Development of a Hindi to Punjabi Machine Translation System
A Doctoral Dissertation

Vishal Goyal, Ph.D.
DEVELOPMENT OF A HINDI TO PUNJABI
MACHINE TRANSLATION SYSTEM

A
THESIS
Presented to the Faculty of Physical Sciences of the
Punjabi University
in Fulfilment of the Requirements for the
Degree of
DOCTOR OF PHILOSOPHY
IN
COMPUTER SCIENCE

Vishal Goyal
Department of Computer Science
Punjabi University, Patiala
February, 2010
CERTIFICATE

This is to certify that this thesis “Development of a Hindi to Punjabi Machine Translation System” embodies the work carried out by Vishal Goyal himself under my supervision and that it is worthy of consideration for the award of the Ph.D. degree.

(Dr. Gurpreet Singh Lehal)
Professor,
Department of Computer Science,
Punjabi University, Patiala.
(Supervisor)
DECLARATION

I hereby affirm that the work presented in this thesis is exclusively my own and there are no collaborators. It does not contain any work for which a degree/diploma has been awarded by any other University/Institution. A part of this thesis has already been published in international & national journals and the proceedings of the IEEE International Conference.

(Vishal Goyal)

Countersigned

(Dr. Gurpreet Singh Lehal)
Professor,
Department of Computer Science,
Punjabi University, Patiala.
Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Gurpreet Singh Lehal, Professor, Department of Computer Science, Punjabi University, Patiala. His advice, his continuous interest, and his support made the thesis ultimately possible. Despite his busy schedule as the Director of the Advanced Centre for Technical Development of Punjabi Language and Culture he has been always available for discussion.

I would like to extend my thanks to all those who have helped in finishing my work on time. I will not go into listing all the names here, but I remember even the slightest instance of support provided to me. I am grateful to all the people at ACTDL LAB and my Computer Science Department, for helping me.

Last, but not the least, I thank all the members of my family for extending me much needed support and inspiration to finish this arduous task.

Vishal Goyal
Patiala, February 2010
Contents

Acknowledgements
Abstract
List of Tables
List of Figures

1 Introduction
  1.1 Machine Translation
    1.1.1 Background
    1.1.2 Approaches used for Machine Translation
      1.1.2.1 Rule-Based Approach
        1.1.2.1.1 Direct MT System
        1.1.2.1.2 Indirect MT System
        1.1.2.1.3 Interlingua MT System
      1.1.2.2 Data-Driven Approach
        1.1.2.2.1 Example Based MT
        1.1.2.2.2 Knowledge Based MT
        1.1.2.2.3 Statistics Based MT
        1.1.2.2.4 Principle Based MT
      1.1.2.3 Hybrid Approaches
    1.1.3 Key Activities
  1.2 Research Questions
    1.2.1 Objectives
    1.2.2 Challenges
    1.2.3 Need and Scope
    1.2.4 Potential Use
  1.3 Approach Applied for Our Machine Translation
    1.3.1 Pre Processing
      1.3.1.1 Text Normalization
1.3.1.2. Replacing Collocations

1.3.1.3. Replacing Proper Nouns

1.3.2 Tokenizer

1.3.3 Translation Engine
   1.3.3.1 Analyzing the words for Translation/Transliteration
      1.3.3.1.1 Identifying Titles
      1.3.3.1.2 Identifying Surnames
      1.3.3.1.3 Lexicon Lookup
      1.3.3.1.4 Resolving Ambiguity
      1.3.3.1.5 Unknown Words

1.3.4 Post Processing

1.4 Thesis Outline

1.5 Summary

2 Survey Of Literature
   2.1 Machine Translation Systems
      2.1.1 Machine Translation System for Non Indian Languages
      2.1.2 Machine Translation System for Indian Languages
   2.2 Summary

3 Comparative Study of Hindi and Punjabi
   3.1 Introduction
   3.2 Comparison of Hindi and Punjabi Language on the basis of orthography
      3.2.1 Family and Status
      3.2.2 Script
         3.2.2.1 Devanagari Script
         3.2.2.1 Gurmukhi Script
      3.2.3 Consonants
         3.2.3.1 Basic Consonants
         3.2.3.2 Dead and Live Consonants
         3.2.3.3 Consonant Conjuncts
         3.2.3.4 Geminate (Doubled) Consonants
3.2.4 Vowels
   3.2.4.1 Full Form
   3.2.4.2 Short Form (or matra)
   3.2.4.3 Inherent ‘a’
   3.2.4.4 Nasalized Vowels
3.2.5 Punctuation marks
3.2.6 Abbreviation
3.2.7 Numerals
3.2.8 Alphabetic Order

3.3 Comparison of Hindi and Punjabi on the basis of Grammar
   3.3.1 Nouns
   3.3.2 Adjectives
   3.3.3 Postpositions
   3.3.4 Pronouns
   3.3.5 Verbs
   3.3.6 Sentence Structure
   3.3.7 Vocabulary

3.4 Comparison of Hindi and Punjabi from Machine Translation point of view
   3.4.1 Language Structure (Syntactic or Analytic)
   3.4.2 Ambiguity
   3.4.3 Gender disagreement
   3.4.4 Problems in identifying Proper Nouns
   3.4.5 Problems related to Collocations
   3.4.6 Problems related to Foreign Words
   3.4.7 Spelling Variations

3.5 Conclusions

4 Pre-processing
   4.1 Introduction
   4.2 Text Normalization
   4.3 Replacing Collocations
   4.4 Replacing Proper Nouns
5 **Tokenizer and Translation Engine**

5.1 Tokenizer

5.2 Translation Engine

5.2.1 Identifying Titles

5.2.2 Identifying Surnames

5.2.3 Word-to-Word translation using Lexicon Lookup

5.2.4 Resolving Ambiguity

5.2.5 Handling Unknown Words

5.2.5.1 Word Inflectional Analysis and Generation

5.2.5.2 Transliteration

5.3 Summary

6 **Postpositions**

6.1 Grammar Corrections

6.2 Pattern Matching and Regular Expressions

6.2.1 Related Works

6.2.2 Our Approach

6.3 Sample Translations

6.4 Illustrative Example

6.5 Summary

7 **Evaluation and Results**

7.1 Introduction

7.2 Related Work

7.3 Our Approach

7.3.1 Selection Set of Sentences

7.3.2 Selection of Tests for Evaluation

7.3.2.1 Subjective Tests

7.3.2.1.1 Intelligibility Test

7.3.2.1.2 Accuracy Test/Fidelity Measure

7.3.2.1.3 BLEU Score

7.3.3 Evaluation based on Quantitative Metrics
7.3.4 Experiments

7.3.4.1 Intelligibility Evaluation
   7.3.4.1.1 Scoring
   7.3.4.1.2 Results
   7.3.4.1.3 Percentage Intelligibility
   7.3.4.1.4 Analysis

7.3.4.2 Accuracy Evaluation / Fidelity Measure
   7.3.4.2.1 Scoring
   7.3.4.2.2 Results
   7.3.4.2.3 Percentage Accuracy
   7.3.4.2.4 Analysis

7.3.5 Error Analysis
   7.3.5.1 Word Error Analysis
   7.3.5.2 Sentence Error Analysis
   7.3.5.3 Error Analysis Conclusion

7.4 Comparison with other existing systems
7.5 Conclusion

8 Summary
8.1 Contributions
8.2 Limitations
8.3 Future Directions

References

Publications Based on the Work Presented in this Thesis

Appendix A: Graphic User Interface and Extended Features
Appendix B: Test Data Set for Intelligibility Test
Appendix C: Test Data Set for Accuracy Test
Abstract

Machine Translation is a task of automatic translation a text from one natural language to another. Even after more than 60 years of research, Machine Translation is still an open problem. Work for the development of Machine Translation systems for Indian languages is still in infancy. This research work is an attempt to develop a Machine Translation system from Hindi to Punjabi language. A number of Machine Translation systems have already been developed though their accuracy needs to be improved. Machine Translation is not a trivial task by nature of translation process itself. But Machine Translation of closely related languages eases the task. We call a language pair to be closely related if the languages have the grammar that is close in structure, contain similar constructs having almost same semantics, and share a great deal of lexicon. By closely related languages, we also mean inductively and morphosyntactically similar languages. Some linguist define closeness between the languages on the basis of features viz. common root, similar alphabets, similar verb patterns, structural similarity, similar grammar, similar religio-cultural and demograpohic contexts and references, a similar clearly displayed ability to blend with foreign tongues. Generally, such languages have originated from the same source and spoken in the areas in close proximity. Hindi and Punjabi belong to same sub group of the Indo European family, thus are sibling languages. It has been analysed that Hindi and Punjabi languages share all features of closely related languages. For such closely related sibling languages, effective word for word translation can
be achieved (Hajic et al., 2000) [90]. Thus for our system, Direct Machine Translation approach which seems promising approach has been used.

The challenges in developing Hindi to Punjabi Machine Translation system lie with major problems mainly related to the non-availability of lexical resources, spelling variations, word sense disambiguation, transliteration, named entity recognition and collocations.

This research work addresses the problems in the various stages of the development of a complete Hindi to Punjabi Machine Translation system and discusses potential solutions. The thesis has been divided into eight chapters.

The first chapter of the thesis introduces general concept of Machine Translation, various approaches to Machine Translation systems and key activities involved in Machine Translation. It also provides a formal description about the research question undertaken for this study. The objectives, need, and scope of the study have also been discussed. Then some of the key application areas of Machine Translation system are explored. Afterwards, the approach followed along with the reasons behind its selection to solve this research problem has been explained in brief. An overview of the design of the Machine Translation system undertaken to develop in this research work is provided later. The chapter concludes by presenting major contributions of this research work and an outline of the study.

Chapter 2 discusses the existing work in the field of Machine Translation in India and outside India. This chapter on literature survey forms the basis of our work on developing the Machine Translation system and later on helps us
in comparing our work with the existing state of the art in Machine Translation system.

Chapter 3 explains and compares Hindi and Punjabi languages with respect to orthography, grammar, and Machine Translation.

Chapter 4 and 5 provide the design and implementation details of various activities involved in the Machine Translation system. Chapter 4 describes the system architecture and preprocessing stage. The chapter starts with the choice of approach and discusses the motivation behind its selection. Then the required resources are discussed followed by description of system architecture. The details of preprocessing phase which involves text normalization, Identifying Collocations, Identifying Proper Nouns are discussed. Then tokenization process is explained. The details of the translation system involving the identifying titles, identifying surnames, lexicon lookup, word sense disambiguation module, transliteration module and post processing modules are discussed in Chapter 5.

Chapter 6 describes the post processing stage of the system. Chapter 7 provides the evaluation of the system and its results. Chapter 8 concludes this thesis by providing a summary of the research work undertaken, contributions of this research work, limitations, and some directions in which this work could be extended in the future. In appendix A, the interface designed for text translation, website translation and email translation has been discussed. Test data set for intelligibility test and accuracy test is available in Appendix B and C respectively. The system has been rigorously evaluated and its accuracy
has been found to be 94% on the basis of intelligibility test and 90.84% on the basis of accuracy test.
List of Tables

3.1 Basic Consonants in Devanagari
3.2 Basic Consonants in Gurmukhi
3.3 Conjunct Consonants
3.4 Vowels in Devanagari and Gurmukhi
3.5 Numerals in Devanagari and Gurmukhi
3.6 Declinable and Indeclinable Hindi Adjectives
3.7 Hindi and Punjabi Pronouns
4.1 Frequency of Occurrence for Possible Spelling Variants of Word अंग्रेजी
4.2 Text Normalization Rules
4.3 Analysis of % word occurrence with spelling variation count
4.4 Text Normalization Database Design
4.5 Sample Entries of Text Normalization database
4.6 Collocation Database Design
4.7 Sample Entries of Collocation Database
4.8 properNoun Database Design
4.9 Sample Entries for properNoun Database
5.1 Titles Database Design
5.2 Sample Entries of Titles Database
5.3 Surnames Database Design
5.4 Sample Entries for Surname Database Design
5.5 HPDictionary Database Design
5.6 Sample Entries for HPDictionary Database Design
5.7 Lexical Category Percentage Distribution of Ambiguous Words
5.8 Example Demonstrating the Ambiguity Resolution
5.9 trigramsMiddle Database Design
5.10 trigramsLeft Database Design
5.11 trigramsRight Database Design
5.12 bigramsLeft Database Design
5.13 bigramsRight Database Design
5.14 Sample Entries for trigramsMiddle Database
5.15 Sample Entries for trigramsLeft Database
5.16 Sample Entries for trigramsRight Database
5.17 Sample Entries for bigramsLeft Database
5.18 Sample Entries for bigramsRight Database
5.19 Contribution of various N-grams in resolving ambiguities
5.20 Inflections in Hindi
5.21 List of correct accepted words in translation after inflectional analysis and generation
5.22 Failure cases during inflectional analysis and generation
5.23 Inflection Rules
5.24 PunjabiUnigram Database Design
5.25 Sample Entries for punjabiUnigrams Database
5.26 Direct Hindi to Punjabi Character mapping
6.1 Grammatical Error Category Wise Regular Expression Distribution
6.2 Percentage Contribution of Regular Expressions on the basis of Grammatical Error Categories

6.3 Percentage Contribution of various MT System modules during translation

7.1 Test data set for the evaluation of Hindi to Punjabi Machine Translation System

7.2 Score Sheet for Intelligibility Test

7.3 Score Sheet for Accuracy Test

7.4 Comparison of our System with other existing systems
List of Figures

1.1 Example Based MT
1.2 MT System General Architecture
1.3 Overview of Hindi to Punjabi Machine Translation System
4.1 Pre-Processing System Design
4.2 Analysis of Percentage Usage of Various Text Normalization Rules
4.3 Analysis of Contribution of Text Normalization Rules
5.1 Flow Chart for Word Inflectional Analysis and Generation
5.2 Flow Chart for Transliteration module
7.1 Percentage Intelligibility for Different Documents
7.2 Percentage Accuracy for Different Documents
7.3 Percentage Distribution of Errors
7.4 Word Error Rate for Different Documents
7.5 Sentence Error Rate for Different Documents
A.1 GUI for Hindi to Punjabi Machine Translation System
A.2 Screenshot for translation facility of the system
A.3 Screenshot for transliteration facility of the system
A.4 Screenshot for website translation facility feature of the system
A.5 Screen shot of Original Website http://www.webdunia.com/ accessed on 27/12/2009 at 08:40 PM IST
A.6 Screen shot of translated version by the system for website shown in Figure A.4
A.7. Screenshot for Email Sending facility of the system
Chapter 1

Introduction

The goal of automatic translation (also called Machine Translation, or MT) is to translate text from one human language into other using computers. MT was one of the first envisioned applications of computers back in the 1950’s. Even after more than 60 years of research, MT is still an open problem. Nowadays, the demand for MT is steadily growing. All over the world, documents have to be translated into all official languages. This multilingualism is considered a part of democracy. Also in the private sector, there is a large demand of MT: technical manuals have to be translated into several languages. An even large demand exists in the World Wide Web. Thus, MT can help to reduce the language barrier and enable easier communication. This research work is an attempt to develop a Machine Translation system from Hindi to Punjabi Language. A number of Machine Translation systems have already been developed though their accuracy needs to be improved. However, there is no Hindi to Punjabi Machine Translation system available at the present.

This chapter introduces general concept of Machine Translation, various approaches for Machine Translation systems and key activities involved in Machine Translation. It also provides a formal description about the research question undertaken for this study. The objectives, need, and scope of the
study have also been discussed. Then some of the key application areas of Machine Translation system are explored. Afterwards, the approach followed along with the reasons behind its selection to solve this research problem has been explained in brief. An overview of the design of the Machine Translation system undertaken to develop in this research work is provided later. The chapter concludes by presenting major contributions of this research work and an outline of the study. This work is based on the Devanagari and Gurmukhi scripts. Thus, the examples given in this thesis work are in Devanagari and Gurmukhi scripts along with their transliteration. For inline examples, transliteration will be provided in italics e.g. गहना (gahnā). The transliteration provided is based on transliteration software – the GTrans, which was developed in the Department of Computer Science, Punjabi University, Patiala, Punjab, India.

1.1 Machine Translation

You feed a story written in Hindi into a computer system and out comes its translation in Punjabi, Oriya, English, Tamil and other languages. It is inexpensive, immediate and simultaneous. The language barriers melt away. The richness of other literatures opens up to everyone. The world is intellectually and culturally united into one. This is the dream of people working in a fascinating area of research called Machine Translation. Thus Machine Translation system is software designed that essentially takes a text
in one language (called the source language), and translates it into another language (called the target language). The source and target languages are natural languages such as English and Hindi, as opposed to man-made languages such as C or SQL. Translation, in its full generality, is a difficult, fascinating, and intensively human endeavor, as rich as any other area of human creativity. Machine Translation is an important sub-discipline of the wider field of artificial intelligence (AI). AI (among other things) deals with getting machines to exhibit intelligent behaviour.

Though Machine Translation has been an interesting area of research since the invention of computers, in India it is relatively young. As a discipline it dates back to the early 1950. The earliest efforts in research date back to late 80s and early 90s. The complexity of the problem was originally underestimated, and some early successful demonstrations of experimental systems led to unrealistic expectations which were hard to fulfill. This led to some skepticism, and funding on MT work almost ceased. In the early eighties, the Japanese Fifth Generation Computing Project revived interest in this work and some of the prominent works in India are the projects at IIT Kanpur, University of Hyderabad, NCST Mumbai and CDAC Pune. The Technology Development in Indian Languages (TDIL), an initiative of the Department of IT, Ministry of Communications and Information Technology, Government of India, has played an instrumental role in funding these projects. Since the mid and late 90’s, a few more projects have been initiated—at IIT Bombay, IIIT Hyderabad, AU-KBC Centre Chennai, Jadavpur.
University Kolkata and Punjabi University Patiala. There are also a couple of efforts from the private sector - from Super Infosoft Pvt Ltd, and more recently, the IBM India Research Lab [1,2].

Machine Translation between closely related languages is easier than between language pairs that are not related with each other. Having many parts of their grammars and vocabularies in common reduces the amount of effort needed to develop a translation system between related languages. In this thesis, we will be discussing the Machine Translation system between closely related languages- Hindi and Punjabi.

### 1.1.1 Background:

Warren Weaver, a director of the Rockefeller Foundation, received much credit for bringing the concept of MT to the public when he published an influential paper on using computer for translation in 1949. The early 1950s were a period of intense research in MT in both the United States and Europe. 1952 saw the first conference on MT, but it was not until 1954 that a translation system was demonstrated in New York. The reaction of public to this MT system was negative because many people thought that perfect MT was close at hand and human translators would be out of their jobs. In 1959, IBM installed an MT system for the United States Air Force, followed by Georgetown University installing systems at Erratum and the United States Atomic Energy Agency. Despite some success of early MT systems, MT research funding was on the verge of serious reduction.

Language in India [www.languageinindia.com](http://www.languageinindia.com) 621
10 : 10 October 2010
Vishal Goyal, Ph.D.

*Development of a Hindi to Punjabi Machine Translation System* - A Doctoral Dissertation
The growing dissatisfaction of research sponsors caused the United States National Academy of Sciences to set up the Automatic Language Processing Advisory Committee (ALPAC) in 1966. ALPAC, whose members were the major sponsors of current MT research projects, was to evaluate the effectiveness, costs, and potential future progress of MT. Their findings, known as the ALPAC Report, concluded that MT was not useful and sufficient goal. The research was rather unsatisfactory to justify further funding from the United States government. The effects of the report rippled to cause most private sponsors of MT projects in the United States to withdraw from future funding. ALPAC also suggested the complete discontinuation of MT research in the United States and the computer aids for translators should be developed instead. So, for several years, MT research was virtually at standstill. 1976 marked a positive turning point for MT research when the country of Canada made public their Mateo System, which translated weather forecasts. Later that year, the European Commission purchased SYSTRAN, a Russian-English system. MT interest and activity has increased ever since, and MT has been established as a legitimate field of research. In the 1980s, MT software for personal computers appeared; the 1990s showed MT implemented as an online service. The 2000s have shown even more research into MT and many new, efficient hybrid algorithms.

The advent of low-cost and more powerful computers towards the end of the 20th century brought MT to the masses, as did the availability of sites on the Internet. Much of the effort previously spent on MT research, however, has
shifted to the development of Computer-Assisted Translation (CAT) systems, such as translation memories, which are seen to be more successful and profitable [3-18,21,22,23].

1.1.2 Approaches used for Machine Translation [19-22]:

There are a number of approaches used for MT. But mainly three approaches are used. These are discussed below:

1. Rule-Based Approaches
2. Data-Driven Approaches
3. Hybrid Approaches

1.1.2.1. Rule-Based Approaches:

The current rule-based architecture of MT can be categorized into three areas:

1. Direct MT
2. Indirect MT
3. Interlingua MT

The Machine Translation has two generations to be considered during its development. The first generation Machine Translations are those which were done in 1960s and are called Direct Machine Translation. They used the direct approach of translation which was based on word-to-word and/or
phrase to phrase translations. Simple word-to-word translation cannot resolve the ambiguities arising in MT. A more thorough analysis of source language text is required to produce better translation. As the major problem of the first generation MT was the lack of linguistic information about source text, researchers therefore moved on to finding ways to capture this information. This gave rise to the development of the indirect MT systems which are generally regarded as second generation MT systems.

1.1.2.1.1 Direct MT System:

The direct method, also known as the “Transformer” method was the strategy adopted by the earliest MT systems. It is the most primitive method and uses a one stage process in which the systems simply translate the source language texts into the corresponding word-to-word or phrase-to-phrase by using the bilingual lexicon. For example - direct translation from Hindi to Punjabi for याभ ने श्माभ को प्माय से गरे लगामा (राम ने श्याम को प्यार से गले लगाया) is तम ने शम नूम पीयर नाल गले लगाया (राम ने श्याम नूम पीयर नाल गले लगाया). The basic characteristic for such type of translation is that it is very simple and one needs to replace a word of source language to a word in target language using a bilingual dictionary.

They also perform some morphological analysis before looking into the bilingual lexicon for the root words. They will then perform the necessary
reordering of the words as according to the target language sentence format. The morphological processes may improve the quality of the translation but they don’t really analyze the structure of the source language. An example of the direct MT system is SYSTRAN. The Direct Machine Translation was the technique developed in 1950s where the computers were in an early stage of technical development with very less speed which resulted in long processing time. Due to these constraints the direct MT used a very straightforward and easy to implement approach. It supports the translation of source language sentences to the sentences of the target language having structures similar to the structure in the source language. As very little effort is made to disambiguate the source language, this technique can’t translate highly ambiguous sentences. The main problem of the direct MT approach is that it doesn’t analyze the linguistic information or the meaning of source sentences before performing the translation. Without this information the resulting MT system can’t always resolve the ambiguities that arise in the source sentence and/or during the translation. As a result, the direct MT systems can’t provide a quality translation of the source language text.

The disadvantage of direct method is that it is unidirectional i.e. if the target is to be translated back into the source language, a different transformer must be used. It uses $n^2$ translation modules for translations among $n$ languages, thus making it exponentially large for multi-language translating. Other problems with the direct method involves are in relation to the structure of
sentences if these are complex. It requires complex grammatical analyses, in the absence of which word ordering in the target language sentence can often be wrong. So direct translation is very inaccurate for complex text, but has been implemented successfully with a limited number of lexical entries. It is to be noted that this direct approach is most suitable choice for language pair that are closely related to each other.

1.1.2.1.2 Indirect MT System:

Owing to the fact that linguistic information helps an MT system to disambiguate source language sentences and to produce better quality target language translation, with the advance of computing technology, MT researchers started to develop methods to capture and use the linguistic information in the translation process. The indirect method occupies the level above direct translation in the MT pyramid and is also known as transfer or linguistic knowledge (LK) translation.

The transfer architecture not only translates at the lexical level, like the direct architecture, it also translates syntactically and sometimes semantically. The transfer method will first parse the sentence of the source language then will apply rules that map the grammatical segments of the source sentence to a representation in the target language. For example:
“Children like to play cricket” will be translated in Hindi as बच्चे क्रिकेट खेलना पसंद करते हैं (bccē krikēṭ khēlnā pasand kartē haiṃ)

In this example Verb Phrase ‘like’ is translated into पसंद करते (pasand kartē), Subject ‘Children’ is translated to बच्चे (bccē).

After syntactically and semantically analyzing the sentence, we can easily translate a sentence even with different structures. In this approach word reordering is also done. Suppose in English the word order in sentence is SVO when translated into Hindi, the word order of the translated sentence will be SOV.

The transfer approach uses $n^2$ transfer modules, $n$ analysis components, and $n$ synthesis components, where $n$ is the number of languages in the translation system. Thus, one of its downfalls is the sheer size of the rules needed for its implementation.

**1.1.2.1.3 Interlingua MT System:**

The Interlingua or pivot approach appears at the apex of the MT pyramid. The main idea behind it is that the analysis of any language should result in a language-independent representation. The target language is then generated from that language-neutral representation. In a pure interlingua system there
are no transfer rules as a representation should be common to all languages used by the system.

This approach requires only one interlingual transfer model whereas the transfer approach requires $n^2$ transfer modules. The interlingual approach requires more analysis and is more abstract. It requires $n$ analysis components, $n$ Interlingua converters, $n$ generation components where $n$ is the number of languages in translation system.

There are few problems with the Interlingua approach. It requires an analyzer for each source language and a generator for each target language. Analysis of source text requires a deep semantic analysis that requires extensive word knowledge. Unfortunately, the true meaning of the sentence cannot always be extracted. Additionally, if a text is analyzed as deeply as is expected, then much of the source author's style will be lost.

1.1.2.2 Data-Driven Approach:

There are four approaches using data driven method:

1. Example Based MT
2. Knowledge Based MT
3. Statistics Based MT
4. Principle Based MT

1.1.2.2.1 Example Based MT:
Example-based translation is essentially translation by analogy. An Example-Based Machine Translation (EBMT) system is given a set of sentences in the source language (from which one is translating) and their corresponding translations in the target language, and uses those examples to translate other, similar source-language sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct again. EBMT systems are attractive in that they require a minimum of prior knowledge and are therefore quickly adaptable to many language pairs.

A restricted form of example-based translation is available commercially, known as a translation memory. In a translation memory, as the user translates text, the translations are added to a database, and when the same sentence occurs again, the previous translation is inserted into the translated document. This saves the user the effort of re-translation that sentence, and is particularly effective when translating a new revision of a previously-translated document (especially if the revision is fairly minor).

More advanced translation memory systems will also return close but inexact matches on the assumption that editing the translation of the close match will take less time than generating a translation from scratch.

wEBMT, ALEPH , English to Turkish, English to Japanese, English to Sanskrit and PanEBMT are some of the example based MT systems.
Knowledge-Based MT:

Knowledge-Based MT (KBMT) is characterized by a heavy emphasis on functionally complete understanding of the source text prior to the translation to the target text. KBMT does not require total understanding, but assumes that an interpretation engine can achieve successful translation into several languages. KBMT is implemented on the Interlingua architecture; it differs from other interlingual techniques by the depth with which it analyzes the source language and its reliance on explicit knowledge of the world.

KBMT must be supported by world knowledge and by linguistic semantic knowledge about meanings of words and their combinations. Thus, a specific language is needed to represent the meaning of languages. Once the source language is analyzed, it will run through the augmenter. It is the Knowledge
base that converts the source representation into an appropriate target representation before synthesising into the target sentence.

KBMT systems provide high quality translations. However, they are quite expensive to produce due to the large amount of knowledge needed to accurately represent sentences in different languages.

English-Vietnamese Machine Translation system is one of the examples of KBMTS.

**1.1.2.2.3 Statistics Based MT:**

By the turn of the century, this newer approach based on statistical models – where in a word or phrase is translated to one of a number of possibilities based on the probability that it would occur in the current context - had achieved marked success. The best examples substantially outperform rule-based systems. Statistics-based Machine Translation (SMT) also may prove easier and less expensive to expand, if the system can be taught new knowledge domains or languages by giving it large samples of existing human-translated texts.

A string of source language, ε, can be translated into a string of target language in many different ways. Often, knowing the broader context in which ε occurs may serve to winnow the field of acceptable target language translations, but even so, many acceptable translations will remain; the choice among them is largely a matter of taste. In statistical translation, the view is
taken that every target language string, $\zeta$, is a possible translation of $\varepsilon$. Every pair of strings is assigned ($\varepsilon \sim \zeta$) a number $Pr(\zeta \mid \varepsilon)$, which then is interpreted as the probability that a translator, when presented with $\varepsilon$, will produce $\zeta$ as his translation. Further the view is taken that when a native speaker of target language produces a string of target language words, he has actually conceived of a string of source language words, which he translated mentally. Given a target language string $\zeta$, the job of the translation system is to find the string $\varepsilon$ that the native speaker had in mind when he produced $\zeta$. Thus the chances of error are minimized by choosing that source language string for which $Pr(\varepsilon \mid \zeta)$ is greatest:

$$\hat{\varepsilon} = \arg \max Pr(\varepsilon \mid \zeta) \quad \text{………………………..(1)}$$

The argmax operation denotes the search problem, i.e. the generation of the output sentence in the target language. The term $Pr(\varepsilon \mid \zeta)$ is termed as true probability distribution that target language sentence $\varepsilon$ is translation of source language sentence $\zeta$.

Bayes' theorem is used:

$$Pr(\varepsilon \mid \zeta) = \frac{Pr(\varepsilon) Pr(\zeta \mid \varepsilon)}{Pr(\zeta)} \quad \text{………………………..(2)}$$

Since the denominator here is independent of $\varepsilon$, finding $\hat{\varepsilon}$ is the same as finding $\varepsilon$ so as to make the product $Pr(\varepsilon) Pr(\zeta \mid \varepsilon)$ as large as possible. Thus, at the Fundamental Equation of Machine Translation is arrived at:

$$\hat{\varepsilon} = \arg \max_{\varepsilon} Pr(\varepsilon) Pr(\zeta \mid \varepsilon) \quad \text{………………………..(3)}$$
Equation (3) summarizes the three computational challenges presented by the practice of statistical translation: estimating the language model probability, \( \Pr(\varepsilon) \); estimating the translation model probability, \( \Pr(\zeta \mid \varepsilon) \); and devising an effective and efficient suboptimal search for the input string that maximizes their product. These are known as the language modeling problem, the translation modeling problem, and the search problem.

Statistical translation systems work in two stages viz. training and translation. In training it “learns” how various languages work. Before translation, the system must be trained. Training is done by feeding the system with source language documents and their high-quality human translations in target language. With its resources, the system tries to guess at documents’ meanings. Then an application compares the guesses to the human translations and returns the results to improve the system’s performance. The whole process is carried out by dividing sentences into \( N \)-grams. While training, statistical systems track common \( N \)-grams, translations most frequently used are learnt and those meanings when finding the phrases in the future are applied. They also statistically analyze the position of \( N \)-grams in relation to one another within sentences, as well as words’ grammatical forms, to determine correct syntax. After their training, the systems are used to process actual phrases and produce the translation from what ever it has learnt in training phase.
Despite some success, however, severe problems still exist: outputs are often ungrammatical and the quality and accuracy of translation falls well below that of a human linguist - and well below demands of all but highly specialized commercial markets. Moses, CASIA, Chinese-to-English, Google translate, LDV-COMBO and MARIE are some of the examples for statistical approach based MT systems.

1.1.2.4 Principle-Based MT:

Principle-Based MT (PBMT) Systems employ parsing methods based on the Principles & Parameters Theory of Chomsky’s Generative Grammar. The parser generates a detailed syntactic structure that contains lexical, phrasal, grammatical, and thematic information. It also focuses on robustness, language-neutral representations, and deep linguistic analyses.

In the PBMT, the grammar is thought of as a set of language-independent, interactive well-formed principles and a set of language-dependent parameters. Thus, for a system that uses n languages, n parameter modules and one principles module are needed. Thus, it is well suited for use with the interlingual architecture.

PBMT parsing methods differ from the rule-based approaches. Although efficient in many circumstances, they have the drawback of language-dependence and increase exponentially in rules if one is using a multilingual translation system. It provides broad coverage of many linguistic phenomena,
but lacks the deep knowledge about the translation domain that KBMT and EBMT systems employ. Another drawback of current PBMT systems is the lack of the most efficient method for applying the different principles.

UNITRAN is one of the examples of Principle based Machine Translation system.

1.1.2.3 Hybrid Approaches:

Hybrid approaches to MT are becoming increasingly popular research subjects. The general idea behind hybrid approaches is to use a linguistic method to parse the source text, and a non-linguistic method, such as statistical-based or example-based, to assist with finding the proper interpretation.

1.1.2.3.1 Example-Based MT and Statistical-Based MT

EBMT works very well translating sentences that are already represented in its translation memory. SBMT can generate sentences with good accuracy, but is generally not successful when it handles idioms, collocations, and long-distance dependencies very well. By combining these two methods, a hybrid EBMT and SBMT system can first query the translation memory for matching sentences. If no close match is found, then a statistical analysis and interpretation of the sentence will be used.
TransEasy is one of the examples for Machine Translation based on Hybrid Approach. It uses Example and Statistical based approaches.

1.1.3 Key Activities [19-23]

Based on the above discussion of the Machine Translation techniques, it is evident that there are some common and key activities, which formulate a typical Machine Translation system.

An overview of such activities is provided below. These activities are usually executed in a sequence. However, depending upon the technique being followed, one or more of these activities may be omitted.

- **Pre-processing:** This module tokenizes the input text into words based on the list of word boundaries. Another major task performed in this phase is filtering. Filtering means detecting and marking certain special expressions like abbreviations, collocations, Named Entities, surnames, titles etc., in the input text. Text Spelling Standardization is another task in Pre-processing in which the words having spelling variations are replaced with the standard spelling words. This task helps in increasing the accuracy of the system. Filtering can be useful as the words or word sequences marked by the filter may not be required to go through the next two stages, namely, morphological analysis and part-of-speech tagging.
Fig 1.2 MT System General Architecture

- **Morphological analysis**: In this stage, morphological analyzer processes every unmarked token in the input text. The purpose of a morphological analyzer is to return root word and grammatical information about all the possible word classes (parts of speech) for a given word. Morphological analysis is essential for Hindi because it has a fairly rich system of inflectional morphology like other Indian languages. Morphological generator does exactly the reverse of
morphological analyzer. Given a root word and its grammatical information (including word class), a typical morphological generator will generate the word form or surface form for that root word.

- **Part-of-speech tagging:** The output of a morphological analyzer is usually ambiguous as it may return more than one POS (part-of-speech) tag for a single word. The reason being that in sentences, same word can be used as a noun or a verb, as a verb or a postposition etc. The job of a part-of-speech tagger is to disambiguate that ambiguous input by making use of the context information in which the word is being used. A part-of-speech tagger is also known as morphological disambiguator or simply a tagger.

- **Phrase chunking:** It is situated between POS tagging and a full-blown grammatical analysis, i.e. parsing. Whereas POS tagging works only on the word level and the grammatical analysis is supposed to build a tree structure of the sentence, phrase chunking assigns tags to word sequences in the sentence. There is no standardization about chunk names and their meanings, like POS tags, anyone can define his/her own chunk names and assign meanings to them. As chunking requires POS tagged text, its accuracy cannot be better than that of a POS tagger used. Chunking process is also known as shallow parsing as it simplifies the task for the next phase, i.e. parsing.

- **Parsing:** A parser is supposed to perform the full syntactic analysis of the given text. For every parsed sentence it is supposed to return a
data structure (mostly a parse tree) describing its syntactic components and their relationships with each other. It outputs the analysis based on the grammar it uses. For analyzing a sentence written in a particular language, the parser needs the grammar of that language. For specifying the grammar rules of a natural language, grammar formalism is required. Grammar formalism provides guidelines for specifying the underlying language’s grammar rules. The parser will then make use of that grammar formalism. There are various grammar formalisms available for use like CFG (Context Free Grammar), GPSG (Generalized Phrase Structure Grammar), and HPSG (Head Driven Phrase Structure Grammar) etc. In simple terms, grammar formalism consists of a lexicon of words associated with their grammatical category and a set of rules specifying the sentence structure or syntax of the language. If a syntax-based parser fails to parse a sentence completely, then that sentence could be marked as incorrect or ungrammatical.

- **Translation and Transliteration:** Having all the necessary information regarding the words in a sentence, the next step is to find the equivalent in the target language. An alternative term for the computation of target texts from intermediate representations is synthesis. This is done with the help of lexicon. In Direct MT technique, this stage involves just dictionary look up. Some local reordering of words is also seen as generation in such systems. Sometime
morphological synthesizers are required to generate the word in target language. In transfer system, the generation phase is generally split into two modules, ‘syntactic generation’ and ‘morphological generation’. In syntactic generation, the intermediate representation which is the output from analysis and transfer resembles a deep structure tree of the older type of transformational-generative grammar. It is converted by ‘transformational rules’ into an ordered surface structure tree, with appropriate labeling of the leaves with grammatical functions and features.

- **Rearrangement of word order:** If the source language and target language have different word order, then this step tries to reorder the words according to the grammar of target language. Any differences between languages can be dealt within the word generation and ordering stages. For example, the word order in English is Subject-Verb-Object. On the other hand, Hindi has relatively free word order. Generally a sentence in Hindi has the order Subject-Object-Verb. So to make the output according to the grammar of target language, some reordering techniques are required.

- **Post Processing:** The main factor which decides the amount of post-editing that needs to be done on a translation produced by machine is the quality of the output. Obviously enough, the difficulty of post-editing and the time required for it correlates with the quality of the raw MT output: the worse the output, the greater the post-edit effort. The post-
editor is a corrector for ill-formed sentences. It is basically tail-end of all the Machine Translation systems. It improves the translation quality by making corrections in the translation generated.

As mentioned earlier, not all of the above-mentioned activities are mandatory for a Machine Translation system. Selection of these activities depends on the approach a Machine Translation is following.

1.2 Research Questions

It has already been said that there is no Machine Translation system for Hindi to Punjabi presently. However, A number of Machine Translation systems between Indian and Non Indian languages have already been developed though their accuracy needs to be improved. Based on the brief introduction of Machine Translation given in section 1.1.1, the problem statement for the present research work has been formulated as below:

“To develop algorithms and lexical resources along with a software package to translate Hindi text to Punjabi text.”

In other terms, the research question is to develop an automated Hindi to Punjabi Machine Translation System that will translate the Hindi text to Punjabi text. In this way, the richness of Hindi literatures opens up to Punjabi knowing people. This system will be helpful in reading the online Hindi newspapers in Punjabi language, Thus, removing the language barrier among
people. The users can type their email in Hindi language and the receiver can receive the email in Punjabi Language, Thus, making the communication in user's native languages possible.

### 1.2.1 Objectives

The objectives of this study are as follows:

1. To study Hindi and Punjabi Languages and their comparison.
2. To develop machine readable Hindi to Punjabi Dictionary for the purpose of translation.
3. To develop algorithm for generating Named Entities from the Corpus and then using this lexicon of Named Entities in translation.
4. To develop lexicon for collocations in Hindi text to be used during translation process.
5. To develop the lexicon and algorithm for handling surnames and titles in input text.
6. To adapt and use the existing lexical resources such as digital dictionary, morph etc.
7. To develop transliteration module for handling out-of-vocabulary words.
8. To develop algorithm for postprocessing tasks.
9. To develop test cases for evaluating the system critically.

### 1.2.2 Challenges
There are number of challenges for developing a Machine Translation system. Some of the major challenges faced in development of Hindi to Punjabi MT system are:

1. Lack of lexical resources such as digital bilingual dictionary, morphological analyzer and generator, POS tagger etc. There is no machine readable dictionary available for Hindi to Punjabi. Morphological Analyzer for Hindi has been developed by IIIT Hyderabad but this can not be used directly into the system and lots of modifications are required for making its use in the system. This is used for handling inflectional words of a word. It is not possible to store all the words including inflected words into the lexicon.

2. Multiple translations in Punjabi for Hindi words. There are many Hindi words which have different meanings depending upon the context in which the word is present in the sentence. The program has to automatically decide the exact translation. We have used n-gram technique for disambiguating the word.

3. Identifying Named Entities present in the text like the word vishal goyal, State Bank of India, S. Parkash Singh Badal, Dr. Parkash.

4. Collection of phrases that cannot be translated word by word and these have different meaning in collection than in individual.

5. Handling grammatical errors after translation i.e. grammatical agreement corrections.
1.2.3 Need and Scope

Machine Translation Systems are in great demand and are widely in use. For the past few years, number of Machine Translation Systems has been developed for Indian and foreign languages but their quality of translation is not up to mark for use in real projects. Thus, at present no such acceptable system is available for most of the Indian languages. The use of computers is gaining popularity in day-to-day tasks of word processing, writing reports, and printing official documents etc. All the documents are written in their regional official language. Thus for making these documents readable and useful for other regions, translation systems must be developed. Therefore, Machine Translation systems are an obvious requirement in such a situation. Recently, “Sampark: Machine Translation System among Indian Languages” has been funded by TDIL, Department of Information technology, Govt. of India, developed by Consortium of Institutions has released the Hindi to Punjabi Machine Translation System on trial basis on 13\textsuperscript{th} August 2009 after spending three years. The translation is not promising and thus present system cannot be used for practical purposes. Indian languages have many features in common, so the present work could be well extendable to other Indian languages that are closely related to each other.

1.2.4 Potential Use [19-22]
The potential application areas of automatic Machine Translation are numerous and have the limits of imagination. Some of them are enumerated in this section.

**Large Scale Translation:** Large scale translation using MT is cost effective – there are many large companies saving time and money with MT. Leaving literature, sociology or legal texts aside (which require high level of publishable quality) MT is a success for technical documents especially within a particular domain. Typical texts are internal reports, operational manuals, repetitive publicity and marketing documents. Operational manuals, in particular, often represent many thousands of pages to be translated, and are extremely boring for human translators, and they are often needed in many languages (English, French, German, Japanese, etc.). But companies want fairly good quality of output as well. Manuals are repetitive, there may be frequent updates; and from one edition to another there may be very few changes. Automation is the obvious answer.

**As an Aid for Translators:** Machine Translation has changed the way translators work. The development of electronic termbanks, the increasing need to adhere to terminology standards, the overwhelming volumes of translation, and above all the development of facilities for using previous examples of translations have meant that translators could see the practical advantages of computerization in their work. Probably the largest users of computer aids for translation are found in the field of software and web localization. Localization means the adaptation of products for a particular
national or regional market, and the creation of documentation for those products. The incentive for computerization is the demand for the localization of publicity, promotional material, manuals for installation, maintenance and repair, etc. These must be available in the country (or countries) concerned as soon as the product itself is ready – often in a matter of days, not weeks.

**Translation of Websites:** A recent development is the appearance of software for translating webpages. Companies must now maintain high-profile presence on the Internet, in order to remain competitive. For multi-national companies, this also means that information on their websites must be made available in multi-languages. One solution is to refer users to online MT services but for many reasons this is unsatisfactory. Another is to engage a localization agency to translate every webpage. A third option which is increasingly adopted is to integrate one of the automatic webpage localization systems offered by many of the vendors of MT systems. Examples are ArabSite, IBM WebSphere, InterTran Website Translation Server, SDL Webflow, SystranLinks, and Worldlingo.

**MT for Assimilation:** Another main use of MT is assimilation, for getting the gist (essence) of the basic message of a text. The recipient does not necessarily require good quality. The main requirement is immediate use. However, the output must be readable; it must be reasonably intelligible for someone knowing the subject background. The wide availability of free translation of webpages makes it possible for companies and organizations to reach potential clients and customers who are unfamiliar with the language of
their websites; and many organizations provide links to such services for users to obtain translations of their websites.

**MT as a Cross Language Information retrieval Tool:** Closely related to the use of MT for translating texts for assimilation purposes is their use for aiding bilingual (or cross-language) communication and for searching, accessing and understanding foreign language information from databases and webpages. In the field of information retrieval there is much research at present on what is referred to as cross-language information retrieval (CLIR), i.e. information retrieval systems capable of searching databases in many different languages. Either they are systems which translate search terms from one language into other, and then do searching as a separate operation, with results presented en bloc to users; or, more ambitiously, translation of search terms or translation of output is conducted interactively with users.

**MT as a Tool for Communication:** It is probably true to say that one of the main applications of personal MT (‘home’) systems is the translation of correspondence (including personal e-mails) and the translation of web pages. Above all, there is oral communication involving translation. Although, we do not yet have speech translation, we do have systems with voice input and output, i.e. where users speak into the system, the spoken word or sentence is converted into text, the written text is translated into another text, and the system then produces spoken output.

**MT for Summarization:** Most people when faced with a foreign language text do not necessarily want the whole text translated, what they want is a
translated summary. There is a clear need for the production of summaries in languages other than the source. Summarization itself is a task which is difficult to automate; but applying MT to the task as well is an obvious expansion, either by translating the text as a whole into another language and then summarizing it, or by summarizing the original text and then translating the summary. The later has usually been the approach of researchers so far.

**MT as Key technology for Cyber Revolution:** Machine Translation can take information technology to the grassroots level and bring about sweeping societal changes through E-governance, E-commerce and E-entertainment leading to E-empowerment of the rural population. Local language information kiosks with computers, printers, Internet and E-mail facility are being set up to connect the Government to the citizen even at the grassroots level. At these kiosks, Machine Translation is essential so that all forms, records and information on the Government web site can be translated instantly into the local language that the people can understand. Similarly, the local language input by the citizens such as E-inquiries and E-grievances, should be machine translated at the click of the mouse, into a language that the concerned bureaucrat or minister can comprehend. [5].

### 1.3 Approach Applied for Our Machine Translation System

There are number of approaches discussed in the literature viz. Direct based, Transformer based, Interlingua based, Statistical etc. The choice of approach depends upon the available resources and the kind of languages involved.
Direct systems do not preclude syntactic or semantic analysis. There is a pragmatic constraint on the analysis, though, that it is subordinated to the translation task. Another difference concerns generation. A pure transfer system relies on a grammar for the target language to derive target sentences, while a direct system uses the word order of the source sentence as the point of departure for deriving a proper word order for the translation. A direct system relying on word-based analysis and transfer, will usually be able to derive some output for every input. The real issue, therefore, is empirical. In general, if the two languages are structurally similar, in particular as regards lexical correspondences, morphology and word order, the case for abstract syntactic analysis seems less convincing. Since the present research work deals with a pair of closely related language, so direct translation system is the obvious choice. The overall system architecture shown in figure 1.3 is adopted for Hindi to Punjabi Machine Translation System. The system is divided into three stages: Preprocessing, Translation Engine, and Post Processing stage. Following is the description of various steps of this architecture.

1.3.1 Pre Processing

The preprocessing stage is a collection of operations that are applied on input data to make it processable by the translation engine. In our current work, we have performed following pre processing steps:

- Text Normalization
- Replacing Collocations
1.3.1.1 Text Normalization

It works on spelling standardization issues, thereby resulting in multiple spelling variants for the same word. The major reasons for this phenomenon can be attributed to the phonetic nature of Indian languages and multiple dialects, transliteration of proper names, words borrowed from foreign languages, and the phonetic variety in Indian language alphabet. The variety in the alphabet, different dialects and influence of foreign languages has resulted in spelling variations of the same word. Such variations sometimes can be treated as errors in writing. During this phase of Pre Processing phase, rules specific to Hindi language which can handle such variations, which could result in more precise performance have been used for making the input text normalized for better accuracy.

For example we found widely used spelling variations for the Hindi word अंग्रेजी (aṅgrējī) as shown below:

अँग्रेजी, अंगरेजी, अन्ग्रेजी, अंग्रेजी, अंग्रेजी, अंग्रेजी

1.3.1.2 Replacing Collocations means finding and replacing those combinations of words in Hindi that cannot be translated word to word and such combinations of words have different word in group rather than their individual. This activity helps a lot in increasing the accuracy of the system.

For example, the collocation उत्तर प्रदेश (uttar pradēsh), if translated word to
word, will be translated as ਜਾਵਾਬ ਰਾਜ (javāb rāj), But it must be translated as ਉੱਤਰ ਪ੍ਰਦੇਸ਼ (uttar pradēsh).

