English to Punjabi Statistical Based Machine Translation System

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

Master of Engineering

in

Software Engineering

By:

Gagandeep Singh

(800831029)

Under the supervision of:

Parteek Bhatia

Assistant Professor

Varinderpal Singh

System Analyst

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2010

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"English to Punjabi Statistical Based 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 this thesis has not been submitted for the award of any other degree of this or any other university.

Gagandeep Singh

(Gagandeep Singh)

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

(RAJESH BHATIA) 23/06/10
Head
Computer Science and Engineering Department,
Thapar University,
Patiala.

(R.K.SHARMA) 5/7/10
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgement

I am extremely thankful to my Guide Mr. Parteek Bhatia, Assistant Professor, and Mr. Varinderpal Singh, System Analyst, Computer Science and Engineering Department of Thapar University, for their valuable advice, motivation, guidance, encouragement, moral support, sincere efforts and attitude with which they solved my queries and make this thesis possible. It has being a great pleasure and experience to work under them. I am honored to work with them.

The support from Dr. Rajesh Bhatia, Head, Computer Science and Engineering Department, Thapar University, Patiala, is immense and invaluable. I am highly thankful for his support, guidance and encouragement. I am indebted to his suggestions and help during my work.

I would like to thank Mrs. Inderveer Kaur (PG Coordinator), Assistant Professor, Computer Science and Engineering Department, Thapar University, Patiala, and other faculty members for their assistance, suggestions and stimulating decisions during the period of this research.

Finally, I would like to thank my dearest family and friends especially Amandeep Kaur, Rupinderdeep Kaur, Divya Pandove, and Jarrar A. Rana for all their love, encouragement and constant support without which I could not complete this work. Last but not least, I would like to thank God for not letting me down in time of crisis and always being with me in my good and bad times.

Gagandeep Singh

Gagandeep Singh

(800831029)

Abstract

Machine Translation (MT) refers to the use of computers for the task of translating automatically from one language to another. The differences between languages and especially the inherent ambiguity of language make MT a very difficult problem. Traditional approaches to MT have relied on humans supplying linguistic knowledge in the form of rules to transform text in one language to another. Given the vastness of language, this is a highly knowledge intensive task. Statistical MT is a different approach that automatically acquires knowledge from large amounts of training data. This knowledge, which is typically in the form of probabilities of various language features, is used to guide the translation process.

This thesis provides an overview of use of Statistical Machine Translation to translate text from English language to Punjabi language. To develop the translation system CMU-Statistical Language Modeling Toolkit, GIZA++, and ISI ReWrite Decoder were used.

CMU-Statistical Language Modeling Toolkit is a set of Unix software tools used to facilitate work related to language modeling in the field of Statistical Machine Translation.

To develop Translation Model, GIZA++ is used. GIZA++ is an open source tool used to develop Translation Models for Statistical Machine Translation systems. GIZA++ works with mkcls that is a tool to generate classes. So GIZA++ and mkcls, both tools were used to develop Translation Model.

Along with these, ISI ReWrite Decoder is used for decoding.

Developed system was tested with the corpus of around 6000 parallel sentences of English and Punjabi language. System worked for simple sentences and can be enhanced in future.

Table of contents

List of Figures

List of Tables

Chapter 1

Introduction

Since, the introduction of computers, they are increasing the impact on humans day by day. As in earlier time, when computers were introduced they were used for some special purposes *i.e.* for some complex calculations, scientific purposes *etc*. But with the passage of time computers entered in the homes and slowly made their place in everyone's life whoever is interacting with them in one or the other way. With the introduction of Internet, people became more dependent on computers for the purpose of gathering information, entertainment *etc*. But Internet is surfed mostly in English and most of the information accessed online is available in English. From science, technology, education to manual of gadgets, advertisements, predominant medium of communication is English. But this world is multilingual, where different languages are spoken in different regions, for example, in a country like India, which is a multilingual country with eighteen constitutional languages recognized, English is understood by less than 5% of the population and English acts as a standard of communication for administration, education and business, and the official language of the country, Hindi is used by more than 400 million people[10, 17], similarly at state level most of the work of local administration is in the native languages of corresponding states *e.g.* in Punjab, where Punjabi is native language of the state, most of the work is documented in Punjabi apart from English. So in this scenario, need of translation of text is felt.

To enable the people at regional levels to use computers and Internet effectively, these language barriers have to fall. Solution to the problem is given by Natural Language Processing. Natural Language Processing is an approach to analyze text, based on set of theories and a set of technologies with the help of computers.

Natural Language Processing is a theoretically motivated range of computational techniques, for analyzing and representing naturally occurring texts at one or more levels of linguistic analysis, for the purpose of achieving human-like language processing for a range of tasks or applications[9, 14].

The goal of Natural Language Processing is to accomplish human like language processing. NLP focuses on following two domains:

- Language Processing
- Language Generation

Language Processing refers to the analysis of language for the purpose of producing a meaningful representation, while the Language Generation refers to the production of language from a representation.

1.1 Applications of Natural Language Processing

Machine Translation is the earliest application of Natural Language Processing (NLP). But recent progress in the field of NLP has found applications in Information Retrieval, Information Extraction, Text Summarization, *etc.*

Various applications of NLP are as follows[10]:

- **Machine Translation**

  Machine Translation refers to translation of text from one language to another with the help of computers.
- **Speech Recognition**

  This refers to mapping of acoustic speech signals to a set of words.
- **Speech Synthesis**

  This refers to automatic generation of speech (utterance of natural language sentences) from text.
- **Natural Language Interfaces to Databases**

  Natural Language Interfaces to Database allow querying a Database in natural language using natural language sentences.
- **Information Retrieval**

  This is concerned to the identification of the documents relevant to a user's query.

- **Information Extraction**

  Similar to the Information Retrieval system, Information Extraction responds to user's need.

- **Question Answering**

  For a given question and set of documents, a question answering system attempts to find precise answer, or at least the precise portion of text in which answer appears.

- **Text Summarization**

  This deals with the creation of summaries of documents. It involves syntactic, semantic, and discourse level of processing of text.

Machine Translation is one of the applications of NLP, which deals with the use of computers to translate text from one language to another. In Machine Translation, we have pair of languages Source Language (SL), Language from which translation is to be done and Target Language (TL), language to which translation is to be done. There are different approaches to translate text using Machine Translation. Various approaches of Machine Translation are as follows:

- Direct Translation
- Rule Based Machine Translation
- Corpus Based Machine Translation
- Knowledge Based Machine Translation

At present, main focus is on the Statistical Machine Translation approach, which is a type of Corpus Based Machine Translation, and makes use of statistical methods to translate the text from one language to another. Statistical methods were proposed by Warren Weaver in 1949, but at that time these methods were abandoned due to technological constraints at that time[17]. But in 1989, this approach was practically implemented and result obtained encouraged the use of statistical methods for translation and hence, Statistical Machine Translation (SMT) became active area of machine translation

research. As compared to other approaches, SMT is little bit complex, but system can be built in less time as compared to other approaches.

We have implemented the translation system for the translation of text from English to Punjabi using Statistical methods. Translation system has three modules; Language Model, Translation Model and Decoder. We have used Carnegie Mellon University (CMU)-Statistical Language Modeling Toolkit to develop Language Model for Punjabi, which is target language for our system, GIZA++ to develop Translation Model for English-Punjabi pair of languages, where English is a Source Language and Punjabi is a Target Language. To perform decoding, ISI ReWrite Decoder is used, which is compatible with CMU-Statistical Language Toolkit and GIZA++.

Chapter 2

Review of Literature

2.1 Machine Translation

Machine Translation is the earliest application of Natural Language Processing, which employs computers to translate text from one language to another. Machine Translation can help to overcome the technological barriers. As Information Technology revolution has led to the plethora of information, which is available in a small subset of languages but not reachable by large portion of society. This has led to another division in the society known as digital divide [23]. For example, in the multilingual country like India where only 5% of the population can understand English, requires Machine Translation System to translate information from English to local language like Hindi, Punjabi, Tamil *etc*.

Machine Translation not only unites the world intellectually and culturally but also unites the world through technology, as justified by Jurafsky and Martin [4].

Translation, in its full generality, is a difficult, fascinating, and intensely human Endeavour, as rich as any other area of human creativity.

2.1.1 Problems in Machine Translation

Problems faced in translating text using computers *i.e.* Machine Translation are as follows:

- **Word Order** The arrangement of words in a sentence varies across languages. For example order of words in English is SVO; Subject, Verb and Object, where as in Indian languages follows the pattern as SOV; Subject, Object and Verb. For example, English sentence *'Ram eats apple'* where *ram* is a subject, eats is a verb, and apple is an object; can be translated to Punjabi as *'RaM Seb khaNda HI'*(ਰਾਮ

ਸੇਬ ਖਾਂਦਾ ਹੈ) where *RaM* (*ਰਾਮ*) is a subject, *Seb* (*ਸੇਬ*) is an object, and *khaNda* (*ਖਾਂਦਾ*) is a verb.

- **Word Sense** The sense of word in one language may translate into a different sense with the words of another language. This creates problem in target language word selection.

- **Pronoun Resolution** Resolving pronominal references is important for Machine Translation. Unresolved references may lead incorrect translation, *e.g.* word *aUH (ਓਹ)* can be used in plural form in *aUH kY kR RHe Hn(ਓਹ ਕੀ ਕਰ ਰਹੇ ਹਨ),* same word can be used as *aUH kY kR RYHa HI(ਓਹ ਕੀ ਕਰ ਰਿਹਾ ਹੈ),* where *aUH* refers in singular form.

- **Idioms** A sentence containing idiomatic expressions is difficult to translate as idioms are composed of words that do not directly contribute to their meaning, sometimes replacing words constituting idioms with words from target language can result to funny and nonsensical translations. For example, idiom like *'an apple of eye',* in English translated literally would make no sense in Punjabi language.

- **Ambiguity** Sometimes same word in a language has different meanings in different contexts. So it is important to resolve the ambiguity depending upon the context in which the word is being used. For example,

  He went to the bank.

  Taj Mahal is situated on the bank of river Yamuna.

  In first sentence word *bank* refers to the financial institute, where as in second sentence *bank* refers to the bank of river

2.1.2 Characteristics of Indian languages

Languages used in India can be categorized in following categories:

- Indo-Aryan
- Dravidian
- Austro-Asian
- Tibetian-Burmese

Structure of the language within the group is similar to great extent. Various characteristics of Indian languages are as follows:

- Indian languages have SOV (Subject Object Verb) structure.
- Indian languages have free word order *i.e.* words within the sentence can be moved freely without changing the meaning of the sentence.
- Indian languages have relatively rich set of morphological variants.

2.2 Machine Translation Approaches

Machine Translations can be classified into following categories as shown in Figure 2.1:

- Direct Machine Translation
- Rule Based Translation
- Corpus Based Translation
- Knowledge Based Translation

2.2.1 Direct Translation

A Direct Translation System carries out word-by-word translation with the help of bilingual dictionary, followed by syntactic rearrangement. Direct Translation Systems involve little analysis of the source language, no parsing, and rely on a large bilingual

dictionary. Besides dictionary translation, the analysis performed in this approach is morphological analysis, preposition handling, syntactic arrangement and morphological generation.

