या है तरीके ?
72	what is your intention?	क्या तुम्हारा इरादे ?
73	shop is closed	दुकान बंद है
74	this is a house	यह एक छोटा सदन

75	friend in need	जैसा आवश्यक मित्र हून ' जांच पर जा रहा है
76	investigation is going on	जांच पर जा रहा है
77	they are not going	नहीं है वह ' जा रहे हैन '
78	children are sweet natured	बच्चे हैं प्रकृति मीत ^ हा
79	national song is nice	अच्छा है राष्ट्रीय गाना
80	can you guess?	क्या तुम अनुमान सकते ?
81	your watch is nice	आपका अच्छा घ . द ^ ई है
82	where is your shirt?	आपका कमीज़ कहान ' है ?
83	what is the standard?	मानद्ण्ड क्या ?
84	he is of humble nature	वह विनम्र प्रकृति का
85	the average height is going down	नीची औसत है ऊँचाई के ' जा रहे हैन '
86	words are more important than actions	शब्दों की कार्यों से अधिक महत्वपूर्ण
87	this property is lost	यह संपत्ति नष्ट हो गई
88	citizens have duties	हैं नागरिक तो संग्रह
89	we are living	हम जीवित हैं
90	this is a small house	यह एक छोटा सदन

Chapter 5

Evaluation

5.1 Evaluation of the System

The proposed English to Hindi SMT system, accepted English language sentences as input and gave Hindi sentences as output. The translation of 90 English sentences was done into Hindi language. There are two ways of evaluating any MT system. Evaluation can be done automatically or manually.

In this thesis, manual evaluation method has been used. The translation was evaluated on the parameters of fluency and adequacy. Adequacy is defined as the degree to which the reference sentence is conveyed in the translation. Fluency refers to the grammatical accuracy of the translated text. Adequacy and Fluency for a given translation has different levels on which it can be evaluated [8]. The levels on which fluency and adequacy were evaluated are given in Table 5.1 and 5.2.

Table 5.1: Levels of fluency

Parameter	Definition	Ranking
Perfect	This indicates good grammar of sentence that is translated	4
Fair	The translated sentence is easy to understand but has lack correct grammar	3
Acceptable	The translated sentence is broken, but is understandable with efforts	2
Nonsense	The translated sentence is not clear	1

Table 5.2: Levels of adequacy

Parameter	Definition	Ranking
All	The translated sentence express all the meaning of the source sentence	4
Most	The translated sentence conveys all the meaning of source sentence	3
Some	The translated sentence conveys some meaning	2
None	The translated sentence does has no meaning conveyed	1

The above mentioned parameters (fluency and adequacy) were evaluated by three persons. The geometric average of the parameters was taken and the results are shown in Table 5.3.

The geometric average of the individual parameters, (fluency and adequacy) was taken and the scores were as shown in Table 5.3.

Table 5.3: Result of SMT evaluation

	Fluency	Adequacy
Geometric average	2.693	2.93

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

In this thesis, English to Hindi SMT system has been developed. The SMT is a part of corpus based MT system which requires parallel corpus before undertaking translation. A parallel corpus of 5000 English and Hindi sentences was used to train the system. The SMT system developed accepts English sentences as input and generates corresponding translation in Hindi. The translation of 90 sentences was evaluated using human evaluation method. On the parameters of fluency and adequacy a geometric average of 2.693 and 2.93 was calculated respectively.

The quality of the translated text can be depends upon the size of the corpus and the quality of the corpus.

6.2 Future Scope

There can be following future directions for English to Hindi SMT system.

- The work can be extended to include multilingual corpus of different languages in the source-target pair. The target and source languages can be increased from present one language.

- The system can also be put in the web-based portal to translate content of one web page in English to Hindi.

- A mobile application can also be developed in which message containing English text is sent to the client in Hindi language.

- The corpus can be preprocessed to change its clause structure for improving the quality of translation.

- The translated text can be reordered and processed to overcome grammatical mistakes which will be part of post-processing. This will improve score of human evaluation.

Research Publication

Paper Published

- Nakul Sharma, Parteek Bhatia, "Statistical Machine Translation for Indian Languages", Published at International Conference in Computer Engineering and Technology, (ICCET) 2010 organized by IEEE and JIET.

Paper Accepted

- Nakul Sharma, Parteek Bhatia, Varinderpal Singh, "English to Hindi Statistical Machine Translation", accepted at International Journal in Computer Networks and Security (IJCNS).