1.3.1.3 Replacing Proper Nouns means finding and replacing those combination of words in the input text that are acting as names of person, bank, river, ocean, days of week, months of year, university, cooperative society etc. For example: ਕਮਲ ਗੋਯਲ (kamal gōyal) is a proper noun.

1.3.2 Tokenizer

The tokenizer takes the text generated by previous text as input. This module, using space, a punctuation mark, as delimiter, extracts tokens (word) from the text and gives it to Translation engine for analysis. This process is repeated for the whole text.
Figure 1.3: Overview of Hindi to Punjabi Machine Translation System

1.3.3 Translation Engine
The translation engine is responsible for translation of each token obtained from the previous step. It uses various lexical resources for finding the match of a given token in target language. Following is the description of how a token is passed through various modules.

1.3.3.1 Analyzing the word for Translation /Transliteration

The token obtained in the previous stage is passed through various stages.

1.3.3.1.1 Identifying Titles:

The token is checked whether it is a title like प्रो (prō), श्रीमती (shrīmtī) etc. If the current token is found to be a title, then the token next to it, should be transliterated instead of translation.

1.3.3.1.2 Identifying Surnames:

The token is checked whether it is a surname like अग्रवाल (agrāvāl), ओबेराय (ōbērāy ) etc. If the current token is found to be a surname, then the token previous to it, should be transliterated instead of translation.

1.3.3.1.3 Lexicon Lookup:

If the token does not satisfy above two steps, then it is looked into the lexicon for a match for direct word to word translation.

1.3.3.1.4 Resolving Ambiguity:

If the token is not present in the lexicon for direct translation, it is looked into the database of ambiguous words. If this token is found to be ambiguous, then disambiguity is resolved with the help of n-gram language modeling. The
system uses bigram and trigram databases, which contains one and two words respectively in the vicinity of an ambiguous word and corresponding meaning for that particular context.

### 1.3.3.1.5 Unknown Words:

If all the above modules fail to analyze the token, it is considered to be foreign/unknown word. Such words first pass through the morphological analysis phase based on the rules for inflections in Hindi words. Morphological generator generates the transliterated word using the inflectional rules and then checks the generated word in the Punjabi unigrams database for its genuinity. If this new generated word is found in the Punjabi unigrams, it is considered for translation otherwise the token is sent to transliteration module for transliteration.

Transliteration Module is the major module in the system that uses various rules specifically designed from the translation point of view.

### 1.3.4. Post Processing

After converting all the source text to target text, there are some of the grammatical errors that need to be corrected. For this purpose, we have formulated the rules for correcting the grammatical errors. Such rules have been implemented using Regular expressions and Pattern matching. This Post Processing phase is responsible for correcting grammatical errors in the generated output.
1.4 Thesis Outline

The study has been undertaken with the following chapter scheme:

In first chapter of this thesis, we introduce Machine Translation and provide details about various types of MT systems. The benefits, applications, and challenges of Machine Translation are described. After elaborating the various approaches used for Machine Translation and stages in a generic MT system we provide a formal description about the research question that we intend to undertake in this thesis work along with the major contribution and achievements of this research.

Chapter 2 discusses the existing work in the field of Machine Translation in India and outside India. This chapter on literature survey forms the basis of our work on developing the Machine Translation system and later on helps us in comparing our work with the existing state of the art in Machine Translation system.

Chapter 3 explains and compares the Hindi and Punjabi languages with respect to orthography, grammar, and Machine Translation.

Chapter 4 and 5 provide the design and implementation details of various activities involved in the Machine Translation system. Chapter 4 describes the system architecture and Pre processing stage. The chapter starts with the choice of approach and discusses the motivation behind its selection. Then the required resources are discussed followed by description of system architecture. The details of Pre processing phase which involves text
normalization, identifying collocations, identifying proper nouns are discussed. Then tokenization process is explained. The details of the translation system involving the identifying titles, identifying surnames, lexicon lookup, word sense disambiguation module, transliteration module and post processing modules are discussed in Chapter 5.

Chapter 6 describes the post processing stage of the system.

Chapter 7 provides the evaluation of the system and its results.

Chapter 8 concludes this thesis by providing a summary of the research work undertaken, contributions of this research work, limitations, and some directions in which this work could be extended in the future.

In appendix A, the interface designed for text translation, website translation and email translation has been discussed. Test data set for intelligibility test and accuracy test are available at Appendix B and C respectively.

1.5 Summary

In this chapter, introduction to Machine Translation, key activities involved, and various approaches for developing Machine Translation have been provided. It is followed by a formal statement for this research work along with its objectives, challenges involved, need and scope, and potential application areas of this system. Further, the approach followed to develop the Hindi to Punjabi Machine Translation System has been discussed along with an overview of the design of this system. The chapter concludes with a brief
outline of this thesis. The next chapter provides a survey of the existing 
literature in the field of Machine Translation.
Chapter 2

Survey of Literature

This chapter presents the state of the art in the field of Machine Translation. First part of this chapter discusses the Machine Translation systems for non Indian languages and second part discusses the Machine Translation systems for Indian languages.

2.1 Machine Translation Systems:

2.1.1 Machine Translation System for non Indian languages

Various Machine Translation systems have already been developed for most of the commonly used natural languages. This section briefly discusses some of the existing Machine Translation systems and the approaches that have been followed.

Georgetown Automatic Translation (GAT) System (1952), developed by Georgetown University, used direct approach for translating Russian texts (mainly from physics and organic chemistry) to English. The GAT strategy was simple word- for-word replacement, followed by a limited amount of transposition of words to result in something vaguely resembling English. There was no true linguistic theory underlying the GAT design. It had only six grammar rules and 250 items in its vocabulary. The translation was done using IBM 701 mainframe computer. Georgetown University and IBM jointly
conducted the Georgetown-IBM experiment in 1954 for more than sixty Russian sentences into English. The experiment was a great success and ushered in an era of Machine Translation research. The Georgetown MT project was terminated in the mid-60s.[8,23]

**CETA (1961)**, included the linguistic theory unlike GAT, for translating Russian into French. It was developed at Grenoble University in France. It is based on Interlingua approach with dependency-structure analysis of each sentence at the grammatical level and transfer mapping from one language-specific meaning representation at the lexical level. During the period of 1967-71, this system was used to translate about 4,00,000 words of Russian mathematics and physics texts into French. It was found that it fails for those sentences for which complete analysis cannot be derived. In 1971, new and improved system GETA based on the limitations of CETA was developed. [24-27]

**METAL (Mechanical Translation and Analysis of Languages) (1961)**, was developed at Linguistics Research Center, University of Texas for German into English. The system used indirect Machine Translation approach using Chomsky’s transformational paradigm. Indirect translation was performed in 14 steps of global analysis, transfer, and synthesis. The performance and accuracy of the system was moderate.[28]

**The Mark II (1964)**, a direct translation approach based Russian to English MT System for U.S. Air Force. It was developed by IBM Research Center. Translation was word by word, with occasional backtracking, Each Russian
item (either stem or ending) in the lexicon was accompanied by its English equivalent and grammatical codes indicating the classes of stems and affixes that could occur before and after it. In addition to lexical entries, processing instructions were also intermixed in the dictionary: ‘control entries’ relating to grammatical processes (forward and backward skips), and also instructions relating to loading and printing routines. There were some 25,000 such ‘control entries’ included in the dictionary. This contained 150,000 entries at the World’s Fair demonstration, and 180,000 in the USAF version. A third of the entries were phrases, and there was also an extensive system of micro glossaries. An average translation speed of 20 words per second was claimed. The examples of Russian-English translations at the World’s Fair were reasonably impressive (Bowers & Fisk (1965)). The Russian-English translations produced by Mark II were often rather crude and sometimes far from satisfactory. The limitations of word by word translation are more evident in the evaluation reports submitted by Pfafflin (1965), Orr & Small (1967), ALPAC(1966). An evaluation, MT research at the IBM Research Center ceased in 1966 (Roberts & Zarechnak 1974).

As one of the first operational MT systems, the IBM Russian-English system has a firm place in the history of MT. It was installed in the USAF’s Foreign Technology Division at the Wright-Patterson Air Force Base, Dayton, Ohio, where it remained in daily operation until 1970. [29]

LOGOS (1964), a direct Machine Translation system for English-Vietnamese language pair was initially developed by US Private firm Logos Corporation.

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
Logos analyzes whole source sentences, considering morphology, meaning, and grammatical structure and function. The analysis determines the semantic relationships between words as well as the syntactic structure of the sentence. Parsing is only source language-specific and generation is target language-specific. Unlike other commercial systems the Logos system relies heavily on semantic analysis. This comprehensive analysis permits the Logos system to construct a complete and idiomatically correct translation in the target language. This Internet-based system allows 251 users to submit formatted documents for translation to their server and retrieve translated documents without loss of formatting. In 1971, It was used by the U.S. Air Force to translate English maintenance manuals for military equipment into Vietnamese. Eventually, LOGOS forged an agreement with the Wang computer company that allowed the implementation of the German-English system on Wang office computers. This system reached the commercial market, and has been purchased by several multi-national organizations (e.g., Nixdorf, Triumph- Adler, Hewlett-Packard). The System is also available for English-French, English-German language pairs. [30-32]

**TAUM-AVIATION (1965)**, a transfer approach based English - French MT System for weather forecasts. It was developed at University of Montreal. After short span of time, the domain for translation shifted to translating aviation manuals by adding semantic analysis module to the system. The TAUM-AVIATION system is based on a typical second generation design (Isabelle et al. 1978, Bourbeau 1981). The translation is produced indirectly,
by means of an analysis/transfer/synthesis scheme. The overall design of the system is based on the assumption that translation rules should not be applied directly to the input string, but rather to a formal object that represents a structural description of the content of this input. Thus, the source language (SL) text (or successive fragments of it) is mapped onto the representations of an intermediate language, (also called normalized structure) prior to the application of any target language-dependent rule. In this system, the dictionaries list only the base form of the words (roughly speaking, the entry form in a conventional dictionary). In March 1981, the source language (English) dictionary included 4054 entries; these entries represented the core vocabulary of maintenance manuals, plus a portion of the specialized vocabulary of hydraulics. Of these, 3280 had a corresponding entry in the bilingual English-French dictionary. The system was evaluated and the low accuracy of the translation by the system forced the Canadian Government to cancel the funding and thus TAUM project in 1981. [33-34]

SYSTRAN (1968) is a direct Machine Translation system developed by Huchins and Somers. The system was originally built for English-Russian Language Pair. In 1970, SYSTRAN System installation at United States Air Force (USAF) Foreign Technology Division (FTD) at Wright-Patterson Air Force Base, Ohio, replaced IBM MARK-II MT System and is still operational. Large number of Russian scientific and technical documents were translated using this system. The quality of the translations, although only approximate, was usually adequate for understanding content. In 1974, NASA also selected
SYSTRAN to translate materials relating to the Apollo-Soyuz collaboration, and in 1976, EURATOM replaced GAT with SYSTRAN. The Commission of the European Communities (CEC) purchased an English-French version of SYSTRAN for evaluation and potential use. Unlike the FTD, NASA, and EURATOM installations, where the goal was information acquisition, the intended use by CEC was for information dissemination - meaning that the output was to be carefully edited before human consumption. The quality for this purpose was not adequate but improved after adding lexicon entries specific to CEC related translation tasks. Also in 1976, General Motors of Canada acquired SYSTRAN for translation of various manuals (for vehicle service, diesel locomotives, and highway transit coaches) from English into French on an IBM mainframe. GM's English-French dictionary had been expanded to over 1,30,000 terms by 1981 (Sereda 1982). GM purchased an English-Spanish version of SYSTRAN, and began to build the necessary [very large] dictionary. Sereda (1982) reported a speed-up of 3-4 times in the productivity of his human translators. Currently, SYSTRAN System is available for translating in 29 language pairs. [35-39]

**CULT (Chinese University Language Translator) (1968)**, is an interactive online MT System based on direct translation strategy for translating Chinese mathematics and physics journals into English. Sentences are analyzed and translated one at a time in a series of passes. After each pass, a portion of the sentence is translated into English. The CULT includes modules like source text preparation, input via Chinese keyboard, lexical analysis, syntactic and...
semantic analysis, relative order analysis, target equivalence analysis, output and output refinement. CULT is a successful system but it appears somewhat crude in comparison to interactive systems like ALPS and Weidner. [40-44]  

**ALPS (1971)**, a direct approach based English into French, German, Portuguese and Spanish for Mormon ecclesiastical texts. It was developed at Brigham Young University. It was started with an aim to develop fully automatic MT System but later in 1973, it became Machine Aided System. It is an Interactive Translation System that performs global analysis of sentences with human assistance, and then performs indirect transfer again with human assistance. But this project was not successful and hence not operational. [45]  

**The METEO (1977)**, is the world’s only example of a truly fully automatic MT System for Canadian Meterological Centre’s(CMC’s) nation wide weather communication networks. METEO scans the network traffic for English weather reports, translates them directly into French, and sends the translations back out over the communications network automatically. This system is based on the TAUM technology as discussed earlier. It is probably the first MT system where translators had involved in all phases of the design, development and refinement. Rather than relying on post-editors to discover and correct errors, METEO detects its own errors and passes the offending input to human editors and output deemed correct by METEO is dispatched without human intervention. This system correctly translates 90-95%, shuttling the other 5-10% to the human CMC translators.[46-47]
An English Japanese Machine Translation System (1982) developed by Makoto Nagao et. al. The title sentences of scientific and engineering papers are analyzed by simple parsing strategies. Title sentences of physics and mathematics of some databases in English are translated into Japanese with their keywords, author names, journal names and so on by using fundamental structures. The translation accuracy for the specific areas of physics and mathematics from INSPEC database was about 93%. [48]

RUSLAN (1985), a direct Machine Translation system between closely related languages Czech and Russian, by Hajic J, for thematic domain, the domain of operating systems of mainframes. The system used transfer based architecture. This project started in 1985 at Charles University, Prague in cooperation with Research Institute of Mathematical Machines in Prague. It was terminated in 1990 due to lack of funds. The system was rule based, implemented in Colmerauer’s Q-Systems. The system had a main dictionary of about 8,000 words, accompanied by transducing dictionary covering another 2,000 words. The typical steps followed in the system are Czech morphological analysis, syntactico semantic analysis with respect to Russian sentence structure and morphological synthesis of Russian. Due to close language pair, a transfer-like translation scheme was adopted with many simplifications. Also many ambiguities are left unresolved due to the close relationship between Czech and Russian. No deep analysis of input sentences was performed. The evaluations of results of RUSLAN showed that roughly 40% of the input sentences were translated correctly, about 40% of
input sentences with minor errors correctable by human post-editor and about 20% of the input required substantial editing or re-translation. There are two main factors that caused a deterioration of the translation. The first factor was the incompleteness of main dictionary of the system and second factor was the module of syntactic analysis of Czech. RUSLAN is a unidirectional system dealing with one pair of language Czech to Russian.[49]

**PONS (1995)**, an experimental interlingua system for automatic translation of unrestricted text, constructed by Helge Dyvik, Department of Linguistics and Phonetics, University of Bergen. 'PONS' is in Norwegian an acronym for "Partiell Oversettelse mellom Nærstående Språk" (Partial Translation between Closely Related Languages). PONS exploits the structural similarity between source and target language to make the shortcuts during the translation process. The system makes use of a lexicon and a set of syntactic rules. There is no morphological analysis. The lexicon consists of a list of entries for all word forms and a list of stem entries, or 'lexemes'. The source text is divided into substrings at certain punctuation marks, and the strings are parsed by a bottom-up, unification-based active chart parser. The system had been tested on translation of sentence sets and simple texts between the closely related languages Norwegian and Swedish, and between the more distantly related English and Norwegian. [50]

**interNOSTRUM (1999)** is a bidirectional Spanish-Catalan Machine Translation system. It was developed by Marote R.C. et al. It is a classical indirect Machine Translation system using an advanced morphological...
transfer strategy. Currently it translates ANSI, RTF (Microsoft’s Rich Text Format) and HTML texts. The system has eight modules: a deformatting module which separates formatting information from text, two analysis modules (morphological analyzer and part-of-speech tagger), two transfer modules (bilingual dictionary module and pattern processing module) and two generation modules (morphological generator and post-generator), and the reformatting module which integrates the original formatting information with the text. This system achieved great speed through the use of finite-state technologies. Error rates range around 5% in Spanish-Catalan direction when newspaper text is translated and are somewhat worse in the Catalan-Spanish direction. The Catalan to Spanish is less satisfactory as to vocabulary coverage and accuracy. [51]

ISAWIKA!(1999) is a transfer-based English-to-Tagalog MT system that uses ATN (Augmented Transition Network) as the grammar formalism. It translates simple English sentences into equivalent Filipino sentences at the syntactic level. [52]

English-to-Filipino MT system (2000) is a transfer based MT System that is designed and implemented using the lexical functional grammar (LFG) as its formalism. It involves morphological and syntactical analyses, transfer and generation stages. The whole translation process involves only one sentence at a time. [53]

Tagalog-to-Cebuano Machine Translation System (T2CMT)(2000) is a unidirectional Machine Translation system from Tagalog to Cebuano. It has three
stages: Analysis, Transfer and Generation. Each stage uses bilingual from Tagalog to Cebuano lexicon and a set of rules. The morphological analysis is based on TagSA (Tagalog Stemming Algorithm) and affix correspondence-based POS (part-of-speech) tagger. The author describes that a new method is used in the POS-tagging process but does not handle ambiguity resolution and is only limited to a one-to-one mapping of words and parts-of-speech. The syntax analyzer accepts data passed by the POS tagger according to the formal grammar defined by the system. Transfer is implemented through affix and root transfers. The rules used in morphological synthesis are reverse of the rules used in morphological analysis. T2CMT has been evaluated, with the Book of Genesis as input, using GTM (General Text Matcher), which is based on Precision and Recall. Result of the evaluation gives a score of good performance 0.8027 or 80.27% precision and 0.7992 or 79.92% recall. [54]

**Turkish to English Machine Translation system (2000)** is a hybrid Machine Translation system by combining two different approaches to MT. The hybrid approach transfers a Turkish sentence to all of its possible English translations, using a set of manually written transfer rules. Then, it uses a probabilistic language model to pick the most probable translation out of this set. The system is evaluated on a test set of Turkish sentences, and compared the results to reference translations. The accuracy comes out to be about 75.6%. [55]

**CESILKO (2000),** is a Machine Translation system for closely related Slavic language pairs, developed by HAJIC J, HRIC J K. and UBON V. It has been
fully implemented for Czech to Slovak, the pair of two most closely related Slavic languages. The main aim of the system is localization of the texts and programs from one source language into a group of mutually related target languages. In this system, no deep analysis had been performed and word-for-word translation using stochastic disambiguation of Czech word forms has been performed. The input text is passed through different modules namely morphological analyzer, morphological disambiguation, Domain related bilingual glossaries, general bilingual dictionary, and morphological synthesis of Slovak. The dictionary covers over 7,00,000 items and it is able to recognize more than 15 million word-forms. The system is claimed to achieve about 90% match with the results of human translation, based on relatively large test sample. Work is in progress on translation for Czech-to-Polish language pairs.[56]

**Bulgarian-to-Polish Machine Translation system (2000),** has been developed by S. Marinov. This system has been developed based on the approach followed by PONS discussed above. The system needs a grammar comparison before the actual translation begins so that the necessary pointers between similar rules are created and system is able to determine where it can take a shortcut. The system has three modes, where mode 1 and 2 enable system to use the source language constructions and without making a deeper semantic analysis to translate to the target language construction. Mode 3 is the escape hatch, when the Polish sentences have to
be generated from the semantic representation of the Bulgarian sentence. The accuracy of the system has been reported to be 81.4%.[57]

**Tatar (2001)**, a Machine Translation system between Turkish and Crimean, developed by Altintas K. et al., used finite state techniques for the translation process. It is in general disambiguated word for word translation. The system takes a Turkish sentence, analyses all the words morphologically, translates the grammatical and context dependent structures, translates the root words and finally morphologically generates the Crimean Tatar text. One-to-one translation of words is done using a bilingual dictionary between Turkish and Crimean Tatar. The system accuracy can be improved by making word sense disambiguation module more robust.[58]

**Antonio M. Corbí-Bellot et. al. (2005)** developed the open source shallow-transfer Machine Translation (MT) engine for the Romance languages of Spain (the main ones being Spanish, Catalan and Galician). The Machine Translation architecture uses finite-state transducers for lexical processing, hidden Markov models for part-of-speech tagging, and finite-state based chunking for structural transfer. The author claims that, for related languages such as Spanish, Catalan or Galician, a rudimentary word-for-word MT model may give an adequate translation for 75% of the text, the addition of homograph disambiguation, management of contiguous multi-word units, and local reordering and agreement rules may raise the fraction of adequately translated text above 90%. [59]
Carme Armentano-oller et. al (2005) extended the idea of A.M.Corbi-Bellot et. al. and developed an open source Machine Translation tool box which includes (a) the open-source engine itself, a modular shallow transfer Machine Translation engine suitable for related languages (b) extensive documentation specifying the XML format of all linguistic (dictionaries, rules) and document format management files, (c) compilers converting these data into the high speed format used by the engine, and (d) pilot linguistic data for Spanish—Catalan and Spanish—Galician and format management specifications for the HTML, RTF and plain text formats. They use the XML format for linguistic data used by the system. They define five main types of formats for linguistic data i.e. dictionaries, tagger definition file, training corpora, structural transfer rule files and format management files. [60]

Apertium (2005), developed by Carme Armentano-oller et. al is an open-source shallow-transfer Machine Translation (MT) system for the [European] Portuguese ↔ Spanish language pair. This platform was developed with funding from the Spanish government and the government of Catalonia at the University of Alicante. It is a free software and released under the terms of the GNU General Public License. Apertium originated as one of the Machine Translation engines in the project OpenTrad and was originally designed to translate between closely related languages, although it has recently been expanded to treat more divergent language pairs (such as English–Catalan).

Apertium uses finite-state transducers for all lexical processing operations (morphological analysis and generation, lexical transfer), hidden Markov
models for part-of-speech tagging, and multi-stage finite-state based chunking for structural transfer. For Portuguese–Spanish language pair, promising results are obtained with the pilot open-source linguistic data released which may easily improve (down to error rates around 5%, and even lower for specialized texts), mainly through lexical contributions from the linguistic communities involved. [61]

**ga2gd (2006)**, a robust Machine Translation system, developed by Scannell K.P., between Irish and Scottish Gaelic despite the lack of full parsing technology or pre-existing bilingual lexical resources. It includes the modules Irish standardization, POS Tagging, stemming, chunking, WSD, Syntactic transfer, lexical transfer, and Scottish post processing. The accuracy has been reported to be 92.72%. [62]

**SisHiTra(2006)** is a hybrid Machine Translation system from Spanish to Catalan. It was developed by Gonzalez et. al. This project tried to combine knowledge-based and corpus-based techniques to produce a Spanish-to-Catalan Machine Translation system with no semantic constraints. Spanish and Catalan are languages belonging to the Romance language family and have a lot of characteristics in common. SisHiTra makes use of their similarities to simplify the translation process. A SisHiTra future perspective is the extension to other language pairs (Portuguese, French, Italian, etc.). The system is based on finite state machines. It has following modules: preprocessing modules, generation module, disambiguation module and post-
processing module. The word error rate is claimed to be 12.5% for SisHiTra system.[63]

2.1.2 Machine Translation Systems for Indian languages

This section will summarize the existing Machine Translation systems for Indian languages that are as follows:

ANGLABHARTI (1991), is a machine-aided translation system specifically designed for translating English to Indian languages. English is a SVO language while Indian languages are SOV and are relatively of free word-order. Instead of designing translators for English to each Indian language, Anglabharti uses a pseudo-interlingua approach. It analyses English only once and creates an intermediate structure called PLIL (Pseudo Lingua for Indian Languages). This is the basic translation process translating the English source language to PLIL with most of the disambiguation having been performed. The PLIL structure is then converted to each Indian language through a process of text-generation. The effort in analyzing the English sentences and translating into PLIL is estimated to be about 70% and the text-generation accounts for the rest of the 30%. Thus only with an additional 30% effort, a new English to Indian language translator can be built. The attempt has been made to 90% translation task to be done by machine and 10% left to the human post-editing. The project has been applied mainly in the domain of public health. [64]
Anusaaraka (1995) was developed at IIT Kanpur, and was later shifted to the Center for Applied Linguistics and Translation Studies (CALTS), Department of Humanities and Social Studies, University of Hyderabad. Of late, the Language Technology Research Center (LTRC) at IIIT Hyderabad is attempting an English-Hindi Anusaaraka MT System. The focus in Anusaaraka is not mainly on Machine Translation, but on Language access between Indian Languages. Using principles of Paninian Grammar (PG), and exploiting the close similarity of Indian languages, it essentially maps local word groups between the source and target languages. Where there are differences between the languages, the system introduces extra notation to preserve the information of the source language. The project has developed Language Accesors for Punjabi, Bengali, Telugu, Kannada and Marathi into Hindi. The output generated is understandable but not grammatically correct. For example, a Bengali to Hindi Anusaaraka can take a Bengali text and produce output in Hindi which can be understood by the user but will not be grammatically perfect. The system has mainly been applied for children's stories.[65]

Anubharati (1995), used EBMT paradigm for Hindi to English translation. The translation is obtained by matching the input sentences with the minimum distance example sentences. The system stored the examples in generalized form to contain the category/class information to a great extent. This made the example-base smaller in size and its further processing partitioning reduces
the search space. This approach works more efficiently for similar languages such as among Indian languages. [66]

**The Mantra (MAchiNe assisted TRAnsl ation tool) (1999)** translates English text into Hindi in a specified domain of personal administration specifically gaze\n\ntes pertaining to government appointments, office orders, office memorandums and circulars. It is based on the TAG formalism from University of Pennsylvania. In addition to translating the content, the system can also preserve the formatting of input word documents across the translation. The Mantra approach is general, but the lexicon/grammar has been limited to the language of the domain. This project has also been extended for Hindi-English and Hindi-Bengali language pairs and also existing English-Hindi translation has been extended to the domain of parliament proceeding summaries.[67]

**MAT (2002),** a machine assisted translation system for translating English texts into Kannada, has been developed by Dr. K. Narayana Murthy at Resource Centre for Indian Language Technology Solutions, University of Hyderabad. Their approach is based on using the Universal Clause Structure Grammar (UCSG) formalism. The input sentence is parsed by UCSG parser and outputs the number, type and inter-relationships amongst various clauses in the sentence and the word groups that take on various functional roles in clauses. Keeping this structure in mind, a suitable structure for the equivalent sentence in the target language is first developed. For each word, a suitable target language equivalent is obtained from the bilingual dictionary. The MAT
System provides for incorporating syntactic and some simple kinds of semantic constraints in the bilingual dictionary. The MAT system includes morphological analyzer/generator for Kannada. Finally, the target language sentence is generated by placing the clauses and the word groups in appropriate linear order, according to the constraints of the target language grammar. Post Editing tool has been provided for editing the translated text. MAT System 1.0 had shown about 40-60% of fully automatic accurate translations. It has been applied to the domain of government circulars, and funded by the Karnataka government. [68]

**An English–Hindi Translation System (2002)** with special reference to weather narration domain has been designed and developed by Lata Gore et. al. The system is based on transfer based translation approach. MT system transfers the source sentence to the target sentence with the help of different grammatical rules and also a bilingual dictionary. The translation module consists of sub modules like Pre-processing of input sentence, English tree generator, post-processing of English tree, generation of Hindi tree, Post-processing of Hindi tree and generating output. The translation system gives domain specific translation with satisfactory results. By modifying the database it can be extended to other domains.[69]

**VAASAANUBAADA (2002)**, an Automatic Machine Translation of Bilingual Bengali-Assamese News Texts using Example-Based Machine Translation technique, has been developed by Kommaluri Vijayanand et. al. It involves Machine Translation of bilingual texts at sentence level. In addition, it also
includes preprocessing and post-processing tasks. The bilingual corpus has been constructed and aligned manually by feeding the real examples using pseudo code. The longer input sentence is fragmented at punctuations, which results in high quality translation. Backtracking is used when the exact match is not found at the sentence/fragment level, leading to further fragmentation of the sentence. The results when tested by authors are fascinating with quality translation. [70]

**ANGLABHARTI-II (2004)** addressed many of the shortcomings of the earlier architecture. It uses a generalized example-base (GEB) for hybridization besides a raw example-base (REB). During the development phase, when it is found that the modification in the rule-base is difficult and may result in unpredictable results, the example-base is grown interactively by augmenting it. At the time of actual usage, the system first attempts a match in REB and GEB before invoking the rule-base. In AnglaBharti-II, provisions were made for automated pre-editing & paraphrasing, generalized & conditional multi-word expressions, recognition of named-entities. It incorporated an error-analysis module and statistical language-model for automated post-editing. The purpose of automatic pre-editing module is to transform/paraphrase the input sentence to a form which is more easily translatable. Automated pre-editing may even fragment an input sentence if the fragments are easily translatable and positioned in the final translation. Such fragmentation may be triggered by in case of a failure of translation by the 'failure analysis' module. The failure analysis consists of heuristics on speculating what might have
gone wrong. The entire system is pipelined with various sub-modules. All these have contributed significantly to greater accuracy and robustness to the system. [71]

The MaTra system (2004), a tool for human aided Machine Translation from English to Indian languages currently Hindi, has been developed by the Natural Language group of the Knowledge Based Computer Systems (KBCS) division at the National Centre for Software Technology (NCST), Mumbai (currently CDAC, Mumbai). The system has been applied mainly in the domain of news, annual reports and technical phrases. This system used transfer approach using a frame-like structured representation. The system used rule-bases and heuristics to resolve ambiguities to the extent possible. It has a text categorization component at the front, which determines the type of news story (political, terrorism, economic, etc.) before operating on the given story. Depending on the type of news, it uses an appropriate dictionary. It requires considerable human assistance in analyzing the input. Another novel component of the system is that given a complex English sentence, it breaks it up into simpler sentences, which are then analyzed and used to generate Hindi. The system can work in a fully automatic mode and produce rough translations for end users, but is primarily meant for translators, editors and content providers. [72]

ANUBHARTI-II (2004) has been generalized to cater to Hindi as source language for translation to any other Indian language. The system used hybrid Example-based Machine Translation approach which is a combination of
example-based approach and traditional rule-based approach. The example-based approaches emulate human-learning process for storing knowledge from past experiences to use it in future. It also uses a shallow parsing of Hindi for chunking and phrasal analysis. The input Hindi sentence is converted into a standardization form to take care of word-order variations. The standardized Hindi sentences are matched with a top level standardized example-base. In case no match is found then a shallow chunker is used to fragment the input sentence into units that are then matched with a hierarchical example-base. The translated chunks are positioned by matching with sentence level example base. Human post-editing is performed primarily to introduce determiners that are either not present or difficult to estimate in Hindi. [71]

Shakti (2004), is a Machine Translation system from English to any Indian language currently being developed at Language Technologies Research Centre, IIIT-Hyderabad. It has already produced output from English to three different Indian languages – Hindi, Marathi, and Telugu. It combines rule based approach with statistical approach. The rules are mostly linguistic in nature and the statistical approach tries to infer or use linguistic information. Although the system accommodates multiple approaches, the backbone of the system is linguistic analysis. The system consists of 69 different modules. About 9 modules are used for analyzing the source language (English), 24 modules are used for performing bilingual tasks such as substituting target language roots and reordering etc., and the remaining modules are used for
generating target language. The overall system architecture is kept extremely simple. All modules operate on a stream of data whose format is Shakti standard format (SSF). [73]

Shiva (2004), is an example based Machine Translation system from English to Hindi developed at IIIT Hyderabad.[73,74]

English-Telugu Machine Translation System has been developed jointly at CALTS with IIIT, Hyderabad, Telugu University, Hyderabad and Osmania University, Hyderabad. This system uses English-Telugu lexicon consisting of 42,000 words. A word form synthesizer for Telugu is developed and incorporated in the system. It handles English sentences of a variety of complexity.[74]

Telugu-Tamil Machine Translation System has also been developed at CALTS using the available resources here. This system uses the Telugu Morphological analyzer and Tamil generator developed at CALTS. The backbone of the system is Telugu-Tamil dictionary developed as part of MAT Lexica. It also used verb sense disambiguator based on verbs argument structure. [74]

ANUBAAD (2004), an example based Machine Translation system for translating news headlines from English to Bengali, has been developed by Sivaji Bandyopadhyay at Jadavpur University Kolkata. During translation, the input headline is initially searched in the direct example base for an exact match. If a match is obtained, the Bengali headline from the example base is produced as output. If there is no match, the headline is tagged and the
tagged headline is searched in the Generalized Tagged Example base. If a match is obtained, the output Bengali headline is to be generated after appropriate synthesis. If a match is not found, the Phrasal example base will be used to generate the target translation. If the headline still cannot be translated, the heuristic translation strategy applied is - translation of the individual words or terms in their order of appearance in the input headline will generate the translation of the input headline. Appropriate dictionaries have been consulted for translation of the news headline. [75]

**Hinglish (2004)**, a Machine Translation system for pure (standard) Hindi to pure English forms developed by R. Mahesh K. Sinha and Anil Thakur. It had been implemented by incorporating additional layer to the existing English to Hindi translation (AnglaBharti-II) and Hindi to English translation (AnuBharti-II) systems developed by Sinha. The system claimed to be produced satisfactory acceptable results in more than 90% of the cases. Only in case of polysemous verbs, due to a very shallow grammatical analysis used in the process, the system is unable to resolve their meaning. [76]

**Tamil-Hindi Machine-Aided Translation system** has been developed by Prof. C.N. Krishnan at AU-KBC Research Centre, MIT Campus, Anna University Chennai. This system is based on Anusaaraka Machine Translation System architecture. It uses a lexical level translation and has 80-85% coverage. Stand-alone, API, and Web-based on-line versions have been developed. Tamil morphological analyser and Tamil-Hindi bilingual dictionary (~ 36k) are the by products of this system. They also developed a prototype of

Language in India [www.languageinindia.com](http://www.languageinindia.com) 681
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
English - Tamil MAT system. It includes exhaustive syntactical analysis. Currently, it has limited vocabulary (100-150) and small set of Transfer rules. [77]

AnglaHindi (2003), a pseudo –interlingual rule-based English to Hindi Machine-Aided Translation System, developed by Sinha et. al. at IIIT, Kanpur. It is a derivative of AnglaBharti MT System for English to Indian languages. AnglaHindi besides using all the modules of AnglaBharti, also makes use of an abstracted example-base for translating frequently encountered noun phrases and verb phrasals. The system generates approximately 90% acceptable translation in case of simple, complex and compound sentences upto a length of 20 words. [78]

IBM-English-Hindi Machine Translation System has been initially developed by IBM India Research Lab at New Delhi with EBMT approach. Now, the approach has been changed to statistical Machine Translation between English and Indian languages. [79-84]

English to {Hindi, Kannada, Tamil} and Kannada to Tamil Language-Pair Example Based Machine Translation (2006) has been developed by Prashanth Balajapally. It is based on a bilingual dictionary comprising of sentence-dictionary, phrases-dictionary, words-dictionary and phonetic-dictionary and is used for the Machine Translation. Each of the above dictionaries contains parallel corpora of sentences, phrases and words, and phonetic mappings of words in their respective files. Example Based Machine Translation (EBMT) has a set of 75000 most commonly spoken sentences
that are originally available in English. These sentences have been manually translated into three of the target Indian languages, namely Hindi, Kannada and Tamil. [79-83]

**Google Translate (2007),** is based on statistical Machine Translation approach, and more specifically, on research by Franz-Josef Och. Before using statistical approach, Google translate was using SYSTRAN for its translation till 2007. Currently, it is providing the facility of translation among 51 language pairs. It includes only one Indian language Hindi. The accuracy of translation is good enough to understand the translated text. [Internet Source: http://translate.google.com/]

**Punjabi to Hindi Machine Translation System (2007)** has been developed by Gurpreet Singh Joshan et. al. at Punjabi University Patiala. This system is based on direct word-to-word translation approach. This system consists of modules like pre-processing, word-to-word translation using Punjabi-Hindi lexicon, morphological analysis, word sense disambiguation, transliteration and post processing. The system has reported 92.8% accuracy. [84]

**Sampark: Machine Translation System among Indian languages (2009),** developed by the Consortium of Institutions. Consortium of institutions include IIIT Hyderabad, University of Hyderabad, CDAC(Noida,Pune), Anna University, KBC, Chennai, IIT Kharagpur, IIT Kanpur, IISc Bangalore, IIIT Allahabad, Tamil University, Jadavpur University. Currently experimental systems have been released namely {Punjabi, Urdu, Tamil, Marathi} to Hindi
and Tamil-Hindi Machine Translation systems. The accuracy of the translation is not up to the mark.[Internet Source:http://sampark.iiit.ac.in]

**Yahoo! Bable Fish (2008),** developed by AltaVista, is a web-based application on Yahoo! that machine translates text or web pages from one of several languages into another. The translation technology for Babel Fish is provided by SYSTRAN. It translates among English, Simplified Chinese, Traditional Chinese, Dutch, French, German, Greek, Italian, Japanese, Korean, Portuguese, Russian, Swedish, and Spanish. [Internet Source: http://babelfish.yahoo.com/]

**Microsoft Bing Translator (2009) is a service provided by Microsoft as part of its Bing services which allow users to translate texts or entire web pages into different languages. All translation pairs are powered by Microsoft Translation (previously Systran), developed by Microsoft Research, as its backend translation software. The translation service is also using statistical Machine Translation strategy to some extent [Internet Source: http://www.microsofttranslator.com/]

**Bengali to Hindi Machine Translation System (2009) is a hybrid Machine Translation system, developed at IIT Kharagpur. This system uses multi-engine Machine Translation approach. It is based on the unfactored Moses SMT system with Giza++ (Josef,2000) derived phrase table as a central element. This system uses dictionary consisting of 15,000 parallel sysnets, Gazeteer list consisting of 50,000 parallel name list, monolingual corpus of 500K words both from source and target languages, suffix list of 100 Bengali
linguistic suffixes. The BLUE score obtained during system evaluation is 0.2318. [85]

2.2 Summary

As we have seen in the above discussion the English to Japanese, GAT (English-Russian), Mark-II (Russian-English), LOGOS (English-Vietnamese), SYSTRAN (English-Russian), CULT (Chinese mathematics and physics journals into English), ALP (English into French, German, Portuguese and Spanish), RUSLAN (Czech and Russian), CESILKO (Czech to Slovak), English-Arabic and Punjabi to Hindi Machine Translation Systems have been developed using direct MT approach for closely related language pairs. Some of these are very successful and popular Machine Translation systems which are still operational. Thus, it is concluded that direct Machine Translation approach is the most appropriate for closely related languages.

Hindi and Punjabi is a case of closely related but distinct languages as these languages are not mutually intelligible, having distinct orthographies, independent lexica and number of important structural differences in terms of syntax. Hindi and Punjabi, being one of the closest pairs of Indo-Iranian languages, are chosen in this study as a model for translation between any pair of close languages. They have most parts of their grammar in common although morphemes and expressions may differ. The use of narration in both languages is almost the same and a narration can directly be translated. But it
is not straightforward to translate some phrases, idioms and even some grammatical structures.

Hence, direct approach is most suitable approach for developing Hindi to Punjabi Machine Translation System. In the next chapter we will discuss about the comparative study of Hindi and Punjabi languages in detail.
Chapter 3

Comparative Study of Hindi and Punjabi

3.1 Introduction

India is a linguistically rich country having eighteen constitutional languages, which are written in ten different scripts. Indian languages can be broadly classified into five groups according to their origin and similarity. These are Indo-Aryan family (Hindi, Bangla, Assami, Punjabi, Marathi, Oriya and Gujarati); Dravidian family (Tamil, Telugu, Kannada and Malayalam); Austro-Asian family and Tibetan-Burmese family and Andamanese (Jha, 2005). Many of them are structurally similar called sibling languages. Within each group, there is high degree of structural similarity. With some efforts effective mapping rules can be created amongst languages within the same group. Indian languages are inflectional with a rich morphology, relatively free word order, and default sentence structure as SOV (Subject Object Verb). It is believed that Machine Translation systems can be developed with less effort and using direct approach between sibling language pairs. [85]

In this chapter, we will discuss the comparative study of the language pair of our Machine Translation system i.e. Hindi and Punjabi. Our motive of comparative analysis is to sort out the closeness between Hindi and Punjabi from Machine Translation point of view and to make the base for deciding about the appropriate approach to be followed for development of our
Machine Translation system. By analysis we mean the identification of bilingual rules for source language and target language so that the transfer of source language to target language can be performed by computers successfully. In order for MT systems to work, source and target languages must be fully analyzed. This kind of study, however, is not adequately covered by theoretical linguistics. V. Geethakumary [86] states that if the source language and the target language both have significantly similar linguistic features on all the levels of their structures then the first step to be adopted is that both languages should be analyzed independently. After the independent analysis, to sort out the different features of the two languages, comparison of the two languages is necessary. The present study has been undertaken keeping in view the Machine Translation system being developed for languages from Hindi to Punjabi. This is not a complete analysis, but rather a comparison to give some idea about Hindi and Punjabi grammar. It covers main aspects of Hindi and Punjabi languages. Details of both Hindi and Punjabi grammar can be found in Michel [87] and Singh and Singh [88] respectively. Following sections will discuss about the comparison between the Hindi and Punjabi Language on the basis of orthography and grammar. This chapter also discusses these languages from Machine Translation point of view.

3.2 Comparison between Hindi and Punjabi Language On the basis of Orthography: [87-102]
3.2.1 Family and Status:

Hindi and Punjabi languages belong to the same subgroup of the Indo-European family i.e. Indo-Aryan family of the languages. Hindi and Punjabi are spoken by about 577 million people and 100 million people all over the world respectively. Hindi and Punjabi have been ranked 4th and 11th widely spoken language in the world respectively (Ethnologue, 2009). In India, Hindi has been accorded the status of ‘official language’ by the central government for use for most administrative purposes, and Punjabi being the official language of the state the Punjab and has been accorded the status of ‘official language’ by the Punjab government for use for most administrative purposes. Both the languages have originated from Sanskrit (Masica 1991). Punjabi language is mostly used in the region of Punjab, Haryana, Delhi, Himachal Pardesh, Jammu & Kashmir and in some areas of Pakistan namely Punjab, Sindh and Blochistan. On the other hand, Hindi is a national language of India and is spoken and used by the people all over the country. But the main regions are Haryana, Uttar Pardesh, Rajasthan, Bihar and Chattisgarh.

3.2.2 Script

3.2.2.1 Devanagari script:

Hindi Language is written in Devanagari Script. It is written Left-to-Right. The Devanagari script, used for writing Sanskrit and other Indian languages had evolved over a period of more than two thousand years. Devanagari emerged around 1200 AD out of the Siddham script, gradually replacing the earlier,
closely related Sharada script (which remained in parallel use in Kashmir). Both are immediate descendants of the Gupta script, ultimately deriving from the Brāhmī script attested from the 3rd century BC; Nagari appeared in approx. the 8th century as an eastern variant of the Gupta script, contemporary to Sharada, its western variant. The descendants of Brahmi form the Brahmic family, including the alphabets employed for many other South and South-East Asian languages.

Nāgarī is in Sanskrit the feminine of nāgara. The feminine form is used because of its original application to qualify the feminine noun lipi "script". There were several varieties in use, one of which was distinguished by affixing deva "divine, deity" to form a tatpurusha compound meaning the "divine urban(e) [script]". However, the widespread use of "Devanagari" is a relatively recent phenomenon; well into the twentieth century, and even today, simply "Nagari" was (and is) also in use for this same script. The rapid spread of the usage of "Devanagari" seems also to be connected with the almost exclusive use of this script in colonial times (particularly by European scholars) to publish works in Sanskrit (held by many to be the language of the gods), even though traditionally nearly all indigenous scripts have actually been employed for this language. This has led to the establishment of such a close connection between the script and Sanskrit that it is erroneously widely regarded as "the Sanskrit script" today.

3.2.2.2 Gurmukhi Script:

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
A unique feature of Punjabi is that it is written in two mutually incomprehensible scripts. In India Punjabi language is written in Gurmukhi script, while in Pakistan it is written in Shamukhi (Urdu) script. Gurmukhi script is written Left-to-Right and Shahmukhi is written right-to-left. Gurmukhi Script derived from the Sharada script and standarized by Guru Angad Dev in the 16th century, was designed to write the Punjabi Language (Gill, Gleason, 1963). The word Gurmukhi is commonly translated as “from the mouth of Guru”. However, the term used for the Punjabi script has somewhat different connotations. The opinion given by traditional scholars is that as the Sikh holy writings, before they were scribed, were uttered by the Gurus, they came to be known as Gurmukhi or the “Utterence of the Guru”. And consequently, the script that was used for scribing the utterence was also given the same name. However, the prevalent view among Punjabi linguists is that as in the early stages the Gurmukhi letters were primarily used by Gurmukhs, or the Sikhs devoted to the Guruy, the script came to be associated with them. Another view is that as the Gurmukhs, in accordance with the Sikh belief, used to meditate on the letter द, ध, झ, ञ which jointly forms दधझञ or God in Sikhism, these letters were called Gurmukhi or the “Speech of the Gurmukhs”. Subsequently, the whole script came to be known as Gurmukhi.

Like most of the north Indian writing systems, the Gurmukhi script is a descendent of the Brahmi script. It is believed that Gurmukhi script was invented by the second Sikh Guru, Guru Angad Dev, However, it would be correct to say that script was standarized rather than invented, by the Sikh
Gurus. E.P. Newton (Panjabi Grammar, 1898) writes that at least 21 Gurmukhi characters are found in ancient manuscripts: 6 from 10th century, 12 from 3rd century BC and 3 from 5th century BC. Apparently, the first Sikh Guru, Guru Nanak Dev also used the Gurmukhi script for his writings. The usage of Gurmukhi letters in Guru Granth Sahib meant that the script developed its own orthographical rules. In the following epochs, Gurmukhi became the prime script applied for literary writings of the Sikhs. Later in the 20th century, the script was given the authority as the official script of the Eastern Punjabi Language. Meanwhile, in western Punjab, a form of the Urdu script, known as Shahmukhi is still in use.

3.2.3 Consonants:

3.2.3.1 Basic Consonants

There are thirty three basic consonants or consonant-like graphs in Devanagari script and thirty-five in Gurmukhi scripts which are as follows.

Table 3.1: Basic Consonants in Devanagari

| क k | ख kh | ग g | घ gh | ङ n |
| च c | छ ch | ज j | झ jh | ङ n |
| ट ṭ | ठ ṭ | ड ṭ | ढ ṭ | ण n |
| त t | थ th | द d | ध dh | न n |
| प p | फ ph | ब b | भ bh | म m |
| य y | र r | ल l | व v |
In addition to basic consonants, there are other consonants that are formed with some of the basic consonants supplemented with a dot diacritic. In Devanagari script these are क़ (k), ख़ (kh), ग़ (g), ज़ (z), फ़ (f), ढ़ (ḍ) and in Gurmukhi script, these are ਷ (ṣ), ਸ (s), ਲ (l), ਵ (v) and ਭ (bh). There is one more such consonant ਫ (ḍ) in Gurmukhi script. But it is not much frequent in clusters. It was a proposal to distinguish consonant ਫ (ḍ) from ਫ (f) by adding a dot diacritic like that used to distinguish ਸ (s) from ਸ (sh). This however has met with no acceptance and is seldom if ever used.
3.2.3.2 Dead and Live Consonants:

Devanagari employs a sign known in Sanskrit as the *virama* or vowel omission sign. In Devanagari and Gurmukhi both, it is called *hal* or *halant*, and that term is used in referring to the virama or to a consonant with its vowel suppressed by the virama. The virama sign (ॆ) nominally serves to cancel (or kill) the inherent vowel of the consonant to which it is applied. When a consonant has lost its inherent vowel by the application of virama, it is known as a *dead consonant*; in contrast, a *live consonant* is one that retains its inherent vowel or is written with an explicit dependent vowel sign.