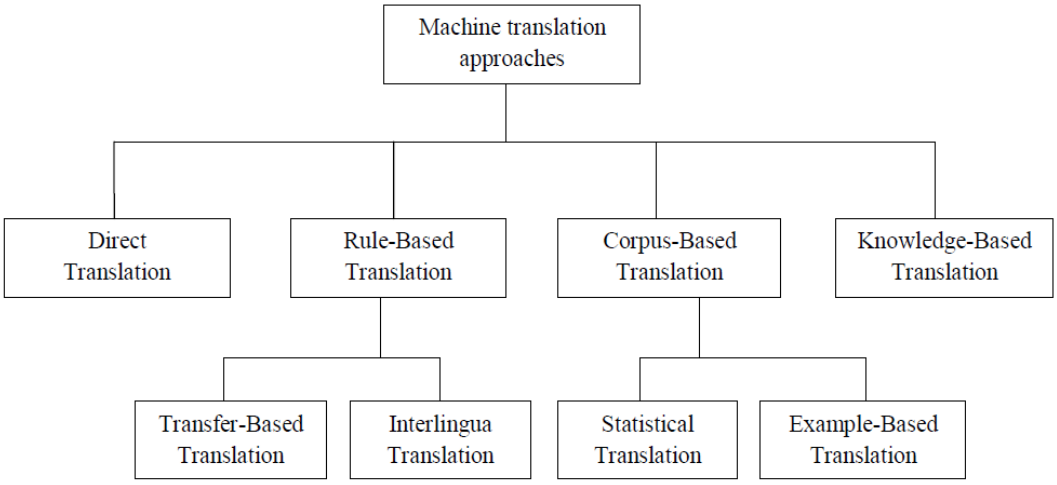


Figure 2.1: Machine Translation Approaches[23]

The general procedure for the direct translation can be summarized as shown in Figure2.2:

- Remove morphological inflections from the words to get the root word from the source language words.
- Look up a bilingual dictionary to get the target-language words corresponding to the source-language words.
- Change word order to that, which best matches the word order of the target language.

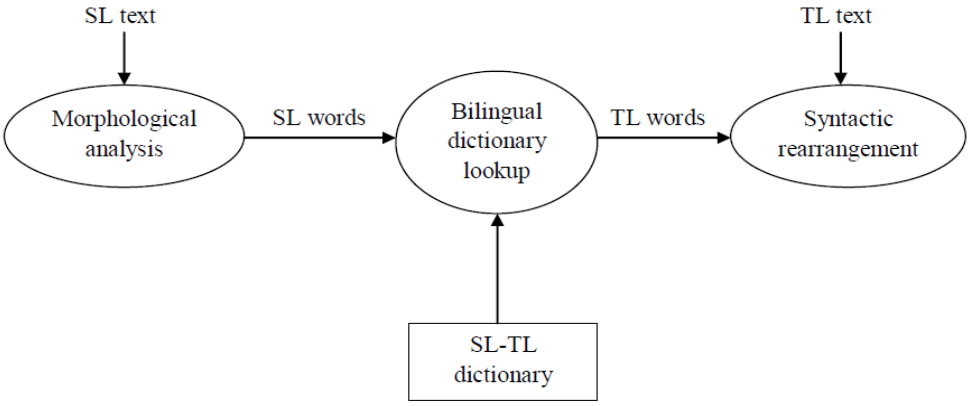


Figure 2.2: Direct Machine Translation[23]

2.2.2 Rule Based Translation

In Rule Based Translation, intermediate representation is produced by parsing the source text, and 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. Depending upon intermediate representation, these systems can be categorized as follows:

- **Transfer Based Machine Translation**

  These models transform the structure of input language to produce a intermediate representation that matches the rules of target language, and this transformation requires understanding of differences between source and target language.

  As shown in Figure 2.3, Transfer-Based Machine Translation has following three components.

  - Analysis- To produce source language structure.
  - Transfer- To transfer the source language representation to target language representation.

- Generation- To generate target language text using target level structure.

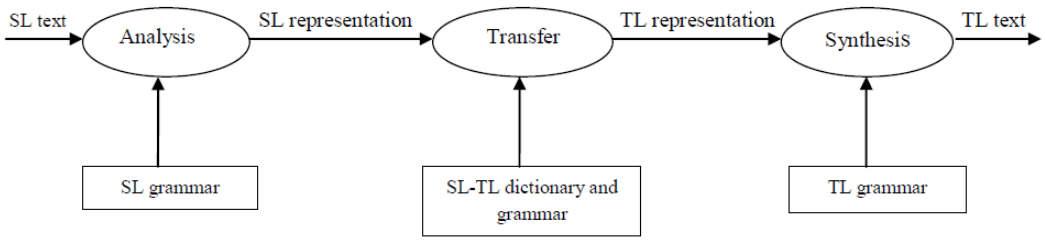


Figure 2.3: Transfer Based Machine Translation[23]

The first stage analyzes the source text and produces a structure confirming the rules of source language. It involves morphological, syntactic, and semantic analyses and involves parsing of source text. The second stage transfers source language representation into target language representation and target language text is generated by third stage.

The main advantage of this approach is its modular structure. The analysis of source language is independent of target language generator. To provide translation capability among set of languages, an analyzer and a generator component for each language and a transfer component of each pair is required.

A second advantage of Transfer Based Machine Translation is that, these can easily handle ambiguities that carry over from one language to another.

- **Interlingua Based Machine Translation**

In Interlingua-Based approach, the source language text is converted into language independent meaning representation called ‘interlingua’.

An Interlingua represents all sentences that mean the same thing in the same way regardless of the source language they happen to be in (Jurafsky and Martin, 2000)[4].

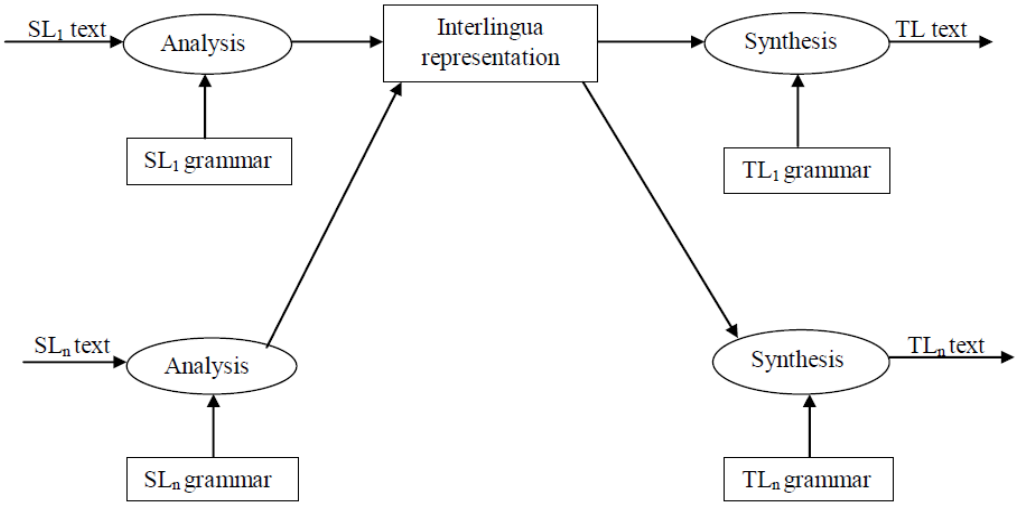


Figure 2.4: Interlingua Machine Translation[23]

Translation in Interlingua Based MT is a two stage process, analysis and synthesis, as shown in Figure 2.4.

In the first stage, source language is represented in Interlingua. In second stage, target language text is generated. The analysis phase is specific to the source language text and synthesis phase is specific to the target language. This makes it convenient to use in the multilingual environment. The same analysis component can be used for more than one language. This means that in order to build a system for *n* number of languages, we need only *n* analysis and *n* generation components as compared to *n(n-1)*complete MT systems in Direct Translation approach.

The amount of analysis in an Interlingua approach is more, as compared to the transfer based system. An Interlingua system has to resolve all the ambiguities so that, translation to any language can take place from Interlingua representation.

Another advantage of Interlingua is that it is a meaning-based representation that can be used in applications like Information Retrieval. The KANT [14] (Knowledge based Accurate Natural language Translation) system that is used to translate technical service information from English into other languages uses this approach.

Difficulty in using this approach is in defining a universal abstract representation which preserves the meaning of a sentence, because different languages conceptualize the world in different ways.

2.2.3 Corpus Based Machine Translation

Corpus Based Machine Translation Systems have advantages that, they are fully automatic and require significantly less human labor than traditional rule-based approaches. However, they require sentence-aligned parallel text for each language pair and cannot be used for language pairs for which such corpora does not exist. Corpus Based approach can be classified into Statistical and Example Based Machine Translation approaches.

- **Statistical Machine Translation**

  In Statistical Machine Translation (SMT), statistical methods are used to develop translation system. System can be described as shown in Figure 2.5. System can be divided into Language Model, Translation Model and Decoder. Decoder maximizes the product of Language Model probabilities and Translation Model probabilities.

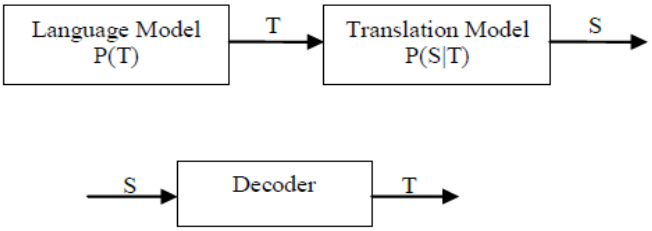


Figure 2.5: Statistical Machine Translation[19]

- **Example Based Machine Translation:** Example Based Machine Translation (EBMT) System uses past translation examples to generate translations for a given input. An Example Based MT system maintains an example-base consisting of translation examples between source and target language. When an input sentence is presented to it, the system retrieves a similar source language (SL) sentence from the example base and its translation. It than adapts the example translation to generate translation of the input sentence. EBMT has two modules, retrieval and adaption as shown in Figure 2.6.

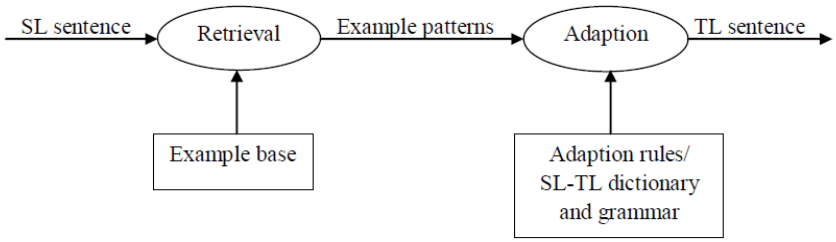


Figure 2.6: Example Based Machine Translation[23]

Retrieval: The task of this module is to retrieve translation examples from the Example base for a given input. The retrieval strategies attempts to find an example from the Example base which is similar to the input sentence. This means that a similarity measure has to be defined for sentence. These similarity measures may be based on word similarity or syntactic and semantic similarity.