References

[1] A. Stolcke, "SRILM-An Extensible Language Modeling Toolkit". *Proc. Intl. Conf. on Spoken Language Processing*, vol. 2, pp.901–904, Denver.

[2] "ngram-count", [Online]. Available: http://www.speech.sri.com/projects/srilm/manpages/ngram-count.1. html/

[3] "Statistical Machine Translation System User Manual and Code Guide", [Online]. Available: http://www.statmt.org/moses/manual/manual.pdf/

[4] S. Charles and S. David, "Overview of Statistical Machine Translation", John Hopkins University, AMTA2006, [Online]. Available: http://www.cs.umass.edu/~dasmith/smt-tutorial.ppt

[5] F.J. Och., "GIZA++: Training of statistical translation models", [Online]. Available at: http://fjoch.com/GIZA++.html.

[6] S. Singh, M. Dalal, V. Vachhani, P. Bhattacharyya, and O. P. Damani, "Hindi generation from Interlingua (UNL)",[Online] Available: http://www.cse.iitb.ac.in/~damani/papers/MTSummit0.pdf

[7] P. F. Brown, S. De. Pietra, V. D. Pietra and R. Mercer, "The mathematics of statistical machine translation: parameter estimation". "*Journal Computational Linguistics"*, vol. 19, no.3, June 1993.

[8] A. Stolcke, "Guide on how-to install and build SRI LM", [Online] Available: http://www.speech.sri.com/projects/srilm/docs/INSTALL.

[9] Charniak and Eugene, "Introduction to artificial intelligence", Boston: Addison-Wesley, 1984.

[10] "Natural language processing",[Online].Available: http://en.wikipedia.org/wiki/Natural_language_processing

[11] "Statistical Machine Translation", [Online]. Available: http://www.comp.nus.edu.sg/~huangyun/ebook/2008_Statistical_Machine_ Translation.pdf

[12] P. Kohen, “Moses: Open Source Toolkit for Statistical Machine Translation.” Proceedings of the *ACL 2007 Demo and Poster Sessions*, pp. 177–180, Prague, June 2007.

[13] V. Goyal and GS Lehal, “Web based hindi to punjabi machine translation system”, Proceeding of *Journal of Emerging Technologies in Web Intelligence*, Vol 2, No 2, May 2010.

[14] D. D. Rao, “Machine Translation A Gentle Introduction”, RESONANCE, July 1998.

[15] S.K. Dwivedi and P. P. Sukadeve, “Machine Translation System Indian Perspectives”, Proceeding of *Journal of Computer Science* Vol. 6 No. 10. pp 1082-1087, May 2010.

[16] G. Athens, “Automated Translation of Indian languages”, ACM News, *Magazine Communications of the ACM*, DOI: 10.1145.

[17] “Machine Translation”, [Online]. Available: http://en.wikipedia.org/wiki/Machine_translation

[18] “Rule-based machine translation”, [Online]. Available: http://en.wikipedia.org/wiki/Rule-based_machine_translation

[19] “Transfer-based machine translation”, [Online]. Available: http://en.wikipedia.org/wiki/Transfer-based_machine_translation

[20] “Interlingual machine translation”,[Online]. Available: http://en.wikipedia.org/wiki/Interlingual_machine_translation

[21] “Dictionary-based machine translation”, [Online].Available: http://en.wikipedia.org/wiki/Dictionary-based_machine_translation

[22] “Example-based machine translation”, [Online].Available: http://en.wikipedia.org/wiki/Example-based_machine_translation

[23] “Statistical machine translation”, [Online]. Available, http://en.wikipedia.org/wiki/Statistical_machine_translation

[24] Tanveer Siddiqui and U.S. Tiwary, “*Natural language Processing and Information Retrieval”*, New Delhi, Oxford Press, 2008.

[25] “Machine Translation”, [Online]. Available, http://faculty.ksu.edu.sa/homiedan/Publications/ Machine%20Translation.pdf

[26] “Machine Translation ”, [Online],Available : http://www.ida.liu.se/~729G11/projekt/studentpapper-10/maria-hedblom.pdf

[27] “CMU toolkit manual”, [Online], Available: http://mi.eng.cam.ac.uk/~prc14/toolkit_documentation.html

[28] G. Singh and G. Singh Lehal,” A Punjabi to Hindi Machine Translation System”, *Coling 2008: Companion volume- Posters and demonstrations*, Manchester, August 2008.