3.2.3.3 Consonant Conjuncts:

The Indic scripts are noted for a large number of consonant conjunct forms that serve as orthographic abbreviations (ligatures) of two or more adjacent letterforms (Michael, 1986). This abbreviation takes place only in the context of a *consonant cluster*. An orthographic consonant cluster is defined as a sequence of characters that represents one or more dead consonants followed by a normal, *live* consonant letter.

In Devanagari, we have four consonant conjuncts namely ज़ (ज +ञ), ध़ (ध +़), ध (ध +र), ज़ (ज +र).
In Gurmukhi, only three types of conjunct consonants are used. In all bases, a modified form of the second consonant is subjoined to the unaltered form of the first. In the first type, a form of ṭ(h) is subjoined. The following table shows the common combinations.

**Table 3.3: Conjugt Consonants**

<table>
<thead>
<tr>
<th>Base</th>
<th>Form</th>
<th>Devanagari Equivalent</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>ṭ (r)</td>
<td>ṭ (rh)</td>
<td>ṭ (r)</td>
<td>प्र (parḥ)</td>
</tr>
<tr>
<td>ṭ (n)</td>
<td>ṭ (nh)</td>
<td>नें nham</td>
<td>देव (nhēr)</td>
</tr>
<tr>
<td>ṭ (l)</td>
<td>ṭ (lh)</td>
<td>ल्ह (lh)</td>
<td>ल्ह (lhā)</td>
</tr>
<tr>
<td>ह (m)</td>
<td>ह (mh)</td>
<td>भ (mh)</td>
<td>म्या (mhaṁs)</td>
</tr>
</tbody>
</table>

In second type of conjunct, a form of ṭ (r) is subjoined to certain consonants, most commonly stops. These occur only in tatsamas (Those words that are directly borrowed from Sanskrit with little or no phonetic alteration) like भ्र , च्र , श्र etc. In Devanagari, when ṭ is served as the second member of a cluster, it is indicated by a small diagonal slash (going in the opposite direction from that of the virama) written under the sign for the first member of a conjunct: क्र . प्र, द्र, व्र

Similarly, in Devanagari, when ṭ is served as the first member of a conjunct, the sound is indicated by a small hook placed on the top of the rekha for the second consonant: क, ठ, ड, म. This hook is deferred until after any matra written to the right side of the conjunct like ठी, मी.
In third type of conjunct, a form of *ਮ* is subjoined. For example: म in Gurmukhi is written as स्व in Devanagari. Similarly मू (svar) in Gurmukhi is written as स्वर (svar) in Devanagari.

Several Devanagari conjuncts are so irregular as to prelude the immediate recognition of their components. The most important of these are क्त, ध, ढ, प, त. The consonant श has a special combining form श that is often used in place of श in some clusters. (e.g. श, ं) . Slightly irregular conjuncts exist in which ह stand as the first element (e.g. हन, हन, हन, हन).

3.2.3.4 Geminate (Doubled) Consonants:

In Gurmukhi, gemination is written by the sign (addak) above and before the consonant to be doubled. In Devanagri, doubled consonant cluster, gemination is written by writing the first component of the consonant cluster as the truncate form of the consonant (which is frequently built from the independent version of the latter consonant by the deletion of the vertical bar that appears on the right side of many Devanagri characters and the second component of the consonant cluster is, the unaltered full symbol for the second consonant. For example: पक्की (pakki) (पक्की (pakki) in Gurmukhi), कच्चा (kaccā) (कच्चा (kaccā) in Gurmukhi). Similarly, in Gurmukhi, clusters of unaspirated stop plus homorganic aspirate stops are written by use of
(addak) before the letter for the aspirate. In Devanagari, this cluster is written with the short form of unaspirated stop plus full form of homorganic aspirate stop. For example: अच्छा (अਕਾ in Gurmukhi), पक्खी (ਪਕਾ in Gurmukhi).

In a small number of cases, the components of a consonant are string out in a horizontal line (e.g. न्न), arranged vertically or juxtaposed in some less regular manner (द्र, ढ). Similarly in the Gurmukhi two geminates /nn/ and /mm/ are written with /tippi/ (ੂੰ). For example: ਣੂੰਢ਱ (ਣੂੰਢ਱ in Devanagari), ਭੂੰਢ (ਪੁੰਨਨ in Devanagari). It must be noted that there are no short forms in Gurmukhi like in Devanagiri for consonants. So, while transliterating the short form of Hindi consonant, it is transliterated into full form of that consonant in Gurmukhi like मग्न (magn) in Devanagari will be transliterated into ਭੂੰਢ (magn).

### 3.2.4 Vowels:

Both the Scripts possess two different forms for each of the vowels - Full form and short form.

#### 3.2.4.1 Full form:
In Devanagari, a full form is employed for a vowel that does not immediately follow a consonant or consonant cluster, i.e. in word-initial position or when the second of a sequence of vowels. Whereas in Gurmukhi, when a vowel is not preceded by a consonant, it is written with one of the three vowel bearers - consonant like sign – ट्र, ण, ध indicating the absence of consonant.

### 3.2.4.2 Short form (or matra):

In Devanagari, short form is used when the vowel immediately follows a consonant or consonant cluster. These short forms consist of lines, hooks or combination of both above, below or to the side of the consonantal characters. These vowels are written around (that is, below, above, to the right, and to the left) the consonant signs.

In Gurmukhi, there are 10 vowel characters, 9 vowel symbols, 2 symbols for nasal sounds and 1 symbol that duplicates the sound of a consonant (Malik 2006, Malik 2005) Whereas in Devanagari, there are 11 vowel characters, 10 vowel symbols, 2 symbols for nasal sounds.

Following table shows both the above form of vowels for both the scripts and their correspondence in the Devanagari and Gurmukhi scripts:

**Table 3.4: Vowels in Devanagari and Gurmukhi**

<table>
<thead>
<tr>
<th>Devanagari</th>
<th>Gurmukhi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Short Form</td>
<td>Full Form</td>
</tr>
<tr>
<td>No Sign</td>
<td>अ (a)</td>
</tr>
<tr>
<td>आ (ā)</td>
<td>आ (ā)</td>
</tr>
<tr>
<td>इ (i)</td>
<td>इ (i)</td>
</tr>
<tr>
<td>औ (ī)</td>
<td>औ (ī)</td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
3.2.4.3 Inherent ‘a’:

One vowel, ‘a’ has no special short form. The absence of a matra adjacent to a consonant suffices to indicate the presence of this vowel. At the end of a word, the inherent ‘a’ is not normally vocalized.

3.2.4.4 Nasalized vowels:

The two signs are used for nasalization. In Devanagari, anusvara (ं m) and anunasika (◌ँ m) also called candrabindu. Indian grammarians have formulated elaborated rules describing when each of these is used. In practice, the distinction between the two notations is often not observed. The first of these, anusvara is always used when the vowel marking (whether short or long form) protrudes above the rekha (e.g. ईं, ऐं, कों, भौं). With other vowel
signs, both *anusvara* and *anunasika* can be used (e.g. मुं (muṃh) / मुँ (muṃh), अंख (āṅkh) / अङ्क (āṅkh)), although some writers take care to consistently employ only anusvara in all contexts. Whereas in Gurmukhi, bindu (ੱ) is used with ਆ, ਚੀ, ਬੋ, ਬੋ, ਓ, ਓ, ਦੇ, ਦੇ, ਦੇ, ਦੇ, ਦੇ, and tippi (ੱ) is used with ਠੱ, ਠੱ, ਠੱ, ਠੱ, ਠੱ, ਠੱ.

3.2.5 Punctuation Marks:

Only viraama (।) or a double vertical line (॥) was used in traditional writing for marking end of sentence and the end of a verse respectively for both Devanagari and Gurmukhi scripts. In modern writings, period, comma, hyphen, semicolon, exclamation sign, question mark and dash have also been used. In the ancient Punjabi, the use of double dandi was customary at the end of the sentences but in contemporary Punjabi, only single Dandi is used.

3.2.6 Abbreviation:

Abbreviations are formed in Hindi by the use of either a small circle (॰) or a dot after the first syllable of the word to be abbreviated: प्रो॰ (prō.), डा॰ (ḍā.), ई॰ (ī.),
म (pū) whereas in Gurmukhi Script, sign (;) is used to mark abbreviation like म (prō), ध (ḍā).

3.2.7 Numerals:

Following chart shows the correspondence between the numerals of both the scripts:

Table 3.5: Numerals in Devanagari and Gurmukhi

<table>
<thead>
<tr>
<th>Devanagari</th>
<th>Gurmukhi</th>
</tr>
</thead>
<tbody>
<tr>
<td>०</td>
<td>ੀ</td>
</tr>
<tr>
<td>१</td>
<td>ੁ</td>
</tr>
<tr>
<td>२</td>
<td>ੂ</td>
</tr>
<tr>
<td>३</td>
<td>੃</td>
</tr>
<tr>
<td>४</td>
<td>੄</td>
</tr>
<tr>
<td>५</td>
<td>੅</td>
</tr>
<tr>
<td>६</td>
<td>੆</td>
</tr>
<tr>
<td>७</td>
<td>ੇ</td>
</tr>
<tr>
<td>८</td>
<td>ੈ</td>
</tr>
<tr>
<td>९</td>
<td>੉</td>
</tr>
</tbody>
</table>

3.2.8 Alphabetic Order:

The alphabetic order of Devanagari is a model of logic and rational design, reflecting a keen understanding of the phonetic properties of the sounds designated by the various characters in the system. In Devanagari, vowels
precede consonants with the latter divided up into groups containing stops and nasals, semi vowels, sibilants, and h respectively.

The full alphabetic order of Devanagari as used for Hindi is as follows:

अ आ इ ई उ ऊ ऋ ऌ ए ऐ ओ औ क (क) ख (ख) ग (ग) घ छ ज झ ञ ट ठ ड ढ ण त थ द ध न प फ भ म य र ल व श ष स ह

The full alphabetic order of Gurmukhi as used for Punjabi is as follows:

ਅ ਆ ਇ ਈ ਉ ਊ ਋ ਌ ਏ ਐ ਓ ਔ ਕ (ਕ) ਖ (ਖ) ਗ (ਗ) ਘ ਛ ਜ ਝ ਞ ਟ ਠ ਡ ਢ ਣ ਤ ਥ ਦ ਧ ਨ ਪ ਫ ਭ ਮ ਯ ਰ ਲ ਵ ਸ਼ ਷ ਸ ਹ

In Hindi, sequence under each consonants is the letter without any symbol, then followed by vowel symbols ा, ि, ी, ु, ू, े, ै, ो, ौ

In Punjabi, Sequence under each consonants is the letter without any symbol, then followed by vowel symbols ਾ, ਿ, ੀ, ੁ, ੂ, ੇ, ੈ, ੋ, ੌ

3.3. Comparison between Hindi and Punjabi on the basis of grammar [87-102]

3.3.1 Nouns

Nouns in Hindi and Punjabi are highly inflected. Hindi and Punjabi both have two genders (masculine and feminine), two numbers (singular and plural) whereas Hindi has three cases (direct, oblique, and vocative) and Punjabi has five cases (direct, oblique, vocative, ablative, and locative/instrumental). The
latter two cases in Punjabi are essentially now vestigial: the ablative occurs only in the singular, in free variation with oblique case plus ablative postposition, and the locative/instrumental is confined to set adverbial expressions.

Nouns in Hindi can be further divided into declensional subtypes, Class I (marked/definite) and Class II (unmarked/indefinite), with the basic difference being that the former has characteristic terminations in the direct singular while the later does not. While Punjabi Nouns may be further divided into extended and unextended declensional subtypes, with the former characteristically consisting of masculines ending in unaccented -ā and feminines in -ī.

### 3.3.2 Adjectives

In Hindi and Punjabi both, adjectives are of two basic kinds, declinable/inflected and indeclinable/uninflected. Declinable adjectives agree with the nouns they modify in gender (masculine vs. feminine), number (singular vs. plural), and case (direct vs. oblique). Indeclinable adjectives possess but a single form when modifying nouns of different genders, numbers, or cases. Indeclinable adjectives are completely invariable, and can end in either consonants or vowels (including ā and ī). These adjectives do not end in any characteristics sound or series of sounds.

<table>
<thead>
<tr>
<th>Table 3.6 : Declinable and Indeclinable Hindi Adjectives</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Declinable</strong></td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
3.3.3 Postpositions

Postpositions denote the relation of noun, pronoun, or verb with the other components of sentence. It is the use of postpositions with a noun or verb that necessitates the noun or verb taking the oblique case. Hindi and Punjabi both have core and compound postpositions. Core postpositions are also known as one word primary postpositions. For example: Some of the core postpositions in Hindi are का, की, के, को, ने, पर, में, तक, से and in Punjabi are ठठ, ठस्ख, ठस्ख, ठस्ख, ठस्ख, ठस्ख, ठस्ख, ठस्ख. Compound postpositions are composed of the genitive primary postposition plus an adverb. These postpositions follow their oblique targets either directly or with the inflected genitive linker. For example: Some of the compound postpositions in Hindi are के लरए, के साथ, के साभने, से ऩहरे and in Punjabi are ठस्ख लस्ख, ठस्ख लस्ख, ठस्ख लस्ख etc.

3.3.4 Pronouns

Hindi and Punjabi languages both have personal pronouns for the first and second persons, while for the third person demonstratives are used, which Language in India www.languageinindia.com 704

10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
can be categorized as proximate and non-proximate. Pronouns distinguish three persons (first, second, and third), two numbers (singular and plural), and two cases (direct and oblique), though not gender.

Table 3.7: Hindi and Punjabi Pronouns

<table>
<thead>
<tr>
<th>Pronouns</th>
<th>First Person</th>
<th>Second Person</th>
<th>Third Person (Proximate)</th>
<th>Third Person (non-proximate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hindi</td>
<td>मैं (maim), हम (ham)</td>
<td>तू (tū), तुम (tum), आप (āp)</td>
<td>यह (yah), ये (yē)</td>
<td>कह (vah), वे (vē)</td>
</tr>
<tr>
<td>Punjabi</td>
<td>ਮੈਂ (maim), ਅਸੀਮ (asīm)</td>
<td>ਤੂਂ (tūm), ਤੁਸੀਮ (tusīm)</td>
<td>ਇਹ (ih)</td>
<td>ਉਹ (uh)</td>
</tr>
</tbody>
</table>

3.3.5 Verbs

In both Hindi and Punjabi, the major grammatical categories that structure the verbal system are those of aspect and tense. The term aspect is to be understood as indicating the nature of the action of a verb as to its beginning, duration, completion, or repetition, but without reference to its position in time. There are three grammatical aspects, the habitual, the progressive (or continuous), and the perfective. Verbal forms indicating one of these aspects is usually further specified for one of four tenses, i.e., the present, past, presumptive, and subjunctive. Like the nominal system, the Hindi and Punjabi
verbs involve successive layers of (inflectional) elements to the right of the lexical base.

Compound verbs, a highly visible feature of Punjabi and Hindi grammar, consist of a verbal stem plus an auxiliary verb. The auxiliary (variously called "subsidiary", "explicator verb", and "vector") loses its own independent meaning and instead "lends a certain shade of meaning" to the main/stem verb, which "comprises the lexical core of the compound". While most verb can act as a main verb, there is a limited set of productive auxiliaries. For example, Some of verbs in Hindi are  रहना (rahnā), होना (hōnā), जाना (jānā), देना (dēnā) and in Punjabi are  रहित (rahiṅā), हेत (hōṅā), जैत (jāṅā), देह (dēṅā) etc.

3.3.6 Sentence Structure

Hindi and Punjabi both are SOV (Subject Object Verb) and free order languages. Structurally both Hindi and Punjabi languages are same. In both languages, sentence is comprised of Subject and Predicate. In both languages, the basic elements are Kaarakā. Both have eight numbers of Kaarakā which by combining with each other create a sentence. The general sequence for transitive Sentence is Karta, Karam, Kria e.g. गणेश खेत में सोता है (gaṇēsh khēt mēṃ sōtā hai) and for intransitive sentence is karta, kriya e.g. गणेश भागा (gaṇēsh bhāgā). In both languages the relation between kaarkā's
are shown by postpositions. Total eight part-of-speeches are recognized in both Hindi and Punjabi. Beside this, both have same types of Nouns, Genders, Number, Persons, Tenses and Cases.

3.3.7 Vocabulary

Joshan and Lehal [84] carried out an experiment to find out the total number of words which use the same alphabets and vowel/vowel sounds and convey the same meaning in both languages. Results showed that about 8% of source language words come under this category. This provides an idea of the overlap of vocabulary across languages. Hence for this study, it strengthens the fact of close relationship between Hindi and Punjabi languages. Moreover, it gives boost to the idea of using transliteration of source text as last option.

3.4 Comparison of Hindi and Punjabi from Machine Translation point of view [87-102]

3.4.1 Language Structure (Syntactic Vs Analytic)

Hindi is both analytic and syntactic in nature. Thus, it is not a purely analytic in nature. It may cause a problem while translating text from Hindi to Punjabi. It can lead to an unacceptable output if left un-dealt.

3.4.2 Ambiguity
Ambiguity is one of the major NLP problems which have been a great challenge for computational linguists. In general, people are unaware of the ambiguities in the language they use because they are very good at resolving them using context and their knowledge of the world. But computer systems do not have this knowledge, and consequently do not do a good job of making use of the context.

Something is ambiguous when it can be understood in two or more possible ways or when it has more than one meaning. If the ambiguity is in a sentence or clause, it is called structural (syntactic) ambiguity. Following example shows the structural ambiguity in Hindi:

परमोद ने खाते हुए चोर को पकड़ा (parmōd nē khātē huē cōr kō pakṛā)

This sentence can be interpreted in two ways viz. Parmod caught the thief while eating or Parmod caught the thief when the thief was eating.

Lexical ambiguity also known as word level ambiguity is a problem in translating Hindi to Punjabi. In Hindi, lexical ambiguity has been found in Nouns, Verbs, and Postpositions etc. The postposition से in Hindi can be translated into number of Punjabi postpositions like ਨੋਂ, ਢੌੰ, ਨਿਵੇ, ਬਨਵੇ and ਸੱਸ depending upon the usage of से in the sentence. Similarly in Verb, like जाना can be translated into ਖ਼ਬ਼ੂਲ and ਖ਼ਬ਼ੂਲਾ. Similarly in case of proper nouns, like प्रकाश (prakāsh) can be translated into ਪ੍ਰਕਾਸ (prakāsh) or ਚਨੇ (cānaṇ).
To illustrate more, consider the following sentence:

राम आम खा रहा है। (rām ām khā rahā hai)

In the above example, word आम in the sentence is lexically ambiguous. Its meaning can be interpreted in two ways – mango (a fruit) and usual (an adjective) as in following examples:

**Usage as Noun:** तोता पेड़ पर बैठकर आम खा रहा है। (tōtā pēḍa par baīṭhakar ām khā rahā hai)

**Usage as Adjective:** ऐसे चोरों से मिलना आम बात है जो चोरी के खिलाफ उपदेश देते हैं।

(aisē cōrōṃ se milnā ām bāt hai jō cōrī kē khailāpha updēś dētē haiṃ)

### 3.4.3 Gender disagreement

During translation, sometimes correct gender of a word is not reflected in the translated language and it causes gender disagreement with verb/postposition in the target language. For example, if we translate the sentence उसको किताब चाहिए (uskō kitāb cāhiē) using direct approach, it will be translated to उसके मिलने किताब काहिदा हाई (usnūṃ kitāb cāhidā hai). Here the word किताब (kitāb) is feminine in nature and thus translation of verb चाहिए (cāhiē) in the sentence
must agree with the feminine nature of किताब (kitāb) and thus be translated into चौहीठी छै (cāhīḍī hai).

3.4.4 Problems in Identifying Proper Nouns

The problem arises when a word in Hindi Sentence which is used as proper name of a person, is translated by the system instead of transliterating it. Such words are required to transliterate rather than translation. For example consider following sentences

दीपक गोयल कहाँ हैं? (dīpak gōyal kahāṃ hai?)

The word दीपक (dīpak) can be translated to दीवा (dīvā). But in this sentence, the word दीपक (dīpak) has been used as a proper noun and thus, must be transliterated to दीपक (dīpak) instead of translated to दीवा (dīvā). This problem is also known as Named Entity Recognition. Thus, Named Entity Recognition (NER) problem is a subtask of information extraction that seeks to locate and classify atomic elements in text into predefined categories such as the names of persons, organizations, locations, expressions of times, quantities, monetary values, percentages, etc.

3.4.5 Problem related to Collocations
Collocation is two or more consecutive words with a special behavior. (Choueka: 1988). Collocation means those combinations of words in Hindi that cannot be translated word to word and such combinations of words have different word in group rather than their individual. These groups of words have a special behavior. The meaning of the collocation can not be predicted from its parts, there is usually an element of meaning added to the parts of collocation. For example, the collocation उत्तर प्रदेश (uttar pradēš) if translated word to word, will be translated as खबरः राज (javāb rāj) But it must be translated as उत्तर प्रदेश (uttar pradēš). Thus, special attention is needed for such combinations of words in Hindi Language.

3.4.6 Problems related to Foreign Words

Modern Hindi includes number of foreign words that are adopted from other languages. These words do not have any meaning in Hindi language and is propagated as such to Punjabi language while translating. So, these words are treated as unknown words and must be transliterated. For example: क्रिकेट (krikēṭ), मैच (maic), जाकेट (jākēṭ) etc.

3.4.7 Spelling variations
The Cambridge Dictionary defines spelling as ‘forming words with the correct letters in the correct order’, or the ability to do this where variation is ‘difference’ or ‘deviation’ in the structure. The existence of the variants does not make much of the difference to the common person who is using the language because it does not come on the way of proper communication of the message but it is much important in case of Machine Translation. The major reasons for spelling variations in language can be attributed to the phonetic nature of Indian languages and multiple dialects, transliteration of proper names, words borrowed from foreign languages, and the phonetic variety in Indian language alphabet. [105]

For example, Following are the possible spelling variations for the Hindi word अंग्रेजी (aṅgrējī):

अँग्रेजी, अंगयेजी, अन्ग्ग्रेजी, अँगरेजी, अंग्रेजी, अंग्रेजी

3.5 Conclusion

In this chapter we have tried to compare Hindi and Punjabi language from the point of view of orthography, grammar and Machine Translation. This study is by no means an exhaustive one. This study was primarily aimed at knowing the closeness between both the languages and thus, to find the appropriate approach for the development of Machine Translation.

We call a language pair to be closely related if the languages have the grammar that is close in structure, contain similar constructs having almost
same semantics, and share a great deal of lexicon. By closely related languages, we also mean inductively and morphosyntactically similar languages. Some linguist define closeness between the languages on the basis of features viz. common root, similar alphabets, similar verb patterns, structural similarity, similar grammar, similar religio-cultural and demograpohic contexts and references, a similar clearly displayed ability to blend with foreign tongues. Generally, such languages have originated from the same source and spoken in the areas in close proximity.

Hindi and Punjabi belong to same sub group of the Indo European family, thus are sibling languages. We have also observed that Hindi and Punjabi languages share all features of closely related languages. For such closely related sibling languages, effective translation can be achieved by word-for-word translation (Hajic et al., 2000) [90]. Thus, it is concluded that direct Machine Translation approach is promising for closely related languages Hindi and Punjabi.
Chapter 4

Pre Processing Phase

The present and the next chapter discuss the design and implementation of the algorithms and structures that formulate our Hindi to Punjabi Machine Translation system. For all the activities, the design of the databases used, if any, along with some sample entries from the databases and the approach followed for that activity have been discussed in detail. While describing the design of the databases used, only the fields or databases directly concerned with performing the activity under consideration have been provided. There may be some additional fields or databases used for proper functioning of this Machine Translation system but have virtually no impact on describing the approach, thus, description of such databases or fields have been avoided. All the activities of this Machine Translation system have been implemented in ASP.Net and their databases are in the MS-Access with Hindi and Punjabi text in Unicode format. This Machine Translation system accepts Hindi text as input and provides output in Gurmukhi script in Unicode.

This chapter provides first activity pre-processing of our Machine Translation system. The remaining activities have been detailed in the next chapters. Chapter 1 has already presented the complete design of this Machine Translation system.

4.1 Introduction

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
The preprocessing stage is a collection of operations that are applied on input data to make it processable by the translation engine. In the first phase of Machine Translation system, various activities incorporated include text normalization, replacing collocations and replacing proper nouns. Figure 4.1 presents the design of this pre-processing system in more detail.

**Figure 4.1: Pre-processing System Design**

The four sub-activities of pre-processing system shown in Figure 4.1 are explained in the following sub-sections.

### 4.2 Text Normalization

Spelling conventions are an important feature of any language that is written.

The Cambridge Dictionary defines spelling as ‘forming words with the correct
letters in the correct order', or the ability to do this where variation is 'difference' or 'deviation' in the structure. The existence of the variants does not make much of the difference to the common person who is using the language because it does not come on the way of proper communication of the message but it is much important in case of Machine Translation. This sub phase works on spelling standardization issues, thereby resulting in multiple spelling variants for the same word. The major reasons for this phenomenon can be attributed to the phonetic nature of Indian languages and multiple dialects, transliteration of proper names, words borrowed from foreign languages, and the phonetic variety in Indian language alphabet. The variety in the alphabet, different dialects and influence of foreign languages has resulted in spelling variations of the same word. Such variations sometimes can be treated as errors in writing. For example, Following are the possible spelling variations for the Hindi word अंग्रेजी (aṅgrējī):

अँग्रेजी, अंग्रेजी, अन्ग्रेजी, अंग्रेजी, अंग्रेजी, अंग्रेजी

But out of these above possible spelling variants, only following are found in the Hindi corpus along with their frequency of occurrence:

**Table 4.1: Frequency of Occurrence for Possible Spelling Variants of Word अंग्रेजी**

<table>
<thead>
<tr>
<th>Word अंग्रेजी</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>अंग्रेजी (aṅgrējī)</td>
<td>87.017%</td>
</tr>
<tr>
<td>अंग्रेजी (aṅgrējī)</td>
<td>8.037%</td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
Following rules specific to Hindi language have been framed which can handle such variations, which could result in more precise performance and for making the input text normalized for better accuracy:

**Table 4.2: Text Normalization Rules**

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Rule</th>
<th>Example</th>
</tr>
</thead>
</table>
| 1.       | Chandrabindu (a half-moon with a dot) and bindu (a dot on top of alphabet) can be used interchangeably. | (i) अंग्रेज़ (aṅgrēja), अंग्रेज़ (aṅgrēja)  
(ii) लौँच (lāñc), लौँच (lāñc) |
| 2.       | There are five consonant characters with nukta (a dot under consonant) viz. क, ख, ग, ज, फ. With this rule, all consonants with nuktas and these consonants without nukta will be considered same. | (i)अंग्रेज़(aṅgrēja), अंग्रेज़ (aṅgrēj)  
(ii) फोटो (phōṭō), फोटो (phaōṭō)  
(iii) तेज (tēj), तेज़ (tēja) |
| 3.       | Hindi and many other Indian languages face the problems of 'schwa' (the default vowel 'a' that occurs with every consonant) deletion. Lots of spelling variations occur due to 'schwa' deletion. In order to normalize such words we delete all the halanth characters in the given word to generate spelling variant. | (i)भगवान(bhagvān), भगवान (bhagvān)  
(ii)अगरज (agaraj), अगरज (agaraj)  
(iii)अक्सर (aksar), अक्सर(akasar) |
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>4.</strong></td>
<td>‘Bindu’ and ‘न्’ can be used interchangeably.</td>
</tr>
<tr>
<td></td>
<td>(i) कन्थ (kanṭh), कंठ (kanṭh)</td>
</tr>
<tr>
<td><strong>5.</strong></td>
<td>‘Bindu’ and ‘म्’ can be used interchangeably for words having ‘म्’ before the labial consonants like प, ब, फ, म, व in the word.</td>
</tr>
</tbody>
</table>
|   | (i) अम्बु (ambu), अंबु (ambu)  
(ii) पम्प (pamp), पंप (pamp) |
| **6.** | There is one supplemental sound occasionally encountered in Hindi. This is the ‘Visarga’, noted in devanagari by the sign (‘ं’). This sign appears only in tatsama vocabulary items. The words having sign (‘ं’) can also be written without it and is treated equivalent. |
|   | (i) अक्रमत (akramat), अक्रमत (akramat)  
(ii) अंतत (antat), अंतत (antat) |
| **7.** | Sometimes in place of ‘ङ्’/‘ण्’/ ‘ञ’ in the words, Chandrabindu (a half-moon with a dot) / bindu (a dot on top of alphabet) can be used and are equally correct. But it is very rare. |
|   | (i) गाँगा (gāṅga), गंगा (gāṅga)  
(ii) ब्राण्ड (brāṇḍ), ब्रॉड (brāṇḍ)  
(iii) पणजा (paṇjā), पंजा (paṇjā) |
| **8.** | ‘ई’ and ‘यी’ can be used |
|   | (i) नाई (nai), नयी (nayī) |
9. ए' and 'ये' can be used interchangeably in words.

(i) लिए (liē), लिये (liyē)

Analysis:

An exhaustive analysis has been done on large Hindi corpus collected from number of online resources for finding most useful rules among above mentioned rules. The Hindi Corpus used for analysis consists of about 1,00,000 words.

As it has been mentioned earlier that there can be a large number of possible spelling variations for a particular word depending upon the above rules, but in real data, among these variations, very less spelling variations are found. Only 1.492% words show the variations in their spellings. Following Table shows that out of these 1.492% words, percentage of words having one, two or three variations:

<table>
<thead>
<tr>
<th>Number of variants</th>
<th>Words (%)</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>99.985</td>
<td>जरूयत (jarūrat), जरुयत (jarurat)</td>
</tr>
<tr>
<td>2</td>
<td>0.010</td>
<td>अँग्रेजी (aṅgrejī), अंग्रेजी (aṅgrejī)</td>
</tr>
<tr>
<td>3</td>
<td>0.005</td>
<td>फ़रक (phark), फरक (phark), फ़रक (phark)</td>
</tr>
</tbody>
</table>

Table 4.3: % Word Occurrence with Spelling Variation Count
Thus, above table represents that, the variations found for majority of the words is just 1 and in worst case, it can go up to 3. And no case has been found with more than three spelling variants.

Following graph represents the importance and usage of different rules during analysis:

![Graph showing the analysis of % Usage of Various Text Normalization Rules]

**Figure 4.2: Analysis of % Usage of Various Text Normalization Rules**

The above graph shows that Rule No 1 and 2 have maximum applicability and rests of the rule are seldom used. Rules other than 1 and 2 are also contributing in standardization but their role is limited.

It is found that only 7.45% text was standardized using the above rules. Following graph shows the analysis of the contribution of various rules Vs the number of words standardized:
Figure 4.2: Analysis of Contribution of Text Normalization Rules

Majority of the standardization is done on the basis of the rules 1 and 2. Rest of the rules play very limited roles.

Database design:

Table 4.4 carries the design of the database used for storing information about text normalization.

Table 4.4: Text Normalization Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nonstandardWord</td>
<td>Stores the non standard Hindi words</td>
</tr>
<tr>
<td>nswFreq</td>
<td>The frequency of the non standard word in the corpus analysed</td>
</tr>
<tr>
<td>standardWord</td>
<td>Hindi Word with standard spellings</td>
</tr>
<tr>
<td>swFreq</td>
<td>The frequency of the standard word in the corpus analysed</td>
</tr>
</tbody>
</table>

Sample database entries:

Table 4.5: Sample Entries of Text Normalization Database
<table>
<thead>
<tr>
<th>nonstandardWord</th>
<th>nswFreq</th>
<th>standardWord</th>
<th>swFreq</th>
</tr>
</thead>
<tbody>
<tr>
<td>फिल्म (phailm)</td>
<td>104</td>
<td>फिल्म (philm)</td>
<td>2165</td>
</tr>
<tr>
<td>हालांकि (hālārkī)</td>
<td>1486</td>
<td>हालांकि (hālārkī)</td>
<td>3120</td>
</tr>
<tr>
<td>हाँ (hāṃ)</td>
<td>2045</td>
<td>हाँ (hāṃ)</td>
<td>4513</td>
</tr>
<tr>
<td>मौका (maukaā)</td>
<td>700</td>
<td>मौका (maukā)</td>
<td>1580</td>
</tr>
<tr>
<td>किराए (kirāē)</td>
<td>600</td>
<td>किराये (kirāyē)</td>
<td>3411</td>
</tr>
<tr>
<td>हूं (hūṃ)</td>
<td>985</td>
<td>हूं (hūṃ)</td>
<td>24910</td>
</tr>
<tr>
<td>एंड (ēṇḍ)</td>
<td>100</td>
<td>एंड (ēṇḍ)</td>
<td>1853</td>
</tr>
<tr>
<td>स्थाई (sthāī)</td>
<td>6</td>
<td>स्थायी (sthāyī)</td>
<td>20</td>
</tr>
<tr>
<td>इंटरनेशनल (intranēshnal)</td>
<td>2</td>
<td>इंटरनेशनल (intranēshnal)</td>
<td>16</td>
</tr>
<tr>
<td>रविन्द्र (ravindr)</td>
<td>1</td>
<td>रविन्द्र (ravindr)</td>
<td>4</td>
</tr>
</tbody>
</table>

**Our Approach:** The small offline module has been developed to generate the database for standardization. The module starts applying the rules discussed above, to the Hindi corpus collected from various sources like Hindi newspaper websites, various literatures available online etc. Thus, storing standard and non standard words extracted during corpus analysis along with their frequency into database. Then, the word having the maximum frequency among its spelling variant words is considered to be standard one. In future, this standard word may also be replaced with some of its other variants if frequency of the new spelling variant exceeds the current standard one. For example: the spelling variations हालांकि (hālārkī) and हालांकि (hālārkī) are equally correct. If in some input text one variation is present more number of times than other, it can become standard one and vice versa. Thus,
database is always in updated mode to accept changes for the existing entries also. The spelling variant(s) among non standards having frequency zero is omitted as they do not have existence in the real text. In this way, only those spelling variations are kept in the database that actually exists in Hindi Vocabulary. In this way, database is generated and presently database consists of 2,00,450 entries. Once this database is generated, during the preprocessing phase, the table lookup is done to replace the non standards words present in the database with the standard ones.

4.3 Replacing Collocations

After passing the input text through text normalization, the text passes through this Collocation replacement sub phase of Pre-processing phase. Collocation is two or more consecutive words with a special behavior. (Choueka :1988). Collocation means those combinations of words in Hindi that cannot be translated word to word and such combinations of words have different word in group rather that their individual. These groups of words have a special behavior. The meaning of the collocation can not be predicted from its parts, there is usually an element of meaning added to the parts of collocation. For example, the collocation उत्तर प्रदेश (uttar pradēsh) if translated word to word, will be translated as जवाब राज (javāb rāj) but it must be translated as उत्तर प्रदेश (uttar pradēsh).
Related works:

Collocation has long been studied by lexicographers and linguists in various ways. Most collocation extraction methods are based on exploiting the various idiosyncrasies exhibited by collocations. The variation in statistical distributional characteristics has been widely employed to test for evidence of a collocation. Point wise Mutual Information is one of the earliest measures of association used for collocations [104]. Word association has also been measured using measures like Jaccard, Odds Ratio, etc [105]. Classical statistical hypothesis tests like Chisquare test, t-test, z-test, Log Likelihood Ratio [106] have also been employed to decide whether the constituents of a collocation are independent of each other. The variation in positional distribution of words in a collocation has also been used to identify significant collocations [107]. Lin [108] and Cruys et.al [109] have used the principle of substitution to extract institutionalized collocations. They measure the difference between the distributional characteristics of the collocation and other similar collocations obtained by lexical substitution. While Lin uses PMI as the base association score, Cruys et.al. [109] use a strength of association measure motivated by the idea of selectional preference of a constituent word for another. Fazly et.al. [110] extract collocation by exploiting their syntactic fixedness. Katz [111] and Baldwin [112] use the context as a bag of words and build context vectors for representing collocations and their constituents. Comparison of the collocation and constituent vectors helps determine if the collocation is non-compositional. Moiron et.al. [113] have used the idea of
translation ambiguity to extract non-compositional MWEs. The noncompositional collocations will have more translation candidates on account of more uncertainty in translation. This uncertainty is measured as translational entropy. Language modeling has been used to extract domain specific phrases, by comparing the distribution of collocations in a general and domain-specific corpus [114]. All the measures mentioned above have modeled the problem as a ranking problem, where the collocations more likely to be MWEs are ranked higher. If an annotated training set is available, the MWE extraction problem can be set up as a classification problem [115]. For Indian languages, automated collocation extraction work has been limited. In fact, both of the existing works [115-117] use some kind of English translation for extracting Hindi collocations. Mukerjee et.al. [116] have used parallel corpus alignment and POS tag projection with parallel English corpus to extract complex predicates. Venkatapathy et.al. [115] use a classification based approach for extracting N-V collocations for Hindi. They use identity of the verb, semantic type of the object, case marker with the object, similarity of the verb form of the object with the verb-object pair under consideration etc. as features in a MaxEnt classifier. Thus, there are number of approaches for extracting Collocations from the corpus Like Frequency Method, Mean and Variance, Hypothesis Testing, t-test, Pearson’s Chi-Square Test, Likelihood Ratio and Point wise Mutual Information.

Our Approach for Extracting Collocations:

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
Our focus is on extracting collocations which can be used for translation into Punjabi and above mentioned approaches are not suitable in our case. We have developed an offline module using t-test for automatically extracting the collocations from the Hindi Corpus. The steps performed for extracting the collocations using the t-test are as follows:

1. Extract all the unigrams, bigrams and trigram from the corpus along with their frequencies of their occurrence in the corpus and store into a database table tbl_Unigram, tbl_bigram, tbl_trigram respectively.

2. Combine all bigrams and their frequencies with their corresponding unigrams and their particular frequencies into the database table tbl_unibi.

3. Combine all trigrams and their frequencies with their corresponding unigrams and their particular frequencies into the database table tbl_unitri.

4. For each entry in table tbl_unibi, Expected mean ($\mu$) is calculated using the formula $P(\text{bigram}) = P(\text{unigram1})P(\text{unigram2})$. Where $P(\text{unigrami}) = \text{Frequency of unigram} / \text{total no of tokens in analyzed corpus}$.

5. For each entry in table tbl_unibi, Observed mean is calculated by dividing the frequency of the particular bigram with the total number of bigrams found during corpus analysis.

6. The variance ($s^2$) is equal to the observed frequency.

7. Now apply the formula $t = (x - \mu)/\sqrt{s^2/N}$. Where $N$ is the total number of bigrams found during corpus analysis.
8. Apply the steps 4 to 7 for trigrams.

9. After applying t-test to all bigrams and trigrams, there are many bigrams and trigrams which are not good candidates for collocations. We removed all the analyzed bigrams and trigrams whose t-value is less than 2.576 (standard value provided by t-test).

The accuracy of the results for collocation extraction using t-test is not accurate and includes number of such bigrams and trigrams that are not actually collocations. Thus, manually such entries were removed and actual collocations were further extracted. The correct corresponding Punjabi translation for each extracted collocation is stored in the collocation table of the database. The collocation table of the database consists of 5000 such entries.

**Database design:** Table 4.6 carries the design of the database used for storing information about collocations.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collocation</td>
<td>Stores the Hindi collocation</td>
</tr>
<tr>
<td>punjabiTranslation</td>
<td>Stores Punjabi translation for corresponding collocation</td>
</tr>
</tbody>
</table>

**Sample database entries:**

<table>
<thead>
<tr>
<th>Collocation</th>
<th>punjabiTranslation</th>
</tr>
</thead>
<tbody>
<tr>
<td>आऩ को (āp kō)</td>
<td>टुहानूँ (tuhānūṁ)</td>
</tr>
</tbody>
</table>
Our approach for replacement of collocation:

In this sub phase, the normalized input text is analyzed. Each collocation in the database found in the input text will be replaced with the Punjabi translation of the corresponding collocation. This step helps a lot in increasing the translation accuracy of the system. It is found that when tested on a corpus containing about 1,00,000 words, only 0.001% collocations were found and replaced during the translation.

4.4 Replacing Proper Nouns

A great proposition of unseen words includes proper nouns like personal, days of month, days of week, country names, city names, bank names, organization names, ocean names, river names, university names etc. and if translated word to word, their meaning is changed. If the meaning is not affected, even though this step fastens the translation process. Once these words are recognized and stored into the proper noun database, there is no need to decide about their translation or transliteration every time in the case of presence of such words in input text for translation. This gazetteer makes
the translation accurate and fast. This list is self growing during each translation. Thus, to process this sub phase, the system requires a proper noun gazetteer that has been compiled offline. For this task, we have developed an offline module to extract proper nouns from the corpus based on some rules. Following sections will explain the process of preparing the proper noun gazetteer and then the use of this gazetteer in pre-processing phase.

4.4.1 Compilation of Proper Nouns Gazetteer:
The gazetteer has been prepared using two approaches. One approach is through an offline module and another is through manual collection from various sources available online. The offline module further needs two databases containing titles like श्री (shrī), श्रीमती (shrīmtī), प्रो (prō) etc. and surnames like अवस्थी (avsthī), आहुजा (āhūjā) etc. The database design of these databases has been explained in following sections. These databases have been prepared manually by collecting the data from various resources. The offline module accepts the Hindi text, applies various rules on it, extracts the proper names, and stores it in proper noun database. Following are the rules for extraction of proper nouns through offline module:

Rule 1: It checks whether the token from input text is matched with any entry in titles database, then the token next to current one is a proper noun like
श्रीमान कमल गोयल (shrīmān kamal gōyal). Here श्रीमान (shrīmān) is a title and thus, कमल (kamal) is a proper noun.

Rule 2: It checks whether the token from input text is matched with any entry in surname database, then the token previous to current one is a proper noun like कमल गोयल (kamal gōyal). Here, गोयल (gōyal) is a surname and thus, कमल (kamal) is a proper noun.

Using above two rules, initial proper nouns gazetteer is prepared from a large Hindi Corpus. Then manual entries are also added into this gazetteer for making it more robust for use by the translations system. After generating this gazetteer, there is need to call transliteration module (explained in the next chapter) for storing the equivalent Punjabi version of this Hindi entry. The database consists of 8000 such entries.

4.4.2 Replacing Proper Nouns:

After passing the input text through text normalization and collocation replacement sub phase of pre-processing, the output text from collocation phase becomes input text for this proper noun replacement sub phase of preprocessing. If there are any tokens in the input text that gets matched with the entries of the proper nouns database, are replaced with the corresponding equivalent Punjabi proper nouns.

Database design: Table 4.8 carries the design of the database used for storing information about proper nouns.
Table 4.8: properNoun Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hindiPropernoun</td>
<td>Stores the Hindi version of proper noun</td>
</tr>
<tr>
<td>punjabiProperNoun</td>
<td>Stores equivalent Punjabi version of the proper noun.</td>
</tr>
</tbody>
</table>

Sample database entries:

Table 4.9: Sample Entries of properNoun Database

<table>
<thead>
<tr>
<th>hindiProperNoun</th>
<th>punjabiProperNoun</th>
</tr>
</thead>
<tbody>
<tr>
<td>अभय लसंह (amar siṃh)</td>
<td>ਅਧ਩ਲਮੂੰੋ (āzād nagar)</td>
</tr>
<tr>
<td>आजाद नगर (ājād nagar)</td>
<td>ਆਜਾਦ ਨਗਰ (ājād nagar)</td>
</tr>
<tr>
<td>इंग्लैंड (īṅglaīnd)</td>
<td>ਇੰਗਲੈਂਡ (īṅglaīnd)</td>
</tr>
<tr>
<td>ਇੰਦਰਾ ਗਾਂਧੀ (indrā gāndhī)</td>
<td>ਇੰਦ੍ਰਾ ਗਾਂਧੀ (indrā gāndhī)</td>
</tr>
<tr>
<td>ਉਤ्तਰ ਭारਤ (uttar bhārat)</td>
<td>ਉੱਤਰ ਭਾਰਤ (uttar bhārat)</td>
</tr>
<tr>
<td>ਜਮਨਾ ਬਾਈ ਸਕੂਲ (jamnā bāī skūl)</td>
<td>ਜਮਨਾ ਬਾਈ ਮਦਰਾਸ਼ਟਰ (jamnā bāī sakūl)</td>
</tr>
</tbody>
</table>

4.5 Summary

In this chapter, pre-processing activity of our Machine Translation system has been provided. Design and implementation details of these activities have been discussed. Along with the database design, some excerpts from the respective databases have been provided to make the design more clear. In the next chapter, the remaining activities of our Machine Translation system, i.e. tokenizer and translation engine are discussed.
Vishal Goyal, Ph.D.

Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
Chapter 5

Tokenizer and Translation Engine

5.1 Tokenizer

Tokenizers (also known as lexical analyzers or word segmenters) segment a stream of characters into meaningful units called tokens. The tokenizer takes the text generated by pre processing phase as input. Individual words or tokens are extracted and processed to generate its equivalent in the target language. This module, using space, a punctuation mark, as delimiter, extracts tokens (word) one by one from the text and gives it to translation engine for analysis till the complete input text is read and processed.

5.2 Translation Engine

The translation engine is the main component of our Machine Translation system. It takes token generated by the tokenizer as input and outputs the translated token in the target language. These translated tokens are concatenated one after another along with the delimiter. Then this generated text is passed on to the postprocessing phase. Translation Engine Phase of the system involves various sub phases that are Identifying titles, Identifying surnames, word-to-word translation using lexicon lookup, Word sense disambiguation and handling out-of-vocabulary words. All the modules have equal importance in improving the accuracy of the system. In this chapter,
these modules are described in detail followed by an example. This phase comprises of following sub phases:

1. Identifying titles
2. Identifying surnames
3. Word-to-word translation using lexicon lookup
4. Word sense disambiguation
5. Handling out-of-vocabulary words
   5 (a) Word Inflectional analysis and generation
   5 (b) Transliteration

5.2.1 Identifying Titles

Title may be defined as a formal appellation attached to the name of a person or family by virtue of office, rank, hereditary privilege, noble birth, or attainment or used as a mark of respect. Thus word next to title is usually a proper noun. And sometimes, a word used as proper name of a person has its own meaning in target language. When this word is passed through the translation engine, it is translated by the system. This cause the system failure as these proper names should be transliterated instead of translation. For example consider the Hindi sentence श्रीमान हर्ष जी हमारे यहाँ पढ़ाई। (sh्रीमान harsh jी hamārē yahāṃ padhārē). In this sentence, हर्ष (harsh) has the meaning “joy”. The equivalent translation of हर्ष (harsh) in target language is khushī. Thus, the sentence will be translated as श्रीमान खुशी जी साधी इत्थे पढ़ाई। (sh्रीमान khushī jī sāḍē itthē padhārē). But actually it must be
translated as मूँभरु उभम ती मझे हिंघे रहपछे। (shrimān harsh jī sāḍē itthē padhāṛē). The reason is straightforward that in this sentence हर्ष (harsh) word is acting as proper noun and it must be transliterated and not translated.

In this system, a small module has been developed for locating such proper nouns where titles are present as their previous word like श्री (shri), श्रीमान (shrimān), श्रीमती (shrimātī) etc. There is one special character ‘.’ in Devanagari script to mark the symbols like डा., प्रो-. If tokenizer found this symbol during reading the text, the word containing it, will be marked as title by setting the IsTitle Flag to true. If isTitle flag has been set to true, the next word generated by tokenizer will be transliterated and not processed for translation. After the word next to title will be transliterated, isTitle flag is again reset to False. The named entities found from the text through this module are also added to the proper nouns database automatically. It improves the systems in two ways – one, it helps in continuously increasing the proper noun coverage, Second, the expansion of proper noun database will increase the speed of translation.