Adaption: This module carries out the necessary modifications in the retrieved example pair to generate the translation of target sentence. This modification may involve addition, deletion or replacement of morphological words, constituent words.

- **Difference between Statistical Machine Translation and Example Based Machine Translation**

  Both Statistical Machine Translation (SMT) and Example Based Machine Translation (EBMT) are type of Corpus Based Machine Translation because both use bilingual corpus of pair of languages to perform translations. But basic difference between SMT and EBMT is that, SMT uses purely statistical methods in aligning the words and generation of texts, whereas EBMT uses a variety of linguistic resources, such as dictionaries and thesauri for the similarity measure, a bilingual lexicon for substitutions, and even parsing or morphological analysis at the analysis stage[24].

2.2.4 Knowledge-Based Machine Translation

Transfer and Interlingua approaches use semantic information; their central component is syntactic analysis. Semantic features are usually attached to syntactic structures and semantic processing occurs only after syntactic structures have been identified. Synthesis and analysis is restricted to sentences. Semantics-based approaches to language analysis have been introduced by AI researchers. The approaches require a large knowledge-base that includes both ontological and lexical knowledge. The basic AI approaches include: semantic parsing, lexical decomposition into semantic networks and resolution of ambiguities and uncertainties by reference to knowledge-bases [22].

2.3 Statistical Machine Translation

In this thesis, we have developed a Statistical Based Machine Translation System to translate text from English to Punjabi.

Statistical Machine Translation (SMT) is an approach to MT, which is characterized by the use of machine learning methods. SMT treats translation as a machine learning problem. This means that we apply a learning algorithm to a large body of previously translated text, known as a parallel corpus, parallel text, bi-text, or multi-text. Thus, SMT can be defined as:

Statistical Machine Translation (SMT) is a Machine Translation paradigm where translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual corpora. [3]

Warren Weaver, suggested the idea of Statistical Machine Translation in 1949[17], but researchers abandoned the concept that time, and it was reintroduced by researchers at IBM's Thomas J. Watson Research Center in 1991 [25].

In 1988 a group from IBM published the results of experiments on a system based purely on statistical methods. The effectiveness of the method was a considerable surprise to many researchers and encouraged the experiments with Statistical methods in subsequent years. Secondly, some Japanese groups began to publish preliminary results using methods based on corpora of translation examples[25].

In the Statistical Machine Translation, there is a pair of languages consisting of Source Language and Target Language; where Source Language is the input language from which translation is to be done and Target Language is the language in which translation to be done.

Statistical Machine Translation (SMT) is based on the following equation:

$$e = argmax\, P(e|f)$$

This equation is known as *Fundamental Equation of Statistical Machine Translation* [17].

Using Baye's theorem:

$$P(e|f) = \frac{P(e)P(f|e)}{P(f)}$$

$$e = argmax \frac{P(e)P(f|e)}{P(f)}$$

$$e = argmax\, P(e)P(f|e)$$

Symbols used in Fundamental Equation of Statistical Machine Translation are for French to English translation system, where *f* is used to represent source language and *e* is used to represent target language.

Replacing *f* by *S*, and *e* by *T*

Equation

$$e = argmax\, P(T)P(S|T)$$

Summarizes the three computational challenges presented by practice of SMT as:

- Estimating Language Model probability, *P(T)*,
- Estimating Translation Model probability, *P(S|T)*,
- Devising an efficient search for the target text that maximizes the product

These challenges corresponds to the problems namely, language modeling problem, translation modeling problem, and search problem. The search problem is also known as decoding. Outline of the system can be as shown in Figure 2.7

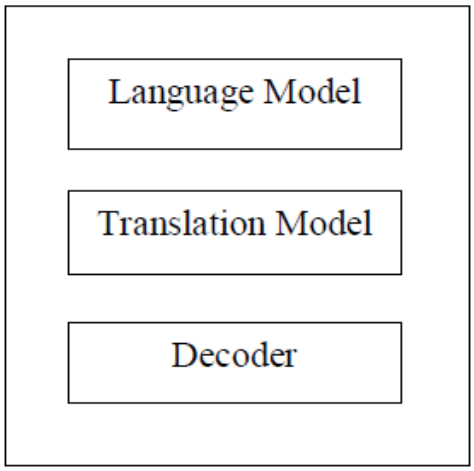


Figure 2.7: Outline of the Statistical Machine Translation system

2.3.1 Language 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. This probability of the sentence can be computed by using n-gram model[17].

This is achieved by decomposing sentence probability into a product of conditional probability using chain rule as follows[13]:

$$P(s) = P(w_1. w_2, w_{3,\ldots..,} w_n)$$

$$P(w_1)P(w_2/w_1)P(w_3/w_1w_2)P(w_4/w_1w_2w_3)\ldots\ldots.P(w_n/w_1w_2w_3\ldots.w_{n-1})$$

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 by the conditional probability given previous *n-1* words only.

If a model limits the history to the previous one word, it is a bi-gram model, for example, bi-gram approximation of P(east/ The Arabian knights are fairy tales of the) is

P(east/the)

If model limits the history to two words, than it is trigram model, *e.g.* P(east/of the)

Special pseudo word <s> is introduced to mark the beginning of the sentence.

Consider the following training set of data [23]

The Arabian Knights
These are the fairy tales of the east
The stories in Arabian knights are translated in many languages

Probabilities for bigram model are as follows:

P(the/<s>) = 0.67	P(Arabian/the) = 0.4	P(knights/Arabian) = 1.0
P(are/these) = 1.0	P(the/are) = 0.5	P(fairy/the) = 0.2
P(tales/fairy) = 1.0	P(of/tales) = 1.0	P(the/of) = 1.0
P(east/the) = 0.2	P(stories/the) = 0.2	P(of/stories) = 1.0
P(are/knights) = 1.0	P(translated/are) = 0.5	P(in/translated) = 1.0
P(many/in) = 1.0	P(language/many) = 1.0	

Probability of sentence; '*The Arabian knights are the fairy tales of the east*', can be calculated as follows:

P(The/<s>) x P(Arabian/the) x P(knights/Arabian) x P(are/knights) x P(the/are) x P(fairy/the) x P(tales/fairy) x P(of/tales) x P(the/of) x P(east/the)

= 0.67 x 0.5 x 1.0 x 1.0 x 0.5 x 0.2 x 1.0 x 1.0 x 1.0 x 0.2

= 0.0067

This gives the probability that sentence, '*The Arabian knights are the fairy tales of the east*', has the probability 0.0067 of occurrence in the above set of data.

2.3.2 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 we break process into smaller units *e.g.* words or phrases and learn their probability. We think of target translation of source sentence as being generated from source word by word.

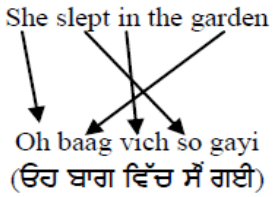
For example, we use the notation (T/S) to represent an input sentence S and its translation T. Using this notation

(aUH bag wYWch S/UN gaYI| she slept in the garden)

(ਓਹ ਬਾਗ ਵਿੱਚ ਸੌਂ ਗਈ | she slept in the garden)

means that '*aUH bag wYWch S/UN gaYI'* (ਓਹ ਬਾਗ ਵਿੱਚ ਸੌਂ ਗਈ) is translation of '*she slept in the garden'*.

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

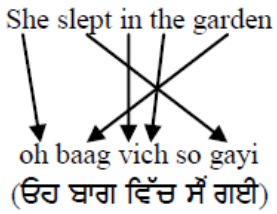


An alignment mark shows the origin of each English word (source) in Punjabi sentence as a target. This alignment can be represented by following notation

(Oh(ਓਹ) baag(ਬਾਗ) vich(ਵਿੱਚ) so(ਸੀ) gayi(ਗਈ)| she(1) slept(4) in(3) the() garden(2)),

in this case ***the*** is not aligned to any Punjabi word.

Another possible alignment for the pair of sentences can be



This alignment can be represented by following notation

(aUH bag wYWch S/UN gaYI| she(1) slept(4) in(3) the(3) garden(2))

From the above discussion, it is clear that for any given source-target sentence pair; a number of alignment is possible. Some of them are more probable and for simplicity we consider translation model as word by word alignment. We denote the set of alignment by 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

$$P(S|T) = sum\ P(S, a/T)$$

The sum is over the elements of A(S, T). We restrict ourselves to the case where each English word has only exactly one connection.

For the alignment,

(aUH bag wYWch S/UN gaYI| she(1) slept(4) in(3) the garden(2))

(ਓਹ ਬਾਗ ਵਿੱਚ ਸੀ ਗਈ|she(1) slept(4) in(3) the garden(2)),

P(ਓਹ ਬਾਗ ਵਿੱਚ ਸੌਂ ਗਈ| she slept in the garden), can be computed by multiplying the translation probabilities T(aUH|she), T(bag | garden), T(wYWch | in), T(null| the), and T(S/UN | slept), and T(gaYI | null).

To generate target sentence from source sentence, we have to follow the steps are as follows[23]:

i. Select the length of S with probability L where L=P[length (S)=*m*] is a constant *i.e.* We assume that all lengths are equally likely with probability *L*.

ii. Select an alignment 'a' with probability P(a|S). There are $(l+1)^m$ possible alignments. Assuming all possible alignments are equally likely, the probability of alignment a, P(a|S), is

$$P(a|S) = L \times 1/(l+1)^m$$

iii. Select the j^{th} English word with probability

$$P(S/a, T) = \prod_{j=1}^{m} T(Sj \mid Taj)$$

The joint likelihood of Punjabi string and an alignment given an English string is:

$$P(S, a/T) = P(a/T) \times P(S/a, T)$$

$$L/(i+1)^m \prod_{j=1}^{m} T(S_j / T_{aj})$$

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, therefore

$$P(S, a/T) = (l+1)^m / L \sum_{aa=0}^{l} \cdots \sum_{am=0}^{l} \cdots \prod_{j=1}^{m} T(S_j / T_{aj})$$

2.3.3 Search

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

$$Pr(S, T) = argmax\ P(T)\ P(S|T)$$

Here problem is the infinite space to be searched. So Brown, et al. (1993)[17] suggested the use of stacked search, in which we maintain a list of partial alignment hypothesis. 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 (aUH bag wYWch S/UN gaYI|*), 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:

(aUH(ਓਹ) bag(ਬਾਗ) wYWch(ਵਿੱਚ) S/UN(ਸੈਂ) gaYI(ਗਈ)| she(1)*),

(aUH(ਓਹ) bag(ਬਾਗ) wYWch(ਵਿੱਚ) S/UN(ਸੈਂ) gaYI(ਗਈ)| *garden(2)*),

(aUH(ਓਹ) bag(ਬਾਗ) wYWch(ਵਿੱਚ) S/UN(ਸੈਂ) gaYI(ਗਈ)| *slept(4))

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

2.4 Tools Used for Implementation of Statistical Machine Translation 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 2.7, shows some open source tools that are available to use.

- CMU-Statistical Language Modeling Toolkit
- SRILM
- GIZA++
- MGIZA
- Moses
- ISI ReWrite Decoder
- Pharaoh

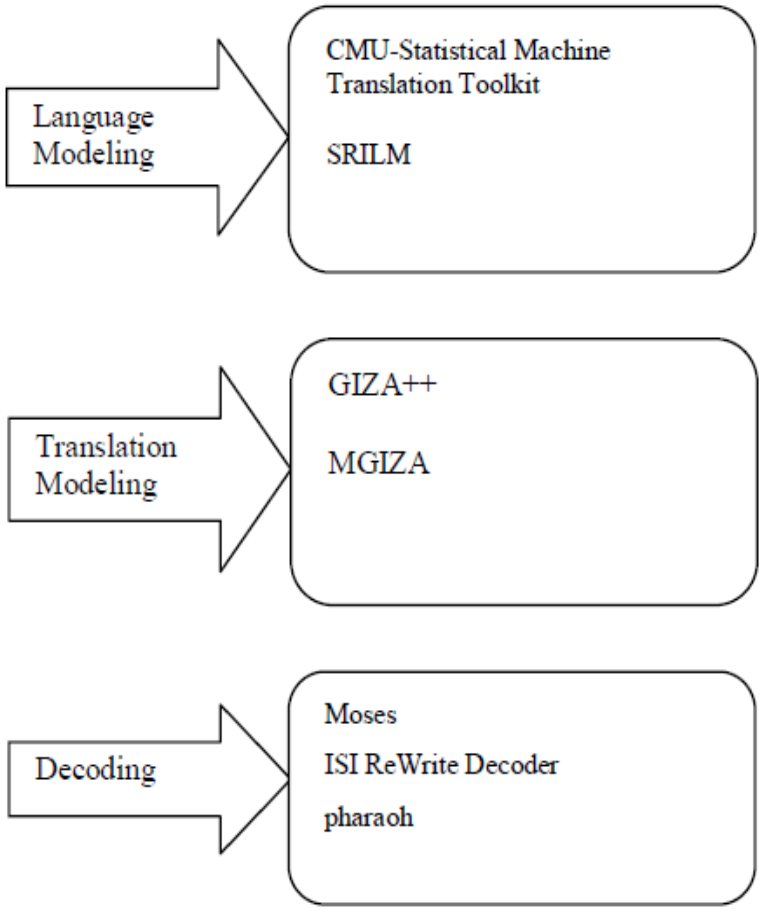


Figure 2.8: Statistical Machine Translation Tools

2.4.1 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[16].

CMUSLM is freely available for download and use at following link:

http://www.speech.cs.cmu.edu/SLM/toolkit.html

2.4.2 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[21].

SRILM is freely available for download and use at the following link:

http://www.speech.sri.com/projects/srilm/download.html

2.4.3 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[6].

GIZA++ is freely available for download and use at the following link:

http://www.fjoch.com/GIZA++.20030930.tar.gz

2.4.4 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 and use at following link:

http://www.cs.cmu.edu/~qing/

2.4.5 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 required is a collection of translated texts (parallel corpus). Moses works with SRILM to develop Language Model, and GIZA++ to develop Translation Model[21].

Moses is freely available for download and use at the following link:

http://mosesdecoder.sourceforge.net/download.php

2.4.6 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[10]. It freely available for download and use at the following link:

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

2.4.7 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[30]. It can be obtained from following link:

http://www.isi.edu/licensed-sw/pharaoh/

2.5 Some Popular Machine Translation Systems

There are Machine Translation systems that are used to translate text from one language to another. Some of the popular Machine Translation systems are:

- Systran,
- Google Translator,
- Bing Translate,
- Hindi to Punjabi Machine Translation System,
- METAL

- **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.

- **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[7]. It is developed by Punjabi University, Patiala. It can translate text file, web-pages from Punjabi to Hindi. It is available to use at following link:
  http://h2p.learnpunjabi.org

- **METAL**

  METAL is a translation system initiated in the late seventies by Siemens-Nixdorf, with the University of Texas[4]. 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 following link:
http://www.lant.be/

2.6 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.

Various Machine Translation projects related to Indian languages are [5]:

- **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 IIT-Kanpur, in collaboration with ER&DCI, Noida, and has been funded by TDIL[22].

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

- **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.

- **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.

- **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.

- **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 langauge using a 'deconverter'.

  Thapar University, Patiala and Punjabi University, Patiala are working on the development of UNL envcoverter for Punjabi language.

- **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.

- **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.

- **English-Hindi Statistical MT:** The IBM India Research Lab at New Delhi has recently initiated work on Statistical MT between English and Indian languages, building on IBM's existing work on Statistical MT.

2.7 Machine Translation Tools and Punjabi Language

Some government, educational organizations and some individual researchers are working for the technological development of Punjabi language. Main organizations working in this field are Thapar University, Punjabi University, C-DAC Noida and IIIT Hyderabad[8].

In 1990, first Punjabi to Hindi direct Machine Translation system was developed by IIIT Hyderabad, under Anusaraka project headed by Dr. Rajeev Sangal. The system performs translations at word level. As word order of Hindi and Punjabi languages more or less same and that the grammar function depends more on inflection than word order.

In 2009, at Punjabi University, Patiala, online Hindi-Punjabi and Punjabi-Hindi Machine Translation system was developed by team of Dr G.S. Lehal, Dr. G.S. Joshan and Vishal Goyal. The system is freely available on the university website[9]. IIIT Hyderabad has also launched online Punjabi-Hindi Machine Translation system. C-DAC Noida is working for development of Hindi-Punjabi and reverse Machine Translation system.

At Thapar university, Patiala, team is working on Punjabi Language Server which includes Punjabi-UNL Enconverter and UNL-Punjabi Deconverter[18]. The main objective of this project is to study Punjabi Language and make the first Punjabi Deconverter. This will convert UNL expressions into Punjabi sentences.

Chapter 3

Problem Statement

Today, World is a global village and information flows from one end to another end, with the speed of light. With the help of Internet one can access the information available online from anywhere in the world. But this world is a multi lingual also, and different regions have different languages to communicate. Most of the information available online is in English language. At the regional level, where vast number of people do not use English, cannot able to access that information. This led to the digital divide in the society. Machine Translation can help to overcome this digital divide in the society.

To bridge this gap Machine Translation can help. Numbers of Machine Translation systems are available to translate text from one language to another. But no system can translate text from English to Punjabi. We have used Statistical Machine Translation, where user feed the input text in English language, and system provides the translated text in Punjabi language.

In this thesis, we have developed the Statistical Based Machine Translation system to translate text from English to Punjabi, where English is a Source Language and Punjabi is a Target Language for the system.

3.1 Gap Analysis

Gaps in the existing technologies are as follows:

- There is no such system which can perform translation of text from English to Punjabi.
- Existing Google translator cannot translate text, web pages neither from English to Punjabi nor from Punjabi to English.
- The existing system is based on rule based approach to translate text from Hindi to Punjabi.

- No attempt to Statistical Based Machine Translation for English to Punjabi has been made so far.

3.2 Objectives

Underlying are the objectives of the thesis.

- To Understand Language Model, Translation Model, and Decoding stages of SMT.
- Creation of Language Model for Punjabi language with the use of CMU-Statistical Language Modeling Toolkit.
- Creation of Translation Model for English-Punjabi language pair with the use of GIZA++ .
- Generation of Punjabi sentences with the use of ISI ReWrite Decoder.
- Testing the system.

3.3 Methodology

We purposed the translation of text from English to Punjabi using Statistical methods *i.e.* Statistical Machine Translation. Our effort is to develop a system that can translate text from English language to Punjabi language without the human efforts. This can be done by making the use of tools available for Statistical Machine Translation. In Statistical Machine Translation there are three major parts of the system, Language Model, Translation Model, and Searching (also known as decoding). Language Model calculates the probabilities related to the target language. Translation Model calculates the probabilities regarding the substitution of target language word with the source language word, along with the fertility and distortion parameters. We have used CMU-Statistical

Language Modeling Toolkit for the development of Language Model, GIZA++ and mkcls to build Translation Model, and ISI ReWrite Decoder to perform searching.

System is based on Unix and accepts sentences in English language in terminal window and produce output in Punjabi.

3.4 Applications of Proposed System

Current research work is focused on the translation of simple sentences of English into Punjabi, later on which can be extended to complex sentences by increasing the size of bilingual corpus of English-Punjabi pair of languages. In future the system can be used to translate web pages from English to Punjabi also.

Chapter 4

Design and Implementation of English to Punjabi Statistical Based Machine Translation System

In this chapter, development of Corpus, Architecture of system, and implementation is discussed.

4.1 Development of Corpus

Statistical Machine Translation system requires parallel corpus of the pair of source and target languages. For our system source language is English, and target langauge is Punjabi. So parallel corpus for English-Punjabi language is required. But unfortunaly no parallel corpus for English-Punjabi was available which could be used with SMT system. Parallel corpus of about 6000 sentences of English-Punjabi pair of language was developed manually by following ways:

- Translation of English text to Hindi text with the help of google translation[15]. Than Hindi text is translated to Punjabi text using Hindi to Punjabi translator[9]. Than alignment of sentences was done manually.
- Development of parallel text by translating kids stories from English to Punjabi manually.
- By aligning the English sentences to sentences translated in Punjabi, in Emilly corpus, because parallel translated text in Emilly corpus cannot be used directly in our system.

4.2 Architecture of English to Punjabi Statistical Based Machine Translation System

Statistical Machine Translation system can be divided into three parts as shown in Figure 4.1.

- Language Model
- Translation Model
- Decoder.

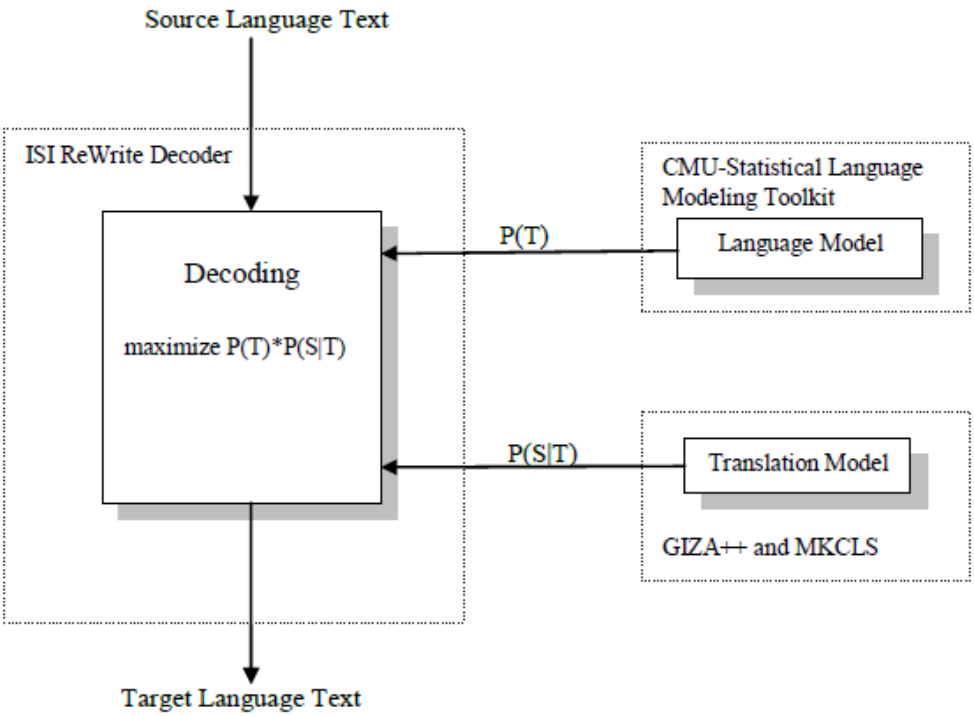


Figure 4.1: Architecture of the System

Language Model for the Punjabi is developed by using CMU-Statistical Machine Translation Toolkit, and GIZA++ and mkcls are used to develop Translation Model for the pair of English-Punjabi languages. ISI ReWrite Decoder, which works with CMU-Statistical Language Modeling Toolkit and GIZA++ is used to perform searching. Probabilities of Language Model and Translation Model are passed to the decoder. To perform translations decoder is started from terminal window. On start up, decoder loads probability tables of Language Model and Translation Model from the values specified in

configuration file. Decoder is set to server mode to perform translations. Decoder accepts the sentence in English language and returns the translated sentence as an output.