**Database design:** Table 5.1 carries the design of the database used for storing information about titles.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>titleInHindi</td>
<td>Stores the titles in Hindi</td>
</tr>
<tr>
<td>titleInPunjabi</td>
<td>Stores the corresponding translated titles</td>
</tr>
</tbody>
</table>
Sample Database Entries:

The title database consists of 14 entries. Following table shows some of the database entries for titles database:

<table>
<thead>
<tr>
<th>titleInHindi</th>
<th>titleInPunjabi</th>
</tr>
</thead>
<tbody>
<tr>
<td>प्रो (prō)</td>
<td>ਪ੍ਰੋ (prō)</td>
</tr>
<tr>
<td>ਸ਼੍ਰੀਮਤੀ (shrīmtī)</td>
<td>ਸ਼੍ਰੀਮਤੀ (shrīmtī)</td>
</tr>
<tr>
<td>ਸ਼੍ਰੀਮਾਨ (shrīmān)</td>
<td>ਸ਼੍ਰੀਮਾਨ (shrīmān)</td>
</tr>
<tr>
<td>ਸ਼੍ਰੀ (shrī)</td>
<td>ਸ੍ਰੀ (shrī)</td>
</tr>
</tbody>
</table>

This database can be extended at any time to allow new titles to be added.

5.2.2 Identifying Surnames

Surname may be defined as a name shared in common to identify the members of a family, as distinguished from each member's given name. It is also called family name or last name. Thus the word previous to surname is usually a proper noun. And sometimes, a word used as proper name of a person has its own meaning in target language. When this word is passed through the translation engine, it is translated by the system. This causes the system failure as these proper names should be transliterated instead of translation. For example consider the Hindi sentence प्रकाश सिंह हमारे यहाँ
In this sentence, प्रकाश (prakāš) is a noun having sense “light”. The equivalent in target language is ਔਡੜਝ (cānaṇ). Thus, the sentence will be translated as ਔਡੜਝ ਲਮੂੰ਒ ਮ਱ਛਸ਼ ਇ ਟਸ਼ ਣਡ਱਩ਸ਼ | (cānaṇ siṅgh sāḍē itthē padhārē). But actually it must be translated as ਣਰਏ਱ਭ ਲਮੂੰ਒ ਮ਱ਛਸ਼ ਇ ਟਸ਼ ਣਡ਱਩ਸ਼ | (prakāš siṅgh sāḍē itthē padhārē). The reason is straightforward that in this sentence प्रकाश (prakāş) word is acting as proper noun and it must be transliterated and not translated.

A small module has been developed for locating such proper nouns where word under consideration is a surname. If it is found to be surname then the word previous to this word is transliterated. If in any case, the previous word has been translated, now it has been corrected by transliteration. This module was also tested on a large Hindi corpus and showed that about 2-5 % text of the input text depending upon its domain is proper noun. Thus, this module plays an important role in translation. But it has also been observed that there were some cases where this module fails on following examples:

(i) आप कुमार से पूछ लें | (āp kumār sē pūch lēṃ).
(ii) उन्होंने सिंह परिवार से रिश्ता जोड़ा | (unhōnnē siṃh parivār sē rishtā jōṛā).
(iii) मैंने गोयल को कहा था कि वो यहाँ ना आये | (mainnē goyal kō kahā thā ki vō yahāṁ nā āyē)

Language in India www.languageinindia.com 737
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
(iv) राम ने सिंह की कुरानी को सराहा। *(राम ने सिंह की कुरानी को सराहा)*.

In the above examples, before the surnames कुमार (*kumār*), सिंह (*siṃh*), गोयल (*gōyal*) the tokens are आप (*āp*), उन्होंने (*unhōnnē*), मैंने (*mainnē*) respectively.

These token were transliterated rather than translated according to this module. Now, this module has been made intelligent to differentiate between proper nouns and other tokens like pronouns, prepositions, adjectives etc and thus only proper nouns will be transliterated. List of such approx. 50 tokens has been prepared manually so that whenever these tokens are found before the surnames, these must not be transliterated and will be translated.

It is not possible to store all the possible proper nouns directly into the database. Thus, the proper nouns found from the input text through this module are automatically added to the proper nouns gazeteer. Hence, through this self learning approach, the system’s accuracy and speed keep on increasing with use.

**Database design:** Table 5.3 carries the design of the database used for storing information about surnames.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>surnameInHindi</td>
<td>Stores the surname in Hindi</td>
</tr>
<tr>
<td>surnameInPunjabi</td>
<td>Stores the corresponding transliterated surnames in Punjabi</td>
</tr>
</tbody>
</table>

**Sample database entries:**

Language in India [www.languageinindia.com](http://www.languageinindia.com) 738
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
The surnames database consists of 654 entries. Following table shows some of the database entries for surnames database:

<table>
<thead>
<tr>
<th>surnameInHindi</th>
<th>surnameInPunjabi</th>
</tr>
</thead>
<tbody>
<tr>
<td>आयोड़ा (arōṛā)</td>
<td>ਅੱ਩ਾਸ਼ਾ (arōṛā)</td>
</tr>
<tr>
<td>कक्ड़ (kakṛ)</td>
<td>ਕੱਕਾਰ (kakkar)</td>
</tr>
<tr>
<td>खुराना (khurānā)</td>
<td>ਖਰਾਰਮ (khurānā)</td>
</tr>
<tr>
<td>जिंदल (jindal)</td>
<td>ਜੀੰਦਾਲ (jindal)</td>
</tr>
</tbody>
</table>

5.2.3 Word-to-Word translation using lexicon lookup

If token is not a title or a surname, it is looked up in the HPDictionary database containing Hindi to Punjabi direct word to word translation. If it is found, it is used for translation. If no entry is found in HPdictionary database, it is sent to next sub phase for processing. For example, token is अइटीसवां (aṛītīsvāṁ), it is looked up in the database and the entry for it is found in the database. Then its translated version is used in the output text i.e. अਚੈੀਲ (aṭhītivāṁ). And no other phase is required for this token. Tokenizer will start generating next token for processing by the translation engine.

Database design: Table 5.5 carries the design of the database used for storing entries for Hindi words to Punjabi words direct translation.

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hindiWord</td>
<td>Stores the Hindi Word</td>
</tr>
</tbody>
</table>
punjabiWord Stores the corresponding translated word in Punjabi

Sample database entries:

The HPDictionary database consists of 54,127 entries. Following table shows some of the database entries for HPDictionary database:

Table 5.6: Sample Entries of HPDictionary Database

<table>
<thead>
<tr>
<th>hindiWord</th>
<th>punjabiWord</th>
</tr>
</thead>
<tbody>
<tr>
<td>अथवा (athvā)</td>
<td>अठ (atē)</td>
</tr>
<tr>
<td>छोड़ (chōḍai)</td>
<td>खड़ (chaḍḍi)</td>
</tr>
<tr>
<td>जायेगे (jāyēṅgē)</td>
<td>खाल (jāṅgē)</td>
</tr>
<tr>
<td>सीखा (sīkhā)</td>
<td>लिखिता (sikkhiā)</td>
</tr>
</tbody>
</table>

This database can be extended at any time to allow new entries in the dictionary to be added.

5.2.4 Resolving Ambiguity

Ambiguity is one of the NLP problems which have been a great challenge for computational linguists. In general, people are unaware of the ambiguities in the language they use because they are very good at resolving them using context and their knowledge of the world. But computer systems do not have this knowledge, and consequently do not do a good job of making use of the context.

Something is ambiguous when it can be understood in two or more possible ways or when it has more than one meaning. If the ambiguity is in a
sentence or clause, it is called structural (syntactic) ambiguity. If it is in a single word, it is called lexical ambiguity.

For the structural ambiguity, consider the sentence “The man saw the girl with the telescope”. This sentence is ambiguous since it can be interpreted in two ways: The man saw the girl who possessed the telescope or, the man saw the girl with the aid of the telescope. However, the sentence “The man saw the girl with a red hat” is not ambiguous for a human reader (people have the knowledge that a hat cannot be used to see), while it has the same ambiguity as the previous example for a computer.

In a Machine Translation application, different senses of a word may be represented with different words in the target language. Consider the following sentence:

याभ आभ खा रहा है। (rām ām khā rahā hai)

In the above example, word आभ in the sentence is lexically ambiguous. Its meaning can be interpreted in two ways – mango (a fruit) and usual (an adjective) as in following examples:

**Usage as Noun:** तोता पेड़ पर बैठकर आभ खा रहा है (tōtā pēḍa par baiṭhakar ām khā rahā hai)

**Usage as Adjective:** ऐसे चोरों से मिलना आभ बात है जो चोरी के खिलाफ उपदेश देते हैं (aisē cōrōṁ sē milnā ām bāt hai jō cōrī kē khailāpha updēsh dētē haiṁ)
In order to correctly translate a text in one language to another, firstly we have to know the senses of the words and then find the best translation equivalent in the target language.

Lexical ambiguity can refer to both homonymy and polysemy. Homonyms are words that are written the same way, but are (historically or conceptually) really two different words with different meanings which seem unrelated. Examples are *suit* (“lawsuit” and “set of garments”) and *bank* (“river bank” and “financial institution”). If a word’s meanings are related, it is called a polyseme. The word *party* is polysemous because its senses can be generalized as “group of people”, that is they are related.

Now let us consider the meaning of the noun *party* in the following sentence:

*Mr. Smith’s party took 38% of the votes in the last election.*

It is clear to a human reader that the noun *party* is in the sense “an organization to gain political power” in the above sentence. Most people are not even aware of the ambiguity contained in the sentence. Humans are so skilled at resolving potential ambiguities that they do not realize they are doing it. There has been research on how people resolve ambiguities; however we still do not know exactly how humans do lexical disambiguation. Therefore, it is a difficult task to teach a computer to do the same thing. The most prominent way to disambiguate a word is examining its context. The context
can be considered as the words surrounding the ambiguous word, which is the noun *party* in our case. Words as *vote* and *election* might be a good clue for the sense of the noun *party*. But context is not the only information available for disambiguation. Syntactic classes of the words in the ambiguous word’s context (whether they are noun, verb or adjective, etc.), whether the ambiguous word plays the role of object or subject in the syntactic structure of the sentence may also be used in the disambiguation process.

In our research problem, we have determined the correct meaning of an ambiguous word which comes across during translation process, namely Word Sense Disambiguation using the context information.

WSD algorithms can be divided into two based on the corpora used for training. These approaches are:

i. Supervised Word Sense Disambiguation

ii. Unsupervised Word Sense Disambiguation

In supervised WSD the training data is sense-tagged whereas in unsupervised WSD the training data is raw corpora which have not been semantically disambiguated. In the following sections these approaches will be explained in detail.

**Supervised Disambiguation**

Supervised disambiguation is an application of the supervised learning approach for creating a classifier. A disambiguated corpus where each occurrence of an ambiguous word is annotated with a contextually appropriate
sense is available for training. The aim in supervised disambiguation is to build a classifier which correctly classifies new cases based on their context of use.

Machine learning algorithms such as Bayesian classifiers [118], decision lists [119], decision trees [120], k-nearest neighbor and neural networks [121] are examples of supervised learning algorithms.

An example of probabilistic algorithms is Naïve Bayes [122] which has been frequently applied in WSD with good results [123]. Gale, Church and Yarowsky [124,125] uses a variant of Bayes ratio on six ambiguous nouns, namely drug, duty, land, language, position, and sentence, and reports 90% accuracy in discriminating between two senses of these words. Mooney [126] reports that Naïve Bayes and neural networks achieved the highest performance with an accuracy of 73% in assigning the correct senses to a corpus of examples of word line which has six senses. The other algorithms in Mooney’s survey were 3-nearest neighbors, perceptron, decision tree, decision list and logic programming variants. Combining various classifiers has also been tested. Florian et al. [127] combined four classifiers namely feature-enhanced Naïve Bayes, Cosine, bag-of-words Naïve Bayes and non-hierarchical decision lists.

Decision lists search for discriminatory features in the training corpus and build a set of rules for disambiguation. Yarowsky [128] makes use of hierarchical decision lists and achieves top performance in the SENSEVAL-1 framework on the 36 test words for which tagged training data was available.
Agirre and Martinez [129] reports that decision lists provide state-of-the-art results with simple and very fast means. This approach is reported to learn with low amounts of data.

Decision lists and Bayesian classifiers are the most popular algorithms in supervised disambiguation. For neural networks Towell and Voorhees [130], for decision trees Black [131] and Pedersen [132], for k-nearest neighbor Ng and Lee [133] and for information-theoretic approaches Brown et al. [134] are some examples of the work done on WSD.

A major problem with supervised approaches is the need for a large sense-tagged training set. Despite the availability of large corpora, manually sense-tagging of a corpus is very difficult and very few sense-tagged data are available now.

The two largest corpora that are available are the SemCor corpus [135] and the SENSEVAL corpus [136-138]. The SemCor corpus, created by the Princeton University, is a subset of the English Brown corpus containing almost 700,000 running words. In SemCor, all the words are tagged by part of speech and more than 200,000 content words are also lemmatized and sense-tagged according to Princeton WordNet 1.6 (mappings for later versions of WordNet are also available). SENSEVAL corpus is derived from the HECTOR corpus and dictionary project. It is a joint Oxford University Press and Digital project which took place in the early 1990s. Another sense-tagged corpus available is the DSO Corpus of Sense-Tagged English (Ng and Lee, 1996) [133]. This corpus contains sense-tagged word occurrences for
121 nouns and 70 verbs which are among the most frequently occurring and ambiguous words in English. These occurrences are provided in about 192,800 sentences taken from the Brown Corpus and the Wall Street Journal and have been hand tagged by students at the Linguistics Program of the National University of Singapore. WordNet 1.5 sense definitions of these nouns and verbs were used to identify a word sense for each occurrence of each word.

There have been several efforts for finding a way to avoid the use of hand-tagged data. Bootstrapping is the most frequently used method for this purpose. Bootstrapping relies on a small number of instances of each sense for each lexeme of interest. These sense-tagged instances are used as seeds to train an initial classifier. This initial classifier is then used to extract a larger training set from the remaining untagged corpus. With each iteration of this process, the training corpus grows and the untagged corpus shrinks.

Hearst [139] generates a seed set by simply hand-tagging a small set of examples from the untagged corpus. However, during the training phase each occurrence of a set of nouns to be disambiguated is manually sense-tagged in several occurrences. Schütze [140-141] proposes a method that avoids tagging each occurrence in the training corpus. Yarowsky [142] proposes an alternative technique by using two constraints named as “One sense per collocation” and “One sense per discourse” and reports an accuracy of 96% on twelve words. “One sense per collocation” argues that nearby words provide strong and consistent clues to the sense of a target word, conditional
on relative distance, order and syntactic relationship. Also, “One sense per discourse” constraint argues that the sense of a target word is highly consistent within any given document. Different bootstrapping techniques are also presented in Mihalcea and Moldovan [143] and Mihalcea [144]. Mihalcea [144] makes a comparison between the results when training is performed on hand-tagged data and the results when training is done using the generated corpus by bootstrapping. She reports that the precision achieved with the generated corpus is comparable, and sometimes better than the precision achieved with hand-tagged corpora.

Another method for avoiding hand-tagged data is using parallel corpora [145]. In this method, bilingual corpora are used since different senses of some words translate differently in another language. By using a parallel aligned corpus, the translation of each occurrence of such words can be used to determine their correct senses automatically. In Dagan and Itai [146], Ide et al. [147] and Ng et al. [148], various uses of parallel corpora for WSD and its disadvantages can be found.

The main problem that supervised disambiguation methods face with is data sparseness. Since the sense-tagged training corpus is finite and very small for WSD, some senses of polysemous words are very likely to be missing and most of them have few examples. For a supervised algorithm to be successful, the training data must ensure that all senses of a polysemous word are covered. Smoothing is used to solve the data sparseness problem. The task of reevaluating some of the zero-probabilities or low-probabilities
and assigning them non-zero values is called smoothing. Some of the smoothing methods are add-one smoothing, Witten-Bell smoothing [149], and Good-Turing smoothing [150]. Gale [151], presented a Good-Turing method for estimating the probabilities of seen and unseen objects in linguistic applications named as Simple Good-Turing method.

**Unsupervised Disambiguation**

In machine learning the distinction between supervised and unsupervised algorithms rests on whether a set of classifications exists. In unsupervised word sense disambiguation, information is gathered from raw corpora which have not been semantically disambiguated.

Yarowsky [152] proposed an approach for marking words with their categories from a thesaurus. He used Roget’s Thesaurus [153]. Training was carried out on an untagged corpus of 10 million words obtained from the electronic version of the Grollier’s Encyclopedia. The important aspect of the approach was that he used a context of 50 words either side so that 100 words were considered in the training examples for each ambiguous word. This method was tested on 12 ambiguous words and reported to achieve 92% accuracy. Yarowsky notes that this method is best for extracting topical information, most successful for nouns. The algorithm presented in Yarowsky [142] is also an unsupervised algorithm making use of a bootstrapping procedure.

McCarthy et al. [154], presents an algorithm that makes use of a thesaurus acquired from raw textual corpora and the WordNet similarity package to find predominant noun senses automatically. The acquired predominant senses
gave a precision of 64% on the nouns of the SENSEVAL-2 all-words task which is a promising result regarding that no hand-tagged data is used.

Some of the unsupervised methods correspond to clustering tasks rather than sense tagging tasks because they do not label words to predefined senses. These algorithms do not make use of an outside source of knowledge to define senses. This is called *Word Sense Discrimination* rather than disambiguation. They divide the occurrences of a word into a number of classes by determining for any two occurrences whether they belong to the same sense or not [155-157]. Schütze’s [155] results indicate that for coarse binary distinctions, unsupervised techniques can achieve results approaching those of supervised and bootstrapping methods. Purandere and Pedersen [156] present a systematic comparison of discrimination techniques proposed by Pedersen and Bruce [156,158,159] and by Schütze [157].

**Knowledge Bases for WSD**

In this section, different kinds of knowledge bases are presented. These knowledge bases can be used in any WSD system, whether it is supervised or unsupervised.

**Machine Readable Dictionaries**

Machine readable dictionaries (MRD) provide a ready-made information source of word senses. The first attempt to use MRD’s came from Lesk (1986)[160]. He starts from the simple idea that a word’s dictionary definitions are likely to be good indicators of the senses they define. By using Oxford Advanced Learner’s Dictionary (OALD), he counts overlapping content words
in the sense definitions of the ambiguous word and in the definitions of context words occurring nearby and selects the sense that achieves the maximum number of overlaps. The accuracy of the method is reported to be 50-70% on short samples of the Jane Austen novel Pride and Prejudice and an Associated Press news story based on very brief experimentation with the program.

Cowie et al. [161] tried to improve Lesk’s approach by optimizing the overlap of all words in a single sentence simultaneously. However, it was found computationally very expensive. Therefore, Cowie et al. [161] used simulated annealing [162] for the first time in natural language processing. They evaluated this approach using a corpus consisting of 50 example sentences taken from Longman Dictionary of Contemporary English (LDOCE) which were disambiguated by hand. 47% of the words were reported to be correctly disambiguated to sense level and 72% to more rough grained senses.

Stevenson and Wilks [163] computed the overlap by normalizing the contribution of a word to the overlap count. Pedersen and Banerjee [164] described a different version of the Lesk’s algorithm by employing glosses contained in WordNet [165]. Because of the fact that dictionaries are created for human use, not for computers, there are some inconsistencies [166-168]. Although they provide detailed information at the lexical level, they lack pragmatic information used for sense determination. For instance, the relation between ash and tobacco, cigarette or tray is very indirect in a dictionary
whereas the word *ash* co-occurs very frequently with these words in a corpus [169].

**Thesauri**

Thesauri provide information about relationships among words. Thesaurus based disambiguation makes use of the semantic categorization provided by a thesaurus or a dictionary with subject categories. The most frequently used thesaurus in WSD is Roget’s International Thesaurus (Roget, 1946) which was put into machine-tractable form in 1950’s [153].

Walker [170] proposed an algorithm as follows: each word is assigned to one or more subject categories in the Thesaurus. If the word is assigned to several subjects, then it is assumed that they correspond to different senses of the word.

Similar to machine readable dictionaries, a Thesaurus is a resource for humans, so there is not enough information about word relations.

**Computational Lexicons**

The usefulness of lexical relations in linguistic, psycholinguistic and computational research has led to a number of efforts to create large electronic databases of such relations. Beginning from the mid-1980’s, construction of semantic lexicons by hand has emerged. Some examples of these lexicons are WordNet [164], CyC [171], ACQUILEX [172], and COMLEX [173]. Each of these lexicons contains different kinds of information.
**WordNet**

WordNet is an online lexical reference system which was developed at Princeton University under the direction of Professor George A. Miller. It combines many features used for WSD in one system. It includes definitions of word senses as in a dictionary; it defines “synsets” of synonymous words representing a single lexical concept; and it includes word-to-word relations.

WordNet consists of three databases: noun database, verb database and one database for adjectives and adverbs. Each database consists of lexical entries corresponding to unique orthographic forms.

The earliest attempts to use WordNet in WSD were in information retrieval field. Voorhees [174] and Richardson and Smeaton [175] created knowledge bases using WordNet’s hierarchy. Li et al. [176] proposed a WordNet-based algorithm for WSD. Disambiguation was done by semantic similarity between words and heuristic rules. Heuristic rules were based on the semantic similarity and the WordNet hierarchy. Leacock et al. [177] used WordNet to counter data sparseness problem. Hawkins [178] built up a WSD system that works with frequency and contextual information that is based on WordNet. Fellbaum et al. [179] proposed a system that made use of syntactic clustering and semantic distinctions extracted from WordNet.

WordNet is mostly used to determine semantic similarity between senses. Resnik [180] computed information content of words which is a measure of the specificity of the concept that subsumes the words in the WordNet hypernym hierarchy. Agirre and Rigau [181] employ WordNet to determine the
conceptual distance among concepts whereas Mihalcea and Moldovan [182] exploit semantic density and WordNet glosses in an all words word sense disambiguation. Lin [183-185] described a semantic similarity measure where similarity between two objects is defined to be the amount of information contained in the commonality between the objects divided by the amount of information in the description of the objects.

Other approaches using WordNet are Jiang [186], Agirre and Agirre et al. [187], Haynes [188] and Banerjee et al. [189]. A combination of MRDs and WordNet has also been tried with some success [190-192].

**WSD in Indian Languages**

Robust Standalone Systems for word sense disambiguation in Indian language are very few. WSD is mostly tackled at the POS tagging and Morphological analysis phase and what ever left is handled with the help of rules. Recently some standalone algorithms have also been developed for WSD in Indian languages.

Anusaaraka is one of the oldest MT systems available in India. It is more a language accessor rather than an MT system [65]. It is based on the assumption that most Indian languages have same origin so most of the words in source language have one meaning in target language. Based on this, it just provides the glosses of source language in the target language. There are cases where the meaning is too general or too specific. Such cases are handled by introducing some special notation to either narrow down or
widen the meaning. An attempt is made to find the underlying thread that connects different senses of the polysemous word. A kind of formula is then evolved that faithfully and unambiguously represents the connection between these different senses. For the English – Hindi system, the current version of Anusaaraka uses a dictionary called Shabdanjali. POS tagger and wasp workbench are used for developing word sense disambiguation rules semi-automatically.

Similarly in AnglaBharati approach a rule base is used for picking up the correct sense of each word in the source language to the extent feasible using interleaved semantic interpreter [65]. Further disambiguation and choice of right construct and lexical preference are generated by the target language text generator module. Many a time, multiple rules may get invoked leading to the multiple interpretation of the input sentence. The rules are ordered in terms of their preference and an upper limit is put on the number of alternatives produced. Most of the disambiguation rules are in the form of syntacto-semantic constraints. Semantics are used to resolve most of the intra-sentence anaphora/pronoun references. Alternative meanings for the unresolved ambiguities are retained in the pseudo target language. The lexical database is hierarchically organized to allow domain specific meanings and also prioritize meanings as per user requirement.

In the example based approach developed by [66] and known as ANUBHARTI, ambiguities in the meaning of the verb phrasal are also resolved using an appropriate distance function in the example base. The
alternate translations are being ranked with respect to the ordering of the rule base.

In ANUBAAD system, sense disambiguation is carried out at various levels [75]. It starts with POS of a word. Some semantic categories are associated with words to identify the inflections to be attached with corresponding words in Indian languages as well as to identify the context in the sentence. Context identification is also done by the recognition of idiomatic expression and using context templates for each word. The context templates have been designed on the basis that meaning of the word may be independent of the context, may depend upon the occurrence of a sequence of words or words with certain semantic categories or may depend on the occurrence of certain keywords or keyword with certain semantic category.

In Matra, rule bases and heuristic approaches are used for word sense disambiguation. A method has been described by Durgesh Rao et al.[1] for mapping prepositions from English to Hindi. Similarly in Saarthak, emphasis is on sentence-level word sense disambiguation, which makes it different from general statistical techniques that use contextual information for the same. At AU-KBC research centre, S. Baskaran [193] presents an approach in which all the occurrences of the ambiguous words are classified into different clusters in such a way that all the occurrences are in the same sense within a cluster. Development of a Prototype of a Frame-based System for the Understanding of Malayalam Language has been carried out by Sumam M. Idicula and David Peter S [194]. In this system, three types of information are
used for word sense disambiguation. They are local word grouping (grouping of words which can collectively perform a syntactic role in a sentence), syntactic information and semantic tags. Prabhakar Pandey et. al. [195] makes use of the Wordnet for Hindi developed at IIT Bombay, for WSD. The accuracy values are reported to be in range from about 40% to about 70%. The system currently deals with only nouns. Ganesh Ramakrishnan [196] introduces the notion of soft word sense disambiguation which states that given a word, the sense disambiguation system should not commit to a particular sense, but rather, to a set of senses which are not necessarily orthogonal or mutually exclusive.

Information Sources for WSD

There are various information sources or feature types used in WSD regardless of the type of the approach. To disambiguate a word, various kinds of information, including syntactic tags, word frequencies, collocations, semantic context, role-related expectations, and syntactic restrictions can be considered.

In Agirre and Martinez [197], a comparison of WSD systems has been made based on the information source they used. Some of these sources are as follows:

**Frequency of Senses:** Frequency information is used to measure the likelihood of each possible sense appearing in the text. Therefore this information is generally used in statistical approaches and it is generally
learned from hand-tagged data such as SemCor corpus. Interestingly very few WSD approaches outperform the “most frequent sense” heuristic. WordNet senses are ordered according to the frequencies of the senses in the SemCor corpus.

**Part of Speech (POS):** Part of speech tagging is regarded as the first step of the disambiguation process if the lemmas have the same orthographic forms but different syntactic classes. It is useful because it reduces the number of possible senses a word can belong to. An orthographic form may even be unambiguous in one syntactic class whereas it has more than one sense in another. For instance, in WordNet 2.0 *handle* has 5 senses as a verb, but only one sense as a noun. The impact of knowledge resources on WSD is examined in Gaustad [198]. The results show that accurate POS information is beneficial for WSD and that including the POS of the ambiguous word itself as well as POS of the context increases the disambiguation accuracy.

**Morphology:** It is defined as the relation between derived words and their roots. For instance, the noun *agreement* has 6 senses, its verbal root *agree* 7. A stemmer tries to reduce various forms of a word to a single stem. Since English is a language with little inflectional morphology, it is not certain that using morphology will lead to significant improvements in WSD. With other languages, such as German or Italian, morphology is of greater influence.

**Collocations:** Collocation is the relationship among any group of words that tend to co-occur in a predictable configuration. Disambiguation relies heavily on collocational information. For example, the noun *match* has 9
senses. However, it has only one possible sense in “football match”. It is observed that collocations are strong indicators if they are learned from hand-tagged corpora. Although they are strong, they should be used with other sources. They should not be treated as rules for sense-filtering alone [199,200].

**Semantic word associations:** These can be classified as follows:

i. *Taxonomical organization:* This refers to the classification of words in a hierarchy and the lexical-semantic relationships holding between words such as a *dog* is a kind of *animal*. This kind of information can be extracted from ontologies like WordNet.

ii. *Situation and Topic:* Information about the situation or topic enables a WSD system to see the ambiguous word in a broader context. For example, if the word *mouse* is used in an office situation and the topic is computer use, the most probable sense of the word *mouse* will be “computer tool”, not “animal”. Semantic word associations around topic and situation are powerful when learned from hand-tagged corpora. Associations learned from MRDs can also be useful.

iii. *Argument-head relations:* These relations provide important clues for disambiguation such as the relationship between *dog* and *bite* in the sentence “the dog bit the man.”

**Subcategorization information:** Subcategorization refers to certain kinds of relations between words and phrases. For example the verb *want* can be...
followed by an infinitive, as in “I want to fly to Istanbul”, or a noun phrase, as in “I want a flight to Istanbul”. But the verb find cannot be followed by an infinitive. For example “I found to fly to Istanbul.” is not a correct sentence.

Verbs have several possible patterns of arguments. A particular set of arguments that a verb can appear with is referred to a subcategorization frame. Subcategorization frames capture syntactic regularities about complements.

Agirre and Martinez [187] made a comparison between the contributions of the above resources to WSD. According to their observations, if learned from hand-tagged corpora, collocations and semantic word associations are the most important knowledge types for WSD, but they also mentioned that syntactic cues are equally reliable. On the other hand, taxonomical information was found to be very weak.

Our Approach

While dealing with related languages like Hindi and Punjabi, structural ambiguity is not a problem at all because the ambiguity in the source sentence is transferred to the target sentence without affecting the underlying meaning. We are not claiming that there is no structural ambiguity in the Hindi language that do not carry over as such in Punjabi language, but we did not come across with any. So, structural ambiguity has not been touched in this research work. To start with, all we have is a raw corpus of Hindi text. So the N-Gram statistical approach is the obvious choice for our purpose. The
following section provides the theory of N-Gram approach and our approach for WSD.

**N-Gram Approach:**

An *n-gram* is simply a sequence of successive *n* words along with their count i.e. number of occurrences in training data [201,202]. An *n-gram* of size 2 is a bigram; size 3 is a trigram; and size 4 or more is simply called an *n-gram* or \((n - 1)\)-order Markov model. An *n-gram* model models sequences of natural languages using the statistical properties of *n-grams*. More concisely, an *n-gram* model predicts \(x_i\) based on \(x_{i-1}, x_{i-2}, \ldots, x_{i-n}\). *n-grams* models are widely used in statistical natural language processing.

The number of words in the local context of ambiguous word makes a window. The size of this window i.e. the value of \(N\) depends on various factors.

a) Larger the value of \(n\), higher is the probability of getting correct word sense i.e. for the general domain; more training data will always improve the result. But on the other hand most of the higher order *n-grams* do not occur in training data. This is the problem of sparseness of data.

b) As training data size increases, the size of model also increases which can lead to models that are too large for practical use. The total number of potential *n-grams* scales exponentially with \(n\). Computer up to present could not calculate for a large \(n\) because it requires huge amount of memory space and time.
c) Does the model get much better if we use a longer word history for modeling an \( n \)-gram?

d) Do we have enough data to estimate the probabilities for the longer history?

Claude E. Shannon [203] established the information theory for finding the value of \( n \) in 1951. This theory included the concept that a language could be approximated by an \( n \)th order Markov model by \( n \) to be extended to infinity. Shannon computed the per letter entropy rather than per word entropy. He gives entropy of English text as 1.3 bits per letter. Since his proposal there were many trials to calculate \( n \)-grams for a big text data of a language. Brown et. al.[204] performs a test on much larger text and give an upper bound of 1.75 bits per character for English language by using trigram model. Iyer et al. [205] investigate the prediction of speech recognition performance for language model in the switchboard domain, for trigram model built on different amounts of in domain and out of domain training data. Over the ten models they constructed, they find that perplexity predicts word error rate well when only in domain training data is used, but poorly when out of domain text is added. They find that trigram coverage or the fraction of trigram in the test data present in training data is a better predictor of word error rate than perplexity.

Chen et al. [206] investigate their language model for speech recognition performance in the Broadcast news domain and concluded that perplexity
correlates with word error rate remarkably well when only considering $n$-gram model trained on in domain data.

Manin [207] performs a study on predictability of word in context and found that unpredictability of a word depends upon the word length. Marti et. al. [208] tested different vocabulary size and concluded that language models become more powerful in recognition tasks with larger vocabulary size. Resnik et. al. [209-210] made several observations about the state of the art in automatic word sense disambiguation and offer several specific proposals to the community regarding improved evaluation criteria, common training and testing resources, and the definitions of sense inventories.

While Kaplan [211] Choueka and Lusignan [212], based on the observation that people don't seem to need very much context, claims that only 5 words to the left and 5 words to the right of the polysemous word are sufficient for WSD but William A. Gale et. al. [213] use a very wide context, 100-words surrounding the polysemous word in question. They find that there are often very useful clues even quite far away from the polysemous word in question. They demonstrated that information is measurable out to 10,000 words away from the polysemous word. They also observed that although contextual clues are measurable at surprisingly large distances, much of this information might not be very useful. In particular, it might have been possible to find the same information at smaller distances. In their words:

“The contribution is largest, not surprisingly, for smaller $d$, but nevertheless, the contribution continues to grow out to at least twenty words,

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
perhaps fifty words, well beyond the ±6 word contexts typically found in many disambiguation studies. Increasing the context from ±6 words to ±50 words improves performance from 86% to 90.”

Among number of approaches for disambiguation, the most appropriate approach to determine the correct meaning of a Hindi word in a particular usage for our Machine Translation system is to examine its context using N-gram approach. After analyzing the past experiences of various authors explained above, we have chosen the value of n to be 3 and 2 i.e. trigram and bigram approaches respectively for our system. Trigrams are further categorized into three different types. First category of trigram consists of context one word previous to and one word next to the ambiguous word. Second category of trigram consists of context of two adjacent previous words to the ambiguous word. Third category of the trigram consists of context of two adjacent next words to the ambiguous word. Bigrams are also categorized into two categories. First category of the bigrams consists of context of one previous word to ambiguous word and second category of the bigrams consists of one context word next to ambiguous word. The disambiguation algorithm starts with the look up in the trigrams databases. All the three trigrams databases are looked up for the some entry corresponding to ambiguous word. If the entry is matched in more than one trigrams databases, the entry of the database with maximum frequency will be considered to be the best match. If the entry is matched only in any one of the three trigrams databases, then that entry is used regardless of the frequency.
If in case, no trigrams database is able to disambiguate the word, then bigram databases are used for disambiguation. Entry is looked up in both of the bigrams databases and if found in both the databases, the entry with maximum frequency will be considered. If the entry is found only in any one of the bigram databases, then that entry is used for disambiguation. In the worst case, if no entry is matched from any of entries in both trigrams and bigrams databases, then this word is assumed to be unknown and out-of-vocabulary module will handle such words. For this purpose, the Hindi corpus consisting of about 2 million words was collected from different sources like online newspaper daily news, blogs, Prem Chand stories, Yashwant jain stories, articles etc. The most common list of ambiguous words was found. We have found a list of 75 ambiguous words out of which the most frequent are से sē and और aur. Following table shows a summary of different lexical categories for these ambiguous words:

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Lexical Category</th>
<th>Ambiguos Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Noun</td>
<td>38%</td>
</tr>
<tr>
<td>2.</td>
<td>Verb</td>
<td>13.5%</td>
</tr>
<tr>
<td>3.</td>
<td>Adjective</td>
<td>6.5%</td>
</tr>
<tr>
<td>4.</td>
<td>Adverb</td>
<td>1.38%</td>
</tr>
<tr>
<td>5.</td>
<td>Postposition</td>
<td>1.38%</td>
</tr>
<tr>
<td>6.</td>
<td>Noun and Verb</td>
<td>8.17%</td>
</tr>
</tbody>
</table>

Table 5.7: Lexical Category % distribution of ambiguous words
<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>7.</td>
<td>Noun and Adjective</td>
<td>21.39%</td>
</tr>
<tr>
<td>8.</td>
<td>Adjective and Conjunction</td>
<td>1.38%</td>
</tr>
<tr>
<td>9.</td>
<td>Adjective and Verb</td>
<td>2.76%</td>
</tr>
<tr>
<td>10.</td>
<td>Noun, Postposition and Conjunction</td>
<td>1.38%</td>
</tr>
<tr>
<td>11.</td>
<td>Noun, Adjective and Adverb</td>
<td>4.16%</td>
</tr>
</tbody>
</table>

Through Corpus analysis and taking this ambiguous word list as base, above mentioned three types of trigrams databases (one word to the left and one word to the right of the ambiguous word, two consecutive words to the left of the ambiguous word, two consecutive words to right of the ambiguous words) and bigrams (one word previous to the ambiguous word, one word to the right of the ambiguous word) were generated from the corpus along with their frequency in the corpus and stored into the databases trigramsMiddle, trigramsLeft, trigramsRight, bigramsLeft and bigramsRight respectively. On analysis, it has been found that in Hindi language, most common words that are ambiguous are post positions like से (sē), घर (par) etc and the conjunction और (aur).

Consider the postposition ‘से’ that can be translated most commonly into ਦੇ, ਦੂ, ਦਾ ਸੀ, ਕੁਝ ਦੋ, ਸੀ, ਨਾਂ ਦੀ, ਕੁਝ ਤੇ ਤਿਨੋਂ , ਕੁਝ ਲਾਇਨਾ. Let us take the example:
It is the task of word sense disambiguator module to find the appropriate meaning/sense for the ambiguous word ‘से’. The algorithm starts with collecting the words surrounding the ambiguous word ‘से’ in the sentence. Thus, it forms following three context bags with window size 3 in which Context Bag 1 contains two context words previous to the ambiguous word से, Context Bag 2 contains one context word previous to and one context word next to the ambiguous word से and Context Bag 3 contains two context words next to the ambiguous word से.

**Context Bag 1**: (मैंने) (राम) से

**Context Bag 2**: (राम) से (पुछा)

**Context Bag 3**: से (पुछा) (आप)

Then the algorithm searches the entry match for Context Bag1, Context Bag2 and Context Bag3 in trigrams databases - trigramsRight, trigramsMiddle, and trigramsLeft respectively. The entries are found in different trigrams databases as shown below:

<table>
<thead>
<tr>
<th>Database</th>
<th>Context Bag</th>
<th>Punjabi Meaning</th>
<th>Frequency</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Database</th>
<th>Context Bag</th>
<th>Punjabi Meaning</th>
<th>Frequency</th>
</tr>
</thead>
</table>

**Table 5.8: Example demonstrating the ambiguity resolution**
Above table shows that, the Punjabi meanings for all the three entries differ, hence the entry of the database with maximum frequency among three, will be used for disambiguation i.e. the entry with frequency 1209 will be used. Reason for using the entry with maximum frequency among these is to further find the most appropriate meaning for that ambiguous word. Finally, the translated text in target language will be

मैं उम्मे थूँ पुच्छिए हुमी विंगे ता चुड़े चे | *(maiṃ rāṃ nūṃ pucchiā tusī kīthē jā rahē hō)*

**Database design:** Table 5.9, Table 5.10, Table 5.11, Table 5.12 and Table 5.13 carries the design of the database used for storing entries for word sense disambiguation.

**Table 5.9:** triGramsMiddle Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambHindiWord</td>
<td>Stores ambiguous word in Hindi</td>
</tr>
<tr>
<td>previousWord1</td>
<td>Stores possible word previous to ambiguous word in text</td>
</tr>
<tr>
<td>nextWord1</td>
<td>Stores possible word next to ambiguous word in text</td>
</tr>
<tr>
<td>punjabiWord</td>
<td>Stores the correct translation for this ambiguous word for this instance</td>
</tr>
<tr>
<td>Freq</td>
<td>Stores the frequency of this trigram in the analyzed corpus</td>
</tr>
</tbody>
</table>
### Table 5.10: triGramLeft Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambHindiWord</td>
<td>Stores ambiguous word in Hindi</td>
</tr>
<tr>
<td>previous1Word</td>
<td>Stores possible word previous to ambiguous word in text</td>
</tr>
<tr>
<td>previous2Word</td>
<td>Stores possible word previous to previous1Word mentioned above in the text</td>
</tr>
<tr>
<td>punjabiWord</td>
<td>Stores the correct translation for this ambiguous word for this instance</td>
</tr>
<tr>
<td>Freq</td>
<td>Stores the frequency of this trigram in the analyzed corpus</td>
</tr>
</tbody>
</table>

### Table 5.11: triGramsRight Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambHindiWord</td>
<td>Stores ambiguous word in Hindi</td>
</tr>
<tr>
<td>next1Word</td>
<td>Stores possible word next to ambiguous word in text</td>
</tr>
<tr>
<td>next2Word</td>
<td>Stores possible word next to next1Word mentioned above in the text</td>
</tr>
<tr>
<td>punjabiWord</td>
<td>Stores the correct translation for this ambiguous word for this instance</td>
</tr>
<tr>
<td>Freq</td>
<td>Stores the frequency of this trigram in the analyzed corpus</td>
</tr>
</tbody>
</table>

### Table 5.12: biGramsLeft Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambHindiWord</td>
<td>Stores ambiguous word in Hindi</td>
</tr>
<tr>
<td>previousWord</td>
<td>Stores possible word previous to ambiguous word in text</td>
</tr>
<tr>
<td>punjabiWord</td>
<td>Stores the correct translation for this ambiguous word for this instance</td>
</tr>
<tr>
<td>Freq</td>
<td>Stores the frequency of this bigram in the analyzed corpus</td>
</tr>
</tbody>
</table>


Table 5.13: biGramsRight Database Design

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ambHindiWord</td>
<td>Stores ambiguous word in Hindi</td>
</tr>
<tr>
<td>nextWord</td>
<td>Stores possible word next to ambiguous word in text</td>
</tr>
<tr>
<td>punjabiWord</td>
<td>Stores the correct translation for this ambiguous word for this instance</td>
</tr>
<tr>
<td>Freq</td>
<td>Stores the frequency of this bigram in the analyzed corpus</td>
</tr>
</tbody>
</table>

Sample database entries:

There is no Hindi-Punjabi parallel corpus available and even no machine readable Hindi-Punjabi dictionary is available. Thus, there is no way to generate the data for word sense disambiguation databases automatically. For this purpose, we have developed a small module for finding the bigrams and trigrams for the ambiguous words from the Hindi Corpus. But the corresponding equivalent meaning in Punjabi for the ambiguous word based on its context is found manually and stored into the database for each entry. It is very time consuming and tedious task. The triGramMiddle database consists of 48,285 entries. The triGramLeft database consists of 46,735 entries. The triGramRight database consists of 49,217 entries. The biGramLeft database consists of 52,456 entries. The biGramLeft database consists of 51,129 entries.

Following tables show some of the database entries for fiveGrams, trigrams and biGrams databases:
<table>
<thead>
<tr>
<th>previous1Word</th>
<th>ambHindiWord</th>
<th>next1Word</th>
<th>punjabiWord</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>तरह (tarah)</td>
<td>से (sē)</td>
<td>समझा (samjhā)</td>
<td>ढऱप (nāl)</td>
<td>5291</td>
</tr>
<tr>
<td>आराम (ārām)</td>
<td>से (sē)</td>
<td>कठ (kaṭ)</td>
<td>ढऱप (nāl)</td>
<td>1683</td>
</tr>
<tr>
<td>देर (dēr)</td>
<td>से (sē)</td>
<td>आना (ānā)</td>
<td>ढऱप (nāl)</td>
<td>4720</td>
</tr>
<tr>
<td>कम (kam)</td>
<td>से (sē)</td>
<td>कम (kam)</td>
<td>द (tōm)</td>
<td>3825</td>
</tr>
<tr>
<td>रखने (rakhnē)</td>
<td>से (sē)</td>
<td>उन्हें (unhēm)</td>
<td>ढऱप (nāl)</td>
<td>853</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>previous1Word</th>
<th>previous2Word</th>
<th>ambHindiWord</th>
<th>punjabiWord</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>भोह (mōh)</td>
<td>भामा (māyā)</td>
<td>और (aur)</td>
<td>अऩ (atē)</td>
<td>4138</td>
</tr>
<tr>
<td>अलावा (alāvā)</td>
<td>कोई (kōī)</td>
<td>और (aur)</td>
<td>देव (hōr)</td>
<td>2960</td>
</tr>
<tr>
<td>अंदर (andar)</td>
<td>ही (hī)</td>
<td>से (sē)</td>
<td>द (tōm)</td>
<td>2105</td>
</tr>
<tr>
<td>ट (ṭūṭ)</td>
<td>गमा (gayā)</td>
<td>और (aur)</td>
<td>भउ (atē)</td>
<td>3198</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ambHindiWord</th>
<th>next1Word</th>
<th>next2Word</th>
<th>punjabiWord</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>और (aur)</td>
<td>मोह (mōh)</td>
<td>माया (māyā)</td>
<td>भउ (atē)</td>
<td>2418</td>
</tr>
<tr>
<td>से (sē)</td>
<td>समरथन (samrthan)</td>
<td>वापस (vāpas)</td>
<td>द (tōm)</td>
<td>3185</td>
</tr>
<tr>
<td>से (sē)</td>
<td>मुलाकात (mulākāt)</td>
<td>की (ki)</td>
<td>ढऱप (nāl)</td>
<td>4190</td>
</tr>
<tr>
<td>कर (kar)</td>
<td>अफिक (ařikit)</td>
<td>अपनी (apnī)</td>
<td>व (kar)</td>
<td>964</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>previousWord</th>
<th>ambHindiWord</th>
<th>punjabiWord</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>और (aur)</td>
<td>मोह (mōh)</td>
<td>माया (māyā)</td>
<td>2418</td>
</tr>
<tr>
<td>से (sē)</td>
<td>समरथन (samrthan)</td>
<td>वापस (vāpas)</td>
<td>3185</td>
</tr>
<tr>
<td>से (sē)</td>
<td>मुलाकात (mulākāt)</td>
<td>की (ki)</td>
<td>4190</td>
</tr>
<tr>
<td>कर (kar)</td>
<td>अफिक (ařikit)</td>
<td>अपनी (apnī)</td>
<td>964</td>
</tr>
</tbody>
</table>
This database can be extended at any time to allow new entries in the dictionary to be added.

**Analysis:**

The analysis was done on a document of 100 pages consisting of 3,58,874 words. It was found that about 8.4% words were ambiguous among these. Out of these 8.4%, approximately 70% words were correctly disambiguated. Following table shows the contributions of various bigrams and trigrams databases mentioned above in disambiguating these words:

**Table 5.19: Contribution of various N-Grams in resolving ambiguity**

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Table Name</th>
<th>Contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>bigramsRight</td>
<td>28.15%</td>
</tr>
<tr>
<td>2.</td>
<td>bigramsLeft</td>
<td>38.95%</td>
</tr>
</tbody>
</table>
Thus, it is clear from above table that context of next two words for an ambiguous word helps the most in disambiguating the sense of the word.

5.2.5 Handling Unknown Words

5.2.5.1 Word Inflectional Analysis and generation

In linguistics, a suffix (also sometimes called a *postfix* or *ending*) is an affix which is placed after the stem of a word. Common examples are case endings, which indicate the grammatical case of nouns or adjectives, and verb endings. Hindi is a (relatively) free word-order and highly inflectional language. Following table shows the Hindi Suffix List:

<table>
<thead>
<tr>
<th>आ</th>
<th>आएं</th>
<th>आता</th>
<th>आने</th>
<th>एगा</th>
</tr>
</thead>
<tbody>
<tr>
<td>इ</td>
<td>आओं</td>
<td>आती</td>
<td>आंगा</td>
<td>एगी</td>
</tr>
<tr>
<td>ई</td>
<td>इया</td>
<td>ई</td>
<td>इंजी</td>
<td>आएगा</td>
</tr>
<tr>
<td>ऊ</td>
<td>इयों</td>
<td>ऊति</td>
<td>आऊंगा</td>
<td>आएगी</td>
</tr>
<tr>
<td>ऋ</td>
<td>आइयां</td>
<td>ऊते</td>
<td>आऊंगी</td>
<td>आया</td>
</tr>
<tr>
<td>ए</td>
<td>आइयों</td>
<td>ऊता</td>
<td>ऊंगे</td>
<td>आए</td>
</tr>
<tr>
<td>ओ</td>
<td>ओं</td>
<td>ऊती</td>
<td>ऊंगी</td>
<td>आई</td>
</tr>
<tr>
<td>एं</td>
<td>इयों</td>
<td>ऊती</td>
<td>ऊंगे</td>
<td>आई</td>
</tr>
<tr>
<td>ओं</td>
<td>आइयों</td>
<td>ऊते</td>
<td>ऊंगी</td>
<td>इए</td>
</tr>
<tr>
<td>आं</td>
<td>ऊता</td>
<td>ऊना</td>
<td>ऊंगी</td>
<td>आओ</td>
</tr>
<tr>
<td>उआं</td>
<td>ऊताएं</td>
<td>ऊनी</td>
<td>ऊंगी</td>
<td>आइए</td>
</tr>
</tbody>
</table>

Table 5.20: Inflections in Hindi
A detailed analysis of noun, adjective, and verb inflections that were used to create this list can be found in McGregor [214] and Rao[215]. A few examples of each type are given below:

**Noun Inflections:** Nouns in Hindi are inflected based on the case (direct or oblique), the number (singular or plural), and the gender (masculine or feminine). For example, लड़का (laḍkā) becomes लड़के (laḍkē) when in oblique case, and the plural लड़के (laḍkē) becomes लड़कों (laḍkōṃ). The feminine noun लड़की (laḍkī) is inflected as लड़क्रकमां (laḍkīyāṃ) and लड़क्रकमों (laḍkīyōṃ), but it remains uninflected in the singular direct case.