4.2.1 Language Model

Language Model is build by using Carnegie Mellon University (CMU)-Statistical Language Modeling Toolkit, written by Roni Rosenfled, released in 1994. It is a set of Unix software tools designed to develop language modeling work in the field of Statistical Machine Translation.

In the development of Language Model, text file of Punjabi is given as input, and numbers of files with different extensions are generated. Various files related to the development of Language Model and there extensions are as shown in Table 4.1.

Following tools are used to develop Language Model:

- **text2wfreq**

  Input to this tool is a simple text file, and it generates word frequency file, which contain list of every word which occurred in the text, along with its number of occurrences.

- **wfreq2vocab**

  It takes word frequency file created by ***text2wfreq*** as input and generates vocabulary file.

Table 4.1: Terminology and File Formats

Name	Description	File extension
Text stream	A file containing text. It may or may not have markers to indicate context cues, and white space can be used freely.	.text
Word frequency file	A file containing a list of words, and the number of times that they occurred. This list is not sorted; it will generally be used as the input to wfreq2vocab.	.wfreq
Vocabulary file	A file containing a list of vocabulary words.	.vocab
Id n-gram file	A file containing a numerically sorted list of n-tuples of numbers, corresponding to the mapping of the word n-grams relative to the vocabulary.	.idngram
Binary language model file	Binary file containing all the n-gram counts, together with discounting information and back-off weights.	.binlm

- **text2idngram**

  This tool use text file and vocabulary file, and creates Idngram file, which contains numerically sorted list of n-tuples of numbers, corresponding to the mapping of the word n-grams relative to the vocabulary.

- **idngram2lm**

  This is a tool to create language model file in binary format. Inputs to this tool are Id n gram file generated by ***text2idngram*** and vocabulary file created by ***wfreq2vocab***.

- **Steps to develop language model**

Binary file of language model *(.binlm)* is built from simple text file *(.text)* of a target language. By following the steps as shown in Figure 4.2, binary file of language model is generated from text file.

In first step text file is converted to word frequency file *(.wfreq)* by using ***text2wfreq***.

Then word frequency file is used by ***wfreq2vocab*** to generate vocabulary file *(.vocab).*

In third step, ***text2idngram*** use text file and vocabulary file to create Id n gram file *(.idngram)*.

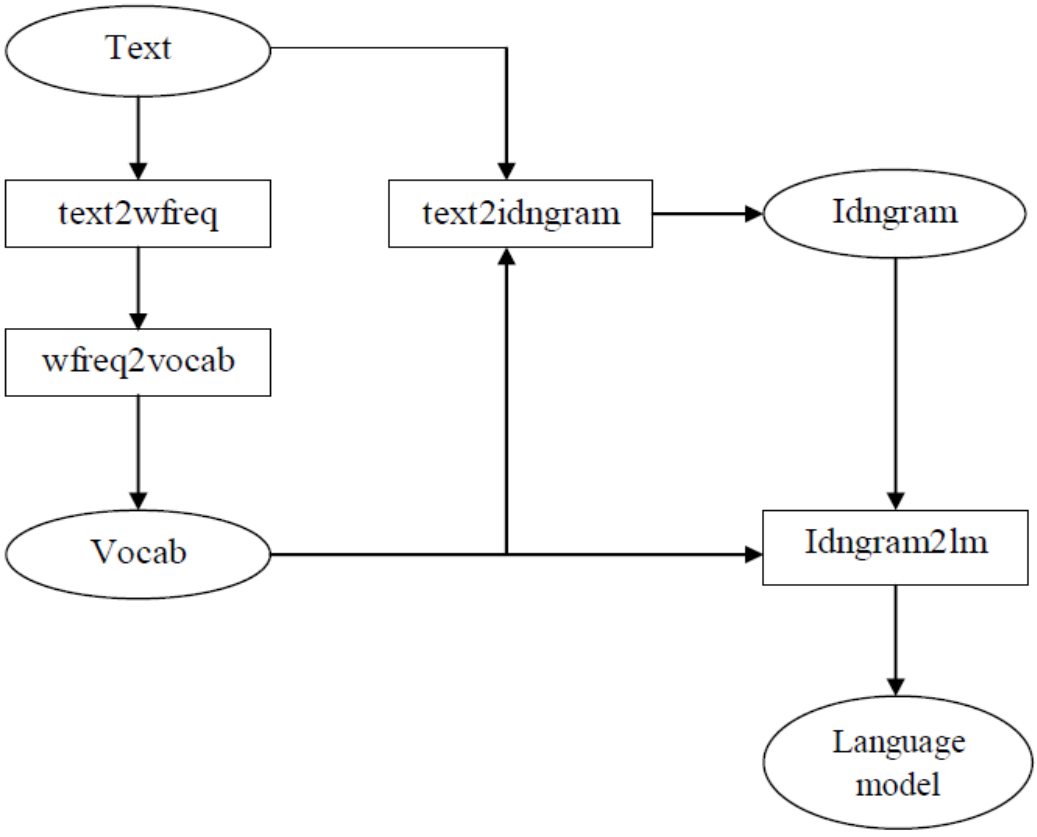


Figure 4.2: Development of Language Model

In fourth step, binary file of language model *(.binlm)* is generated by ***idngram2lm*** by using id n gram file *(.idngram)* and vocabulary file *(.vocab)*.

Implementation of steps

I. Conversions of text file to word frequency file, that contains list of every word occurring in the text, and its frequency of occurrence.

```
[gagan@localhost CMU-Cam_Toolkit_v2]$ ./bin/text2wfreq
<punjabi.text> punjabi.wfreq
```

Here:

punjabi.text, is text file of language for which language model is to be build, and

punjabi.wfreq, is Word frequency file, containing the list of words, along with their occurrences as given below

```
[gagan@localhost CMU-Cam_Toolkit_v2]$ head -10
punjabi.wfreq
ਕਰਨਾ 48
ਕਰਨੀ 12
ਕਰਨੇ 3
```

Here first column represents word occurring in the text, and second column represents the total number of occurrences of the word *i.e.* word ***ਕਰਨਾ*** occurs 48 time in the text.

II. Generation of vocabulary file from word frequency file.

```
[gagan@localhostCMU-Cam_Toolkit_v2]$./bin/wfreq2vocab
<punjabi.wfreq> punjabi.vocab
```

Here

punjabi.vocab, is Vocabulary file, containing the list of vocabulary words. Structure of vocabulary file is as follows.

ਯਾਰ
ਰਾਸ਼ਟਰ
ਹੁਣੇ

It is a single column file, containing the list of all the words occurring in the text.

III. Generation of idngram of the training text, based on the vocabulary.

```
[gagan@localhost CMU-Cam_Toolkit_v2]$ ./bin/text2idngram -
vocab punjabi.vocab <punjabi.text> punjabi.idngram
```

Here:

punjabi.idngram is Id n-gram file.

Input to *text2idngram* is *.text* and *.vocab*. It generates *.idngram*.

IV. Conversion of idngram into binary format of the language model,

```
[gagan@localhost CMU-Cam_Toolkit_v2]$ ./bin/idngram2lm -
idngram punjabi.idngram -vocab punjabi.vocab -binary
punjabi.binlm
```

Here:

punjabi.binlm binary language model file, which contains n-gram counts.

4.2.2 Translation Model

Translation Model is developed by using GIZA++. GIZA++ is an extension of the program GIZA (part of the SMT toolkit EGYPT (http://www.clsp.jhu.edu/ws99/projects/mt/toolkit/) which was 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 (CLSP/JHU). GIZA++ includes a lot of additional features. The extensions of GIZA++ were designed and written by Franz Josef Och[6].

Features of GIZA++:

- 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 (the parameter of the Poisson-distribution of the sentence lengths is computed automatically from the used training corpus)

GIZA++ contains following tools:

- GIZA++ : Tool to develop Translation Model files
- Plain2snt.out: Tool to transform simple text into GIZA text.
- Snt2plain.out: Tool to transform from GIZA text into plain text.

- **Steps to Develop Translation Model**

  To develop Translation Model, two tools are used, GIZA++ and mkcls. Translation Model is developed in three steps, as shown in Figure 4.8. In first step, plain text is converted into GIZA format using ***plain2snt.out***. In second step, ***mkcls*** is used to

generate classes. And finally plain text file of source and target languages and bitext file in GIZA format are used to develop Translation Model files by using ***GIZA++***.

I. Convert plain text into GIZA++ format,

```
[gagan@localhost bin]$ ./plain2snt.out english.txt punjabi.txt
```

Input to ***plain2snt.out*** is text file of the source and target language aligned corpus. Output files are Vocabulary files (*english.vcb, punjabi.vcb)* and bitext files in GIZA format *(english_punjabi.snt, and punjabi_english.snt)*.

Here:

english.vcb, consists of each word from the English corpus in the front of the unique id assigned to the English word followed by frequency of occurrence of the word in the text file as follows.