**Adjective Inflections:** Adjectives which end in आ (ā) or आं (āṃ) in their direct singular masculine form agree with the noun in gender, number, and case. For example, the singular direct अच्छा (acchā) is inflected as अच्छे (acchē) in all other masculine forms, and as अच्छी (acchē) in all feminine forms. Other adjectives are not inflected.

**Verb Inflections:** Hindi verbs are inflected based on gender, number, person, tense, aspect, modality, formality, and voice. Rao [215] provides a complete list of verb inflection rules.
Because of same origin, both languages have very similar structure and grammar. The difference is only in words and in pronunciation e.g. in Hindi it is लड़का (laḍkā) and in Punjabi the word for boy is ਮੰਚਾ (munḍā) and even sometimes that is also not there like घर (ghar) and ਘਰ (ghar). The inflection forms of both these words in Hindi and Punjabi are also similar. In this activity, inflectional analysis without using morphology has been performed for all those tokens that are not processed in the previous activities of pre-processing and translation engine phases. Thus, for performing inflectional analysis, rule based approach has been followed. For this purpose, inflectional rules are also derived from the morphological analysis developed by IIIT, Hyderabad. This morphological analyzer works for Linux platform. First it was converted to work on Windows platform and then inflection rules were extracted from it for Hindi language. These rules were used for writing the rules for equivalent Punjabi inflections. These inflection rules resulted for Hindi to Punjabi translation purpose are implemented using regular expressions. The suffix separation module is based on the Hindi stemmer presented in Ananthakrishnan and Rao [216], and works by separating from each word the longest possible suffix Hindi Suffix List. When the token is passed to this sub phase for inflectional analysis, If any pattern of the regular expression (inflection rule) matches with this token, that rule is applied on the token and its equivalent translation in Punjabi is generated based on the matched rule(s). There is also a check on the generated word
for its correctness. We are using correct Punjabi words database for testing the correctness of the generated word. This generated Punjabi word is matched with some entry in punjabiUnigrams database. The database punjabiUnigrams is a collection of about 2,00,000 Punjabi words from large Punjabi corpus analysis. Punjabi corpus has been collected from various resources like online Punjabi newspapers, blogs, articles etc. If there is a match, the generated Punjabi word is considered a valid Punjabi word. If there is no match, this input token is forwarded to the transliteration activity.

The advantage of using punjabiUnigrams database is that ingenuine Punjabi words will not become the part of translation. If the wrong words are generated by inflectional analysis module, it will not be passed to translation rather it will be treated as out-of vocabulary and will be transliterated.

It has been analyzed that when this module was tested on the Hindi corpus of about 50,000 words, approx. 10,000 distinct words passed through this phase. And out of these 10,000 words, approx. 7,000 words were correctly generated and even accepted by Punjabi unigrams database. But rest was either generated wrong and was simply transliterated. Following table shows the correct accepted words generated by the inflectional analysis:

<table>
<thead>
<tr>
<th>रिवाजों (rivājōṃ)</th>
<th>खिद्द्वा (rivājāṃ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>समाजवादियों (samājvādiyōṃ)</td>
<td>समाजवादियों (samājvādiyōṃ)</td>
</tr>
</tbody>
</table>

Table 5.21: List of Correct accepted words in translation after inflectional analysis and generation

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
Following table shows the failure cases through inflectional analysis:

**Table 5.22: Failure cases during inflectional analysis and generation**

<table>
<thead>
<tr>
<th>Input Token</th>
<th>Translation generated</th>
</tr>
</thead>
<tbody>
<tr>
<td>धमक्रियां (dhamkiyāṃ)</td>
<td>धमक्रियां (dhamkiyāṃ)</td>
</tr>
<tr>
<td>उपलब्धियां (uplabdhiyāṃ)</td>
<td>उपलब्धियां (uplabdhiyāṃ)</td>
</tr>
<tr>
<td>तहकें (tāktēṃ)</td>
<td>उजाम (tāktēṃ)</td>
</tr>
<tr>
<td>जाभें (jāōgī)</td>
<td>जाभें (jāōgī)</td>
</tr>
<tr>
<td>निकलें (niklēṅgē)</td>
<td>निकलें (niklēṅgē)</td>
</tr>
<tr>
<td>सीमाओं (sīmāōm)</td>
<td>सीमाओं (sīmāōm)</td>
</tr>
<tr>
<td>उठा (uṭhā)</td>
<td>उठा (uṭhā)</td>
</tr>
<tr>
<td>पहुँचें (pahuńctē)</td>
<td>पहुँचें (pahuńctē)</td>
</tr>
<tr>
<td>दिखाए (dikhāiē)</td>
<td>दिखाए (dikhāiē)</td>
</tr>
<tr>
<td>प्रतियोगियां (pratiyōgiyōm)</td>
<td>प्रतियोगियां (pratiyōgiyōm)</td>
</tr>
<tr>
<td>जाभें (jāōgī)</td>
<td>जाभें (jāōgī)</td>
</tr>
<tr>
<td>नवाजा (navājā)</td>
<td>नवाजा (navājā)</td>
</tr>
<tr>
<td>जलाक (jalākar)</td>
<td>जलाक (jalākar)</td>
</tr>
<tr>
<td>धुआं (dhuāṃ)</td>
<td>धुआं (dhuāṃ)</td>
</tr>
<tr>
<td>टटोरने (ṭaṭōrē)</td>
<td>टटोरने (ṭaṭōrē)</td>
</tr>
<tr>
<td>आँका (āṅkā)</td>
<td>आँका (āṅkā)</td>
</tr>
<tr>
<td>Punjabi Words</td>
<td>Correct Hindi Words</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>लडकियाँ (&lt;i&gt;laḍkiyāṁ&lt;/i&gt;)</td>
<td>ठडकिया (&lt;i&gt;laḍkiām&lt;/i&gt;)</td>
</tr>
<tr>
<td>तारेगना (&lt;i&gt;tārēgnā&lt;/i&gt;)</td>
<td>उत्तरेगनिभा (&lt;i&gt;tārēgniā&lt;/i&gt;)</td>
</tr>
<tr>
<td>शुजा (&lt;i&gt;shujā&lt;/i&gt;)</td>
<td>मुनिभा (&lt;i&gt;shujīā&lt;/i&gt;)</td>
</tr>
<tr>
<td>मुंधा (&lt;i&gt;mugdhā&lt;/i&gt;)</td>
<td>मुंगधियभा (&lt;i&gt;mugdhiā&lt;/i&gt;)</td>
</tr>
<tr>
<td>किराणा (&lt;i&gt;kirānā&lt;/i&gt;)</td>
<td>बिराविभा (&lt;i&gt;kirāniā&lt;/i&gt;)</td>
</tr>
<tr>
<td>उत्रवा (&lt;i&gt;utravā&lt;/i&gt;)</td>
<td>उत्रविभा (&lt;i&gt;utraviā&lt;/i&gt;)</td>
</tr>
<tr>
<td>तारेगना (&lt;i&gt;tārēgnā&lt;/i&gt;)</td>
<td>उत्तरेगनिभा (&lt;i&gt;tārēgniā&lt;/i&gt;)</td>
</tr>
<tr>
<td>पटाखा (&lt;i&gt;paṭākhā&lt;/i&gt;)</td>
<td>पटाकियभा (&lt;i&gt;paṭākhiā&lt;/i&gt;)</td>
</tr>
</tbody>
</table>

The above Punjabi words are not correct and are not present in the Punjabi language vocabulary. Thus, these words have not been passed by punjabiUnigrams database and thus rejected. This step helps in improving the accuracy of the translation system.
Following flow chart presents its working:

Word generated by tokenizer as input

Look for a match of a rule among all the rules related to inflectional analysis

Is there a match?

Yes

Apply the rule and generate Punjabi Word

Is this generated Punjabi word found in punjabiUnigrams database?

No

Forward this word for transliteration activity

Yes

Generated Punjabi word will be added to output

Figure 5.1: Flow Chart for Word Inflectional Analysis and generation
Following table shows various inflectional rules, each illustrated with example:

Table 5.23: Inflection Rules

<table>
<thead>
<tr>
<th>Rule No.</th>
<th>Substring at the end of Hindi Word</th>
<th>Hindi Example</th>
<th>String to be replaced</th>
<th>Punjabi Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ििूगा (iyūṅgā)</td>
<td>पियूगा (piyūṅgā)</td>
<td>िीटच्चा (īvāṅgā)</td>
<td>पीटच्चा (pīvāṅgā)</td>
</tr>
<tr>
<td>2.</td>
<td>ििूगी (iyūṅgī)</td>
<td>पियूगी (piyūṅgī)</td>
<td>िीटच्छी (īvāṅgī)</td>
<td>पीटच्छी (pīvāṅgī)</td>
</tr>
<tr>
<td>3.</td>
<td>िेयूगा (iıyūṅgā)</td>
<td>जीयूगा (jīyūṅgā)</td>
<td>िीटच्छा (īvāṅgā)</td>
<td>पीटच्छा (pīvāṅgā)</td>
</tr>
<tr>
<td>4.</td>
<td>िेयूगी (iıyūṅgī)</td>
<td>जीयूगी (jīyūṅgī)</td>
<td>िीटच्छी (īvāṅgī)</td>
<td>पीटच्छी (pīvāṅgī)</td>
</tr>
<tr>
<td>5.</td>
<td>िेयूगा (iıyūṅgā)</td>
<td>जीयूगा (jīyūṅgā)</td>
<td>िीटच्छा (īvāṅgā)</td>
<td>पीटच्छा (pīvāṅgā)</td>
</tr>
<tr>
<td>6.</td>
<td>िेयूगी (iıyūṅgī)</td>
<td>जीयूगी (jīyūṅgī)</td>
<td>िीटच्छी (īvāṅgī)</td>
<td>पीटच्छी (pīvāṅgī)</td>
</tr>
<tr>
<td>7.</td>
<td>िेयॉगा (iīyūṅgā)</td>
<td>जीयॉगा (jīyūṅgā)</td>
<td>िीटच्छा (īvāṅgā)</td>
<td>पीटच्छा (pīvāṅgā)</td>
</tr>
<tr>
<td>8.</td>
<td>िेयॉगी (iīyūṅgī)</td>
<td>जीयॉगी (jīyūṅgī)</td>
<td>िीटच्छी (īvāṅgī)</td>
<td>पीटच्छी (pīvāṅgā)</td>
</tr>
<tr>
<td>9.</td>
<td>िेयॉगा (iīyūṅgā)</td>
<td>जीयॉगा (jīyūṅgā)</td>
<td>िीटच्छा (īvāṅgā)</td>
<td>पीटच्छा (pīvāṅgā)</td>
</tr>
<tr>
<td>10.</td>
<td>िेयॉगी (iīyūṅgī)</td>
<td>जीयॉगी (jīyūṅgī)</td>
<td>िीटच्छी (īvāṅgī)</td>
<td>पीटच्छी (pīvāṅgā)</td>
</tr>
<tr>
<td>11.</td>
<td>िेयॉइंगी (iīyēṅgī)</td>
<td>जीयॉइंगी (jīyēṅgī)</td>
<td>िीटच्छा (īvāṅgī)</td>
<td>पीटच्छा (pīvāṅgī)</td>
</tr>
<tr>
<td>12.</td>
<td>िेयॉइंगे (iīyēṅgē)</td>
<td>जीयॉइंगे (jīyēṅgē)</td>
<td>िीटच्छा (īvāṅgī)</td>
<td>पीटच्छा (pīvāṅgī)</td>
</tr>
<tr>
<td>13.</td>
<td>ााईंगीं (āaiṅgīṃ)</td>
<td>पाईंगीं (pāiṅgīṃ)</td>
<td>ााआंगीं (āavāṅgīṃ)</td>
<td>पााआंगीं (pāavāṅgīṃ)</td>
</tr>
<tr>
<td>14.</td>
<td>ााआंगा (āyēgā)</td>
<td>पााआंगा (pāyēgā)</td>
<td>ााआंगा (āvēgā)</td>
<td>पााआंगा (pāvēgā)</td>
</tr>
<tr>
<td>15.</td>
<td>ााआंगी (āyēgī)</td>
<td>पााआंगी (pāyēgī)</td>
<td>ााआंगी (āvēgī)</td>
<td>पाा�ंगी (pāvēgī)</td>
</tr>
<tr>
<td>16.</td>
<td>ााआंगा (āyēgā)</td>
<td>पााआंगा (pāyēgā)</td>
<td>ााआंगा (āvēgā)</td>
<td>पााआंगा (pāvēgā)</td>
</tr>
<tr>
<td>17.</td>
<td>ााआंगी (āyēgī)</td>
<td>पाा�ंगी (pāyēgī)</td>
<td>ााआंगी (āvēgī)</td>
<td>पााआंगी (pāvēgī)</td>
</tr>
<tr>
<td>18.</td>
<td>ााआंगी (āyēgī)</td>
<td>पााआंगी (pāyēgī)</td>
<td>ााआंगी (āvēgī)</td>
<td>पााआंगी (pāvēgī)</td>
</tr>
<tr>
<td>नं</td>
<td>विषय (विषय)</td>
<td>विषय (विषय)</td>
<td>विषय (विषय)</td>
<td>विषय (विषय)</td>
</tr>
<tr>
<td>-----</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
<td>---------------</td>
</tr>
<tr>
<td>19.</td>
<td>जिएगा (jięgā)</td>
<td>दीजिएगा (diįjįęgā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>20.</td>
<td>जीएगा (jięgā)</td>
<td>दीजीएगा (diįjįęgā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>21.</td>
<td>जियेगा (jįyįgā)</td>
<td>दीजियेगा (diįjįyįgā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>22.</td>
<td>येंगी (yęrıgı)</td>
<td>जीयेंगी (jiyęrıgı)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>23.</td>
<td>येंगी (yęrıgı)</td>
<td>जीयेंगी (jiyęrı gı)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>24.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>25.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>26.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>27.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>28.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>29.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>30.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>31.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>32.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>33.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>34.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>35.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>36.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>37.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
<tr>
<td>38.</td>
<td>निंगा (iūngā)</td>
<td>पिंगा (piūngā)</td>
<td>दी (ô)</td>
<td>दी (diô)</td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com) 10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
<table>
<thead>
<tr>
<th></th>
<th>Hindi</th>
<th>Punjabi</th>
<th>Translation</th>
<th>Transcription</th>
</tr>
</thead>
<tbody>
<tr>
<td>39.</td>
<td>वाना (vānā)</td>
<td>पकवाना (pakvānā)</td>
<td>वाना (vānā)</td>
<td>पकवाना (pakvānā)</td>
</tr>
<tr>
<td>40.</td>
<td>आएगा (āēgā)</td>
<td>पकाएगा (pakāēgā)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>41.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>42.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>43.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>44.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>45.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>46.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>47.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>48.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>49.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>50.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>51.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>52.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>53.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>54.</td>
<td>आएआइ (āēai)</td>
<td>पकाएआइ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
<tr>
<td>55.</td>
<td>आएआइ (āēai)</td>
<td>पकाए�इ (pakāēai)</td>
<td>आइ (āē)</td>
<td>पकाई (pakāē)</td>
</tr>
</tbody>
</table>
Database Design for punjabiUnigrams:

Following table shows the database design for punjabiUnigrams:

<table>
<thead>
<tr>
<th>Punjabi Unigram</th>
<th>Frequency</th>
<th>Stores the Punjabi word</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punjabi Unigram</td>
<td>Stores the frequency of this word in the analyzed Punjabi corpus.</td>
<td></td>
</tr>
</tbody>
</table>

Sample Entries for punjabiUnigrams database:

Following table shows the sample entries for punjabiUnigrams database:

<table>
<thead>
<tr>
<th>Punjabi Unigram</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;atē&quot; (atē)</td>
<td>81897</td>
</tr>
</tbody>
</table>
5.1.5.2 Transliteration

With the advent of new technology and the food of information through the Web, it has become increasingly common to adopt foreign words into one's language. This usually entails adjusting the adopted word's original pronunciation to follow the phonological rules of the target language, along with modification of its orthographical form. This phonetic “translation” of foreign words is called transliteration. Transliteration is a process that takes a character string in a source language and generates equivalent mapped character string in the target language. One of the most frequent problems translators must deal with is translating proper names and technical terms. Such terms are not translated rather are transliterated. Transliteration maps the letters of source script to letters of pronounced similarly in target script. Transliteration is particularly used to translate proper names and technical terms from languages. For example the word विशाल (vishāl) is transliterated as ਵਿਸ਼ਵਾਲ (vishāl) whereas translated as ਵਦਦਰ (vaddā). There must be some method in every Machine Translation system for words like technical terms and proper names of persons, places, objects etc. that cannot be found in
translation resources such as Hindi-Punjabi bilingual dictionary, surnames database, titles database etc and transliteration is an obvious choice for such words. It is the process of converting characters in one alphabet into another alphabet.

**Principles for Transliteration**

Following are some of the principles for a transliteration system:

- **Partial Reversibility:** Two segments of text in the target script, arising from the same source script, are to be the same if, and only if, the segments in the source script are either identical or use alternative orthography.

- **Uniformity:** Two segments of text in the target script, arising from different source scripts, are to be the same if, and only if, the two segments in the source scripts correspond precisely, according to comparative linguistics.

- **Compromise:** One symbol may have different meanings if its interpretation is never in doubt. Compromise is also necessary whenever two of these principles conflict.

- **Readability:** Even if the text is meant for computer processing, it needs to be read easily.

- **Economy:** This will also improve readability.
Approximation: The symbol used should remind one of sound, and of the transliteration scheme used for printing.

Transliteration Guidelines

The following lists the general guidelines for transliterations:

Complete: Every well-formed sequence of characters in the source script should transliterate to a sequence of characters from the target script.

Predictable: The letters themselves (without any knowledge of the languages written in that script) should be sufficient for the transliteration, based on a relatively small number of rules. This allows the transliteration to be performed mechanically.

Pronounceable: Transliteration is not as useful if the process simply maps the characters without any regard to their pronunciation. Simply mapping by alphabetic order could yield strings that might be complete and unambiguous, but the pronunciation would be completely unexpected.

Unambiguous: It is possible to recover the text in the source script from the transliteration in the target script. That is, someone that knows the transliteration rules would be able to recover the precise spelling of the original source text.

Partial Reversibility: In script transliteration, there are cases where all characters in the source script may not have one-to-one mapping for transliteration in the target script. To preserve pronunciation these characters
may mapped to some character or sequence of characters that may produce a similar sound. In such cases reversibility will be incomplete.

**History of Transliteration**

- **1885** — The American Library Association [ALA] creates a system for representing Cyrillic characters. No diacritics are used. (e.g. zh, kh, tch, sh, shtch, ye [for jat], yu, ya) Reverse transliteration is not considered.

- **1898** — The Prussian Instructions (Preussische Instruktionen [PI]) are created, which use a system of transliteration based on the Croatian model (with diacritics).

- **1909** — The ALA and British Library Association [BLA] allow for two systems, the ALA system and one based on Croatian.

- **1905** — Library of Congress creates their system, which is virtually identical to what is used today.

- **1917**— The British Academy creates its own system. Like many other systems. It does not take into account reverse transliteration.

- **1930s**— Central European and Scandinavian countries adopt the Prussian Instructions [PI]. This system was based on the Croatian model. Exceptions were made for German speaking countries, where "ch" was used instead of "h" for Cyrillic "x"

- In France the Bibliotheque Nationale adopts a purely phonetic rendering following French spelling conventions (transcription rather than transliteration).

- **1953** — The British Royal Society [BRS] creates another system,
covering Russian, Serbian & Bulgarian (but not Ukrainian, Macedonian or Belorusian).

- **1954** — The International Organization for Standardization [ISO] creates ISO/R9. Based on Croatian, this transliteration system is very close to the PI system.

- **1959** — The British Standards Institution [BS/BSI] rejects ISO/R9 (because of its reliance on multiple diacritics) and comes up with its own system: BS 2979. Very close to the British Royal Society system. (This system is used by Chemical Abstracts).

- **1976** — The American National Standards Institute [ANSI] publishes their system, nearly identical to the BSI system.

- **1968** — ISO/R9:1968 is relaxed to allow for the ANSI and BS 2979 systems (in certain countries).

- **1995** — ISO/R9:1995 reverts to to its initial standards, doing away with allowing "ch" or "kh" for Cyrillic "x."

Transliteration Models:

Four machine transliteration models have been proposed by several researchers: grapheme-based transliteration model \( (\Psi_G) \), phoneme-based transliteration model \( (\Psi_P) \), hybrid transliteration model \( (\Psi_H) \) and correspondence-based transliteration model \( (\Psi_C) \). These models are classified in terms of the units to be transliterated. The \( \Psi_G \) is sometimes referred to as the *direct method* because it directly transforms source language graphemes into target language graphemes without any phonetic
knowledge of the source language words. The $\Psi_P$ is sometimes referred to as the *pivot method* because it uses source language phonemes as a pivot when it produces target language graphemes from source language graphemes.

The $\Psi_H$ and $\Psi_C$ make use of both source language graphemes and source language phonemes when producing target language transliterations. Hereafter, we refer to a source language grapheme as a *source grapheme*, a source language phoneme as a *source phoneme*, and a target language grapheme as a *target grapheme*.

The transliterations produced by the four models usually differ because the models use different information. Generally, transliteration is a phonetic process, as in $\Psi_P$, rather than an orthographic one, as in $\Psi_G$. However, standard transliterations are not restricted to phoneme-based transliterations.

A review of the archives of Indian language documents on the Internet reveals several other schemes of Transliteration and fonts. The Indology site in England has electronic texts of Sanskrit Documents prepared in CSX format, a special input method recommended in 1990 for Sanskrit data entry using a DOS feature called Code page switching. ITRANS which is more recent offers conversion facilities to convert from CSX to the ITRANS format. The Tamil archives of the Institute of Indology and Tamil Studies in Germany (IITS) has an archive of texts of Tamil Sangam literature and many Sanskrit documents. These archives are based on the transliteration scheme recommended by the University of Madras, a fairly well known and accepted standard.
Transliteration among Indian scripts is easily achieved using ISCII (Indian Script Code for Information Interchange). ISCII has been designed using the phonetic property of Indian scripts and caters to the superset of all Indian scripts. By attaching an appropriate script rendering mechanism to ISCII, transliteration from one Indian script to another is achieved in a natural way. Transliteration schemes have to face the problem of letters present in one language and not in the other. Thus, unless a superset of letters from all the Indian Languages is formed, uniform transliteration is ruled out. Ram Viswanadha [217] has the view that when characters do not have any appropriate transliteration they should be consumed and not replaced with any other character. This results in partial loss of reversibility.

**Our Approach**

Although Hindi and Punjabi are closely related languages and for except few cases all alphabets of Devanagri script are present in Gurmukhi script, the task of transliteration from Hindi to Punjabi is not trivial. The Unicode encoding has eased the problem to some extent. In our system besides direct character mappings from alphabet in one script to another, rule based transliteration useful for a translation system is also employed to improve its accuracy. Using only direct character mapping, it shows that this word is out-of-vocabulary for our system and has been displayed in the output by changing the script and is unknown to our system. Both direct character
mapping and complex rules employed for transliteration are explained in the following sections.

**Direct Character Mappings:**

Both Hindi and Punjabi languages are phonetic languages and their scripts represent the phonetic repository of their respective languages. These phonetic sounds are used to determine the relations between the characters of two scripts. On the basis of this idea, character mappings are determined. With this system every alphabet can be uniquely mapped to the corresponding alphabet as shown in following table. Taking into account the similarity of both the scripts, letter to letter mapping is the obvious choice for baseline. Following table 5.26 shows the direct mapping of Hindi to Punjabi alphabets:

**Table 5.26: Direct Hindi to Punjabi Character Mapping**

<table>
<thead>
<tr>
<th>Hindi Character</th>
<th>Decimal Code</th>
<th>Punjabi Character</th>
<th>Decimal Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>ए</td>
<td>2305</td>
<td>ॐ or ॐ</td>
<td>2562 or 2672</td>
</tr>
<tr>
<td>ए</td>
<td>2306</td>
<td>ॐ or ॐ</td>
<td>2562 or 2672</td>
</tr>
<tr>
<td>ए</td>
<td>2307</td>
<td>ॐ or ॐ</td>
<td>2562 or 2672</td>
</tr>
<tr>
<td>अ</td>
<td>2309</td>
<td>अ</td>
<td>2565</td>
</tr>
<tr>
<td>आ</td>
<td>2310</td>
<td>आ</td>
<td>2566</td>
</tr>
<tr>
<td>ह</td>
<td>2311</td>
<td>हि</td>
<td>2567</td>
</tr>
<tr>
<td>है</td>
<td>2312</td>
<td>है</td>
<td>2568</td>
</tr>
<tr>
<td>उ</td>
<td>2313</td>
<td>उं</td>
<td>2569</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>ऊ</td>
<td>2314</td>
<td>ऊं</td>
<td>2570</td>
</tr>
<tr>
<td>ऋ</td>
<td>2315</td>
<td>ऋं</td>
<td>2608+2623</td>
</tr>
<tr>
<td>ए</td>
<td>2317</td>
<td>एं</td>
<td>2575</td>
</tr>
<tr>
<td>ऐ</td>
<td>2318</td>
<td>ऐं</td>
<td>2575</td>
</tr>
<tr>
<td>ए</td>
<td>2319</td>
<td>एं</td>
<td>2575</td>
</tr>
<tr>
<td>ऐ</td>
<td>2320</td>
<td>ऐं</td>
<td>2575</td>
</tr>
<tr>
<td>ओ</td>
<td>2321</td>
<td>ओं</td>
<td>2566</td>
</tr>
<tr>
<td>औ</td>
<td>2322</td>
<td>औं</td>
<td>2580</td>
</tr>
<tr>
<td>औ</td>
<td>2323</td>
<td>औं</td>
<td>2579</td>
</tr>
<tr>
<td>औ</td>
<td>2324</td>
<td>औं</td>
<td>2580</td>
</tr>
<tr>
<td>क</td>
<td>2325</td>
<td>कं</td>
<td>2581</td>
</tr>
<tr>
<td>ख</td>
<td>2326</td>
<td>खं</td>
<td>2582</td>
</tr>
<tr>
<td>ग</td>
<td>2327</td>
<td>गं</td>
<td>2583</td>
</tr>
<tr>
<td>घ</td>
<td>2328</td>
<td>घं</td>
<td>2584</td>
</tr>
<tr>
<td>ङ</td>
<td>2329</td>
<td>ङं</td>
<td>2585</td>
</tr>
<tr>
<td>च</td>
<td>2330</td>
<td>चं</td>
<td>2586</td>
</tr>
<tr>
<td>छ</td>
<td>2331</td>
<td>छं</td>
<td>2587</td>
</tr>
<tr>
<td>ज</td>
<td>2332</td>
<td>जं</td>
<td>2588</td>
</tr>
<tr>
<td>झ</td>
<td>2333</td>
<td>झं</td>
<td>2589</td>
</tr>
<tr>
<td>ञ</td>
<td>2334</td>
<td>ञं</td>
<td>2590</td>
</tr>
<tr>
<td>ट</td>
<td>2335</td>
<td>ट</td>
<td>2591</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>ठ</td>
<td>2336</td>
<td>ठ</td>
<td>2592</td>
</tr>
<tr>
<td>ड</td>
<td>2337</td>
<td>ड</td>
<td>2593</td>
</tr>
<tr>
<td>ढ</td>
<td>2338</td>
<td>ढ</td>
<td>2594</td>
</tr>
<tr>
<td>ण</td>
<td>2339</td>
<td>ण</td>
<td>2595</td>
</tr>
<tr>
<td>त</td>
<td>2340</td>
<td>त</td>
<td>2596</td>
</tr>
<tr>
<td>थ</td>
<td>2341</td>
<td>थ</td>
<td>2597</td>
</tr>
<tr>
<td>द</td>
<td>2342</td>
<td>द</td>
<td>2598</td>
</tr>
<tr>
<td>ध</td>
<td>2343</td>
<td>ध</td>
<td>2599</td>
</tr>
<tr>
<td>न</td>
<td>2344</td>
<td>न</td>
<td>2600</td>
</tr>
<tr>
<td>ल</td>
<td>2345</td>
<td>ल</td>
<td>2600</td>
</tr>
<tr>
<td>प</td>
<td>2346</td>
<td>प</td>
<td>2602</td>
</tr>
<tr>
<td>फ</td>
<td>2347</td>
<td>फ</td>
<td>2603</td>
</tr>
<tr>
<td>ब</td>
<td>2348</td>
<td>ब</td>
<td>2604</td>
</tr>
<tr>
<td>भ</td>
<td>2349</td>
<td>भ</td>
<td>2605</td>
</tr>
<tr>
<td>म</td>
<td>2350</td>
<td>म</td>
<td>2606</td>
</tr>
<tr>
<td>य</td>
<td>2351</td>
<td>य</td>
<td>2607</td>
</tr>
<tr>
<td>र</td>
<td>2352</td>
<td>र</td>
<td>2608</td>
</tr>
<tr>
<td>ल</td>
<td>2353</td>
<td>ल</td>
<td>2608</td>
</tr>
<tr>
<td>ठ</td>
<td>2354</td>
<td>ठ</td>
<td>2610</td>
</tr>
<tr>
<td>ढ</td>
<td>2355</td>
<td>ढ</td>
<td>2611</td>
</tr>
<tr>
<td>थ</td>
<td>2356</td>
<td>ढ</td>
<td>2611</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>व</td>
<td>2357</td>
<td>व</td>
<td>2613</td>
</tr>
<tr>
<td>श</td>
<td>2358</td>
<td>म</td>
<td>2614</td>
</tr>
<tr>
<td>ष</td>
<td>2359</td>
<td>म</td>
<td>2614</td>
</tr>
<tr>
<td>स</td>
<td>2360</td>
<td>म</td>
<td>2616</td>
</tr>
<tr>
<td>ह</td>
<td>2361</td>
<td>उ</td>
<td>2617</td>
</tr>
<tr>
<td>ॠ</td>
<td>2364</td>
<td>ॠ</td>
<td>2620</td>
</tr>
<tr>
<td>ा</td>
<td>2366</td>
<td>ा</td>
<td>2622</td>
</tr>
<tr>
<td>ि</td>
<td>2367</td>
<td>ि</td>
<td>2623</td>
</tr>
<tr>
<td>ी</td>
<td>2368</td>
<td>ी</td>
<td>2624</td>
</tr>
<tr>
<td>ू</td>
<td>2369</td>
<td>ू</td>
<td>2625</td>
</tr>
<tr>
<td>ृ</td>
<td>2370</td>
<td>ृ</td>
<td>2626</td>
</tr>
<tr>
<td>े</td>
<td>2375</td>
<td>े</td>
<td>2631</td>
</tr>
<tr>
<td>े</td>
<td>2376</td>
<td>े</td>
<td>2632</td>
</tr>
<tr>
<td>ो</td>
<td>2377</td>
<td>ो</td>
<td>2622</td>
</tr>
<tr>
<td>ो</td>
<td>2378</td>
<td>े</td>
<td>2635</td>
</tr>
<tr>
<td>ो</td>
<td>2379</td>
<td>े</td>
<td>2635</td>
</tr>
<tr>
<td>ौ</td>
<td>2380</td>
<td>ौ</td>
<td>2636</td>
</tr>
<tr>
<td>ू</td>
<td>2381</td>
<td>ू</td>
<td>2637</td>
</tr>
<tr>
<td>ृ</td>
<td>2384</td>
<td>छभ</td>
<td>2579+2606</td>
</tr>
<tr>
<td>क</td>
<td>2392</td>
<td>न</td>
<td>2581</td>
</tr>
<tr>
<td>ख</td>
<td>2393</td>
<td>ध</td>
<td>2649</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>श</td>
<td>2394</td>
<td>ष</td>
<td>2650</td>
</tr>
<tr>
<td>ष</td>
<td>2395</td>
<td>स</td>
<td>2651</td>
</tr>
<tr>
<td>ट</td>
<td>2396</td>
<td>ट</td>
<td>2652</td>
</tr>
<tr>
<td>ठ</td>
<td>2397</td>
<td>ठ</td>
<td>2652+2637+2617</td>
</tr>
<tr>
<td>फ</td>
<td>2398</td>
<td>फ</td>
<td>2654</td>
</tr>
<tr>
<td>घ</td>
<td>2399</td>
<td>घ</td>
<td>2607</td>
</tr>
</tbody>
</table>
Following flowchart explains the working of the transliteration phase:

1. Word generated by the tokenizer as input
2. Read one character at a time
3. Is any rule matched for the character?
   - Yes: Apply the rule
   - No: Apply direct character mapping for the character
4. Is the full word read?
   - Yes: Is the transliterated word found in punjabiUnigrams database?
     - Yes: iteration
     - No: Apply direct character mapping for transliteration without using any complex rule.
   - No: iteration
Rule Based Mapping:
Although direct character mapping can produce successfully the transliterated output in Punjabi which can represent the source word in target language, but we can improve the results by making them nearer to target language in term of spellings and choice of alphabets by using some set of rules. A quite reasonable improvement can be achieved by small amount of dependency or contextual rules. Following are the rules for alleviating some of the problems not solved by direct character mapping.

1. या at the end of the words of length greater than 3 will be transliterated into ਆ. For example: विकिपीडिया (vikīpīḍiyā) : ਵਿਕੀਪੀਡੀਆ (vikīpīḍā), ਵੇਬਦੁਨੀਆ (vēbduniyā): ਵੀਬੁਦਨੀਆ (vēbduniā)

2. Substring आ in the word of length greater than 3 will be transliterated to ਆਨ. For example: आयाम (āyām): ਆਯਾਮ (āyām), आयातित (āyātit): ਆਯਾਤਿਤ (āyātit).

3. यी at the end of the words will be transliterated into ਇ. For example: वाजपेयी (vājpēyī): ਵਾਜਪੇੜੀ (vājpēī), ਎ਮਪ्लाई (ēmplāyī): ਅਮਪਲਾਈ (amplāī).
4. य at the end of the word will be transliterated into ḍ. For example: अक्षय (akshay) : अक्षय (akshaē), समुद्रपारीय (samudrpārīy) : समुद्रपारीय (samudrapārīē).

5. या preceded by consonant or halant in the word, will be transliterated to ṭ + ṣ + ा. For example: प्यास (pyās) : प्यास (piās), ब्यास (byās): ब्यास (biās).

6. यू preceded by consonant or halant in the word, will be transliterated into ṭ + उ. For example: अब्हिमन्यू (abhimnyū) : अब्हिमन्यू (abhimniū), एविन्य (ēvinyū) : एविन्य (ēvinīū).

7. यू preceded by consonant or halant in the word, will be transliterated into ṭ + ṭ. For example: इम्युनोलॉजी (immynōlājī) : इम्युनोलॉजी (immiunōlājī), मेक्युलम (maikyulam) : मेक्युलम (maikiulam).

8. यू preceded by consonant or halant in the word, there is a consonant, will be transliterated to ṭ + े. For example: क्रियेटिव (kriyēṭīv) : क्रियेटिव (kariēṭīv), एनडीब्युर (ēndibyēr) : एनडीब्युर (ēndibēr).
9. यू preceded by consonant or halant in the word, will be transliterated into ञ + स्ये. For example: कहिल्यू(kahilyai): लालिल्यू(kahiliiai), मोंदग्यू(mōdgapyai): भेंदाप्तिल्यू(mōdgapiiai).

10. यू within the words of length greater than 2 and not at the beginning of the word, will be transliterated into े. For example: थुयेर (thuyēr):

    थुयेर (thuyēr).

11. यू within the words and not at the beginning of the word, will be transliterated into ी and ु. For example: बयूरा (bayūra): बियूरा (biūrā), कॉम्यूटर (kampyūtar): लेबिटर (kampiūtar).

12. यू preceded by matra ा and followed by consonant, will be transliterated into ी. For example: रसायनिक (rasānyik): उम्मीटिन (rasāniś), डायबिटीज (dāyiṭīj), डायबिटीन (dāibiṭīn).

13. यू at the end of the word will be transliterated into ठान्या. For example: समझाया (samjhāyā): समझान्या (samjhāiā), ठाराया (thahrāyą): ठारान्या (thahrāiā).
14. या within the word and not at the end या. For example: सजायाफ्ता
(sajāyāphtā): मन्नाफ्ता (sajāyāphātā), नायाब (nāyāb): राजाब (nāyāb).

15. य preceded by the matra य and followed by consonant in the word, will
be transliterated भ. For example: कैलिशियम (kailshiyam): कैलिशियम
(kailshiam), स्टेडियम (sṭeḍiyam): स्टेडियम (sṭeḍiām).

16. य preceded by the matra झ and followed by the matra ल in the word,
will be transliterated श. For example: मुथियाग (muthaiyā):
मुथियात (muthiā), फैयाज (phaiyāja): फैयाज (phiāj).

17. Dead Consonant क (क) followed by live consonant ख in the word, will
be transliterated into क. For example: मक्खान (makkhan): मक्खान
(makkhan), मधुमक्खी (madhumkkhī): मधुमक्खी (madhumkkhī).

18. Dead Consonant च (च) followed by live consonant छ in the word, will
be transliterated into च. For example: इच्छा (icchā): इच्छा (icchā),
अक्छ (akacch): अक्छ (akacch).
19. Dead consonant ट(ट) followed by live consonant ठ, will be transliterated into ਟ. For example: मट्ठी (matthi) : ਫਾਠੀ (bhatthā):

20. Dead consonant ग(ग) followed by live consonant घ in the word, will be transliterated into ਗ. For example: मग्घर (magghar), 

लक्कड़बग्घा (lakrabgha) : ਧੱਚ਼ਲ਼ (magghar).

21. Dead consonant ज(ज) followed by live consonant झ in the word, will be transliterated into ਜ. For example: निज्झर (nijjar), उज्झड़ (ujjar) :

22. Dead consonant त (ट) followed by live consonant थ in the word, will be transliterated into ਤ. For example: पत्थर (patthar), कत्था (katthā).

23. Dead consonant द (द) followed by live consonant ध in the word, will be transliterated into ਦ. For example: सिद्धार्थ (siddhārth), अनिरुद्ध (aniruddh).
24. Dead consonant ड(ड) followed by live consonant ढ in the word, will be transliterated into ੱ. For example: गडढा (gaddhā): गੱਢਾ (gaddhā),
बुडढा (budḍhā): बੱਢਾ (budḍhā).

25. Conjunct consonant ज in the word will be transliterated in Gurmukhi into ਲ + ਅ + ਏ. For example:खगोलवैज्ञानिक (khagolvaigānīk):

खोजीशिशिष्य (khagolvaigānīk), मनोविज्ञानी (manovigānī): ਮੰਤਰੀਕਾਲੀ ਮਾਨੋਵਿਜਾਤੀ (manovigānī).

26. Dead consonant म (म) followed by any labial consonant (प, फ, म, व),

will be transliterated into ॐ. पम्मी (pammi): ਪਤੀ (pammi), ਪਮਿ (pamp):

ਪੰਗ (pamp).

27. Dead Consonant followed by the live consonant of its preceding dead consonant in the word, will be transliterated to ॐ. This rule is not applicable for consonants म and न. For example: दिग्गज (diggaj):

दिग्गज (diggaj), छज्जा (chajjā): ਛੱਣਾ (chajjā).
28. Dead Consonant न (न) followed by live consonant न in the word, will be
transliterated to ں. For example: पन्ना(pannā) : ण्ण(panṇā), अन्नामलाई
tannāmlāi(annāmlāi), अन्नामलाई (annāmlāi).

29. ा or (ा) preceded by consonant क in the word will be transliterated
क्षभ. For Example: क्रिकेट(krikēṭ), विकेट(krikēṭ), संस्कृत(saṃskrit):
मंसव्रूत(saṃskrit).

30. (ृ) preceded by consonant प in the word will be transliterated ृव. For
Example: प्रयोगशाला(prayōgshālā): भूवेश्वर(prayōgshālā), प्रिंटिंग
(printīṅg) : प्रिंटिंग(printing).

31. Conjunct consonant ् in the word will be transliterated to वम. For
example: क्रेमेन्द्र राव(kshēmēndr rāv): क्रेमेन्द्र उद्र (kshēmēndr rāv),
कुरुक्षेत्र(kurukṣēṭr), तुल्कुमेंद्र(kurukṣētar).

32. Conjunct consonant ् त् in the word will be transliterated to उव. For
example: त्रिपला(triphlā): त्रिपला(triphlā), तंत्रिका(tantrikā): तंत्रिका(tantrikā).
33. Conjunct consonant श्र in the word will be transliterated to ਭ਩. For example: श्रवण (shravana) : ਸਾਵਨ (sharvan), श्रीलंकाई (shrilanka) : ਸ਼ਰੀਲੰਕਾਈ (shrilanka).

34. अं or ऑ preceded by either आ or ई ए or ओ or ऋ or ऊ or ऎ or ऒ or ੂ or ੃ or ੄ or ੅ or ੆ or ੇ or ੈ in the word will be transliterated to ਅ. For example: ਹਿਊਮੋਰ (hūmōr) : ਹਿਊਮੋਰ (hūmōr), ਏਂਟਰੀ (aiṇtri) : ਏਂਟਰੀ (aiṇtri).

35. अं or ऑ preceded by either ए or अ or उ or ऊ or इ or ा in the word will be transliterated to ਅ. For example: ਇੰਟਰਨੇਸ਼ਨਲ (intraishnal) : ਇੰਟਰਨੇਸ਼ਨਲ (intraishnal), ਏਂਟਰੀ (aiṇtri) : ਏਂਟਰੀ (aiṇtri).

36. अं or ऑ between two consecutive consonants in the word, will be transliterated to ਅ. For example: ਸੰਦੀਪ (sandīp) : ਸੰਦੀਪ (sandīp), ਸਟੂਇਡੰਟਸ (satūḍāntas) : ਸਟੂਇਡੰਟਸ (satūḍāntas).

When translation was performed on a document of 100 pages consisting of about 3,58,874 words, we found that about 24% of total words gets transliterated during translation process. Thus, above transliteration module plays a major role in translation.
5.3 Summary
This chapter presented the detailed working of tokenizer and the working of sub phases of translation engine phase of our Machine Translation system. The translation engine plays an important role in selecting the correct target language. Identifying surnames and titles modules detect the proper names of a person and pass them to the transliteration module. N-gram approach is employed for word sense disambiguation. The solution for handling out of vocabulary words using inflectional analysis without using morphology and transliteration activities are discussed in depth. The output generated by this phase is further refined by the post-processing phase. This phase is discussed in detail in next chapter.
Chapter 6

Post-Processing

6.1 Grammar Corrections

In spite of the great similarity between Hindi and Punjabi languages, there are still a number of important grammatical divergences: gender and number divergences which affect agreement. The grammar is incorrect or the relation of words in their reference to other words, or their dependence according to the sense is incorrect and needs to be adjusted. This phase is the tail end of our Machine Translation system. It is a sentence level post-processing module that improves the translation quality by making corrections in the translation generated. In other words, it can be said that it is a system of correction for ill-formed sentences. The output generated by the translation engine phase becomes the input for post-processing phase. This phase will correct the grammatical errors based on the rules implemented in the form of regular expressions discussed in the next section. In this section, we will discuss error categories which include those mistakes that lead to ungrammaticality, and thus need to be corrected. It is not possible to fully remove all the grammatical errors but to some extent. In each of the examples given in each error category, sentence marked with the asterisk (*) is
ungrammatical sentence and the other sentence without the asterisk (*) is the one corrected grammatically by the post processing phase.

1. Within Verb Phrase Agreement:

In a typical Punjabi sentence, within verb phrase, all the verbs must agree in gender and number.

1(a) * ਲਣਕਪਸ਼ ਦੀ ਅਬਵਾਸ ਮੁਰਦੇਹ ਦੀ ਇੱਸਟੀ ਊਠ।

(nirmalā dī avāj suṇdē hī bhajdī han.)

ਲਣਕਪਸ਼ ਦੀ ਅਬਵਾਸ ਮੁਰਦੇਹ ਦੀ ਇੱਸਟੀ ਊਠ।

(nirmalā dī avāj suṇdē hī bhajjdīm han.)

1(b) * ਉਢਾਂ ਦੂਰੀਅਂ ਊ ਸਿਕਾਦਾ ਸਿਖਾਕੁਸਲ ਬਚਾਗਾ ਤੇ ਝੁਦਰ ਦੁੱਚ ਮਿਤਵਾ ਅਪਤੀ ਅਮਨਮਲੀਅਣ

ਦੂਰੀਅਂ ਬੋਂ-ਬਾਨ ਵਲੀ ਊਠ।

(unhāṃ kampnīāṃ nūṃ jiādā viākul karēgā jō bhārat vicc sathit āpī anushṛṅgiāṃ)

ਉਢਾਂ ਅੱਪਾਂ ਊ ਸਿਕਾਦਾ ਸਿਖਾਕੁਸਲ ਬਚਾਗਾ ਤੇ ਝੁਦਰ ਦੁੱਚ ਮਿਤਵਾ ਅਪਤੀ ਅਮਨਮਲੀਅਣ

ਦੂਰੀਅਂ ਬੋਂ-ਬਾਨ ਵਲੀ ਊਠ।

(unhāṃ kampnīāṃ nūṃ jiādā viākul karēgā jō bhārat vicc sathit āpī anushṛṅgiāṃ duārā kamm-kāj karāndīm han)

1(c) *ਧਿੱਕੜੇ ਮਾਸ ਦੀਆਂ ਮਕੋਂ ਦੂਰੀਅਂ ਕੱਪ੍ਲਿਆ ਤੇ 2,76,580 ਮੇਗਾਪ੍ਰੋਸੀਲਾ ਦੇਵੀ ਮਾਲ।
2. Noun’s Oblique Form before Postpositions

In the Punjabi Sentence, the noun before the postpositions take the oblique form.

2(a) *(Mad mē) mē  appBar)*

(sabh samāṃ tē āō)

Mad mē mē appBar

(sabh samēṃ tē āō)

2(b) *(jana tē pūndīṃ) bāu apāthēṃ dīn)*

(rājā nē mantrīāṃ nūṃ ādēsh dittā.)

Jana tē pūndīṃ bāu apāthēṃ dīn

(rājē nē mantrīāṃ nūṃ ādēsh dittā)

3. Subject Verb Agreement

In Punjabi Sentence, the subject must agree with verb.

3(a) *(tusī kōī vī musmmī phal lai sakadē han)*

(tusī kōī vī musmmī phal lai sakadē han)

Language in India www.languageinindia.com 807
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
4. Verb Object noun phrase agreement if there is cāhīdā in verb phrase

In Punjabi Sentence, verb must agree with the object noun phrase if there is cāhīdā in the verb phrase.

4(a) *(prīkhiā vicc ih gall sāhmaṇē āī hai ki sarīktgrasat bairīk āph amrīkā nūṁ sabhṭūṁ jiādā 33 . 9 arab ḍālar dī pūnjī cāhīdā hai .)