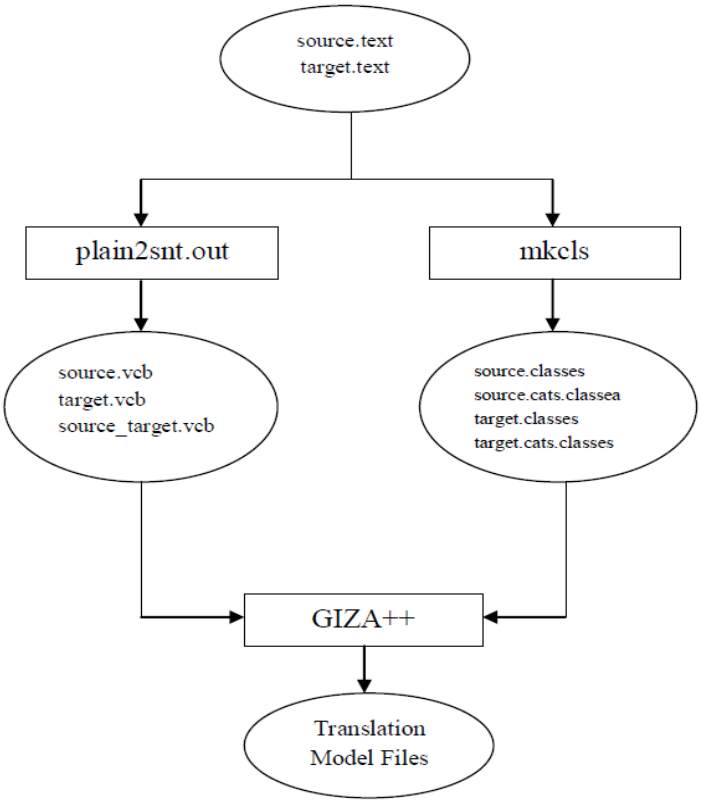


Figure 4.3: Development of Translation Model

```
[gagan@localhost bin]$ head -10 english.vcb
2 the 4856
3 passengers 2
4 objected 1
5 to 2716
6 it 395
7 . 4480
```

Here, file *english.vcb* signifies that, unique id *2* is assigned to word *the*, which is occurring 4856 times in the text.

punjabi.vcb consists of each word from the Punjabi corpus in the front of unique id assigned to the Punjabi word followed by the frequency of occurrence of the word in the text file as follows:

```
[gagan@localhost bin]$ head -10 punjabi.vcb
2 ਮੁਸਾਫਿਰਾਂ 1
3 ਨੇ 96
4 ਇਸ 652
5 ਉੱਤੇ 68
6 ਏਤਰਾਜ਼ 1
```

Here, in file *punjabi.vcb* word ਮੁਸਾਫਿਰਾਂ is assigned unique id 2, and is occurring one time in the text, and word ਨੇ is assigned unique id 3, and is occurring 96 times in the text.

english_punjabi.snt consists each sentence from parallel English and Punjabi corpus translated into the unique number for each word. First line, specifies the number of times the sentence pair occurred. Second line, is the source sentence where each token is replaced by its unique integer id from the vocabulary file and the third is the target sentence in the same format.

```
[gagan@localhost TM]$ head -10 En2Pn.snt
1
8298 139 26 195 7315 28 20
2 4947 10740 7 29 3
1
30 743 1676 2 21 1877 8688 768 1842 6381 6 7814 284 7 210
12 2 41 6 1665 37 132 27 4
2964 8160 618 18 125 748 8158 7473 116 2 8377 5 1169 6 38
713 2181 5 450 3
1
29 365 1893 7 1130 200 13 4
12 52 313 11 1942 6 625 163 3
```

Here, first line in the file specifies the number of occurrences of the sentence pair in the text *i.e.* 1 represents that sentences pair is occurring single time, and second line is sentence of English sentence where each token is replaced by unique id, and third line is target sentence where each token is replaced by its unique id.

II. Generate word vs frequency (classes) and frequency vs words (cats) files:

```
[gagan@localhost bin]$ ./mkcls -penglish.txt
-Venglish.vcb.classes

[gagan@localhost bin]$ ./mkcls -ppunjabi.txt
-Vpunjabi.vcb.classes
```

Input to *mkcls* is also text files of the English and Punjabi text files of the aligned corpus. Execution of *mkcls* results in generation of *english.vcb.classes, english.vcb.classes.cats, punjabi.vcb.classes,* and *punjabi.vcb.classes.cats.*

Here:

english.vcb.classes file consists an alphabetical list of all words (including punctuation) in the *english.text* file corresponding to frequency count of the word.

```
[gagan@localhost bin]$ head -10 english.vcb.classes
!  62
"  33
Let      37
The      99
care     75
fixed    76
without  13
```

Here, file e*nglish.vcb.classes* signifies that word *Let* is occurring 37 times, *care* is occurring 75 time, and so on.

punjabi.vcb.classes file consists an alphabetical list of all words (including punctuation) in the *punjabi.text* file corresponding to frequency count of the word.

```
[gagan@localhost bin]$ head -10 punjabi.vcb.classes
!  40
"  88
ਯਾਰ 85
ਰਾਸ਼ਟਰ    70
ਹੁਣੇ 81
%  101
```

Here, in file *punjabi.vcb.classes*, third line signifies that word *ਯਾਰ* is occurring 85 times, *ਰਾਸ਼ਟਰ* is occurring 70 times, and so on.

english.vcb.classes.cats, file contains list of frequency and set of words of the e*nglish.text,* for the corresponding frequency.

punjabi.vcb.classes.cats, file contains list of frequency and set of words of the *punjabi.text,* for the corresponding frequency.

III. Finally run GIZA++:

```
[gagan@localhost bin]$ ./GIZA++ -t punjabi.vcb -s english.vcb
-c english_punjabi.snt
```

Inputs to the GIZA++ are the vocabulary files of source and target language *(English.vcb, punjabi.vcb)* and bitext file *(english_punjabi.snt).*

Here:

t is paRameter for target language,

s is paRameter for source language,

c is paRameter for bitext file in GIZA++ text(*.snt*) format file.

Successful execution of the above steps results in creation of following files:

- **Translation Tables (T TABLE (*.t3.*))**

  Translation tables are the files containing information about the probabilities related to the alignment of the tokens of source and target words.

  - **t3.final:** This file contains three columns, first column represents unique ids of source words, second column represents unique ids of target words, and third column represents probability of alignment.

Figure 4.4: t3.final

  - **ti.final:** This file contains word alignments from the bitext corpus. Word alignments are in the specific words unique id, followed by the probability of the alignment.

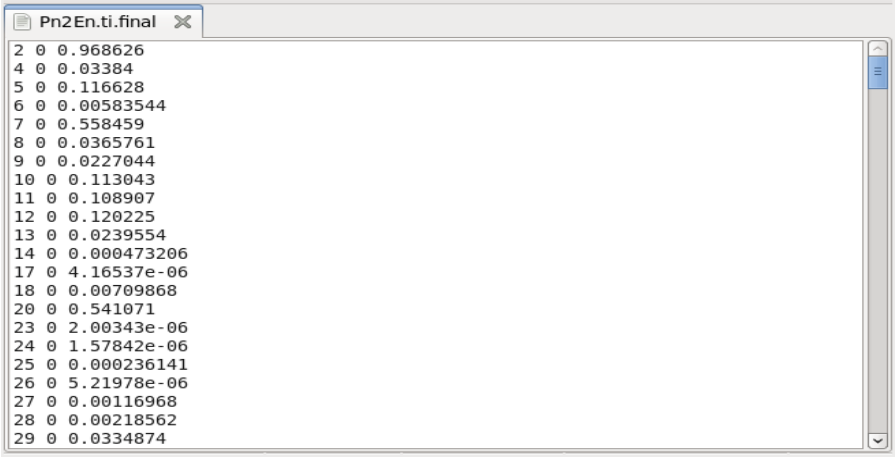


Figure 4.5: ti.final

Here first column specifies English words replaced by unique ids of token, second column specifies Punjabi words replaced by unique ids, and third column specifies the probability of replacing source word by target word.

- **Fertility Table (N TABLE (*.n3.*))**

  This file contains the values of probabilities related to fertilities of the source words.

  n3.final: This file contains probability of a source token having zero fertility, one fertility, …..*N* fertility, in the following format:

  s_id p*0* p*1* p*2*……p*N*

  Here:

  s_id is a unique id of source token

  p*0* is the probability that source token has zero fertility,

  p*1* is the probability that source token has one fertility,

  .

  p*N* is the probability that source token has maximum possible fertility.

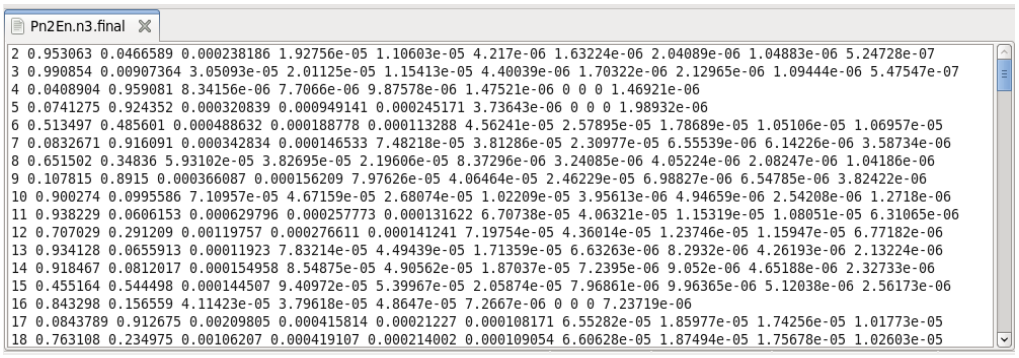


Figure 4.6: n3.final

Here in Figure 4.5 first line of file represents that unique id 2 has following probabilities

0.953063 that token 2 has fertility 0,

0.0466589 that token 2 has fertility 1,

0.000238186 that token 2 has fertility 2, and so on.

- **P0 TABLE (*.p0*)**

  This file contains only one line with real number which represents the probability of inserting a NULL after source token.

- **A TABLE (*.a3.*)**

  This file contains the values of distortion probability of source words.

  This contains a table with the following format:

  i j l m P (i / j, l, m)

  Where:

  j = position of target sentence,

  i = position of source sentence,

  l = length of the source sentence,

  m = length of the target sentence, and

  P(i / j, l, m) represents the probability that a source word in position i is moved to position j in a pair of sentences of length l and m

Figure 4.7: a3.final

- **Alignment File (*.A3.*)**

  This file contains matches of the source sentence to the target sentence and gives the match an alignment score. This file is represented by three lines for each sentence pair.

  First line contains information about the sentence serial number in training corpus, sentence length and alignment probability.

  Second line contains source sentence.

  Third sentence contains target sentence. Each token in the target sentence is followed by a set of numbers. These numbers represent the positions of the source words to which this target word is connected, according to alignment.

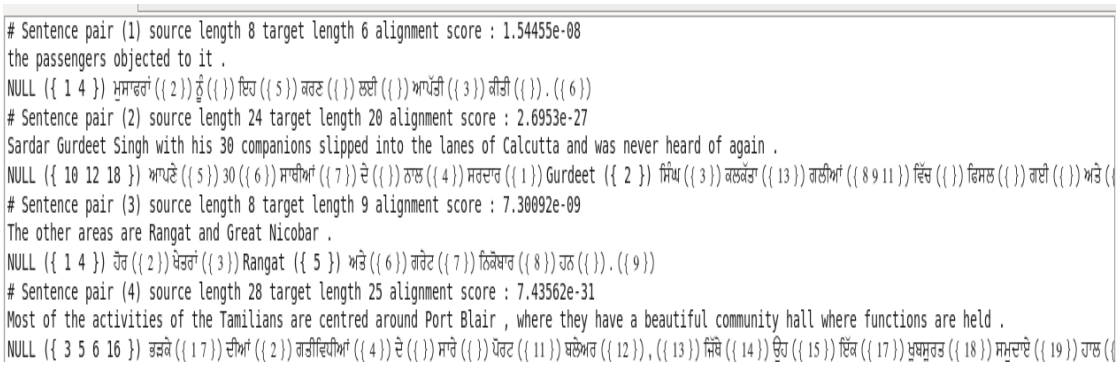


Figure 4.8: Alignment File

Here first line specifies that in sentence pair 1, length of source sentence is 8, and that of target sentence is 6, and alignment score is 1.54455e-0.6.