(प्रीक्षित विच इह गॉल सामंत आई है कि सार्रक्षक्रसत बैर्क आफ अमरीका नूम साख्टूम जिादा 33 . 9 अरब झालर दी पुण्जी चाहिदा हा।)
6.2 Pattern matching and Regular Expressions:

Pattern matching is a searching technique employed normally on a string containing text in order to locate a portion or all of the specified data based on a specific search pattern criterion. A regular expression is a special text string for describing a search pattern. Regular expressions are a ‘way to describe text through pattern matching’ (Stubblebine 2003: 1). Regular expressions provide a powerful, flexible, and efficient method for processing text. The extensive pattern-matching notation of regular expressions allows to quickly parsing large amounts of text to find specific character patterns; to extract, edit, replace, or delete text substrings. The idea of using regular expressions for natural language processing is widely known. By using them, the most complex and repetitive linguistic errors can be identified and replaced with the right text in the MT output. The syntax of regular expressions can be simple or highly complex, depending on the pattern.

6.2.1 Related Works

The concept of automated Post-Editing was first introduced by Knight and Chander[218] and further explored by Allen and Hogan with a view to fix systematic errors committed by an MT system [219]. When these MT errors cannot be fixed with advanced User Dictionary coding techniques, they may be fixed using powerful global search and replace patterns. Roturier [220-223] used regular expressions for post-processing module of their Machine
Translation system. Kartunnen [224] suggests applying finite automata and transducers that represent regular expressions, for natural language texts. Oflazer [225] shows the use of regular expressions for tokenization, shallow parsing or morphology analysis. Hasan [226] describes the use of regular expressions for sentence clustering in SMT. Number of hybrid experiments have been conducted by combining rule-based MT (RBMT) systems with Statistical Post-Editing (SPE) systems. Two experiments were carried out for the shared task of the ACL 2007 Workshop on Statistical Machine Translation, combining a raw SYSTRAN system with a statistical post-editing (SPE) system. One experiment was run by NRC using the language pair English<>French in the context of Automatic Post-Editing systems using the PORTAGE system. The second experiment based on the same principle was run on the German > English and Spanish > English language pairs using the Moses system. The objective was to train a SMT system on a parallel corpus composed of SYSTRAN translations with the referenced source aligned with its referenced translation. A detailed evaluation of these experiments was then conducted and presented in [227,228]. They concluded that the SYSTRAN+SPE experiments demonstrated very good results – both on automatic scoring and on linguistic analysis. Their detailed comparative analysis provided directions on how to further improve these results by adding “linguistic control” mechanisms.

6.2.2 Our Approach
Here, we present the use of regular expressions in a translation system for doing post editing. The grammatical categories discussed in Section 6.1 are corrected using pattern matching through regular expressions in the MT output. It is a two step process:

(i) Pattern mentioned in regular expressions are matched with text.

(ii) If some pattern(s) matches with the strings in the text, it is replaced with the required one mentioned in pattern matched regular expression.

We have formulated 28 regular expressions for correcting such grammatical errors. Following table shows the distribution of regular expressions on the basis of error categories discussed above:

**Table 6.1: Grammatical Error Category wise Regular Expression Distribution**

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Grammatical Error Category</th>
<th>Regular Expression</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Within Verb Phrase agreement</td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>2.</td>
<td>Noun’s Oblique Form before Postpositions</td>
<td></td>
<td>02</td>
</tr>
<tr>
<td>3.</td>
<td>Subject Verb Agreement</td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>4.</td>
<td>Verb Object noun phrase agreement if there is ਚਾਹੀਣਾ (cāhīdā) in verb phrase</td>
<td></td>
<td>01</td>
</tr>
</tbody>
</table>
For instance, the following example shows the subject verb agreement through regular expression:

**MT output before post-processing**: अमीं दसं दी ठीव ठेंधवण वज मेंढें ज़ा।

*(asīṃ vālāṃ dī ṭhīk dēkhbhāl kar sakkṛ han.)*

**Search pattern**: (अमीं)(((?!\u0A05\u0A38\u0A40.).)+)(यढ)

**Replace pattern**: $1$2 Yellowstone

**MT output after post-processing**: अमीं दसं दी ठीव ठेंधवण वज मेंढें ज़ा।

*(asīṃ vālāṃ dī ṭhīk dēkhbhāl kar sakkṛ hāṃ)*

In the above example, the pattern is matched for a sentence in which subject is अमीं and verb is ज़ा. Even the regular expression has been written in complex way to handle similar nested patterns also. $1$ and $2$ are the environment variables to store the intermediate substrings matched within the pattern.

The analysis was done on a document consisting of 35500 words. It was found that 6.197% of the output text has been corrected grammatically using these regular expressions. Following table shows the contributions of various regular expression categories in correcting the grammatical errors:

**Table 6.2: % Contribution of Regular Expressions on the basis of Grammatical Error Categories**

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Grammatical Error Category</th>
<th>Regular Expression</th>
</tr>
</thead>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
### 6.3 Sample Translations:

Following are some sample translations obtained from the system:

**Input:** संभावना है कि जल्दी ही कर्नाटक के राजनीतिक भविष्य का कोई फैसला हो जाएगा

$sambhāvnā hai ki jaldī hī karnāṭak kē rājnitik bhavishy kā kōī phaaislā hō jāēgā$

**Output:** मंजरी है वि नस्त्री झी बलरूद्वर दे जननीविव इल्लिंगद एस्टी देमल्ल दे नटेरा।

$(sambhāvnā hai ki jaldī hī karnāṭak dē rājnitik bhavikkh dā kōī phaaislā hō jāvēgā)$

**Input:** लाला साहब ने अपना जीवन इसी काम के हेतु अर्पण कर दिया है।

$(lālā sāhab nē apnā jīvan isī kām dē hētu arpaṇ kar diyā hai.)$

**Output:** सल्सा मदिर ते आरप्पा चौक्क हिम वेम दे तल्ली आरप्पा बुड़ टिंद्र दे।

$(lālā sāhib nē āpanā jīvan is kamm dē laī arpaṇ kar dittā hai)$. 

**Input:** अपना बिस्तर खिड़की से सटाकर कभी भी न लगाएँ।

$(apnā bistar khirki sē sāṭākar kabhi bhī na lagāēm.)$

**Output:** आरप्पा बिमुद्वच बिडली दें मटवच वटे दौ तभी सुगाएँ।
6.4 Illustrative Example:
For illustration purpose of how the input is passed through various phases consider the following sentence in Hindi:

**Input:** हम अपने दोस्त दीपक शर्मा से पूछे कि क्या वो हमारे साथ वन डे मैच खेल कर गरीब बच्चों की मदद करना चाहेगा जैसे हम करना चाहते हैं।

*(ham apnē dōst dīpak sharmā sē pūchē kyā vō hamārē sāth van ē maic khaēl kar garīb baccōṃ kī madad karnā cāhēgā jaisē ham karnā cāhtē haiṃ).*

**Pre Processing Phase:**
- During the Text Normalization, खेल (khaēl) will be replaced with खेल (khēl)
- No named Entity is found.
- वन डे (van ē) is collocation, thus it will be replaced with रन डे (van ē) rather than its actual word to word translation जारिगल डे (jaṅgal ē).

**Intermediate Output:** हम अपने दोस्त दीपक शर्मा से पूछे कि क्या वो हमारे साथ रन डे मैच खेल कर गरीब बच्चों की मदद करना चाहेगा जैसे हम करना चाहते हैं।

*(ham apnē dōst dīpak sharmā sē pūchēngē kyā vō hamārē sāth van ē maic khēl kar garīb baccōṃ kī madad kārnā cāhēgā jaisē ham kārnā cāhtē haiṃ).*
**Tokenizer:** It will generate हम, अपने, दोस्त, दीपक, शर्मी, से, पूछेंगे, कि, क्या, वो, हमारे, साथ, दल, जे, मैच, खेल, कर, गरीब, बच्चों, की, मदद, करना, चाहेगा, जैसे, हम, करना, चाहते, हैं,। These tokens are generated one by one and passed to translation engine for processing one after another.

**Translation Engine:**
- No Title is found
- Surname शर्मी is found and thus दीपक शर्मी will be translated to दीपक शर्मा rather than दीपक मक्खा
- हम, अपने, दोस्त, पूछेंगे, कि, क्या, वो, हमारे, साथ, खेल, गरीब, बच्चों, की, मदद, करना, चाहेगा, जैसे, हम, करना, चाहते, हैं,। tokens will be translated using lexicon lookup and will be translated to अमीं, आपने, दीपक शर्मा भाषा राष्ट्र, भेंट, गरीब, बच्चों, वी, भाषा, वि, वी, धुंध, मदद, राष्ट्र, भेंट, गरीब, बच्चों, वी, भाषा, वि, दीपक शर्मा वि वी धुंध मदद राष्ट्र
- Tokens से and कर are ambiguous words and hence using word sense disambiguation approach, these will be translated to वे and वे respectively.
- मैच will be transliterated by the transliteration module to मैच.

**Intermediate Output:** अमीं आपने दोस्त दीपक मक्खा दे पूछेंगे वि वी धुंध मदद राष्ट्र
दे मैच मैच वे गरीब बच्चों वी भाषा बच्चा चाहेगा निहे अमीं बच्चा चाहेंगे गल।

(asīṃ āpaṇē dōsat dīpak sharmā tōm pucchāṅgē ki kī uh sāḍē nāl van ṅē maic khēl kē garīb bacciāṃ dī madad Karnā cāhēgā jivēṃ asīṃ Karnā cāhundē han.)
Post Processing:
Pattern matching is done using regular expressions for correcting grammar. Thus, this output will be made grammatically correct using the regular expressions implementing the grammatical agreements. In this example, subject verb agreement will be done and ਚਲਵਾ will be replaced with ਜਾਂ.

Final Output: ਅਸੀਮ ਅਪਨੇ ਟੀਮ ਚੀਫ ਮੈਸੂਰ ਜਾਂ ਪੁਰਾਣੀ ਬਿ ਦੀ ਖੇਡ ਮਹੜੀ ਚੋਰ ਦੇ ਮੇਹੇਲ ਵੇ ਰਾਜੀ ਖੋਚਨ ਲੀ ਚੋਰ ਬਚਨਾ ਨਾਲੇ ਅਸੀਮ ਬਚਨਾ ਬਚਨੇ ਜਵਾਂ।

(asīm āpaṇē dōsat dīpak sharmā tōm pucchāṅgē ki kī uh sādē nāl van ṛē maic khēl kē garīb bacciām dī madad karnā cāhēgā jivēṃ asīm kamā cāhundē hāṃ.)

Evaluation:
The evaluation document set consisted of documents from various online newspapers news, articles, blogs, biographies etc. This test bed consisted of 35500 words and was translated using our Machine Translation system. Following table shows the contribution of various important modules of the system during translation:

<table>
<thead>
<tr>
<th>Module</th>
<th>Submodule</th>
<th>Contribution (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preprocessing</td>
<td>Text Normalization</td>
<td>1.121%</td>
</tr>
</tbody>
</table>

Table 6.3 : % Contribution of Various MT System Modules during Translation
<table>
<thead>
<tr>
<th>Translation Engine</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Replacing Collocations</td>
<td>0.281%</td>
<td></td>
</tr>
<tr>
<td>Replacing Proper Nouns</td>
<td>1.408%</td>
<td></td>
</tr>
<tr>
<td>Identifying Titles</td>
<td>0.005%</td>
<td></td>
</tr>
<tr>
<td>Identifying Surnames</td>
<td>3.380%</td>
<td></td>
</tr>
<tr>
<td>Word-to-word translation using Lexicon Lookup</td>
<td>50.949%</td>
<td></td>
</tr>
<tr>
<td>Word Sense disambiguation</td>
<td>7.140%</td>
<td></td>
</tr>
<tr>
<td>Word Inflectional analysis and generation</td>
<td>6.45%</td>
<td></td>
</tr>
<tr>
<td>Transliteration</td>
<td>23.239%</td>
<td></td>
</tr>
<tr>
<td>Post Processing</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grammar Correction</td>
<td>6.197%</td>
<td></td>
</tr>
</tbody>
</table>
उत्तरी भारत में ठंड का प्रकोप जारी है. राज्य सरकार के बयान में कहा गया है कि हिमाचल प्रदेश और कश्मीर राज्य में बर्फबारी हुई है और कई इलाकों में बारिश ज्यादा हुई है. बिहार, हरियाणा और उत्तर प्रदेश में भी कदाक हुए ठंड रहते हैं. यह बारिश वर्षाओं से हुई है. इससे कई राष्ट्रीय खुलासों दो रुपये भर कर निकल रहे हैं. उत्तर प्रदेश में तापमान सामान्य से दो दिग्री कम चल रहा है.

गिरता पारा

बर्फ़ी राजधानी दिल्ली में भी काफ़ी सर्दी है. दिल्ली में किरदार 20 ट्रेनें रुक कर दी गई हैं. रायवार को न्यूनतम तापमान 9.5 डिग्री सेल्सियस तथा धूँच रूपक को तापमान 8.4 रहा. उधर अमृतसर में रायवार रात को पारा मात्र 0.8 डिग्री सेल्सियस रहा जबकि चंडीगढ़ में तापमान 9.4 डिग्री सेल्सियस था. ठंड की वजह से कई जगह में फिस्कियों बादा रही गई है. कोहरे के कारण कई जगह दुर्घटनाएँ भी हो रही है. शनिवार को दो रेल हादसों में कम से कम 10 लोग मारे गए थे. कोहरे से विज्ञानी आर्थिक भी प्रभावित हुई थी और उत्तरी बिहार फेल हो गया था.

उत्तरी भारत में मौसम थान्ड का प्रकोप जारी है. सरकारी टिल्लियाँ के मुताबिक थान्ड का कारण अब तक 100 लोगों की मौत हो चुकी है. हिमालय प्रदेश और कश्मीर राज्य में बर्फबाली हुई है और कई इलाकों में बारिश ज्यादा हुई है. पंजाब, हरियाणा और उत्तर प्रदेश में भी कदाक हुए ठंड रहते हैं. यह बारिश वर्षाओं से हुई है. इससे कई राष्ट्रीय खुलासों दो रुपये भर कर निकल रहे हैं. उत्तर प्रदेश में तापमान सामान्य से दो दिग्री कम चल रहा है.
Output text:

उत्तरी भारत विस्तार तख्त देख देखा जाना है। ब्रह्मचारी टेसीविलोकन ने भूमभाष पढ़े ने बच्चे घर ऊंच 100 हजरत ची नंदे देखे चूंकी है। दिवसभर पुस्तक लेने जाने हाथी दिख बदलया देखी है जब देखी दिखारी दिन संयुक्त वैज्ञानिक कॉम्प्यूटर डायरेक्शन. पत्रिका, उपविष्टार अने भुजुट पुस्तक दिख से बच्चे पे देखे है। भिड़े हुं दिखाए हुं इसे बेबले की दस्तू बच्चे देखके उड़ टैक्स तंत्र बागते बाही है। भुजुट पुस्तक दिख उपविष्टार दिखे पिशें दे दे 10 दिवसार मौत सटे दिख है।

उठाया भारत

उठे दी नस्वाती हैन्टी दिख दी बाही मदवी है। हैन्टी दिख बाही 20 टैक्स हें बत हैन्टी बाही महत्व पूर्व का। भुजुट पूर्व वेलस उपविष्टार 9.5 डिग्री मेंमिस्ट्री मी तत्त्व ये मदवीता उंच उपविष्टार 8.4 दिख। भुजुट भारतीय संपर्क वेलस पूर्व उंच का परिवह 0.8 डिग्री मेंमिन्स्ट्री दिख मर्य से वार्डित दिख उपविष्टार 9.4 डिग्री मेंमिन्स्ट्री मी। खंड ही दस्तू बच्चे देखी हाथी सबसे देख है। भुजुट पूर्व दे बने वार्डित मेंमिन्स्ट्री दिख पूर्व दे पूर्व 10 हजरत भागे बाही पाल। भले हो दे विस्तार अनुपाती दी पूर्वविष्ट वेलस मी अने भुजुटी वार्डित हें दे इस्मी मी।

उत्तरी भारत विस्तार तख्त देख कहीं जाती है। सरकारी तेलिविजन दे मुताबक तख्त दे कारण हुं तक्क 100 लोकम दी मृत हो तुक्की है। हिमाल प्रदेश अते कश्मीर घाटी विस्तार बर्थबारी होई है अते कालोमिम विस्तार सार्था तोहऱा कोह्रात कहाँ होई हैं। पानजाह, हराना अते उत्तर प्रदेश विस्तार वी कार्केड तख्त पाई हो। पिछले कुजह दिनाम तोम चाहे कोह्रे दी वाहार कार्केइ उर्मेन हैर तारेन्म रत्द कारन पाइम है। उत्तर प्रदेश विस्तार कीमों इक्की जीहे तोम दो तभी 10 दिग्री घाट्ट कॉल रही है।

द्गग्दा परा

उठी ही राज्यहाँ दिली विस्तार वी काफी सार्डी है। दिली विस्तार 20 तार्कनम रत्द कार दिती गाख दे है। आत्म कुम हेथ्ला ताप्मान 9.5 दिग्री सेल्सियस सी जादों की शान्तोरुम तभी ताप्मान 8.4 रही। उद्धार अतिमित्व विस्तार आत्म कुम दोपरा 8.4 दिग्री सेल्सियस ही जादों की चांदगार विस्तार 9.4 दिग्री सेल्सियस सी। तख्त दी वाहार कार्केइ काल जाग्हा साकुल्य विस्तार चुट्ठिमां वात्त दिती गाख है। कोह्रे दे कारण काल जाग्हा दुर्गातनवाम ही हो रही है। शान्तोरुम नुम दे रत्द हादसिाम विस्तार गाष्ट्ट तभी गाष्ट्ट 10 लोक मरे गाए सान। कोह्रे तोम बिजी अपुर्ती वी प्राप्तवी होई सी अते उत्तरी गारी घेल हो गिया सी।
6.5 Summary
This chapter discusses in detail the post-processing phase. This phase involves the rules that are applied on the output produced by previous phases. The various grammatical errors corrected by this phase are also discussed. The implementation of whole system is also discussed along with illustrative example.

Translations for text from various sources like news items, stories, blogs, office orders, articles etc. are obtained from this system and made available to the evaluators for the evaluation purpose. In the next chapters, we will discuss the evaluation and results of our system, for the language pair of Hindi and Punjabi.
Chapter 7

Evaluation and Results

7.1 Introduction

Evaluation of a MT system is as important as the MT itself, answering the questions about the accuracy, fluency and acceptability of the translation and thus artifying the underlying MT algorithm. Evaluation has long been a tough task in the development of MT systems because there may exist more than one correct translations of the given sentence. The problem with natural language is that language is not exact in the way that mathematical models and theories in science are. While there is general agreement about the basic features of Machine Translation (MT) evaluation (as reflected in general introductory texts Lehrberger & Bourbeau, 1988; Hutchins & Somers, 1992; Arnold et al., 1994), there are no universally accepted and reliable methods and measures, and evaluation methodology has been the subject of much discussion (e.g. Arnold et al., 1993; Falkedal, 1994, AMTA, 1992). As in other areas of NLP, three types of evaluation are recognised:

- Adequacy evaluation to determine the fitness of MT systems within a specified operational context. It is typically performed by potential users and/or purchasers of systems (individuals, companies, or agencies). Adequacy evaluations usually include the testing of systems with sets
of typical documents. But these are necessarily restricted to specific domains.

- Diagnostic evaluation to identify limitations, errors and deficiencies, which may be corrected or improved (by the research team or by the developers). It is the concern mainly of researchers and developers.
- Performance evaluation to assess stages of system development or different technical implementations. It may be undertaken by either researchers/developers or by potential users.

MT evaluations typically include features not present in evaluations of other NLP systems: the quality of the raw (unedited) translations, e.g. intelligibility, accuracy, fidelity, appropriateness of style/register; the usability of facilities for creating and updating dictionaries, for post-editing texts, for controlling input language, for customisation of documents, etc.; the extendibility to new language pairs and/or new subject domains; and cost-benefit comparisons with human translation performance.

7.2 Related Works:

Several researchers have worked on evaluation techniques of Machine Translation systems and many measures and methods have been developed for this purpose. Attempts have been made to produce well designed and well founded evaluation schemes. Initially, MT evaluation was seen primarily in terms of comparisons of unedited MT output quality and human translations, e.g. the ALPAC evaluations [3] and those of the original Logos system.
Later, systems were assessed for quality of output and usefulness in operational contexts, e.g., the influential evaluations of Systran by the European Commission [102]. SYSTRAN [35,227,228] and Logos [30,31] have developed internal evaluation methods to compare results given by different versions of their own systems. Palmira Marrafa and Antonio Ribero [92] proposed quantitative metrics for evaluations based on the number of errors in an evaluation and the total number of possible errors. Rita Nüebel [229] presents a blueprint for a strictly user-driven approach to MT evaluation within a net-based MT scenario, which can also be adapted to developer-driven evaluations. The Van Slype report for the European Commission [102] provided a very thorough critical survey of evaluations. Eagles Evaluation Group [230] also worked to establish standards in the field to come up with a theoretically sound framework for evaluation of a Machine Translation system. However, no consensus has ever been reached in defining one single evaluation procedure, applicable to a Machine Translation system in all circumstances. Valuable contributions to MT evaluation methodology have been made by Rinsche (1993) in her study for the European Commission, and by the JEIDA committee (Nomura & Isahara, 1992), which proposed evaluation tools for both system developers and potential users. The evaluation exercise by ARPA (White et al., 1994) compared the unedited output of the three APRA-supported experimental systems (Pangloss, Candide, Lingstat) with the output from 13 production systems from Globalink, PC-Translator, Microtac, Pivot, PAHO, Metal, Socatra XLT, Systran, and
Winger. The initial intention to measure the productivity of systems for potential users was abandoned because it introduced too many variables. Evaluation, therefore, has concentrated on the performance of the core MT engines of systems, in comparison to human translations, using measures of adequacy (how well a text fragment conveys the information of the source), fluency (whether the output reads like good English, irrespective of accuracy), and comprehension or informativeness (using SAT-like multiple choice tests covering the whole text). Roudaud [231] discusses in detail the procedure for the evaluation and improvement of an MT system by the end users. He describes the different types of problems encountered and categories them. Simone Wagner [232] suggested four methods viz. percentage of correct sentences, no. of errors, Intelligibility, Accuracy, and time taken to do post editing, which concentrates on linguistic performance of the system. He claims that these evaluation methods do not require expert linguistic knowledge and can be performed in quite short time. However, not all of them were equally suited for a comparative evaluation. Keiji et al. [233] evaluates the translation output by measuring the similarity between the translation output and translation answer candidates from a parallel corpus. Yasuhiro et al. [234] use multiple edit distances to automatically rank Machine Translation output by translation examples. While the IBM BLEU method Papineni et al. [235] and the NIST MT evaluation [236] compare MT output with expert reference translations in terms of the statistics of word N-grams. Melamed et al. [237] adopt the maximum matching size of the translation and the
reference as the similarity measure for the score. Nieben and Och [238] score a sentence on the basis of scores of translations in a database with the smallest edit distance. Yokoyama et al. [239] propose a two-way MT based evaluation method, which compares output Japanese sentences with the original Japanese sentence for word identification and the correctness of the modification.

7.3 Our Approach:

For our purpose following steps have been performed for evaluating the system that is discussed in detail as follows:

7.3.1 Selection Set of Sentences:

It is very important aspect in MT evaluation to make appropriate selection of the sentences for evaluating the Machine Translation system. According to Lorna Balkan [240,241], There are basically three types of test materials:

Test Corpora: It is a collection of naturally occurring text, increasingly in electronic form.

Test Suites: It is a collection of usually artificially constructed inputs, where each input is designed to probe a system’s treatment of a specific phenomenon or set of phenomena. Inputs may be in the form of sentences, sentence fragments, or even sequences of sentences. Test suites are useful for presenting language phenomena and combinations of phenomena in an exhaustive and systematic way. Furthermore, negative data can be derived systematically from positive data by violating grammatical constraints associated with the positive data item.
Test Collections: It is a set of inputs associated with a corresponding set of expected outputs. This type of test material is increasingly common and has been used in the evaluation of parsers and other Natural Language Processing applications. The problem with test collections is that of being able to specify an appropriate output for a system. Output from parsers can be many and varied. The Parseval project, in common with other parser evaluation projects, uses hand-produced ideal parses of sentences from the Penn Treebank, a parsed corpus, to compare parser output against. Machine Translation shares a similar problem - there is no one correct output. While at present no test collections exist for MT, it is possible to imagine producing an ideal translation, in the same way as an ideal parse.

There are several issues involved in the selection of set of sentences for a comprehensive evaluation. For example, the set could be constant, variable or a mixed one; the number of sentence may be small or large, the collection of sentences may be domain specific or generic. It is obvious that there is no guarantee that even the bulkiest sample will include all the possible syntactic structures of the source language. Elliott et al. [242] describes the text limit to include in a corpus for MT evaluation, given the general hypothesis that more text would lead to more reliable scores. The author, on the basis of an empirical assessment of score variation, estimates that systems could be reliably ranked with around 40 texts (ca. 14,000 words). Zhang and Vogel [243] also studied the influence of the amount of test data on the reliability of automatic metrics, focusing on confidence intervals for
BLEU and NIST scores. Estrella P. et. al. [244] show that for human or automatic evaluation about five documents from the same domain—with ca. 250 segments or 6,000 words—seem sufficient to establish the ranking of the systems and about ten documents are sufficient to obtain reliable scores.

For our Machine Translation system evaluation, we have used benchmark sampling method for selecting the set of sentences. Input sentences are selected from randomly selected news (sports, politics, world, regional, entertainment, travel etc.), articles (published by various writers, philosophers etc.), literature (stories by Prem Chand, Yashwant jain etc.), Official language for office letters (The Language Officially used on the files in Government offices) and blogs (Posted by general public in forums etc.). Care has been taken to ensure that sentences use a variety of constructs. All possible constructs including simple as well as complex ones are incorporated in the set. The sentence set also contains all types of sentences such as declarative, interrogative, imperative and exclamatory. Sentence length is not restricted although care has been taken that single sentences do not become too long. Following table shows the test data set:

**Table 7.1: Test data set for the evaluation of Hindi to Punjabi Machine Translation System**

<table>
<thead>
<tr>
<th></th>
<th>Daily News</th>
<th>Articles</th>
<th>Official Language Quotes</th>
<th>Blog</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Documents</strong></td>
<td>100</td>
<td>50</td>
<td>01</td>
<td>50</td>
<td>20</td>
</tr>
</tbody>
</table>

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System* - A Doctoral Dissertation
7.3.2. Selection of Tests for Evaluation

There are number of tests available for evaluating the Machine Translation systems. Van Slype [102] describes that the selection of tests for MT evaluation depends upon the target users of the MT system. The main aim of our system is effective transfer of information from Hindi to Punjabi language. Thus, Subjective tests and Error diagnosis/analysis have been selected for our MT System evaluation. Subjective Tests include intelligibility test, Accuracy Test / Fidelity rating and BLUE Scoring. Some Quantitative Metrics have also been evaluated through error analysis / diagnosis by calculating Sentence Error Rate (SER) and Word Error Rate (WER). These tests are discussed in detail in following sections.

7.3.2.1 Subjective Tests

7.3.2.1.1 Intelligibility Test

This test is used to check the intelligibility of the MT System. Van Slype Georges [102] describes intelligibility as a measure of how understandable the sentence is. Intelligibility is measured without the reference to the original sentence. It tells the degree of comprehensibility and clarity of the translation. Intelligibility is effected by grammatical errors, mis-translations, and untranslated words. The scoring methodology for intelligibility test has been adopted described by Van Slype Georges [102]. Each evaluator receives a
series of sentences in sequence i.e. sentence in their context. Literature shows variations in selecting the point scales. It has been observed that a scale comprising a very low number of points seems insufficiently discriminatory. On the other hand, a scale comprising a high number of points, assessment of which remains in the final analysis subjective, involves too wide a scatter of the ratings. Furthermore, to clarify in detail each of the possible values of the scale, there is a risk of introducing elements not germane to intelligibility. Thus, it is concluded that four points scale is most adequate, in that it measures intelligibility only, has a low scatter and is of a sufficiently discriminatory character since the evaluation covers several hundreds of sentences and the average calculated as a percentage is sufficiently precise. Hence, a four point scale is made in which highest point is assigned to those sentences that look perfectly clear and intelligible and lowest point is assigned to the sentence which is non understandable. The scale looks like:

**Table 7.2 Score Sheet for Intelligibility Test**

<table>
<thead>
<tr>
<th>Score</th>
<th>Significance</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>The sentence is perfectly clear and intelligible. It is grammatically correct.</td>
</tr>
<tr>
<td>2</td>
<td>The sentence is generally clear and intelligible. Despite some inaccuracies, one can understand the information to be conveyed.</td>
</tr>
<tr>
<td>1</td>
<td>The general idea is intelligible only after considerable study. The sentence contains grammatical errors and/or poor word choice.</td>
</tr>
</tbody>
</table>
7.3.2.1.2 Accuracy Test / Fidelity Measure

Accuracy Test or Fidelity measure is a measure of how much information the translated sentence retained compared to the original. It is measured indirectly. The evaluator is asked to gather whatever meaning he could from the translation sentence and then evaluate the original sentence for its "informativeness" in relation to what he had understood from the translation sentence. Thus, a rating of the original sentence as "highly informative" in relation to the translated sentence would imply that the latter was lacking in fidelity/accuracy. Halliday [245] define it as the Measurement of the correctness of the information transferred from the source language to the target language. Van Slype Georges [102] describes it as a subjective evaluation of the measure in which the information contained in the sentence of the original text reappears without distortion in the translation. Analogous to the Intelligibility test, the methodology described by Van Slype Georges [102] is adopted for the accuracy test also. A Four point scale is made in which highest point is assigned to those sentences that are completely faithful and lowest point is assigned to the sentence which is un-understandable and unacceptable. The scale looks like:

<table>
<thead>
<tr>
<th>Score</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>The sentence is unintelligible. The meaning of the sentence is not understandable.</td>
</tr>
</tbody>
</table>

Table 7.3 Score Sheet for Accuracy Test
These both scales i.e. Intelligibility and Accuracy test scales have already been used in the evaluation of the SYSTRAN English-French MT system acquired by the Commission of the European Communities.

### 7.3.2.1.3 BLEU Score

Bilingual Evaluation Understudy or BLEU [246] is one of the most popular metric for automatically evaluating Machine Translation system output quality. The central idea behind this metric is the closer a Machine Translation is to a professional human translation, the better it is. The primary programming task in a BLEU implementation is to compare n-grams of the candidate with the n-grams of the reference translation and count the number of matches. These matches are position-independent. The more the matches, the better is the candidate translation. The metric calculates scores for individual segments, generally sentences, and then averages these scores over the whole corpus in order to reach a final score. It has been shown to correlate highly with human judgments of quality at the corpus level. The quality of translation is
indicated as a number between 0 and 1 and is measured as statistical
closeness to a given set of good quality human reference translations.
Therefore, it does not directly take into account translation intelligibility or
grammatical correctness. The metric works by measuring the n-gram co-
ocurrence between a given translation and the set of reference translations.
Then the weighted geometric mean is calculated.

7.3.3 Evaluation based on Quantitative Metrics

Rather than using broad indicators as guides to score assignments, we must
also focus on the errors made by the MT system. Quantitative metrics play
major role in it. It includes the technique of error analysis that tries to establish
how seriously errors affect the translation output. The error analysis includes
calculating Word Error Rate (WER) and Sentence Error Rate (SER). Word
Error Rate (WER) is defined as percentage of words which are to be inserted,
deleted, or replaced in the translation in order to obtain the sentence of
reference. Sentence Error Rate (SER) is defined as percentage of sentences,
whose translations have not matched in an exact manner with those of
reference.

7.3.4 Experiments

It is also important to choose appropriate evaluators for our experiments.
Thus, depending upon the requirements and need of the above mentioned
tests, 50 People of different professions were selected for performing
experiments. 20 Persons were from villages that only knew Punjabi and did not know Hindi and 30 persons were from different professions having knowledge of both Hindi and Punjabi. Average ratings for the sentences of the individual translations were then summed up (separately according to intelligibility and accuracy) to get the average scores. Percentage of accurate sentences and intelligent sentences was also calculated separately by counting the number of sentences.

7.3.4.1 Intelligibility Evaluation

The evaluators do not have any clue about the source language i.e. Hindi. They judge each sentence (in target language i.e. Punjabi) on the basis of its comprehensibility. The target user is a layman who is interested only in the comprehensibility of translations. Intelligibility is effected by grammatical errors, mis-translations, and un-translated words.

7.3.4.1.1 Scoring

The scoring is done based on the degree of intelligibility and comprehensibility. A four point scale is made in which highest point is assigned to those sentences that look perfectly alike the target language and lowest point is assigned to the sentence which is un-understandable. Detail is as follows:

**Score 3:** The sentence is perfectly clear and intelligible. It is grammatical and reads like ordinary text.
**Score 2:** The sentence is generally clear and intelligible. Despite some inaccuracies, one can understand immediately what it means.

**Score 1:** The general idea is intelligible only after considerable study. The sentence contains grammatical errors &/or poor word choice.

**Score 0:** The sentence is unintelligible. Studying the meaning of the sentence is hopeless. Even allowing for context, one feels that guessing would be too unreliable.

### 7.3.4.1.2 Results

The response by the evaluators were analysed and following are the results:

- 70.3 % sentences got the score 3 i.e. they were perfectly clear and intelligible.
- 25.1 % sentences got the score 2 i.e. they were generally clear and intelligible.
- 3.5 % sentences got the score 1 i.e. they were hard to understand.
- 1.1 % sentences got the score 0 i.e. they were not understandable.

So we can say that about 95.40 % sentences are intelligible. These sentences are those which have score 2 or above. Thus, we can say that the direct approach can translate Hindi text to Punjabi Text with a considerably good accuracy.

### 7.3.4.1.3 Percentage Intelligibility:

Language in India [www.languageinindia.com](http://www.languageinindia.com) 834

10 : 10 October 2010

Vishal Goyal, Ph.D.

*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
Following graph shows that percentage intelligibility of individual documents:

![Graph showing percentage intelligibility](image)

**Figure 7.1**: Percentage Intelligibility for Different Documents

7.3.4.1.4 Analysis

The main reason behind less accuracy for literature documents is due to the language dialect used by the writer of the stories. Some writers use Rajasthani language, some use Haryanavi dialect. And this resulted in less translation accuracy for this category. Otherwise for rest of the four categories, the quality of translation is better than other systems which will be discussed in following sections.

7.3.4.2 Accuracy Evaluation / Fidelity Measure

The evaluators are provided with source text along with translated text. A highly intelligible output sentence need not be a correct translation of the source sentence. It is important to check whether the meaning of the source
language sentence is preserved in the translation. This property is called accuracy.

7.3.4.2.1 Scoring:
The scoring is done based on the degree of intelligibility and comprehensibility. A four point scale is made in which highest point is assigned to those sentences that look perfectly like the target language and lowest point is assigned to the sentence which is not understandable and unacceptable. The scale looks like:

Score 3 : Completely faithful

Score 2: Fairly faithful: more than 50 % of the original information passes in the translation.

Score 1: Barely faithful: less than 50 % of the original information passes in the translation.

Score 0: Completely unfaithful. It doesn’t make any sense.

7.3.4.2.2 Results
Initially Null Hypothesis is assumed i.e. the system’s performance is NULL. The author assumes that system is dumb and does not produce any valuable output. By the intelligibility of the analysis and Accuracy analysis, it has been proved wrong.

The accuracy percentage for the system is found out to be 87.60%

Further investigations reveal that out of 13.40%:
- 80.6% sentences achieve a match between 50 to 99%
- 17.2% of remaining sentences were marked with less than 50% match against the correct sentences.
- Only 2.2% sentences are those which are found unfaithful.

A match of lower 50% does not mean that the sentences are not usable. After some post editing, they can fit properly in the translated text.

### 7.3.4.2.3 Percentage Accuracy:

Following graph shows that percentage accuracy of individual documents:

![Percentage Accuracy for Different Documents](image)

**Figure 7.2: Percentage Accuracy for Different Documents**

#### 7.3.4.2.4 Analysis

The overall performance accuracy test of the system is quite good. But for Blog it is less than others. The reason is the use of slang which causes the failure of the translation software as the slang available in one language is not present in the other language. Also un-standardized language causes more ambiguities.
7.3.5 Error Analysis

Error analysis is done against pre classified error list. All the errors in translated text were identified and their frequencies were noted. Errors were just counted and not weighted. In the following sections, the experiments conducted for Word Error Analysis and Sentence Error Analysis will be explained.

7.3.5.1 Word Error Analysis

After robust analysis, Word Error rate is found to be 4.58% which is comparably lower than that of general systems, where it ranges from 9.5 to 12%[231,237,238]. Following figure shows the percentage type of errors out of the errors found:

![Pie Chart showing percentage distribution of errors]

**Figure 7.3: Percentage Distribution of Errors**

From the above figure, it is concluded that majority of the errors are due to wrong choice of words, means the WSD module of the system must be improved.
improved. Further, the bilingual dictionary improvements can reduce the wrongly translated and untranslated words errors.

**Word Error Rate Percentage:**

Following graph shows the Word Error Rate for different articles:

![Word Error Rate Graph](image)

**Figure 7.4: Word Error Rate for Different Documents**

**7.3.5.2 Sentence Error Analysis:**

The Sentence error rate comes out to be 28.82%. Following graph shows the Word Error for different articles:
7.3.5.3 Error Analysis Conclusion

As discussed earlier, the WER and SER of un-standardized matter i.e. Blog and Literature is higher than the standardized matter. It strengthens the fact that better input gives the better output. If some pre editing of the text is performed then better results may be expected.

7.4 Comparison with Other Existing Systems:

Following table shows the comparison among various existing systems with our system:

<table>
<thead>
<tr>
<th>MT SYSTEM</th>
<th>Accuracy</th>
<th>Test Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUSLAN</td>
<td>40% correct 40% with minor errors. 20% with major error.</td>
<td>Intelligibility Test</td>
</tr>
</tbody>
</table>

![Figure 7.5: Sentence Error Rate for Different Documents](image)

Language in India [www.languageinindia.com](http://www.languageinindia.com)
<table>
<thead>
<tr>
<th>CESILKO (Czech-to-Slovak)</th>
<th>90%</th>
<th>Intelligibility Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech-to-Polish</td>
<td>71.4%</td>
<td>Accuracy Test</td>
</tr>
<tr>
<td>Czech-to-Lithuanian</td>
<td>69%</td>
<td>Accuracy Test</td>
</tr>
<tr>
<td>Punjabi-to-Hindi</td>
<td>92%</td>
<td>Intelligibility Test</td>
</tr>
<tr>
<td><strong>Hindi-to-Punjabi</strong></td>
<td><strong>94%</strong></td>
<td><strong>Intelligibility Test</strong></td>
</tr>
<tr>
<td></td>
<td><strong>90.84%</strong></td>
<td><strong>Accuracy Test</strong></td>
</tr>
</tbody>
</table>

From the above table, it is clear that the system is outperforming in comparison to others. Thus system can be acceptable for practical use.

### 7.5 Conclusion

Human evaluation is Holy Grail for MT evaluation, but due to lack of time and money it is becoming impractical. Thus, many automatic MT evaluation techniques have been developed. We have evaluated our system based on the subjective tests and quantitative metrics. From the above analysis, it is concluded the overall accuracy of Hindi to Punjabi Machine Translation system is found to be 94% on the basis of intelligibility test and 90.84% on the basis of accuracy test. The accuracy can be improved by improving and extending the bilingual dictionary. Even robust Word Sense Disambiguation module and Post Processing of the system can improve the system to greater extent. This system is comparable with other existing systems and its accuracy is better than those.
Chapter 8

Summary

We have developed robust Hindi to Punjabi Machine Translation system. It is available to use for free at website http://h2p.learnpunjabi.org. With online version, a user can translate a text by typing it in a box provided at webpage or one can submit a file containing text in Unicode. A user can also translate any Hindi website like http://bbc.co.uk/hindi/ and can view it in Punjabi. An E-mail option is also included whereby a user can type his message in Hindi and send the translated text or typed text to an email id submitted by him. To the best of the knowledge, the current system is one of the best Machine Translation System from one Indic language to another.

In this chapter, we will summarize the achievements and limitations of the present research work. Directions for further research that can help to enhance this Machine Translation system have also been included.

8.1 Contributions

- The survey of various existing Machine Translation systems has been presented. Based on this survey, it has been concluded that direct Machine Translation approach is suitable for closely related language pair. We call a language pair to be closely related if the languages have the grammar that is close in structure, contain similar constructs having almost same semantics, and share a great deal of lexicon. By
closely related languages, we also mean incentively and morphosyntactically similar languages. Generally, such languages have originated from the same source and spoken in the areas in close proximity. Thus, being Hindi and Punjabi closely related language pair [250], direct approach has been used for developing Machine Translation system for this language pair.

- The closeness between Hindi and Punjabi has been devised by comparing these languages on the basis of orthogonality, grammar and from machine translation point of view. It has also been proved using corpus based measures by Anil and Harshit [250].

- As the Statistical Machine Translation approach is actively used among researchers nowadays, the scarcity of the resources of language pair like non availability of any annotated or parallel corpus in question limited the choice of translation approach to conventional direct method. The required resources are developed from scratch and used to develop a Machine Translation system.

- The system has to tackle the named entity recognition problem as there are the chances when a token in input text having its translated meaning in target language need to be transliterated rather than translated because it acts as proper noun. Thus, module has been developed for handling Proper Nouns successfully.
As there is no dictionary available for the language pair for Machine Translation purpose, Hence, Hindi to Punjabi lexicon for Machine Translation has been developed.

Word sense disambiguation is done by using language modeling techniques. N-grams can successfully model the disambiguation of Hindi language.

Transliteration is the option for the out-of-vocabulary words. A successful transliteration module has been developed that uses large number of developed rules in addition to direct mapping of characters.

Transfer rules are desirable for handling the grammatical and some structural deviations.

The development is aimed to make a robust system for translating the input text without failure or going blank. The system was evaluated formally and informally both ways. In informal evaluation, the system has been made online at website http://h2p.learnpunjabi.org. The system was introduced to all the researchers working in this area through emails. Even the announcement of this Machine Translation system was also done through media (newspapers, Television and FM Radio). All the Major newspapers like The Tribune, Indian Express, Hindustan Times, Ajit daily, Jagbani, Dainik Jagran, Dainik Bhaskar, Amar Ujala, Rozana Rashtriya Sahara (Urdu Newspaper), Punjab Newsline etc. have published the news of launch of this system at prominent positions of their newspaper. Number of readers have used
the system and sent us the feedback about the quality of the system. Now, it is regularly being used by several newspaper publishers for translating their news, book publishers etc. In formal evaluation, the system is evaluated by both objective and subjective tests. The accuracy is figured out as 94% on the basis of Intelligibility test and 90.84% on the basis of accuracy test. In the quantitative tests the Word Error Rate is found out to be 4.58% whereas Sentence Error Rate is 28.82%.

- The development of this system is an effort to bring the Punjabi on the map of Machine Translation. The system can be integrated to other existing translation system like English to Hindi (facility provided by Google) to produce a system that will translate the text from English to Punjabi. In fact the integration of our system with Urdu to Hindi transliteration system is on chart where the Hindi text produced by transliteration system is fed into our translation system thereby producing Punjabi text from the text in Urdu.

8.2. Limitations

Although system shows good accuracy but the system still fails at some points. Some common errors are explained with examples:

8.2.1 Named Entity Recognition Failure:
There are foreign names in the text like फाश की लीनी (bāsh kī līnī). It will be translated into घम दी लीनी (bāsh dī līnī).

b. There are proper nouns having multiple translations in Punjabi which do not have title or surname surrounding them in the sentence. For example:

दीपक कहाँ है (dīpak kahāṃ hai)

दीवा कित्तेहै (dīvā kitthē hai)

8.2.2 Modifier and Noun Agreement: All the modifiers must agree with the word that they modify in a noun phrase. But it fails in some of the cases as shown in following example:

बैंक खेतीबारी के कर्ज़ के लिए घटी दरें लागू करेगा (bāṅk khētībāṛī kē kārjē kē liē ghāṭī darēṃ lāgū kārēgā)

8.2.3 Subject/Object and Verb Agreement: All the verbs must agree with the Subject/Object in the sentence. But it fails in some of the cases as shown in following example:

हम नयी भाषा कर नौकरी में उत्पन्न कर सकते हैं (ham nayī bhāshā kār naukṛī mēṃ unntī kār saktē haiṃ)


8.2.4 Resolving meaning of ambiguous words: For some of the cases, the system fails to resolve the meaning of the word among its multiple meanings. For example:

दो हफ्तों के लिए बढ़ दिया है (dō haphtōṃ kē liē baḍhā diyā hai)

8.2.5 Noun phrase in oblique case form before postposition: In the sentence, if Noun phrase is present before the postposition, then it will come in oblique case. In some cases, it fails as in the following example:

वह खुश होकर अपने पास चली गयी (Vah khush hōkar apnē pās calī gayī)

8.2.6 Agreement of subject noun phrase having नौकरियाँ (naukāriāṃ) with verb phrase: In Punjabi, all the Verb phrases in the sentence must agree with the Subject Noun phrases having नौकरियाँ (naukāriāṃ) like नौकरियाँ. Sometimes, it fails in cases as shown in the following example:

मुझे दवाई चाहिए (mujhē davāī cāhiē)
8.2.7 ठ �postposition agreement before Verb phrase: In Punjabi, all the
Verb phrases in the sentence must agree with the postposition ठ. But in
some cases, it fails as shown in following example:
यह शव कंवलजीत सिंह का है (yah shav kamvijit singh ka hai)
ठह रगम वंदजीनी मिन्थ ठ वै। (ih lāsh kamvijit singh dā hai.)

8.2.8 ठ ठ postposition and following Noun phrase agreement: In Punjabi,
the postposition ठ ठ must agree with the following Noun phrase in the
sentence. But in some cases, it fails as shown in following example:
रात वाला झगडा चल रहा है (rāt vālā jhagdā cal rahā hai)
ठउ ठनह ठरहाँ चेंल विज वै। (rāt vālā lařāī call riḥā hai.)

8.2.9 Noun Verb Agreement: In Punjabi, the postposition ठ ठ must agree
with the following Noun phrase in the sentence. But in some cases, it fails as
shown in following example:
वे कोई भड़काऊ भाषण नहीं देंगे (vē kōī bhaḍkāū bhāshaṇ nahīṃ dēṅgē)
ठह बेएी ठजबरूँ ब्यमट ठरी ठेलर (uh kōī bhaṛkāū bhāshaṇ nahīṃ dēvāṅgē)
Similarly, it is not possible to cover all the hindi words in the dictionary and as when we use the application for use, we come across words that are missing and can be added in parallel to its use.

8.3 Future Directions

Although our system is showing good results using the direct translation approach but still there is lot of scope for improvement. Following are some of the future directions:

• More Data
The most obvious way to improve a data-driven approach like presented here is of course to utilize more data. Database entries for bilingual dictionary, proper noun gazetteers, surnames, titles, bigrams and trigrams for WSD need to be extended.

• Resource Development
Statistical Machine Translation approach has now often been used by the researchers. The only requirement for this approach is the availability of high quality parallel corpus. Thus, with the development of this system, parallel corpus for Hindi-Punjabi Language pair must be developed for use in future researches in these languages.

• Better Models
Despite using more data, improved models can lead to better translation quality. Using parallel corpus for the language pair, it is of great interest to combine automatic techniques for various tasks with direct approach to
develop a more robust and accurate Machine Translation system. Even use of full parsers in the Machine Translation Systems can show better results.

- **Public Corpora and Tools**

There are initiatives by various NLP research groups for releasing the corpora publicly. Some of NLP tools are also available for various tasks. Using such corpora and tools will help in reducing the development time and effort of the system. Such practice will also help the researchers’ efforts in redoing the tasks that have already been done.

- **Better Evaluation Metrics**

Automatic evaluation metrics are important for a rapid development cycle. During the development and tuning phase, the quality of the MT system is evaluated several (hundred) times. The parameters of the MT system are adjusted to achieve a high score of a given automatic evaluation metric. Nevertheless, the ultimate goal is to improve the translation quality using this parameter tuning. Therefore, automatic evaluation metrics should have a high correlation with human judgment of translation quality. Furthermore, it should not be possible to cheat the metric, i.e. to improve the score without improving translation quality. Current metrics have their limitations as pointed out in [Callison-Burch & Osborne+ 06] for the BLEU score. As MT systems are tuned toward a specific metric, improved MT evaluation metrics will lead to better Machine Translation quality.

- **Integration with other systems**
The system developed can be integrated with other systems to deal with more complicated tasks. The system can be integrated to translate any Language to Hindi and further to Punjabi. Thus, we can say that any language text can be translated to Punjabi language text using this system. For this purpose, Google translation APIs can also be used.
References


Theoretical and Methodological Issues in Machine Translation, July 23-25, St. John’s College, Santa Fe, New Mexico, USA. pp. 19-30.


Quarterly in Artificial Intelligence, Vol. 10, No. 3, NCST, Bangalore, India, pp. 22-25.


Conference on Recent Advances and Future Trends in IT (RAFIT),
Punjabi University, Patiala, Punjab. pp.156-59.

machine-aided translation system, MT Summit IX, New Orleans.

English to Hindi EBMT. In proceedings of MT SUMMIT IX. New
Orleans, Louisiana, USA. pp.157-162.

[80] Sivaji Bandyopadhyay. 2002. Teaching MT – An Indian
Perspective. In proceedings of the 6th EAMT Workshop on

Interlingua-based English- Hindi Machine Translation and
251-304.


[83] Murthy, B. K., Deshpande, W. R. 1998. Language technology in
India: past, present and future. In proceedings of MLIT Symposium
pp. 134-137.


[131] Black, E. 1988. An Experiment in Computational Discrimination of 
32. pp.185-194.

[132] Pedersen, T. 2002. Evaluating the effectiveness of ensembles of 
decision trees in disambiguating Senseval lexical samples. *In 
Proceedings of the ACL 2002 Workshop on Word Sense 
Disambiguation: Recent Successes and Future Directions.* 

[133] Ng, H. T. and Lee, H. B. 1996. Integrating multiple knowledge 
sources to disambiguate word sense: An exemplar-based 
approach. *In 34th Annual Meeting of the Association for 

1991. Word sense disambiguation using statistical methods. *In 

electronic lexical database and some applications.* Cambridge. MA.: 
MIT Press. pp. 198-203.

disambiguation programs. *In proceedings of the First International 
Conference on Language Resources and Evaluation (LREC 1998), 
pp. 581-588.*


workshop on Automatic Speech Recognition and Understanding
Santa Barbara, CA. pp. 254-261.


Conference on Natural Language Processing (ICON-07),


Publications Based on the Work Presented in this Thesis

Journals

- Vishal Goyal and Gurpreet Singh Lehal, “Web Based Hindi to Punjabi Machine Translation System”, Journal of Emerging Technologies in Web Intelligence, Volume 2, Number 2, May 2010 (Accepted, to be Published).

Conference

Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
Appendix A

Graphic User Interface and Extended Features

Information technology in the current scenario is evolving as an effective tool for making information wide spread and available on-line to several communities at large. On one hand, the increased use of ICT is enabling people across the globe to participate in the knowledge network; at the same time larger populations in the rural areas of developing country like India are being deprived of the benefits of the use of ICT. One of the main reasons behind this seems to be the language barrier. For such cause, Hindi to Punjabi Machine Translation system can play an important role to reduce digital divide due to the language barrier. This lessening of digital divide and increasing the accessibility of information present in the Internet happens to be one of the objectives of our work among the various aims of this research.

We have made our Hindi to Punjabi Machine Translation available online free of cost for use world wide. Our System is capable of doing following tasks:

1. Text translation from Hindi to Punjabi
2. Text transliteration from Hindi to Punjabi
3. Translating Websites
4. Sending Email in Punjabi Language originally written in Hindi language.
Above tasks will be discussed in detail in following sections. Following Screenshot shows the GUI for the Hindi to Punjabi Machine Translation System:

![GUI for Hindi to Punjabi Machine Translation System](image)

**Figure A.1: GUI for Hindi to Punjabi Machine Translation System**

**Text translation from Hindi to Punjabi**

This facility enables the users to translate the input text into Punjabi language text on clicking the Translate button. The text can be input in the textarea there through various modes viz. browsing and reading the text file, typing the text using keyboard and by using the typing pad provided on the interface. It is also an added feature that the text can be entered in a mixed way i.e. English text can also be embedded in between the Hindi text. For the ease of users,
the text can also be typed in four different ways. Care has been taken for professionals or users who are habitual of typing Hindi text in Krutidev or AnmolLipi Font. As it is clear from the above GUI, there is an option of choosing keyboard mapping for typing showing three values in the drop down besides it. The dropdown has the values Krutidev, AnmolLipi and Roman. For instance, if the user chooses the Krutidev option for keyboard mapping style, it enables the user to type the text using the keyboard mapping similar to Krutidev character mapping on the keyboard. For example – on pressing the key ‘v’, the character ‘अ’ will be typed. The difference is the typed text will be in Unicode encoding rather in Krutidev font. It is very interesting that if the user chooses the Roman option, it facilitates the user to type a word just the way it sounds in Hindi language using English letters and once the typing of a word is finished, hit the SPACE bar, the word will be converted to Hindi language script. For example, typing “hamesha” transliterates into Hindi as: “हमेशा “. We have enabled it on our System interface using the Google Indic Transliteration APIs available at the Google website. Another way for entering in input text is by using the typing pad available on the interface. User needs to just click the appropriate buttons corresponding to each character to be typed. Now the text has been entered using any one of the options mentioned above and on pressing the “Translate” button, the input text is translated into Punjabi language. Following screen shot demonstrates text translation facility in our system:
Text transliteration from Hindi to Punjabi:

Transliteration is the process of converting a word written in one language into another language. Transliteration is distinct from translation, which involves a change in language while maintaining the meaning of the word; transliteration instead converts the sound of the word from one language to another. The option of a transliteration component is to enable the well developed poetic verse in the Hindi language to be available to the Punjabi literate public. The transliteration facility in our system can be used in similar manner as explained above for translation facility. The only difference is that user will click the transliterate button for transliterating the text from Hindi to Punjabi text. Following screen shot demonstrates the transliteration feature of the system:

![Transliteration Screenshot](image.png)
Website Translation

Using this facility, user can translate an entire web page directly, simply by entering the URL and clicking Translate button. This facility is available on the home page of our system displayed at the lower right most corner of the GUI. The user can submit a URL of a Hindi website of his/her interest for translation, then clicks the translate button present besides the textbox where user has entered the URL. Then the translation request is processed at the web server and after translation, translated website is displayed to the user. The important aspect in this feature is that the format of the website is retained after translation. On translated webpage, the links can be further clicked to process them similar to the one that has been translated. It gives the user a feeling that they are browsing Punjabi website. The implementation
of this task includes modules - retrieving and parsing HTML Page, translating, combining the translation unit with HTML Codes, altering the links in webpage, displaying the result. Retrieving and Parsing HTML Page includes first downloading the html source code from the server and then extracting the text out of the html tags for processing. Then the text present in the HTML tags is translated using the text translation module mentioned above. Altering the links in webpage is very important process in it. Here, all the links in webpage are replaced, so that the next links must redirect the request through our translation service. By this step, user does not need to enter URLs or take any other action if user wants to translate the linked page. The user simply needs to click on the given link. Translated webpage is then forwarded to the client in the same format in which the original page had appeared. Following screen shots demonstrate the website translation feature:

![Figure A.4: Screenshot for website translation facility of the system](image-url)

Translate Website in Hindi to Punjabi

Enter the URL of the website in the text box below OR Click on any website listed below to translate:

http://webdunia.com/


Figure A.4: Screenshot for website translation facility of the system
Figure A.5: Screen shot of Original Website http://www.webdunia.com/
accessed on 27/12/2009 at 08:40 PM IST
Sending Email in Punjabi originally written in Hindi language:

This facility enables the user to write the email (text) in Hindi language and this text can be emailed to the recipient either in same language or in Punjabi language after translating the original text. It has very real application in sense that sender knows Hindi and wants to communicate some information to target recipient who knows only Punjabi. For this purpose, the sender can write the email in Hindi language and while sending the email, can send the email in Punjabi by clicking the option of sending the email after translating into Punjabi. Thus, recipient will receive the email in Punjabi. The message is communicated as per the
convenience of the sender and recipient both. Following screen shot demonstrates this feature:

![Screenshot for Email Sending facility of the system](image)

**Figure A.7: Screenshot for Email Sending facility of the system**
Appendix B

Test Data Set for Intelligibility Test

Intelligibility Evaluation:
The evaluators do not have any clue about the source language i.e. Hindi Language. They judge each sentence (in target language i.e. Punjabi) on the basis of its comprehensibility. The target user is a layman who is interested only in the comprehensibility of translations. Intelligibility is effected by grammatical errors, miss-translations, and un-translated words.

Scoring:
The scoring is done based on the degree of intelligibility and comprehensibility. A Four point scale is made in which highest point is assigned to those sentences that look perfectly alike the target language and lowest point is assigned to the sentence which is un-understandable. Detail is as follows:

Score 3: The sentence is perfectly clear and intelligible. It is grammatical and reads like ordinary text.

Score 2: The sentence is generally clear and intelligible. Despite some inaccuracies, one can understand immediately what it means.

Score 1: The general idea is intelligible only after considerable study. The sentence contains grammatical errors &/or poor word choice.

Score 0: The sentence is unintelligible. Studying the meaning of the sentence is hopeless. Even allowing for context, one feels that guessing would be too unreliable.
<table>
<thead>
<tr>
<th>S.No.</th>
<th>Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ਧਾੰਥੀ । ਬਿਨਾਂ ਖੋਜੇ ਦੇ ਸੂਚਨਾਂ ਹੁੰਦੀਆਂ ਦੀ ਵਿਚ ਵੀ ਮੰਗਣਾਵਾਂ ਅਧਾਰਵਾਂ ਦੇ ਬੁਧਦਾਨ ਹੁੰਦੇ ਹਨ ਇੱਥੇ ਭਾਸ਼ਾ ਰਾਹੁਤ ਦੇ ਦੀਰਤਾ ਦੁਆਰਾ ਮਿਲਤਾ ਹੈ ਪੁਰਾਤਨ ਯੂਸਵ ਦੀ ਬਣਾਵਤਾ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>ਹਵਾਲਤ । ਅਭਾਸਕੀ ਖੋਜੇ ਦੇ ਸੂਚਨਾਂ ਦੇ ਪ੍ਰਗਟਾ ਅਧਾਰਵਾਂ ਲਈ ਵਿਚਕਾਰ ਨਹੀਂ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>ਤਹਾਂ ਖੋਜੇ ਹੁੰਦੀ ਚੀਜ਼ ਹੈ । ਤਹਾਂ ਖੋਜੇ ਦਾ ਦਸਤਕਰ ਹੁੰਦਾ ਹੈ ਸੁਵਿਧਾ ਦਾ ਮਹਾ ਦੁਆਰਾ ਦੀਸਕ ਵਿਚਕਾਰ ਹੈ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>ਖੋਜੇ ਦੀ ਹਿੰਦੀ ਅਧਾਰਵਾਂ ਪ੍ਰਗਟਾ ਦੀ ਸਤਿਕਾਰਾ ਦੇ ਮਹਾ ਮੁਕਾਮ ਬਣਦਾ ਹੈ ਨਿਯਮਿਤੀ ਦੇ ਸੁਵਿਧਾਂ ਦੀ ਖਿੱਡਾਈ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>ਅਧਾਰਵਾਂ ਦੇ ਵਿਚਕਾਰ ਵਾਲਾ ਹੈ ਵੀ ਅਧਾਰਵਾਂ ਦੁਆਰਾ ਹੀ ਮੁਕਾਮ ਬਣਦਾ ਹੈ ਮਹਾ ਮੁਕਾਮ ਸਕਾਰਿਆ ਉਦਰ ਨੁਹੀਂ ਹੈ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>ਪ੍ਰਗਟਾ ਦੇ ਪ੍ਰਗਟਾ ਦਾ ਦੀਸਕ ਦੀ ਬਣਾਵਤ ਦੀ ਸਤਿਕਾਰਾ ਵਿਚਕਾਰ ਖਰੀਦ ਦੀ ਬਣਾਵਤ ਨਹੀਂ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>ਲਹਿ ਦੀ ਬਣਾਵਤ ਖੋਜੇ ਦੀ ਮੁਕਾਮ ਬੀਜਾ ਦੀ ਸੰਗਵਾਂ ਹੈ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>ਹਿੰਦੀ ਮਹਾ ਦਸਤਕਰ ਦੀ ਅਭਾਸਕੀ ਦਰਸ਼ਾ ਸਿੱਟੀ ਦੇ ਵਿਚਕਾਰ ਵੀ ਦੀਸਕ ਵੀ ਸੁਵਿਧਾ ਦਾ ਬਣਾਵਤ ਦੁਆਰਾ ਹੀ ਪ੍ਰਗਟਾ ਦੀ ਸਤੀਕਾਰਾ ਦੀ ਸੰਗਵਾਂ ਹੈ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>ਹਿੰਦੀ ਮਹਾ ਦੋਸਤਕਰ ਦੀ ਸਤੀਕਾਰਾ ਦੀ ਸੰਗਵਾਂ ਦੇ ਸੁਵਿਧਾ ਦੀ ਬਣਾਵਤ ਨਹੀਂ ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10. धिम भाषा देश विस्तार बहुभाषी इंटरनेट पर वनस्पती मववज्ञान ते पृथ्वी क्षेत्र एवं दे मटेकम टेस्ट दे रचना।

11. बूढ़ा गायी जे धीरीजीडी सिंच भूमिकालिंग दे मिलां उद्यरप खान दे मल।

12. हिंदी मालिक दे घनगा दिन आंदोलन दे वाणिज्य दे रिवॉल्यूशन डूंड़ बूढ़ा गायी दे मिलां देवबालय सेवा देवी सी।

13. भविष्य वी हिंदी माल जाती मास 2004 - 05 सिंच वेबसाइट ते उन मववज्ञान 2831.82 वेबसाइट उपभोक्ता दो जाती हिंदी वाणिज्य बच्ची।

14. सफ़ जाती भिन्न देश दिन विंडी 32 दिन जाती 15 मीटर घनगा बाढ़ चुहड़ी दो, विंडी दिन जाती यज्ञ वनस्पती मीटर दी गन - सिंघ उं वाणिज्य वेबसाइट देवताह दो वनस्पती डूंड़ दो वाणिज्य नगरसं।

15. दुईने उन्नयन तेज़ ठूंड़ दी हिम उठां दे विभाग लोग चेंड दी माल दिंदी। हिमवेद रात्रि माल जाता दे दिनियाँ वैकंट दिन जातीया दे मान बाधें रता ए भेंट दिव स्विमिट।

16. वणुकूल ते बिंग़, भालभुज रूप बाहेर पृथ्वीभूमि चल, हुंद बूझों से दी पृथ्वीभूमि दै।

17. वणुकूल गायी दे डिंक दे चेंड पृथ्वी डूंड़ उन्नयन दिन चल।
18. अनुवाद मंच पर उम्मीदें इंतज़ार की मांग़रेखा अंकवर दर्शा हिंद टेम्प दिंग अभिज्ञ दे 10 दंडे बैठे बेह दिखवने रहे।

19. घंघरी दी हिंदे बालें दें मूंग रते छवसवर्ती मानिसवर सं मदवरी गूड़ीडूडीजी मुं हिंदे निते मेंढ़चं हिंद परिवारवर वींगे मिलें अभिज्ञ इंतरी अभ्यास स्मार्ट हिंद मदद दिखाने।

20. जिन्हें विच विज उन्नत उन्नत हिंद मूंग चरी भेंट गरने नज दे माफ़ मधवसुली हिंद मतभेदवर मदवर जुद वादशाही बतय दि दिवाग्व वच रही है।

21. मेहमान में भुज़िवर, टाटम 100 मंबवर दिंग माही जिंज़र दें नाम दिंदे रहे ने मारी टुलीबर हुं महें निकाल हूं हंदियं वत्ते रहे।

22. मेहमान मंच भी उउर बंधुळ दे केवल अभेज मेंखयता दवन दि गुणवत बतय दे दह वी अभेज मनषवर दवन दि ही मधुबी हुनण दी वीर।

23. हिंद दिख्दा हूंप भूपभांद्री मधवीज बलभड़ दे दी पिंटफ्लेक्ट हंज़े बही पीडियाल घड़ छंड़ी है।

24. जस़द दी हिंद रह़न भूषणवर दे माज़ इंदःवर दी दिलभ नेसु हिंद हिंद वधावन दी विराग्व वीर।

25. हिंदेस दी, बौद्ध झं वीव है लेबिक बतय दे जस़द दे घापत झुड़टू तवी संवाधा वि दववर दि झुड़टू वींगे दे लुब हिंद मदवार बतय वेटू भसमवर वेदेबा।

26. जिन्हें दी सेवावर चेक हिंद मिंड दि चढ़ वीर है।
27. दिल्ली मदराजा विनियमित वी घर बनना

28. मैं दिन नवदं शू दिन तीस हमसंग वन ,
वृत्ति वर्तु व्यास अभी वे , दूर भनिये
तरंग बंटे आजी वे

29. वर्ती धूले मिर राख ठाक यहे भजन बुंदे जं बंटे
भैलाजी बढ़ भमजी बनानी ठान भली

30. लेखावर यहे वर्ती वे अच्छे ते तर वी मानिये
दिन मैं ते वी अच्छे वी धमाके सजायी ना तरी मी

31. जल दिन भेदमेंट दिन देवी देवी बच्चणी दिन 18
अवधारणी भरे वाले मत

32. विश्वजय वेद ते बि पाठमुद्र ते प्राचीनेँउदिन
हरेहर दिन धिल्ले जुंग उदयीएं दिंग देन बहे
उदेसुँत वर्तमनीतीएं ते दिंग दीपत मंचवत
सेहिया ते

33. दिल्लीबाहर ते। बालमुद्र ते गृह भंडी
बहुसमूह भिन्न ते मेहराब तू विद भिन ते
रेशेल प्राचीनेंउदिन वेद दिंग चंस एवे दिंग दें
देवी बालमुद्र दिंग बरीं 700 आविष्कार भान्वीएं
तू भक्त भजानिया विभाग ते भैल मार भान्वीएं ते
भरभ देख उंग बुंदे देवी बच्चणी रीती देवी

34. गृह भंडी ते विद भिन धूले देव दिंग दिंग
भजानिया ते

35. बालमुद्र ते सीमापत्रीवीं हिस्ट्रीवीं दिंग भमाशीर
भमण बुंदे जंटे एव बहे दिंग देवी बालमुद्र भमाशीर
ते हिस्ट्री धमाके दिंग ते.
36. पुरुषर्मेंटली दे विभाजन दी घटना तें दी अवेंशन बीजी दै। दिल्ली पुलिस दे पुरुषर्मेंटली आवेदनी मिले ते बिंदा दि पुरुषर्मेंटली दे भत दी वाहिं पुरुषर्मेंटली बीजी दै।

37. पुरुषर्मेंटली दे मानवती भत ते ढेरें हिंद मोहरें हिंद महानैतिक हिंद महान्यूनर्मेंटली दि महानैतिक रहें ते बचाव ते धर्मत दे मंडली मानवा हिंद नामशेषनी दि हिंदन्व लगानिश मी मिलें हिंद तें हिंद ही वैभव बीजी दै।

38. गुरुवार हिंद 2002 हिंद तें संवैधानिक हिंद हिंद महात्मा दे अन्यत्र हिंद दसता डेव अपनाता माहि भवत दे मंडली वेतन दे मानव नैले हिंद हर दे हिंद भूवीर्य हिंद दै।

39. भैंस हिंदन्व ते भैंस हिंद भत हिंदमृती दी महात्मा भीं बेंद्रु देव ते बेंद्रु दी मंडली नवथी दै।

40. हिंदन्व दी हिंद मदें मेंती घाटवली मार्ग हिंद हिंद हेंद्र दर्शा भार्तवदूव दी उच दे।

41. हिंदन्व दर्शा दी नवनव मनी।

42. हिंदन्व दे अन्यत्र पार्श्विक दे हिंदन्वाली दे हिंद दे मंडल प्रेषेंट प्रेषेंट डिग्री।

43. हिंदन्व बांसी दर्शनी भतंग दूरदर्शनिकी [ हिंदन्व ] दे हिंदन्वाली में हिंद दर्शा भत पत्र मेंती पत्रिकी देन दी मूलिक भिनता तन बनी।

44. वेई ते कान्हवती पत्नी भेंडी।

45. हिंदन्व भतंग महत्ता उद्धेश्यव टीम दी भेंडी तत स्थापित लसी मिलुएं भत अपो स्वरूप हिंद हिंदन्व तें हिंद अद्भुत।
46. त्र्यम्रत्रय डी गुल्मिगुल्म चेरट ब्रह्म तवी भये दिवंग रचना दिवंग ब्रह्म अमित डे घरबाट दुमटे मटे बैलियान अपत्र दिखेट मुटे ठले ठले।

47. मानने चैरटिटस चैरटिटस है विभाजन कांड May 08, 06:45 pm

48. भवत्र घराने गुटे उन्न भाव तवी मेंट अचिन्त है देत दर्द दिवंग मटे ठले मचान पन है।

49. टे जोह डे दामे रागुण्ड टे घराने जीने दें रल 2 - 2 डे दें रल भेड़टे टे घराने ब्रह्म साधन है देविन्वत वव रामी दिवंग मेंदर्दित है टेटू टे घराने दे विभाग निंदे रल जिय घराने ब्रह्म िम दे दिवंग हुम्बट डी चुरु चुरु टे विभाग।

50. मारपत्रहर्ष भेड़टे लहेरी जीवीभीभी धूंरे हे दी देवदी डुंटे दिय ब्रह्म दे दिवंग दीमटे उन्न बटा दिय ब्रह्म है। तय हट तल 21 दली डे भगु ठेटेह।
### Intelligibility Test - Literature

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ਨੇਕਸ ਕੈਂਡ ਦੋੱਢ ਚੀਨ ਲਹੀ ਮੁਹਿ ਜੇ ਅਧਾਰਤਰ ਕੈਂਡ ਰੂਬੱਧ ਲਹੀ ਵਕਰਤ , ਬਲਦਰ ਮੁਲਿ ਥਾ ਮਿਚਾ ਢੌੰਜੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>ਹਿੰਦੀ ਮਹਾਰਾ ਨੇ ਅਧਾਰਤਰ ਬਹਿਸ਼ ਢੂੰਜ ਹੁੰਦ ਚੁੱਖ ਬਠ ਹਿੰਦ ਬੇਠੇ ਹੁੰਦ ਢਿੱਖ ਦੋਘੱਭ ਹੇਠ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>ਮਨ ਦਾ ਮਾਰ ਮੀ , ਹਿਹਾਮਲ ਇੱਕ ਪਥ ਭਰਤਕ ਵਿਰੋਧੀ ਬੈਠੀ ਅਵਾਜ ਬੀਜੀ ਦੇਵ ਦੁਸ਼ਿਤ ਨੇ ਅੰਦੋਂ ਦੇ ਦੇਸ਼ ਹੁੰਦੀ ਮੀ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>ਹਿਸਹਲਮ - ਹੇ ਉੱਤਰਮਿਲਤ ਕਾਲ ਜੂ ਬੀਚਾ - ਜੂ ਤਾਂ , ਮੀ ਲਹੀ ਨਾਹੀ ਲੀ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>ਬਣਾ ਹੁੰਦ ਦੁਆ ਜਿਥੇ ਦੇ ਮਹਿ - ਮੁਹੀ - ਮੁਹੀ ਮੂਰਣੀ ਅਲਹੀ ਮੀ। ਦੇਵ ਦੀ ਮੀਲੀ ਭਾਗ ਮੀਲੀ ਢੂੰਖ ਦੁੰਡ ਮੀ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>ਹੀਦਾ ਵਿਕਰੇਦੀ ਬਹਿਸ਼ ਹੇ ਹੁੰਦ ਹਦਾਦ ਬਗਦੀ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>ਹੁੰਮਾਸੀ ਜੀ ਬਸਕ ਵਕਰ ਵਹੀ ਸਾਹਿਤ ਹਿਸਹਲਮ ਚੁਰਕ ਦੀ ਚੱਕਰ ਪਥ ਭਤੀ ਮੀਚ ਦੁੰਡ ਮੀ। - ਪਛਟਉ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>ਮਾਲੁ - ਵਹੀ ਅਲਹ ਹੁੰਦਾ ਕਾਲਾ , ਕੁਹਾਦ ਪਰ ਬਿਖੇ ਤੇ ਹੇ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>ਸੀਕ ਹੁੰਦ ਹਿਸਹਲਮ ਹੇ ਮਿਸਤਾਨੁਮ ਢੁੰਝ ਪੀ ਹਿਸਹਲਮ ਹੁੰਦੀ ਅਧਾਰਤਰ ਬਹਿਸ਼।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>ਮਾਲੁ - ਦੁਰ ਟੇ ਕੇਸ! ਜੂ ਪੱਥ ਮੀ ਟਹਾਕੜੀ ਕੇ ਟੇ ਟੇ ਦੁੰਡ ਮੀਤ ਦੇ ਕਮਤ ਮਹਿ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>ਬੀ ਬੀਜੀ ਜਾਲ ਹੁੰਦ ਦੇ ਕੇਕ ਦੀ ਦਰਕ ਮਿਚਾ ਢਾਉਂਦੇ ਤੇ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>ਹੁੰਮ ਹੁੰਦ ਦੇ ਹਿਸਹਲਮ ਦਾ ਵਾਣਾ - ਵੱਢਾ ਦਰਕਰ ਸਹਾ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>ਬੀ ਹੁੰਦ ਹੁੰਦ ਮਿਸਤਾਨ ਬੇਲੀ ਬੀਸਲ ਦੇਣ ਤੇ ਹੁੰਦ ਤੇ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>ਹਿਸਹਲਮ - ਉਹ ਮੀ ਦੁਰ ਬਿਖੇਦੀਅ ਤੇ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>ਬੇਕਰਾਹੇ ਹੁੰਦੇ ਹੂਂ ਦੁਰ - ਹੁੰਦ ਨਾਹਾਟਿਆ ਵਰਤੀ ਟੇ। ਹੁੰਮੀ ਮੀ ਤੇ ਹੁੰਮੀ ਅਧਾਰਤਰ ਹੇ ਢਾਉਂਦ ਬਹਤਾ ਮਹਿ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>ਬਕਰੁਸ ਹੁੰਦ ਹੁੰਮੀ ਟੈਜ਼ਮ ਟਰਕ਼ ਦੁੰਡ ਮੁਖ ਢੁੰਝ ਹੁੰਦ ਅਧਾਰਤਰ ਮੇਵਾ ਦੇਸ਼ਿਆ। ਬੇਕਰਾਹੇ - ਪਾਵ , ਮੀ ਤੇਲ ਦੇ ਵਹਾਨਾ ਤੇ। ਬੀ ਹੁੰਮ ਹੁੰਮੀ ਹਿਸਹਲਮ ਹੁੰਦੀ ਵਹੀ ਤੇ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
17. टर्नाम उं मित डोंग्रफ बदे अपां भवन हुई उड़ाए ताए भव भर निजी बेहतर दीया, धुंती दें उं एन उव खुलन भजो वहो।

18. भौं बी, तन भर वृद्ध बाँधे ते डिब्बे अपां फिक्स भरे उं में डिब्बे जरे डिब्बे में भरे उं एन ती गुण दिन।

19. लगा गहरीपृथ्वी भर्ती दे घड़ दे डिब्बे बिंच मल बादे तेवर डिब्बे हुई विराम धिंड बत तवी मल उं घर मल तें दिवी मर। टेंडी बिंच भुंते - बुजी दे बिवाद दी गोल्फ वेंशी ने गाय भी।

20. भुजी ते भाजी दी मल मजबीय हुई जबे उं गटा दिंडा, धूम हुई दिउं मुलीशन वें बाले बाल गलाच दा बीटा भूं बर भवनीय। भेकी - भेकी टेंडी उसके मजबरे, मल भवनीय भुंदुदा दिंडी।

21. भिमटां बत्तमा— (भुं बतु एंड देवले) टीबू उत्तरा भगत विमानी जां बि जुमी लेके मूर्त दाहर जवाब तर्कियो।

22. सुलम दिंडा बि भवनीय टे हमे टे में जार जनक दूरथ भिमटा। भुजी गहरीपृथ्वी टे फिक्स दे घड़ जनक दिंडा ने डिब्बे डी टेंडी दे डिब्बे जरे तिक्स भवनीय मी।

23. बज— (भवनीय) चुआ। ‘भवनीय ही बेसी बिंडी है।

24. अपां फिक्स दा नाह भवनीय टेंडी टीबू बिंच टेंडी दिंडा विमान। वि जाम भवनीय टे विमान मल भर तवी बाधै। धुंती दें उं टेंडी उं टेंडी ली देंफ बेटी तवी भवन नते टीबू हुई हो तवी बाधै उं डिउं टेंडी टिस वंश मम्रे सग। टेंडी विमानीय चे टेंडी देसी चोटवे भव भवनीय भुं एंड भर नपते उं वंदे बदे सवाले दही तवी। दिव भवनीय भवनीय टे उं एन भवन के तैल नाही।

25. भवनीय— (टेंडी नजर हें) टीबू मल चकाच में देवले टेंडी।

Language in India www.languageinindia.com
10 : 10 October 2010
Vishal Goyal, Ph.D.
Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>26.</td>
<td>ਅੱਮੂਦੋ—ਧੈ ਗੁਰ ਬੈਟ ਵਿਸ਼ਵਵ ਜਗਾਤ ਦੇ। ਮੇ ਉੱਤ ਵੱਡੇ ਵਿਸ਼ਵ ਜਗਾਤ ਦੇ।</td>
</tr>
<tr>
<td>27.</td>
<td>ਨੀ ਜੁਰੇ ਦੇ ਕੇਸੀ ਭਰ-ਵੜੀਤ ਕੋਠ ਤਾਫ਼ ਛਾਣਵੁਂ।</td>
</tr>
<tr>
<td>28.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਸੌ ਦੇ ਹਿਸਾਬੇ ਪੱਛਮ ਮੇ ਅਪ੍ਰਤੇ ਦੁ ਵਿਸ਼ਵਵ ਬਵਾ ਵਿਸ਼ਵਵਵਾਂ।</td>
</tr>
<tr>
<td>29.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਵਿਸ਼ਵਵਵਾਂ।</td>
</tr>
<tr>
<td>30.</td>
<td>ਵੱਡੀ ਹਿਸਾਬੇ ਬੁੰਗਾਲੀ ਅੱਠਲੋਖਾ ਅੱਠਲੋਖਾ ਬੁੰਗਾਲੀ ਦੇ ਖਪਠਲ਼ ਦੇ ਪਤ ਰੱਖ - ਚੰਗ ਦੁ ਵਾਇਤਰਾ ਸਰਾਂ ਕੱਚ ਹੀਦ ਮੀ।</td>
</tr>
<tr>
<td>31.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਵਿਸ਼ਵਵਵਾਂ।</td>
</tr>
<tr>
<td>32.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਵਿਸ਼ਵਵਵਾਂ।</td>
</tr>
<tr>
<td>33.</td>
<td>ਚੰਗ ਦੇ ਹਿਸਾਬੇ ਵਿਸ਼ਵਵਵਾਂ।</td>
</tr>
<tr>
<td>34.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>35.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>36.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>37.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>38.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>39.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>40.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>41.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>42.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
<tr>
<td>43.</td>
<td>ਹੀ ਜੁਰੇ ਦੇ ਹਿਸਾਬੇ ਇੱਕ ਜਾਲੀ।</td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com) 910
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
44. भनिए बूढ़े उत्साह भरे गायक, निर्देश पुलिस ते संबंध दें भूलना दी बि गिरे ही घटे, छूट विलेख ली छुट दित दें तिमिरजी चढ़ा लिए।

45. तुम दिच्छ मिरध घवउ दंडें लेखव दें से, तिरठे दंडें बि माती हुतीभा दुवळा नारदी है।

46. तबीतब ठवउ दी दिंच दंडी दी गीत देंत दसी उचला है।

47. देवदूर चला बिभा भरे वाक्से दिल संदे इन दवडिस्वा दंडे। दुमें दंच दिच्छ दुरु बंदीभा दी मुखी मी।

48. आच्छा है बुर्री है तिरला भेद छुंदा है।

49. बिखर दं भुरस्वं दिल बि उत आच्छा आतीरी मधुस्वं दे अतमर मां हैं। बे हे मुकुट लक्ष्मण दे अतमर पड़े।

50. आमी आक बाधे दं बि भागव शिम विलंघं है दंडे चढ़ राध भलों, दुमें दंच दी भागवए। तिहए दंच उपयुक्त भूलता देंत दो।
### Intelligibility Test - Articles

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ਅਨੀਮੀਆ (Anemia) ਦੇ ਵਾਲਾ ਮੋਉੱਟ ਹੁਣਿਆ ਹੋਇਆ ਮਰਨੀ ਸੀ, ਰੁੱਚੀ ਬੋਟਨ ਦੇਣ ਮੰਨੀ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਸੀ, ਬੋਟਨ ਦਾ ਭਲਾ ਰ ਦੀ ਖਾਤਮਕਾ ਮੰਨੀ ਹੋਇਆ ਹੋਇਆ ਸੀ, ਰੁੱਚੀ ਹਿੰਡ ਪਾੜਤਾ ਦੀ ਦਿਸ਼ਾ - ਮਾਮ ਦੇਡ ਹੁਣਿਆ ਉਪਲਬਧ ਮੰਨੀ ਹੋਇਆ ਸੀ। ਹੌਲ ਦੰਦਾ ਉਪਰ ਵਾਦੀਆਂ ਹੁਣਿਆ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਸੀ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| 2.    | ਪੇਰੀਗੈ (pregnancy) ਦੇ ਵਾਲਾ ਹੁਣਿਆ ਹੋਇਆ ਜਿੜਾ ਲੇਣ ਦੇਣ ਪੇਟੀ ਦੀ ਜਿੜ ਦੇਣ ਦੇਣ ਲੈਖ ਦੇ ਲੋਕੀ ਰੋਕੀਆਂ 
       | ਮੰਨੀ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਸੀ। |
| 3.    | ਮਾਮਾ ਵਾਲੀ : 150 ਗਾਂਧ ਰਾਜਾ ਦੰਦ, 20 ਗਾਂਧ ਮੂਰਨਾਂ ਦੰਦ, 100 ਗਾਂਧ ਪ੍ਰਦ, 100 ਭਾ. ਰਹੀ। 
       | ਦੇਦਾ, 15 ਗਾਂਧ ਦਾਸ਼ਾਲੀ, 3 ਗਾਂਧ ਜੀਂਗ, 5 
       | ਗਾਂਧ ਜੀਮਲੀ, 5 ਗਾਂਧ ਭੂਕ ਮੂੰਦ, 5 ਗਾਂਧ ਰੁੱਦ, 
       | 3 ਗਾਂਧ ਰੱਲਦੀ, 5 ਗਾਂਧ ਲਾਵੀਕੀਪਾਨ, 5 ਗਾਂਧ, 
       | 3 ਗਾਂਧ ਵਲੀਪ੊ਂਪਾ, 3 ਗਾਂਧ ਲਾਭਜ਼ (ਵਿਸ਼ਣ ਲਾਭਜ਼), ਮਾਸ਼ਨ ਚਾਰੀ ਵਾਲੀ ਤ੍ਰੀਧਾ। |
| 4.    | ਅਧਿਮਵਲੀਭ ਦੇ ਦੇਣ : - ਨੇਵਰੀ ਅਧਿਮਵਲੀਭ ਦੇ 
       | ਦੇਣ ਬਹੁਣੇ ਹੁਣਾ ਕਾਦੇ ਅਨਮੀਆ ਦੇ ਪਧੀ ਪੱਧੀ। |
| 5.    | ਦੰਦਾ ਹੀ ਬੀਖ ਸਹਿਯੋਗ ਮੰਨਾ ਮਹਤੀ ਦੇ ਰੱਸ ਹੋਣਾ ਹਿੰਦਾ 
       | ਦੰਦ ਭੂਸਿੱਟਾਂ ਦੇ ਬੀਖ ਦੇ ਮਹਤੀ ਦੇਣ ਦੇਣ ਹੋਇਆ ਹੋਇਆ ਸੀ। |
| 6.    | ਮੇਟੀ ਦੇ ਨੇਵਰੀ ਦੇ ਰਹਿਮੀ ਚਲਾਕੀ ਮੇਟ ਦੇ ਮਹੱਤ ਤੇ ਪ੍ਰਤੀ ਦੇਣ ਨੇ ਨੇ ਦੇਣ |
       | ਚਲਾਕ ਦੇਣਾ ਹੈ। |
| 7.    | ਪੈਦੇ ਹੀ ਹੁਣਾ ਮੰਨੀ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਹੋਇਆ ਸੀ। |
8. अभि वेधियाँ भविष्यांत तेजवीऽथर नवीऽं दत समाभय निरचन मां हत जह धरण तुजहदीऽं कह मैयुव भवरतः हिंदू नाँ निनहुँ दे मामे हिंदू भद दे बचते दे दूराँ हु घचच म दी देश्वर दी महीम रुड़ी तुड़ी दे भव देश्वर भवरतः हिंदू मां दे दलबत सन हु दे घचच घचच म दी देश्वर रुड़ी तुड़ी दी निनहुँ भविष्यांत अध्ययन हक़च म दे सिव आध्यात्मिक ही उत्साह बनती है।

9. अध्ययन है हीयेन्सिया का नसल भाँद दो हमें पूर्ण बन्ह लाही बुढ़ आजन्योगिक देश्वर , हिंदी मदलख है वेंस पुलिस रतम है

10. जवाँ दा जवाँ दे , मरुभ दा वेंस दे , मदलख दा मलख दे , विंद दा जवां रतम है मिल्ह हु दी पेवा विन्त विभाग है

11. भिननव के हिंदू 64 मस्त दा भिनवी नवेश की 50 पवलीन हाँ मां 100 बांदे गर । भिननव के सेवरांगा 80 बीभ दुल ववरांगा पिंड दा भिनवी नवेश अध्ययन हे पवलख है 180 दे सिवभायत मदलख दे हां भवजी दे मबदे बढ़े पवलख दे म्यम दे खुप हिंद नवविक नांग है।

12. हिंद बेठा बचन दुवे घचच हूँ , नेवर दित हु मुका भों निविविक उं बी देविगा , दुःसे घचच के नलख हिंदू , विजनि दं विल घचच नांगिया।

13. दीववचल हं उचवर धर दी आजड़ मी लेबिल भवचल (Akbar) ल ही मदर हिंदू दित भवचल हे उचवर है घें हिंद है सा हूँ बाध्य हूँ घें देखवे दित दीववचल हिंद देबिगा उचवर हिंदे दी हुजी चीन है , गाय उव हिंद हं तृती बदने ।
14. दिन में फलीचौरी चीजों खाने तर बिटूने हुए हेज खाए जाने नहीं अब चाहिए तुम्हीं उन भेजवट न हो हारी है ।

15. तुमें भूल अपनिविष मिलियिंग में आउँधम उत्ती वे घराँ दिखा आ चोदे है । हिम सधी सतर्की है वि आप दी एक में लोगों दिखा सजन हलिण भिजाए भूलियों उठां बदलो लटो , आमी उंद हिम बेम तु हिम उड़ीवें दे बढ़ते आ उड़े है भूले अभिन्न ही बढ़ेँ ।
<table>
<thead>
<tr>
<th>S.No.</th>
<th>Sentence</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ਮੂਲਐਣਾ ਢੋਙ ਯਸਚਾਂ ਲਠ ਞ਱ ਲ਑ਆ ਯ਷</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ਪੁੰਚਾਲ ਹਿਪਸਾ ਦੀ ਬਣੀ ਨਾ ਚਖੀ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ਵਾਲੇ ਹਿਪਸਾ ਜੀ ਬੇਨੀ ਨਾ ਸੁੱਚੀ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ਹੁੰਦੇ ਬਵੇ ਅਹਾਂ ਘਟਦਾਂ ਬੀਜੀ ਨਾਂ ਦੇਖੇ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ਭਭਤੇ ਬੀੜ ਬਵਦਾਂ ਬੀਜੀ ਨਾ ਲਾਏ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>ਪਹਾਣ ਦੀ ਹੀਂ ਚਾਰਾਂ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>ਹਿਪਸਾ ਦੇ ਪ੍ਰਕਾਸ ਬਚਣ ਨਵੂੰ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>ਬੇੜੀ ਦੀ ਉਰੈ ਦੇ ਨਾਟੂ ਬਚਣਾ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>ਬਲੰ - ਮੁੱਢੀ ਰਲ ਬੇਨੀ ਨਾ ਦਾਖਲੀ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>ਅਸਾਸੀ ਹੋਵਾ ਬਰਾ ਦੀ ਨਾਖੀ ਗਾਇ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ਸੀ ਵੇ ਹੇ ਮੁੰਬਦ ਦੇ ਸਾਥੀ ਜੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>ਘਣਟ ਬੀਂ - ਡੀ ਬਾਲ ਹਿਪਸਾ ਲਾਖ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>ਝਨਟ ਹੀਂ ਹਿਪਸਾ ਦੇ ਸਾਥੀ ਜੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>ਮਹੀਂ ਤਰਹ ਬੀਜੀ ਬੇਣਾ ਨਾ ਮਹਲਣ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>ਭਭਤੇ ਬੀੜ ਼ਤਾਂ ਚੱਲ ਵਖੀ ਹੈ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Intelligibility Test - Blogs

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>ਦੈਂ ਅਲਖਯਲ ਇਮਪੀ ਲਪਐ ਲ਩ਯਲ ਯਾਂ ਲਏ ਧਸ਼਩ਲ ਇ ਏ ਠਸਮਙ ਖਸ ਧਸ਼਩ਲ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>ਜਨਨਸਮ , ਹੰਡੀ ਹੰਡੀਅਂ ਦੇ ਸਿਖਾਉਉਲਾ ਧਤਚ ਸਿੰਘ ਲਿਸ਼ਤ ਼ਰਜਮੋਂ ਦੀ ਤਲਜ , ਹੁੰਦੇ ਹੈਅਂ ਦੀ ਸੁੰਨੀ ਦੇ ਧਤਚ ਦਿਸਤਗਲ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>ਉੱਜ ਦੇਵ ਮੁਚ ਭਰਾਂ ਪੁਰਾਤ ਕਿਸਾ ਕੁੱਠ ਵਮ - ਦਰਤ ਦੇਨ ਸਿੰਘ ਉਂ ਤੁਮੀ ਦੀ ਬਲ ਬੰਕੈ ਵਲ ! ਪਥਵਾਲ ਦੀ ਬਾਲੀ ਦੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>ਸਿੰਘ ਦੇਵ ਮੇ ਸੰਸਾਰ ਵਾਂ , ਹੰਡੀ ਦੇ ਸਿੰਘ ਬੱਚੇ ਘਿੱਲ ਦੀ ਸੌਮਿਤ ਮਲੋਟਲਾ ਧਿ ਸਿਕੀ ਤਨ ਉਤੇ ਜੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>ਸਿੰਘ ਉੱ ਮੱਚ ਦੀ , ਹੰਡੀ ਦੀ ਬਲੀ ਪਸੇਟਪੰ ਦੋ ਹਿੱਲ ਤੇਰਾਲਾਂ ਖਹਾਂ ਦੱਖਹ ਨਾ ਖੱਚੀ ਨਾ ਖੱਚੀ ਮੱਚ ਦੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>ਭਾਪ ਤ੍ਰਹਲ ਖੁਭਾਂ ਕਿਸਾ ਪੁਰਾਤ ਧਾਰਨ ਮਾਦਰਾ ਦੇਵਪਲਾ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>ਖ਼ੁਝ ਸਵੀ ਸਿਹਿਤ ਹੈ ਤੁਮ ਦਾਸ ਖੁਦਾ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>ਤੁਮ ਨਹੀ ਹੀਸਤ ਛੋਕੜੀ ਬਲਟ ਚਾਰਟਲਾਂ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>ਖ਼ੁਝ ਸਤੀ ਭੇਟ ਦੇਵ ਡਾਟਕਿਸਲਾਈ ਕੀ ਦੋਹਾਂ ਟੇਮਲਾ ਮੈਂ ਕਰਕੀ ਦੱਸਤ ਦੱਸਤੇਸ਼ ਦੱਸਤ ਅਨੁਸਾਰ ( ਮਾਦਰਾਂ ) ਬਲਟ ਮੱਚ ਉੱ ਧਾਰਨ ਮਾਦਰਾ ਦੇ ਖ਼ੁਝ ਬਲਟ ਦੇ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>ਜਿਸਦੀ ਜਾਂਦਾ ਬਲ ਹੰਡੀ ਦੀ ਦੀ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>ਸਿੰਘ ਦੇਵ ਮੱਚ ਦੇ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
12. चिल्ल टीव्ह - टीव घटाशिश विषय है
13. विभ दिशा चले हिंदी वडीवड़ के दिशा चले।
14. उमी निंदे देवे दा मादी भीम से नूँ भूती विभाग वि मरिशन हूँ धुराफिंद उखां।
15. अछू अमल वेमिमल लहरी दपाड़ी। हिंद सरलफ ठेमट दिशा भव बैल विभाग दूं।
Appendix C

Test Data Set for Accuracy Test

Accuracy Evaluation:
The evaluators are provided with source text along with translated text. A highly intelligible output sentence need not be a correct translation of the source sentence. It is important to check whether the meaning of the source language sentence is preserved in the translation. This property is called accuracy.