Second line is source sentence of English corpus file, and third line specifies tokens of Punjabi sentence followed by number, specifying the word to which it is connected *e.g.* token ਮੁਸਾਫਿਰਾਂ is connected to word passengers.

- **Perplexity File (*.perp)**

  This file is generated at the end of the training. It summarizes perplexity values for each training iteration.

#trnsz	tstsz	iter	model	trn-pp	test-pp	trn-vit-pp	tst-vit-pp
5553	0	0	Model1	14057.9	N/A	341225	N/A
5553	0	1	Model1	119.305	N/A	603.659	N/A
5553	0	2	Model1	66.6503	N/A	211.984	N/A
5553	0	3	Model1	50.7803	N/A	122.798	N/A
5553	0	4	Model1	45.6	N/A	95.6724	N/A
5553	0	5	HMM	43.1604	N/A	84.0774	N/A
5553	0	6	HMM	33.2121	N/A	45.0295	N/A
5553	0	7	HMM	24.5048	N/A	29.5671	N/A
5553	0	8	HMM	21.0917	N/A	24.3053	N/A
5553	0	9	HMM	19.5923	N/A	22.0557	N/A
5553	0	10	THTo3	16.9565	N/A	18.0157	N/A
5553	0	11	Model3	38.2347	N/A	40.6686	N/A
5553	0	12	Model3	33.6771	N/A	35.213	N/A
5553	0	13	Model3	32.1813	N/A	33.3651	N/A
5553	0	14	Model3	31.466	N/A	32.5091	N/A
5553	0	15	T3To4	31.025	N/A	32.0069	N/A
5553	0	16	Model4	22.8124	N/A	23.3047	N/A
5553	0	17	Model4	21.0842	N/A	21.4775	N/A
5553	0	18	Model4	20.4427	N/A	20.825	N/A
5553	0	19	Model4	19.939	N/A	20.3156	N/A

Figure 4.9: Perplexity File

- **Revised Vocabulary files (*.src.vcb, *.trg.vcb)**

  The revised vocabulary files are similar in format to the original vocabulary files. The only exceptions is that the frequency for each token is calculated from the given corpus (*i.e.* it is exact), which is not required in the input.

- **Final paRameter file (*.gizacfg)**

  This file includes all the paRameter settings that were used in order to perform the training.

4.2.3 Decoder

To perform decoding ISI ReWrite Decoder is used. ISI ReWrite Decoder works with CMU-Statistical Language Modeling Toolkit and GIZA++. It uses the paRameters for computing P(T) provided by CMU-Statistical Language Modeling Toolkit, and the paRameters for computing P(S|T) provided by GIZA++ to translate new sentences. For example to translate English sentence *e*, we require Punjabi sentence *p* which maximizes the product of these two terms, known as decoding.

ISI's ReWrite Decoder needs access to

- A monolingual Language Model for the target language, as produced by the CMU-Cambridge Language Modeling Toolkit, and
- Information about the Translation Model, as produced by the *GIZA* Toolkit.

The location of the Language Model file can be passed to the decoder with the switch **`--lmfile`**. Information about the Translation Model is stored in the *Translation Model Config File*, the location of which is passed to the decoder with either the switch **`--config`** or **`–tmfile.`**

Decoder config file is used to pass the paRameters to the decoder, so that paRameters do not have to be specified on the command line.

Pn2En.Decoder.config

```
LanguageModelFile = /home/gagan/Desktop/SMT_QuickRun1.2/MY_SMT/LM/Pn2En.binlm
TM_RawDataDir = /home/gagan/Desktop/SMT_QuickRun1.2/MY_SMT/TM
TM_BinDataDir = /home/gagan/Desktop/SMT_QuickRun1.2/MY_SMT/TM
Source.vcb = Pn2En.trn.src.vcb
Target.vcb = Pn2En.trn.trg.vcb
Source.classes = p.vcb.classes
Target.classes = e.vcb.classes
TTable = Pn2En.t3.final
InverseTTable = Pn2En.ti.final
NTable = Pn2En.n3.final
D3Table = Pn2En.d3.final
D4Table = Pn2En.D4.final
ATable = Pn2En.a3.final
PZero = Pn2En.p0_3.final
Model = 4 // can be changed to 3 if we want to run model 3 decoder
Top = 1 // can be changed to 2, 3, ...10, if you want to see top x translations
LanguageModel = trigrams // unigrams, bigrams
WeightTM = 0.5 // other values between 0 and 1.
Output = oneSentencePerLine // target-source pairs
ShowProbabilitiesInOutput = yes // no
KeepAsciiFiles = yes // no
FZeroWords = ZeroFert.rw
```

Figure 4.10: Decoder.config File

The ISI ReWrite Decoder uses the paRameters for computing $P(p)$ provided by the CMU-Cambridge Language Modeling Toolkit, and the paRameters for computing $P(e \mid p)$ provided by GIZA++ to translate new sentences. To translate a English sentence e, we need the Punjabi sentence p which maximizes the product of those two terms. This process is called *decoding*. It is impossible to search through all possible sentences, but it is possible to inspect a highly relevant subset of such sentences. The ISI ReWrite Decoder do that, and produces the single sentence from the subset that it inspects which best maximizes $P(p)P(e \mid p)$.

Translation can be performed as follows:

To translate sentence from English to Punjabi, decoder is started in server mode and Language model file and Translation model files are loaded to the decoder. Then sentence in English is given to the decoder to translate it into Punjabi.

Steps to translate are as follows:

- Step 1

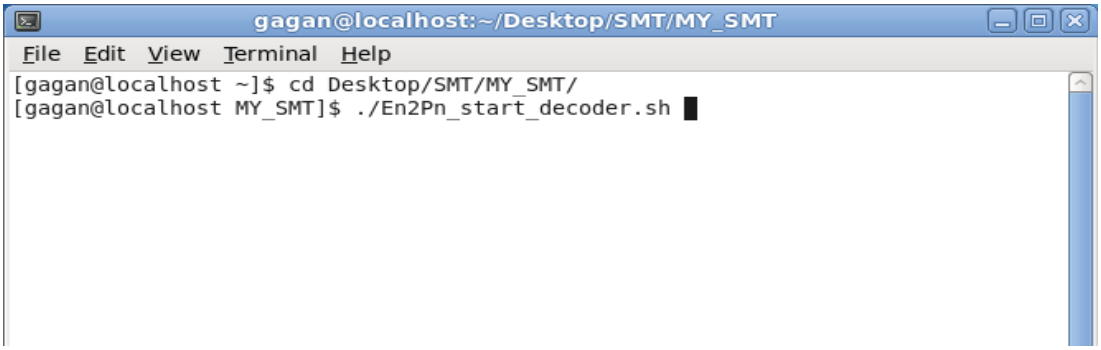


Figure 4.11: Starting the Decoder

Start decoder by executing *En2Pn_start_decoder.sh* shell script. This will start the decoder in server mode, and load Language model and Translation Model files.

- Step 2

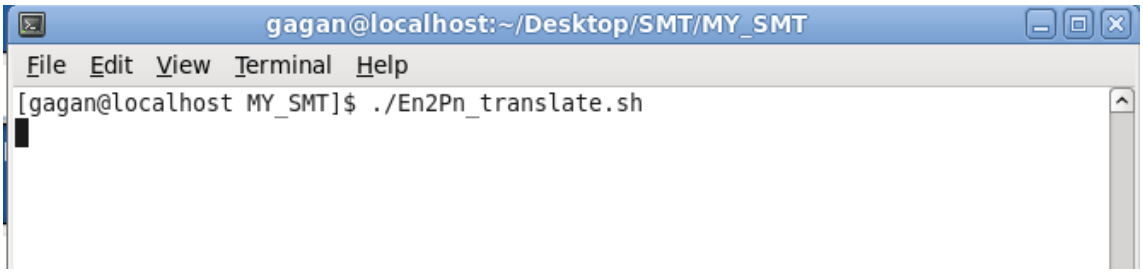


Figure 4.12: Starting Translation

Execute *En2Pn_translate.sh* shell script to perform translation. After execution of shell script, enter the sentence to translate.

- Step 3

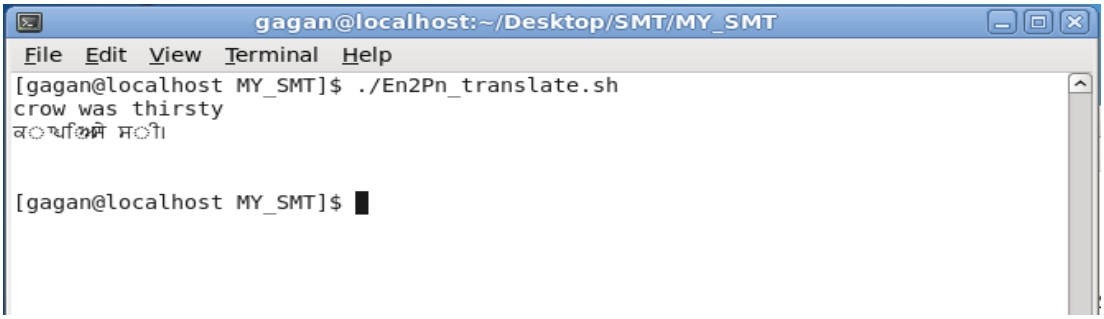


Figure 4.13: Performing Translation

After executing step 3, to retrieve output press “Ctrl+d”. It will show translated sentence in Punjabi in the terminal window.

- Step 4

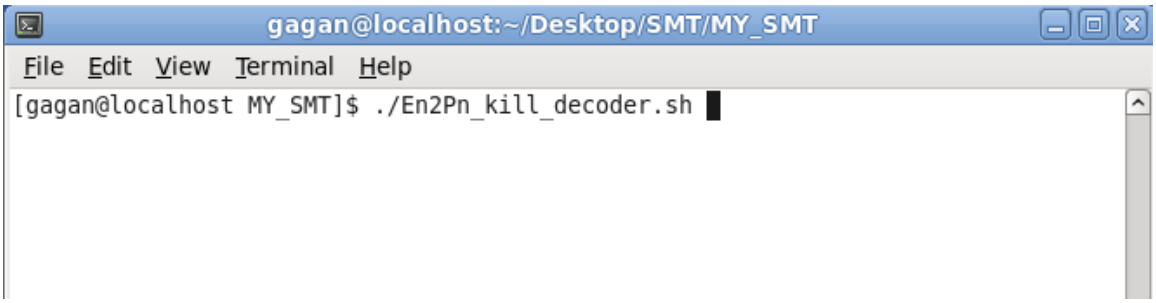


Figure 4.14: Stopping the Decoder

Finally to stop decoder execute *En2Pn_kill_decoder.sh.* This will stop the decoder.

Chapter 5

Testing of English to Punjabi Statistical Based Machine Translation

In this chapter we will look into the results of the translation system developed. We have used aligned text of about 6000 sentences of parallel corpus of English-Punjabi language pair.

For the given corpus, following results were obtained

Table 5.1: Sentence and Word Count of English and Punjabi corpus

	English	Punjabi
Total sentences	5714	5714
Total words	101193	116376
Unique words	11855	9746

As depicted in the Table 5.1, in the given corpus, there are 5714 total sentences in the English text file of the corpus, with total words count of 101193 and 11855 unique count *i.e.* there are total 5714 sentences in English file, which contains 101193 total words, out of which 11855 are unique words. Similarly in Punjabi text file there were 5714 total sentences with total word count of 116376, out of which 9746 are unique words.

Training of Translation Model took about 150 seconds.

Quality of translation depends upon the size and quality of corpus. Corpus used is not large enough to produce quality translation, but translations for simple sentences can be performed with the corpus used. We have tested the system for simple sentences, which resulted in translations of the English sentences into Punjabi. Accuracy and quality of translation can be enhanced in future by increasing the size of corpus, and quality of translation in corpus.

Translation of some simple sentences is as shown in Table 5.2.

Table 5.2: Results of translation performed by system

Input English Sentence	Output generated by system
india is a country	ਇੰਡੀਆ ਇੱਕ ਦੇਸ਼ ।
india is a democratic country	ਇੰਡੀਆ ਇੱਕ ਲੋਕੰਤਰਿਕ ਦੇਸ਼ ।
today is a holiday	ਅੱਜ ਇੱਕ ਛੁੱਟੀ ।
today is holiday	ਅੱਜ ਛੁੱਟੀ ।
punjab is a part of india	ਪੰਜਾਬ ਇੱਕ ਹਿੱਸਾ ਇੰਡੀਆ ।
delhi is a capital of india	ਦਿੱਲੀ ਇੱਕ ਰਾਜਧਾਨੀ ਇੰਡੀਆ ।
i am a boy	ਮੈਂ ਇੱਕ ਮੁੰਡੇ ।
crow was thirsty	ਕਾਂ ਪਿਆਸੇ ਸੀ ।
crows were thirsty	crows ਪਿਆਸੇ ਸਨ ।
crow saw a pitcher of water	ਵੇਖਿਆ ਕਾਂ ਪਾਣੀ ਦੀ ਇੱਕ ਕੀਤਾ, ।
lions live in jungle	lions jungle ਵਿੱਚ ਰਹਿੰਦੇ ਹਨ ।
what is your name ?	ਕੀ ਤੁਹਾਡੇ ਨਾਮ ?
meet you tomorrow	ਪੂਰਾ ਮਿਲਦੇ ਹਨ ।
what is your dream ?	ਕੀ ਤੁਹਾਡੇ ਸੁਫ਼ਨਾ ?
what are your dreams ?	ਕੀ ਤੁਹਾਡੇ dreams ?
what is your age ?	ਕੀ ਤੁਹਾਡੇ ਉਮਰ ?
i am a student	ਮੈਂ ਹਾਂ, ਇੱਕ ਵਿਦਿਆਰਥੀ ।
what is your age?	ਕੀ ਤੁਹਾਡੇ ਕਿੰਨੀ ।
what is your age ?	ਕੀ ਤੁਹਾਡੇ ਉਮਰ ?
lion is an animal	lion ਇੱਕ ਜਾਨਵਰ ਹੈ ।
this is a small cottage	ਇਹ ਇੱਕ ਛੋਟੇ ਝੌਂਪੜੀ ।
bill is passed in parliament	ਬਿਲ parliament ਵਿੱਚ ਪਾਰਿਤ ।
bill is signed by president	ਪ੍ਰਧਾਨ ਬਿਲ ਦੁਆਰਾ ਹਸਤਾਖਰ ।

Chapter 6

Conclusion and Future Scope

Issue of translating text from one language to other with the help of computers is not easy to solve. But with effort we can overcome the problem of using computers to translate text automatically. Statistical Machine Translation approach, is a machine learning technique and can be enhanced by improving the size and quality of the bilingual aligned corpus for the pair of languages.

6.1 Conclusion

From our experience, in development of English to Punjabi Statistical Based Machine Translation system, we conclude that, although it requires less time to develop translation system using statistical methods as compared to other approaches of Machine Translation. It is complex to understand the approach, but yields result in short duration of time. It takes less time to develop translation system using Statistical methods, but it is not easy to achieve accuracy. Quality of translation depends on the size of corpus and quality of corpus. So, quality of translations will be good if size of corpus is large enough and quality of corpus is considerably good.

Traditional MT techniques require large amounts of linguistic knowledge to be encoded as rules. Statistical MT provides a way of automatically finding correlations between the features of two languages from a parallel corpus, overcoming to some extent the knowledge bottleneck in MT.

A major drawback of the Statistical Machine Translation is that it pre-supposes the existence of a sentence-aligned parallel corpus. For the translation model to work well, the corpus has to be large enough, so that, the model can derive reliable probabilities from it.

Statistical MT techniques have not so far been widely explored for Indian languages.

6.2 Future Scope

In future, the Quality of translation can be improved by increasing the size and quality of bilingual corpus of the English-Punjabi pair of languages. System can be enhanced to translate document, text file, and web pages from English to Punjabi. System can be used translate text messages received from mobile phone; send back the translated text message to the mobile phone. More Indian languages can be added to the system by developing the parallel corpus for the pair of languages.

References

[1] ADAM LOPEZ, "Statistical Machine Translation", ACM Computing Surveys, Vol. 40, No. 3, Article 8, Aug 2008.

[2] Daniel Jurafsky and James H. Martin,"Speech and Language Processing", Pearson Education Inc, 2000.

[3] Definition of SMT at *http://en.wikipedia.org/wiki/Statistical_machine_translation* accessed on 06/02/2010.

[4] Durgesh D Rao, "Machine Translation A Gentle Introduction", RESONANCE, July 1998.

[5] Durgesh Rao; "Machine Translation in India: A Brief Survey".

[6] Franz Josef Och., "GIZA++: Training of statistical translation models" available at: *http://fjoch.com/GIZA++.html* accessed on 26/03/2010

[7] Gurpreet Singh Josan, Gurpreet Singh Lehal, "A Punjabi To Hindi Machine Translation System", Coling 2008: Companion volume – Posters and Demonstrations, Manchester, August 2008.

[8] Gurpreet Singh Lehal, "A Survey of the State of the Art in Punjabi Language Processing", Language in India, oct 2009.

[9] Hindi to Punjabi Translation system available at *http://h2p.learnpunjabi.org* accessed on 03/04/2010

[10] ISI ReWrite Decoder User's Manual, Version 0.2, available at *http://www.isi.edu/~germann/software/ReWrite-Decoder/isi-decoder-manual.html* accessed on 12/03.2010

[11] Jamie G. Carbonell, Teruko Mitamurs, Eric H. Nyberg, "The KANT Perspective: A Critique of Pur Transfer (and Pure Interlingua, Pure Statistic,….)"

[12] Jayprasad J Hegde, Ananthakrishnan R, Kavitha M, Chandra Shekhar, Ritesh Shah, Sawani Bade, Sasikumar M, "MaTra: A Practical Approach to Fully-Automatic Indicative English-Hindi Machine Translation".

[13] Jean Senellart, Péter Dienes, Tamás Váradi, "New Generation Systran Translation System", MT Summit VIII, Sept 2001.

[14] Liddy, E. D., "Encyclopedia of Library and Information Science", 2nd Ed. Marcel Decker, Inc.

[15] On line Translation System available at: *www.translate.google.com* accessed on 03/04/2010.

[16] Online manual of CMU Statistical Language Modeling Toolkit available at: *http://mi.eng.cam.ac.uk/~prc14/toolkit_documentation.html* accessed on 15/03/2010.

[17] P. Brown, S. Della Pietra, V. Della Pietra, and R. Mercer "The mathematics of statistical machine translation: parameter estimation". Computational Linguistics, 19(2), 263-311. (1993).

[18] Parteek Bhatia, Sandeep Singh, "Punjabi Deconverter Architecture", National Seminar on Creation of Lexical Resources for Indian Language Computing and Processing, CDAC Mumbai, March 26-28, 2007.

[19] Peter F. Brown, John Cocke, Stephen A. Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin, "A Statistical Approach to Machine Translation", ComputationalLinguistics, 16(2), pages 79–85, June 1990.

[20] Philipp Koehn, "MOSES, Statistical Machine Translation System User Manual and Code Guide", Cambridge University Press, 2009.

[21] Philipp Koehn, Pharaoh, a Beam Search Decoder for Phrase-Based Statistical Machine Translation Models" USC Information Sciences Institute, Aug 2004.

[22] Sinha, R. M. K., Jain, and A. Jain, "An English to Hindi machine-aided translation system based on ANGLABHARTI technology"; 2002.

[23] Tanveer Siddiqui, U. S. Tiwary, "Natural Language Processing and Information Retrieval", Oxford University Press, 2008

[24] Veritas L S, "Statistical Machine Translation and Example-based Machine Translation", 06/30/2009, available at: *http://www.proz.com/translation-articles/articles/2483/* accessed on 23/05/2010.

[25] W. John. Hutchins; "MACHINE TRANSLATION: A BRIEF HISTORY", The encyclopedia of languages and linguistics, Oxford: Pergamon Press, vol. 5, 1994

[26] W. Weaver (1955). Translation (1949). “ Machine Translation of Languages”, MIT Press, Cambridge, MA, Feb 2009.

[27] Warren Weaver, Translation, “Machine Translation of Languages: Fourteen Essays”, William Locke and Donald Booth (eds), pages 15–23, 1955.

Research Publications

Research Paper Published

- Gagandeep Singh, Parteek Bhatia, "Statistical Machine Translation for English to Punjabi Translation", Published at National Conference on NGC-2010 GITM, Gurgaon. (Mar 20, 2010)

Research Paper Communicated

- Gagandeep Singh, Parteek Bhatia, "English to Punjabi Statistical Machine Translation System", communicated in International Journal of Translation (IJT)

APPENDIX A

Punjabi Alphabet	English mapping
ਸ	S
ਹ	h
ਕ	k
ਖ	kh
ਗ	g
ਘ	gh
ਙ	Mg
ਚ	ch
ਛ	chh
ਜ	j
ਝ	jh
ਞ	Mj
ਟ	T
ਠ	Th
ਡ	D
ਢ	Dh
ਣ	n
ਤ	t
ਥ	th
ਦ	d
ਧ	dh

ਨ	n
ਪ	p
ਫ	ph
ਬ	b
ਭ	bh
ਮ	M
ਯ	Y
ਰ	R
ਲ	L
ਵ	w
ੜ	R
ਸ਼	Sh
ਖ਼	Kh
ਗ਼	G
ਜ਼	Z
ਫ਼	F
ਸ	S
ਹ	H
ਉ	Au
ਊ	auU
ਈ	aYI
ਇ	aY
ਆ	A
ਏ	E
ਐ	Ay

ਅੌ	a/U
ਓ	aU
ਅ	A
ਾ	A
ਿ	Y
ੀ	YI
ੇ	E
ੈ	I
ੋ	U
ੌ	/U
ੁ	U
ੂ	uU
ੰ	W