Scoring:
The scoring is done based on the degree of intelligibility and comprehensibility. A Four point scale is made in which highest point is assigned to those sentences that look perfectly alike the target language and lowest point is assigned to the sentence which is un-understandable and unacceptable. The scale looks like:

Score 3 : Completely Faithful
Score 2: Fairly faithful: more than 50 % of the original information passes in the translation.
Score 1: Barely faithful: less than 50 % of the original information passes in the translation.
Score 0: Completely Unfaithful. Does not make sense.
## Accuracy Test - News

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hindi Sentence</th>
<th>Punjabi Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>मुंबई। रिजर्व बैंक ने शुक्रवार को कहा कि वैश्विक आर्थिक संकट के प्रभाव से देश की अर्थव्यवस्था के उबरने के बाद वह मुद्दास्पदति के अनुमानों और मध्यम काल में इसके परिणामों के प्रबंधन पर ध्यान देगा।</td>
<td>भंपई। जिनबहुत बैंक ने मूल्यांकन के किस्म के संशोधन के लिए विश्व संघ की भागीदारी में पाकिस्तान में व्यवस्था के आगमन के प्रबाव से देश की अवस्था के उफयने के फाद वह भरास्पीत के अन भानों और भड़मभ कार भें इसके रूरणाभों के प्रफंधन रूप में देगा।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>वाशिंगटन। अमेरिकी बैंकों के स्ट्रेस टेस्ट का नतीजा आखखयकाय आ ही गया।</td>
<td>हासिल किया। अमेरिकी बैंकें ने मात्रेम टेस्ट दे तीजार अभियंताएँ भा ती विभाग।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>जिन बैंकों को पूंजी की आवश्यकता है, उन्हें योजना बनाने के लिए 8 जून तक का समय दिया गया है।</td>
<td>निल्लें बैंकें ने पूंजी की रेट्र देने, हैल्सें बैंकें योजना घोषित की 8 जून तक उंच दा मभ डिंगु विभाग दे।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>बैंकों को यह योजना अपने नियमों पर मांजर करानी होगी।</td>
<td>बैंकें ने एक योजना अपने विभागों के भरमाण लवकर ती किया।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>अधिकारियों का कहना है कि आर्थिक स्थिति में सुधार के लिए मजबूत बैंकिंग तंत्र जरूरी है।</td>
<td>अधिकारियों का कहना है कि आर्थिक स्थिति में सुधार के लिए मजबूत बैंकिंग तंत्र जरूरी है।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>इस टेस्ट से निवेशकों में यह भरोसा लोटेगा कि सारे बैंक कम्युनिकेशन नहीं हैं।</td>
<td>धिम टेस्ट डें डिजाइन इंच हिंदेहि जेलम्बा पहुंचेगा जि मर्हे बैंक अभिनेत तत्त्व गह।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>साथ ही कम्युनिकेशन बैंकों में भी सुधार किया जा सकता है।</td>
<td>रथ दी अभिनेत केंद्र इंच ही मुआद्व सीओ न मल्पत है।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>विक्रम पंडित की अगुवाई वाली सिद्दी ने कहा कि वह 5.5 अरब डॉलर अतिरिक्त पूंजी जुटाने के लिए पब्लिक एक्सचेंज आफर का दायरा बढाएगी।</td>
<td>डिजाइन भड़उं दी भगुवारी ह्थी मिटी ई बिग बि ह्येर 5.5 अरब डॉलर डिस्ट्रेक्शन पूंजी मुटट रसी पिचलिय शेल्म्बेंस अब्दु दा लाद्विर डिजिसी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>न्यूयॉर्क। अमेरिकी शेयर बाजार बृहस्पतिवार को गिरावट के साथ बंद हुए।</td>
<td>न्यूयॉर्क। अभवीजी मेस्वार घानच डीवर्ड ह्यू डिक्ट डे रथ बैंड डें।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>उधय, बृहस्पतिवार को बाजार बंद होने के बाद सरकार ने प्रमुख बैंकों के स्ट्रेस टेस्ट के नतीजे घोषित किए जिनके मुलाकक देश के 10 प्रमुख बैंकों को अपने बचाव के लिए और नकदी एकत्रित करनी पड़ेगी।</td>
<td>उधय, डीवर्ड ह्यू घानच बैंड डें घानच मर्वाद डे घानच ह्यू बैंड डें माटवें डेंडर मिटी बीडुं दे भुगुवार बैंड डे 10 पूंजी मुटट ह्यू अथै मल्पत रसी देव लगाडी श्वेती बलडी पड़े।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>वरूण माँधी ने पीलीभीत में मुसलमानों के खिलाफ अंडकाज भाषण दिए थे</td>
<td>ड्वूट गांपी के पीलीडीजु हिंच भ्रमालण्ड दे भिलाद डॉउट पु डागम्ट डिंडू मर।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

12. इस मामले के प्रकाश में आने के बाद चुनाव आयोग के निर्देश पर वस्तु गांधी के खिलाफ एफआईआर दर्ज हुई थी।

13. पहले ही वित्तीय वर्ष यानी वर्ष 2004-05 में केंद्र ने राज्य सरकार को 2831.82 करोड़ रुपए की राशि उपलब्ध कराई।

14. संपा के लिए पिछले चुनाव में जीती 32 में से अपनी 15 सीटों को बचाने की चुनौती है, जिनमें से लगभग आधा दर्जन सीटों की हार-जीत तो कल्याण कैंडर का कसौटी पर ही कसा जाएगा।

15. उन्होंने भाजपा नेताओं को इस तरह के बयान न देने की सलाह दी। इसके साथ ही शरद यादव ने विदेशी बैंकों में भारतीयों के जमा कार्ड धान का मुद्दा फिर उठाया।
| 16. | राहुल ने कहा, "मनमोहन सिह हमारे प्रधानमंत्री हैं, तो यूपीए के भी प्रधानमंत्री हैं।" | याह र ने कहा, "भतभेन्द्र मिथ्य मठे भूपरंभेडूली घर . दवु झुपोड़े घर है भूपरंभेडूजी घर।" |
| 17. | राहुल गांधी दो दिन के चुनाव प्रचार पर राजस्थान में हैं। | याह र गांधी दो हदन के नाव प्रचाय याजस्थान भें हैं। |
| 18. | आधिक संकट का दबाव झोलने की क्षमता आकर वाले इस टेस्ट में अमेरिका के 10 बड़े बैंक बेडम निकले हैं। | आधिक मंबट ए सघन्ध लेट दी मंभोक अंबट हसे हिम टेमट हिंच भभतीजी है 10 दौं घेव केरन फिवडे घर। |
| 19. | कंपनी की इस पहल से उसे अतिरिक्त सरकारी सहायता या सरकारी प्रतिभूतियों को सामान्य शेयरों में परिवर्तित किए बगैर अपना पूंजी आधार बढ़ाने में मदद मिलेगी। | कंपनी की दीम बधिल दौ सघन्ध लेट दी मंभोक मंजिडू नं मंभोक मुजीजुडीजी हैं दिखे फिने मेनार्ड हिंच भविधबूडू वीजे दिखं भरपट पूंजी अथान ह्यापुंड हिंच हँटट भिलेजी। |
| 20. | उन्होंने कहा कि राजम द्वारा इसे चुनावी मुद्दा बनाए जाने के बाद मजबूरी में मनमोहन सरकार अब कार्यवाह करने का दिखावा कर रही है। | उन्होंने कहा कि राजम द्वारा इसे चुनावी मुद्दा बनाए जाने के बाद मजबूरी में मनमोहन सरकार अब कार्यवाह करने का दिखावा कर रही है। |

|   | मंगलवार दे भुजाघर ,
|   | टाइम 100 मीमवरल हिंदू भभी छ्रड़ुँ तेज़ द रहभ कहि ठहर ने मध्ये हटवीभा हुँ मझूँ निमित्त भुजाघर बवठे छे । |
| 22. | मेगास्टार अभित्राब बच्चन ने नो अनन संवैधानिक दायित्व का नियम करने के साथ ही अपने सामाजिक दायित्व का भी बख़्ती निवाह किया । |
|   | भेगास्टाय अलभताब फच्चन ने कर अऩने संव धाननक दानमत्व का ननवाकह कयने के साथ ही अऩने साभािजक दानमत्व का बी फख फी ननवाकह क्रकमा । |
| 23. | इसके अलावा उप मुझमंत्री सूजकीर बादल ने भी इंटरनेट पर कई प्रोफाइल बना रखी हैं । |
|   | दिमेह दिल्ली हुँध भक्कभेड़वी मध्यवर्त घटन ने ही हंटरेटेट हुँधे बयी पेशा घटन तेंधी है । |
| 24. | हाल ही में लता मंगेशकर ने मधुर भंडारकर की फिल्म 'जेल' में एक धार्मिक गीत रिकॉर्ड किया है । |
|   | उए दी हिंदू संग्रेम भेजीखर ते महुद ढांढुबव दी हिंदु 'नेलु' हिंदू हिंदू पर्वाभिं भी विकरुद बोझ दे । |
| 25. | हिमेश जी, बात तो सही है लेकिन कर्ज के हिश के बाद आपको नहीं लगता कि दर्शकों का आपको हीरो के रूप में स्वीकार करना थोड़ा मुश्किल होगा । |
|   | दिमेह ती , जो म्ज हीरव ते केहिं बवत दे गँठ दे ब्लभर उगँठे तरी हमारे वि चंबल द उगँठे तोवे दे बुध हिंदू भवीवर बवठ बेठे भुजवत ठेड़ीगा । |
26. उन्होंने भी लोकसभा चुनाव में जीत का दावा किया है।

<table>
<thead>
<tr>
<th>उन्होंने भी लोकसभा चुनाव में जीत का दावा किया है।</th>
</tr>
</thead>
</table>

27. फिल्म स्टार्डिंग स्लिमियनेयर की बाल किलाकार खीना अली काफी प्रसिद्ध हो गई है।

<table>
<thead>
<tr>
<th>फिल्म स्टार्डिंग स्लिमियनेयर की बाल किलाकार खीना अली काफी प्रसिद्ध हो गई है।</th>
</tr>
</thead>
</table>

28. मैं इस परिवार को पिछले बीस सालों से जानता हूँ, रफीक बहुत शरीफ आदमी है, वो ऐसी हरकत कभी नहीं करेगा।

<table>
<thead>
<tr>
<th>मैं इस परिवार को पिछले बीस सालों से जानता हूँ, रफीक बहुत शरीफ आदमी है, वो ऐसी हरकत कभी नहीं करेगा।</th>
</tr>
</thead>
</table>

29. कीना पूरे दिन घास से बने मधान पर तो कभी बैलगाड़ी पर मस्ती करती नजर आई।

<table>
<thead>
<tr>
<th>कीना पूरे दिन घास से बने मधान पर तो कभी बैलगाड़ी पर मस्ती करती नजर आई।</th>
</tr>
</thead>
</table>

30. शुकवार को कीना के आने के साथ ही शहर में सेफ के भी आने की उम्मीद लगाई जा रही थी।

<table>
<thead>
<tr>
<th>शुकवार को कीना के आने के साथ ही शहर में सेफ के भी आने की उम्मीद लगाई जा रही थी।</th>
</tr>
</thead>
</table>

31. हाल में मोहम्मद में हुई सैन्य कार्रवाई में 18 चरममंथी मारे गए थे।

<table>
<thead>
<tr>
<th>हाल में मोहम्मद में हुई सैन्य कार्रवाई में 18 चरममंथी मारे गए थे।</th>
</tr>
</thead>
</table>

32. गाँवतलब है कि पाकिस्तान के पश्चिमी इलाक़े में बुनर में पिछले कुछ हफ्तों में सेना और तलबाला चरममंथियों के बीच भीषण संघर्ष हुआ है।

<table>
<thead>
<tr>
<th>गाँवतलब है कि पाकिस्तान के पश्चिमी इलाक़े में बुनर में पिछले कुछ हफ्तों में सेना और तलबाला चरममंथियों के बीच भीषण संघर्ष हुआ है।</th>
</tr>
</thead>
</table>
33. इस्लामाबाद,। पाकिस्तान के गृह मंत्री रहमान मलिक ने सोमवार को कहा कि देश के अर्शात पत्थरोत्स्पर्श क्षेत्र में चल रहे एक बड़े सैन्य अभियान में करीब 700 तालिबान आतंकीयों को मार गिराया गया है और सभी आतंकीयों का खत्मा होने तक यहां सैन्य कार्रवाई जारी रहेगी।

34. गृह मंत्री ने कहा कि यह पूरे देश के लिए एक परीक्षा है।

35. पाकिस्तान के सीमावर्ती इलाकों में मिसाइल हमले होते रहे हैं और इसके लिए पाकिस्तान अमेरिका पर आरोप लगाता रहा है:

36. प्रधानमंत्री के बयान की भाजपा ने भी आलोचना की है। दिल्ली प्रदेश के महामंत्री आरपी सिंह ने कहा कि प्रधानमंत्री ने पद की गरिमा धूमिल की है।
| 37. प्रधानमंत्री के सरकारी आवास पर हुई यह बातचीत | पुष्परंभजी के मध्यवर्ती प्रति जैसी हिंदी दर्शन पर हिंदी तैयारी के लिए अन्य भाषाओं पर हिंदी भए हैं। भारतीय भाषा के लिए मलयालम पत्रकार अरविंद बिन्दियार ने अपने अंतिम साल के दौरान हिंदी में विश्वभर के सभी के लिए अन्य भाषाओं पर हिंदी भक्ति वर्तमान \n
| 38. गुजरात में 2002 में हुए दंगों के कुछ मामलों में प्रतिदिन सुनवाई के आधार पर प्रक्रिया शुरू | गुजरात दिनेश 2002 दिनेश गौरव की दिनेश गौरव के लिए अन्य भाषाओं पर हिंदी भए हैं। गौरव की दिनेश गौरव के लिए अन्य भाषाओं पर हिंदी भक्ति वर्तमान \n
| 39. मौसम विभाग ने अंडमान के सागर में मानसून की सालाना बारिश ठोड़ा देर से होने की आशंका जताई है। | भूभाषा की दिनेश 2002 दिनेश गौरव की दिनेश गौरव के लिए अन्य भाषाओं पर हिंदी भए हैं। गौरव की दिनेश गौरव के लिए अन्य भाषाओं पर हिंदी भक्ति वर्तमान \n
| 40. विष्णु की तीन सबसे मोटी बर्फीली परतों में से एक प्रथमी अंतर्क्षितक की तह | तिथिदारी की दिनेश 2002 में भिंडू विभाषा की दिनेश 2002 में भिंडू विभाषा के लिए अन्य भाषाओं पर हिंदी भए हैं। भिंडू विभाषा की दिनेश 2002 में भिंडू विभाषा के लिए अन्य भाषाओं पर हिंदी भक्ति वर्तमान \n
<p>| 41. फिलहाल डरने की जस्त नहीं | विष्णु की तीन सबसे मोटी बर्फीली परतों में से एक प्रथमी अंतर्क्षितक की तह। | तिथिदारी की दिनेश 2002 में भिंडू विभाषा की दिनेश 2002 में भिंडू विभाषा के लिए अन्य भाषाओं पर हिंदी भए हैं। भिंडू विभाषा की दिनेश 2002 में भिंडू विभाषा के लिए अन्य भाषाओं पर हिंदी भक्ति वर्तमान |</p>
<table>
<thead>
<tr>
<th>हाइलाईट किए गए कॉलंक</th>
<th>हिंदी</th>
<th>पंजाबी</th>
</tr>
</thead>
<tbody>
<tr>
<td>42. उनके और पाइट के विद्यार्थियों के बीच का संवाद बेहद रोचक रहा।</td>
<td>उनके औय ऩाइट के ववद्यागथकमों के फीच का संवाद फेहद योचक यहा।</td>
<td></td>
</tr>
<tr>
<td>43. इंदिरा गांधी राष्ट्रीय मुक्तिविधायत अनुसंधान अड्डी मैनेजर ऑफिस [इडॉल्डॉ] के छात्रों को उनकी मांग पर घर बैठे परीक्षा देने की सुविधा मिलने जा रही है।</td>
<td>हिंदी गांधी राष्ट्रीय मुक्तिविधायत अनुसंधान पुरस्कार विशेष [हिंदू] दे हिंदिमधवियाँ खूं-खूं ची भनेगा भन भन बैठे भलीनिभा रेन दी मुखिया भलकट नआ रही है।</td>
<td></td>
</tr>
<tr>
<td>44. धोनी ने कसनी पारी खेली</td>
<td>धोनी ने कप्तानी खेरी डस्डल्डेण णारल्ड्स्ल फ ठे बनाई देने की स्वाध्य लभरने जा यही है।</td>
<td></td>
</tr>
<tr>
<td>45. इसके बाद सचिन तेंदुलकर टीम की नेता पार लगाने के लिए मैदान पर अभाए लेकिन वह अधिक रन नहीं बना सके।</td>
<td>दिमटे घब्बट ममक्त उदक्षगत टीम ची देंगी। भव सच्चाईट देंगी मेंटर भव अण्टे हेविन हुए</td>
<td>निमंत्रण वह तरी बहड मचे।</td>
</tr>
<tr>
<td>46. राजस्थान की शुरुआत बेहद खराब रही और एक बार दबाव में आने के बाद उसके सभी बल्लेबाज अपना विकेट फैंकर चलते बने।</td>
<td>जर्नमल्डकह ची मुखभाष भेड़ट भलठ दी भजे हिंद रान एश्म दिंह भारी दे घब्बट फ्रिमेट मचे। बल्लेबाज अभाए रिबेट मुटटे चलटे बने।</td>
<td></td>
</tr>
<tr>
<td>47. विश्व बैडमिंटन चैंपियनशिप को तैयार भारत May 08, 06:45 pm</td>
<td>मंगल वैडमिंटन वैधिकोस्चिंच है तिमल्डव व्होट May 08, 06 : 45 pm</td>
<td></td>
</tr>
<tr>
<td>48.</td>
<td>मोहन बागान अभी तक खाता नहीं खोल पाया है और ग्रुप में सबसे निचले स्थान पर है।</td>
<td>मेहरबन घराने घुटे डेंव धावा रूही मेहरबन परिवार तैयार है गैर बालुध डिंच मड़ूं डे ठठे मजबूर वह है।</td>
</tr>
<tr>
<td>49.</td>
<td>दो गोल से बढ़त बनाने के बावजूद चीन के साथ 2-2 से हाराने के बाद भारत मंगलवार के बाद यात्रा कप हाई में सेमीफाइनल की दीड़ से बाहर हो गया जिससे खिताब बनाने का उम्मीद बी चूर चूर हो गया।</td>
<td>देखो तू हुये बड़ी टुटुटे दे बचने चीत दे रुख 2 - 2 ये खड़े ये वघाट दे घरभांद जरूर भत्ती रुख देखी डिटी सैड़ी दी टैंडर में घरभांद दे जिम्मा फिस्ट पल भिख बचने की वैकट डेंव जामर मे डुंगर है जिससे खखता फयकयाय यखने का उसका ख्वाफ बी चूर चूर हो गया।</td>
</tr>
<tr>
<td>50.</td>
<td>सर्वजनिक क्षेत्र के आईडीबीआई बैंक ने भी एफडी पर व्याज दरें आधा से एक फीसदी तक घटा दी है। नई दरें 21 मई से लागू होंगी।</td>
<td>महेंद्रनाथ भेटव दे आईडीबीआई बैंक दे दो देढ़ हंटें दिवसी तरंग भांग में डिंग दीमातों उंच धर्मों डिंडीर गढ़। ठहरी रंगं 21 मई दे लागू देनेगी</td>
</tr>
</tbody>
</table>

### Accuracy Test - Literature

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hindi Sentence</th>
<th>Punjabi Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>अगर तुझे वह चीज न मिले तो खरददार इधर रुख न करना, त्वह सूली पर खिचवा दूंगी।</td>
<td>नेक उठू घुं चीज चीन लुई बिहे उं भचहक स रे पु तं दिनगी कई चित्र, चित्र मुझी भल विचर दूंगी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>इसकी सूचना ने अजन बलिका को मुंह ढप कर एक कोने में बिठा रखा है।</td>
<td>हिममी मुंघल रे भविभूय कहिना है भट दंग वर दिँव बैठे दंघ बिठा वकिना है।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>संध्मा का समय था, निममला छत पर जानकर अकेरी बैठी आकाश की और तृषित नेत्रों से ताक रही थी।</td>
<td>समझ दे मभी मी, सिंभभी हंड बह नहरे दिवंसी बैठी अभभ बीडी देव जूहोड़ रेंज़ धे ठेख जूआ मी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>निममला- ने उदासीन भाव से कहां-तू जा, मैं न जाऊंगी।</td>
<td>हिवभभ - के हंदमीहै बढ हूँ विच - ूँ ना, मैं तुझी नाप्नी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>बाग मे पूल खिले हुए थे। मीही-मीही सुगन्ध आ रही थी। चैल की शीतल मण्ड समीर चल रही थी।</td>
<td>बाग देंछ हुड़ धिे देते मल । बिनी - बिनी मूंह बह ऋ वजी मी। चेह की दी जीउँग भें मी ही चेह वजी मी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>यह कहकर कल्याणी कमरे के बाहर निकल गई।</td>
<td>हिव बजते वजस्वीहै वाबजे दे घर भिचल गई।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>मुंशीजी ने भीजन करने गये और निममला द्वारा की चौखट पर खड़ी सोच रही थी। - भगवान।</td>
<td>भूमीजी हं बैनेट बब नाप्ने न्दे नहेन देख दी चेतु भल भधी मेभ नदी मी - उजाहर।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>साध- कभी आ जाऊंगा बच्चा, तुम्हारा घर कहां है?</td>
<td>मापु - बंदे बा नाप्ना बंछ , उजांग बह बिंघे है ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
<table>
<thead>
<tr>
<th>No.</th>
<th>Question</th>
<th>Hindi</th>
<th>Punjabi</th>
</tr>
</thead>
<tbody>
<tr>
<td>9.</td>
<td>एक दिन निम्नला ने सियास को घी लाने के लिए बाजार भेजा।</td>
<td>दिन दिन निम्नला ने मिसालब टू पी मिसालब सही घटना देखि।</td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>माता-झूठ ने बोल। तूने पाँच सौ रुपये के नोट नहीं भेजे थे?</td>
<td>माता - झूठ ने बोल। झूठ देना तो ध्यान में चुभे दे रेट रही देने मत।</td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>क्या मेरी दशा को और भी दारू बनाना चाहते हो?</td>
<td>क्या भेंजी ढाँढा हूँ देख हो चाँढा ढाँढा चाँढे ने?</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>उस दिन से निम्नला का घंटा- दंग बदलने लगा।</td>
<td>झ्म दिन देने निम्नला रं देगा - देगा घटना रहा।</td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>क्या इन्हें सच्ची कोई भीषण रोग हो रहा है?</td>
<td>क्या हिडना हूँ मचाम हेटी जीतम देगा देना मिरा है?</td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>निम्नला- तो में झूठ कहती हूँ?</td>
<td>निम्नला - ऊँ में झूठ बिंदी वं?</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>बेचारे नड़के को बार-बार दौड़ाया करती है। सौंतेली मां है न! अपनी मां हो तो कुछ खायाल भी करे।</td>
<td>बेचारे भेरे दुँ देना - देन टेंज़लिसा बजी। माँ उँची भी दे रही। आपकी भी दे उँ बेड़ दिख़ा देश हो चेत।</td>
<td></td>
</tr>
<tr>
<td>16.</td>
<td>दाननाथ को ऐसी उत्साह स्पष्ट को न सुनने का अत्यंत शोक हुआ। बोले-यार, मैं जम का अभागा हूँ। क्या अब फिर कोई व्याख्यान न होगा?</td>
<td>दाननाथ हु मैंनी हेन भुजे महीने हु भुजी मुड़क डा महिमें में तेंज़लिसा। बेतरे-जाव, मैं ताम दे भेंखा वं। क्या चुट्टिया बेदी दिखिकाद रजी देशे?</td>
<td></td>
</tr>
<tr>
<td>संख्या</td>
<td>वाक्य</td>
<td>हिन्दी अनुवाद</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>17.</td>
<td>दाननाथ तो यह बातचीत करके अपने मकान को रखना हुए और अमृतराय उसी ऑर्थरे में, बड़ी देर तक युपवाप खड़े रहे।</td>
<td>दाननाथ तो यह बातचीत वलढ़े आपड़े मकान को रखना हुए और अमृतराय उसी ऑर्थरे में, बड़ी देर तक युपवाप खड़े रहे।</td>
<td></td>
</tr>
<tr>
<td>18.</td>
<td>आज भी, जब अमृतराय ने उससे अपने इरादे जाहिर किये तब उसने सच्चे दिल से उनको समझाकर ऊँच नीच सुझाया।</td>
<td>आज भी, जब अमृतराय ने उससे अपने इरादे जाहिर किये तब उसने सच्चे दिल से उनको समझाकर ऊँच नीच सुझाया।</td>
<td></td>
</tr>
<tr>
<td>19.</td>
<td>लाला बदरिप्रसाद अमृतराय के बाप के दोस्तों में थे और अगर उससे अधिक प्रतिष्ठा न थे तो बहुत हेठे भी न थे। दोनों में लड़के-लड़की के ब्याह की बातचीत पकड़ी हो गयी थी।</td>
<td>लाला बदरिप्रसाद अमृतराय के बाप के दोस्तों में थे और अगर उससे अधिक प्रतिष्ठा न थे तो बहुत हेठे भी न थे। दोनों में लड़के-लड़की के ब्याह की बातचीत पकड़ी हो गयी थी।</td>
<td></td>
</tr>
<tr>
<td>20.</td>
<td>पूर्णा ने आते ही सब स्पर्शों को वहाँ से छोड़ दिया, प्रेमा को इतने सुगन्ध के सबड़े और गुलाब का छीटा मुख पर मारा। धीरे धीरे उसके तलवे सहलाये, सब खिड़कियाँ खुलवा दी।</td>
<td>पूर्णा ने आते ही सब स्पर्शों को वहाँ से हटा दिया, प्रेमा को इतने सुगन्ध के सबड़े और गुलाब का छीटा मुख पर मारा। धीरे धीरे उसके तलवे सहलाये, सब खिड़कियाँ खुलवा दी।</td>
<td></td>
</tr>
<tr>
<td>नं.</td>
<td>हिन्दी अनुवाद</td>
<td>पंजाबी अनुवाद</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>----------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>21.</td>
<td>‘मिस्टर शर्मा—(मूँझों पर हाथ फेरकर) वह ताजा खबर लाया हूँ कि आप लोग सुनकर फड़क जायः।'</td>
<td>‘भिमाता सहम— (भूड़े भव चंब देववे ) हूँ उसन कन्हव सिकारिश नं विन हुमी ढेव महवे दउयचैट नारिइयो।’</td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>‘खुलासा यह कि अमृतराय को यहॉ से सतत हजार सुप्ता मिला। मुँही बदरीप्रसाद ने अकेले बारह हजार दिया जो उनकी उम्मेद से बहुत ज्यादा था।’</td>
<td>‘वहुमत धिह विन हुमौठवे हूँ धिहे के माउठ वालव दुष्प्ल भिविभां। भुतुमी घटतीपूजां दे डिबके खङज़ उनब धिँदे ने हूँ दो हुँमें डे बङइउ लिहाँट मी।’</td>
<td></td>
</tr>
<tr>
<td>23.</td>
<td>‘राम—(मुस्कराकर) चुप। ऐसा भी कोई कहता है।’</td>
<td>‘मभ— (भभबचवे) चुप। भभिल हो वेशी वलीये है।’</td>
<td></td>
</tr>
<tr>
<td>24.</td>
<td>‘अपने दिल का परिचय उसको एक दिन दो मिलाकि बाबू अमृतराय निमंत समय पर नहीं आये। थोड़ी देर तक तो वह उनकी राह देखती रही मगर जब वह अब भी न आये तब तो उसका दिल कुछ मसोसने लगा। बड़ी व्याकुलता से दोही हुई दीवाजे पर आयी और आध घंटे तक कान लगाये बड़ी रही, फिर भी तर आयी और मन मारकर बैठ गयी।’</td>
<td>‘आपते ढिल द नाह पापिचर हूँढ़ धिल ढिल धिंद भिविभा विन बचछ भिमू दीह निहान भमां वत जही अभी। बेडी ढेव उंच उं ढुँढ ढुँढ ही डेवम देबकी जवी भजव नते ढुँढ ढुँढ ही जवी अभी उंच उं ढुँढ हिल वँच भमानते डरा। ढेवी धिढीहोूरूगँ डे ढेवी ढेवी ढीलने भव भजी भवे भोय ढंगे उंच वंच सतापे ढजी जवी , ढिल भंदब भजी भवे भत भजवे बैठ गयी।’</td>
<td></td>
</tr>
<tr>
<td>No.</td>
<td>Hindi Text</td>
<td>Punjabi Text</td>
<td></td>
</tr>
<tr>
<td>-----</td>
<td>------------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>25.</td>
<td>अभित्यम—(दबी जबान से) वह सब कहार मेरे नौकर हैं।</td>
<td>अभित्यम— (दबी तंगर दें) ठिक मजब वजह भेदे ठेवन दें।</td>
<td></td>
</tr>
<tr>
<td>26.</td>
<td>अभित्यम—देखे अब कब भाग्य जागता है। मैं तो बहुत जल्दी मधा रहा हूँ।</td>
<td>अभित्यम—देखे वृट बरें विभिन्न महान तैयार है। मैं उंग घरूं तरलतिं भूल दिख जां।</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>मैं तुमसे कई अनुचित बात नहीं चाहता।</td>
<td>मैं उठते देव वेशी भंट-वैछित बौल तरीं चचुंट।</td>
<td></td>
</tr>
<tr>
<td>28.</td>
<td>उनके जरा से इशारे पर मैं अपने को निष्कार कर सकती हूँ।</td>
<td>दूल्हे दे तन देव दिमारे काल मैं अपने देव दिलचस्स वज मजदूर दें।</td>
<td></td>
</tr>
<tr>
<td>29.</td>
<td>बिल्लो—दे क्यों नहीं गया।</td>
<td>बिंधुं—दे बिंघु तरीं विभित।</td>
<td></td>
</tr>
<tr>
<td>30.</td>
<td>कुछ दिनों से झड़ीढ़ औरचाँद आदि ने भी पूर्ण के बनाव-चुनाव पर नाक-भों चढ़ाना छोड़ दिया था।</td>
<td>बेंग दिल्ली ने भंडाफर अवदेशवापित भाषि ते दी रमर्दी दे वजना क्षेर- चेट पठ ठंब - डब्बेंट चुर्टर दूल्ह दिल्ला में।</td>
<td></td>
</tr>
<tr>
<td>31.</td>
<td>उसने आते ही हकम दिया कि भीड़ हटा दी जाय।</td>
<td>धूतें भधटे दी तुनम डिल्ला वि जीजू ङटा डिल्ली नहँ।</td>
<td></td>
</tr>
<tr>
<td>32.</td>
<td>शादी के चारे दिन बाद पूर्णा बैठी हुई थी कि एक औरत ने आकर उसके एक बंद लिफाफा दिया।</td>
<td>दिवाल से छेंघे दिल्ली घबरे दुमनी बैठी भेदी मी वि दिल्ली भेदें ते आवे ठूले दिल बंद रितरा डिल्ल।</td>
<td></td>
</tr>
</tbody>
</table>
| 33. | चम्भन—चौधरी कह गये हैं कि आज इनके काम न छोड़ देहाँ तो टाट बाहर कर दीन जैनी। | चिंतित पैबली—बड़ी साथ दल बिनाम पित वेद वेदि डेव डेव दुः टॉट घाँठ वव घीरु मैरी।
| 34. | एक दिन कुजरानी सुवामा के सिरहाने कैथी पंखा झल रही थी। | पिंठ पैठ दुनवाली मुड़बा दे मियकटर बैती पैका घल उद्धी मी।
| 35. | प्रताप-तो भई, एक दिन मुझे भी नेवता दो। | भूज़ - उं बरी, पिंठ पैठ में दी ठेकट जे।
| 36. | नवीन मिठी की मीठी-मीठी सुगन्ध आ रही है। | रचिर मिठी की मिठी - मिठी मुखप आ जो त मी।
| 37. | सेवती-में तो बिन गीत सुने आज तुम्हारा पीछा न छोइंगी। | पिंठा सुषाध - में उं हिला जीजू सूदे 'भौ डाँड' पिंठा रुँगी ब्रह्मजी।
| 38. | राजा ने कहा “अच्छी बात है!” | जुपा ते पिग “चौथी बाँठ दे।”
| 39. | तीन दिन बीतने पर बुझिया फिर वहाँ पहुँची। | पिंठ पैठ वुस्तकर भल झूंठी पिंठ पैठ पहुँची।
| 40. | सी रोने लगी। एक मुसाफिर उधर जा रहा था। | श्रमजी वेक संजी। पिंठ भ्रमचर झुँपव ना जिंग मी।
| 41. | सेठ अपने जमाई से मिलकर बड़े प्रसन्न हुए और उन्होंने उसे बड़ी अच्छी तरह से घर में रखा। | मेंठ अपघ तलमक ते भिंचवे झूंठे पुथ वेदे भजे झूंठगे झूंठे झूंठे झूंठे चंजी उर्लुं उं पुत पिंठ वैडिया।
<table>
<thead>
<tr>
<th>No.</th>
<th>Sentence in Hindi</th>
<th>Sentence in Punjabi</th>
</tr>
</thead>
<tbody>
<tr>
<td>42.</td>
<td>लगता है यह दरजी लोभी है।</td>
<td>रगता ह  मह दयजी रोबी ह ।</td>
</tr>
<tr>
<td></td>
<td>यह हमको खिलाना नहीं चाहता, इसलिए यह सारा नाटक कर रहा है।</td>
<td>मह हभको खखराना नहीं चाहता , हिमसाती रिह मच' बछभा वच दिह है ।</td>
</tr>
<tr>
<td>43.</td>
<td>राजा मुझसे दर गया।</td>
<td>राजा भेजे में दर दिखा।</td>
</tr>
<tr>
<td>44.</td>
<td>पारदती को दया आ गई, और उन्होंने शंकर से विनती की कि जैसे भी बने, वे गिलहरी को फिर से खी बना दें।</td>
<td>पखवजी हू मच' उदम भा गारी , भउं डूँ आं स्वंब है भूववचर ची वि निए दी घट है , छी किलहरी हूं दिह है विमसाती घट दिए ।</td>
</tr>
<tr>
<td>45.</td>
<td>स्स में एक बहुत बड़े लेखक हुए हैं, इतने बड़े दो सारी दुनिया उन्हे जानती है।</td>
<td>सुम सिंह रिंग पहुंड देंदे लेखक देंदे हैं , दिह है दिखे वि माजी लुडींग पहुंड हूड़ हूड़ नलहदी है ।</td>
</tr>
<tr>
<td>46.</td>
<td>सचीन्द्र ठाकुर की एक बड़ी ही सीख देने वाली रचना है।</td>
<td>उन्हीं ठाकुर छी डिंग देंदी छी मोह रेंक बही जनर है ।</td>
</tr>
<tr>
<td>47.</td>
<td>देवदत चला गया और अगले दिन जब वह लौटा तो उसके हाथ में उन आदमियों की सूची थी।</td>
<td>देवदत लछ बिखा 'भउं भवले दिल नटे छी पवित्रं उं डूमटे रं डिंग दिहुं बंधौं दी मूझी मी</td>
</tr>
<tr>
<td>48.</td>
<td>आदमी को भूमि से कितना मोह होता है।</td>
<td>अचहरी हूं छुभी दे बिंटा भेज भूं है ।</td>
</tr>
<tr>
<td>49.</td>
<td>कहने का मतलब यह कि हर आदमी अपनी क्षमता के अनुसार काम करे और जरूरत के अनुसार पाये।</td>
<td>वरिष्ठ दा भजुलख्च दिल वि उज भजही भप्तही मरहभंग दे अभमान बंभ बड़े देव नलकुंड दे अभमान बढ़े ।</td>
</tr>
<tr>
<td>हम आशा करते हैं कि पाठक</td>
<td>भाषी भाषा विद्वान जी विभाग\nधिम विवादों तू हैं कि\nताल भवानी, तुम्हें मे रूस \भवानी अनेक रुपों चालु करा वी कुछ</td>
<td></td>
</tr>
</tbody>
</table>
### Accuracy Test - Articles

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hindi Sentence</th>
<th>Punjabi Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>एनीमिया (Anemia) के कारण महिलाओं में थकान, ऊजन बढ़ने और खेड़ होने में चक्र आना, काम करने का मन न करना, शरीर में तापमान की कमी, तत्त्वा में पीलापन, दिल में असामान्य धंडक, सांस लेने में तकलीफ, सीने में दर्द, तलवों व हथेलियों में ठंडापन और लगातार रहने वाला सिर में दर्द होता है।</td>
<td>तैलिमिया (Anemia) दे बौछर भोजन हिंच भवर, हिन्दी घटौं देख भोज देख हिंच चबर हाड़, बौह बौध दे मन तोची बौध, महीन हिंच उपद्रव दी बारी, उदच हिंच फिस्ट्यूल, हिंच हिंच गौड़ - भाषाली पदवर, माम भैंड हिंच उबलीद, मीई हिंच चदर, उदर ह भैं उपेशीय हिंच घड़खड़ देह संगाद विद्य हइस भैं हिंच चदर झूंघतूं दै।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>गर्भवत्स्था (pregnancy) के दौरान देट में तीव्र दर्द और योगी से रक साव होने लगे तो इसे हंगारता से सै तथा डॉक्टर को तथाकाल बताएं।</td>
<td>गर्भवान्मया (pregnancy) दे डॉक्टर हिंच हिंच जीवत दर्द भाद्र डॉक्टर व लोग दे बौध भनौं देख हेंस उंडिम्हूं बांडीबड़ दे बौध भनौं हूं ऊंचवर चेंज़ी।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)

10 : 10 October 2010

Vishal Goyal, Ph.D.

*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
3. सामग्री: 150 ग्राम अर्द्ध दाल, 20 ग्राम सूंगफली, 100 ग्राम गुड़, 100 मि. ली. तेल, 15 ग्राम दालचीनी, 3 ग्राम हांग, 5 ग्राम ईमली, 5 ग्राम अदरक, 5 ग्राम नमक, 3 ग्राम हल्दी, 5 ग्राम हरीमिर्च, 5 ग्राम, 3 ग्राम करीमचीना, 3 ग्राम नारियल (कसा हुआ), धनिया बारीक कटी हुई।

मभावी: 150 गाज़म भजवन चरस, 20 गाज़म भवली, 100 गाज़म खुट, 100 भ. रूढ़ि, 15 गाज़म दालचीनी, 3 गाज़म हलदी, 5 गाज़म हल्दी, 5 गाज़म छीमीले, 5 गाज़म बछु, 5 गाज़म लख, 3 गाज़म ब्लांड़, 5 गाज़म जी, 5 गाज़म, 3 गाज़म बछु, 3 गाज़म हलदी (बीमी भेडिया), परिभाषा घजीव बटी तुरी।

4. आइसक्रीम के दाग - अगर आइसक्रीम के दाग कपड़ों में लग जाए तो अमोनिया का घोल डालो।

आइसक्रीम बे दाग : नेवर भागिताबिब बे दाग बधट लिंग संग नाँद उन भागिताबिब बे बेर भाग।

5. चावल की खीर बनाते समय शकर के साथ थोड़ा सा नमक मिलाकर से खीर का स्वाद और बढ़ा जाता है।

चावल दी खीर बहरपूर्वे मभाव मज़व दे तस बेढ़ निंबु लहर प्रानु चे तो भूल दे भजवन चे खज तमप भज तमप दै।

6. सोने दे जेवर पर पिसी हल्दी लगा कर मसलने से वे चमकने लगते हैं।

मेद दे नेवर दे मिसी हल्दी भजवा बे भमलते दे छुए चभवन छजवाते है।

7. कभी भी इसका स्पिनर खाली न चलने दें।

चते दी धिमाह मभीतव भजी हतो छत्त टिक।
8. आजकल की महिलाएं नौकरीपेशा वाली हैं इसलिए ज्यादा समय घर से बाहर बिताती हैं संयुक्त परिवारों में या जिन के माता पिता घर पर रहते हैं उन्हें बच्चों की देखभाल की समस्या नहीं होती है परंतु एकल परिवारों में मां के दफ्तर जाने के बाद बच्चों की देखभाल के लिए कोई नहीं रहता इसलिए महिलाएं अपने बच्चों के लिए आया का इंतजाम करती हैं।

9. अपने जीवन उदेश्य को जानना और उसे प्राप्त करने के लिए ठीक आत्मविश्वास रखना, यही सफलता की ओर पहला कदम है।

10. भक्त का भगवान से, मानव का ईश्वर से, व्यक्ति का समस्या से, पिण्ड का ब्रह्मण से मिलन को ही योग कहा गया है बरवा दे बरवा दें, महीने दे चंचल दें, नदेहर दे मारे दें, धिंदी दे घरकुंभ दे भिलाई दें ही लेता विल विल विल विल विल।
11. मिजोरम के एक 64 वर्षीय व्यक्ति जियां की 50 पत्तियां और 100 बच्चे हैं। मिजोरम से लगभग 80 किमी दूर बर्तावांग गांव का निवासी जियां अपने परिवार के 180 से अधिक सदस्यों के साथ पृथ्वी पर सबसे बड़े परिवार के मुखिया के रूप में जाना जाता है।

12. एक छोटा बच्चा दुसरे बच्चे से, अगर दिन की सूर्य न मिलता है तो क्या करेगा?

दुसरे बच्चे ने जवाब दिया, "विजय का बिन बढ़ जाएगा।"

13. बीरबल को तम्बाकू खाने की आदत थी लेकिन अकबर (Akbar) न खाते थे। एक दिन अकबर ने तम्बाकू के खेत में गधे को घास खाते देखकर कहा, बीरबल ये देखा तम्बाकू कैसी कुरी चीज है, गधे तक इस को नहीं खाते। धीरजिय हूँ उंचाई भाँति की भांडू में से लेबिल भव्य (Akbar) ने भी मर्मत धिंधे बियार भव्यत हूँ उंचाई छोटे भुजे भाँतिये देखकर धीरजिय हूँ किया उंचाई बियार इसी गठी चीन है, गधे उंच हिंदू भाँति भटे।
| 14. | ये सब कहने की बातें हैं कि उन को छोड़ बैठे हैं। जब आंखें चार होती हैं मौहकत माँ ही जाती है।। | उन्हें मव बबिट चींच बूथ कर बिछु तुंठे बैठे बैठे बैठे।। साथें बेब्हें चाव ठोलों कर भेववथ भा जी नहीं है।। |
| 15. | नए आधुनिक डिज़ाइन के अन्तरगत तेजी से बाज़ार में आ रहे हैं। इस के लिए ज़रूरी है कि आप भी अपने नीतियों में बदलाव लाएं और यहीं न करते रहें, "हम तो इस काम को इसी तरीके से करते आ रहे हैं और ऐसा ही करेंगे।" | उन्हें अहें भागवत दिनिन्धिन टे अउरफ देती हें भज्जन हिंच भा उठे है।। धिम सली लहल है विभ दी आपटे ठीज़ दिविंग फरेंग भड़ह हिंच रद्दी बढ़ी बढ़ी वड़े , 'भी म वह धिम बेब्ह तुंठे धिम उबोबे हें बढ़ भा उठे है भड़ भाजव जी बढ़ें है।। |
# Accuracy Test – Official Language Quotes

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hindi Sentence</th>
<th>Punjabi Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>संक्षेप नोट नीचे दिया गया है</td>
<td>मध्यप्रदेश रैट गेथे डिंडू सिल्बा दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>प्रचलित नियमों के अनुसार</td>
<td>पूरछेडा हिस्सियां दे मध्यप्रदेश</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>पावती पहले ही बेजी जा चुकी है</td>
<td>उमीद पहिली ले खोजी ता चुंबी दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>ऊपर के अनुसार कार्रवाई की जाए</td>
<td>हुईं वज्रे महत्व वांछनी बीजी सड़े</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>मामले में कार्रवाई की जा चुकी है</td>
<td>भभरे हिंद वांछनी बीजी ता चुंबी दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>पत्र की एक प्रतिलिपि</td>
<td>भेंडि छा हिंद वनब</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>पहले से प्रबंध करना जरूरी है</td>
<td>पवित्र दे पृथ्वी वनब न्यूविय है</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>आगे की प्रगति से अवगत कराएं</td>
<td>भोंदे छा उदवल दे न्यूत वनबिय</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>कार्य-सूची साथ बेजी जा रही है</td>
<td>वरन - मृगी मथू बेजी ता बेजी दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10.</td>
<td>अपील खारेज कर दी गई है</td>
<td>भभील भविन वन डिंडी गाडी दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11.</td>
<td>जहां तक संभव हो</td>
<td>भिंबे उंच महद दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>बिल सही-सही बनाया गया है</td>
<td>बिल ठीच - ठीच घटकशिक्षा विश्व दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13.</td>
<td>बजट में व्यवस्था है</td>
<td>बजट हिंद विदमभ दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14.</td>
<td>स्वीकार नहीं किया जा सकता</td>
<td>महीना रौं बीजा ता मबट</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>मामले की जांच चल रही है</td>
<td>भभरे छा नंच चेल उदवी दे</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
Vishal Goyal, Ph.D.

Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation
### Accuracy Test - Blogs

<table>
<thead>
<tr>
<th>S.No.</th>
<th>Hindi Sentence</th>
<th>Punjabi Sentence</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>मैं ऐसा इस्लिए लिख रहा हूँ कि मेरा एक दोस्त जो मेरा ब्लाग पढ़ता है उससे मुझे कहा कि यह नारद का एकाधिकार खत्म होगा!</td>
<td>में अभिनेत निमथली लिख विर उं वि भेज हिंच हेमउ ते भेज धल्ला भट्टा दे दिवमत भेजू विर वि निश लट्ट से थेबपिण्‍च धउभ वेदेहा</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.</td>
<td>राजेश, हिंदी चिह्नों के ज्यादातर पाठक चिह्न लिखने वाले ही हैं, आपके मित्र की श्रेणी के पाठक फिलहाल कम हैं। अभी तक हिंदी के गैर चिह्नेका पाठक आपका अखबार ही पढ़ते हैं। वैसे पाठक कोई भी हों, कितने ही क्यों न हों किसी बहस अन्जाम से पहले छोड़ना अन्य माध्यम भी नहीं चाहते।</td>
<td>वर्णसेन , किंची हिंदीमं वे निमथपाल निरन विने किवाल रटे दी गइं , दउजें भिंउद दी मूंटी दे निरन दिलिशस भंट गइं। उदं किंची दे गैव हिंदी० निरन निन निरन निरन गइं। दैं वैं निरन वेंटी ही रेड , किंची दी किंची रुटी रेड लंगम विन्हिम भालम दे पविल्ज हंडर देत भावभ से गइंजी छंपर हॉट</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.</td>
<td>कुछ लोगों को मजा आता उसी विषय को बार-बार फँटने में तो आप क्या कर सकते हैं! अभिव्यक्ति की स्वतंत्रता है।</td>
<td>वंजेह सेवज हूँ भना अभिव्यक्ति हूँ हिंच हूँ दुर - दुर ढंटें हिंच उं दुभी बी बूं मंचें गइं! भवमग़त दी भनती है।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>संख्या</td>
<td>हिंदी में</td>
<td>संक्षिप्त अर्थ</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>----------</td>
<td>----------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.</td>
<td>जहाँ तक मैं समझता हूँ, हिंदी के चिंते अभी बहुत ही सीमित विषयों पर लिखते जा रहे हैं। ऐसे विषय, जिनमें अधिकांश नेट प्रयोगकारों की कोई दिलचस्पी नहीं है। जब तक यह हाल रहेगा, शायद ही हिंदी चिंतन का पाठकवर्ग विकसित हो सके।</td>
<td>सिंखे उंच में मूंगुटिया गं, छिंटी ते छिंटे गुटे घबर जी मौजित भन्ने भन छिंथे ता चढ़े जा। 'भनिंधे रिमा', सिंखा छिंट मार्ग रेट पूछेबुझदाँटी बेठी छिंधलसभी तर्की है। तत्ते उंच छिंट ग़ल बदेबाटा, माहिति दी छिंटी छिंटीभां दा भाटवदाँटा रिविलम दे मावे।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.</td>
<td>यह तो सच है, हिंदी की कई पोस्टों को बिना बैंकएवाँद जाने कोई समझ नहीं सकता।</td>
<td>छिंट उंग में छिंट है, छिंटी दी बसी पेमटं हु छिंटं बैंक़बुझदाँटी नटे बेठी मौंगुट में।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.</td>
<td>आप द्वारा उठाया गया प्रश्न बहुत महत्व रखता है।</td>
<td>अप्ड दुभाग चूनिप्फा विभाग धृतत हलेउट भंजर लेखन है।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.</td>
<td>बहुत सही लिखा है आपने, साधुवाद!!</td>
<td>घबरू ठीक लिखित है उमी, मपुरुङ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.</td>
<td>अब मैं आपसे एक नवेदन करना चाहूँगा।</td>
<td>घबरू में घबरू हिंट घेरूँगी बन्न सतवदाँ।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9.</td>
<td>अगर हिंदी तथा अन्य भारतीय भाषाओं के लिये एक बड़ी टेक्स्ट एनालिसिस का औजार(साफटवेयर) बना सकें तो भारतीय भाषाओं का बहुत भला हो।</td>
<td>नेवेत हिंटी भाँ चेत बाउटीरामं ठीक द्वारिग टेक्स्ट एनालिसिस दा भोजन ( मांडटेंटिव) घट मववट उंग बचैडी इमारं दा घबरू जला थे।</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Language in India [www.languageinindia.com](http://www.languageinindia.com)
10 : 10 October 2010
Vishal Goyal, Ph.D.
*Development of a Hindi to Punjabi Machine Translation System - A Doctoral Dissertation*
| 10. | अपील खारिज कर दी गई है | अपील भट्टिन बत दिंदी वाही दे |
| 11. | जहां तक संभव हो | निंचे उंच मंडद दे |
| 12. | बिल सही-सही बनाया गया है | बिल ठीच - ठीच घटाई में विभा दे |
| 13. | कृपया इस बारे में गम्भीरता से विचार करें। | क्रूपा डिम बरे डिंच बंजीउं दे डिंच बरे। |
| 14. | आप जैसे लोगों का हमारी भाषा के प्रति प्यार हि सबको उत्साहित रखता है। | डवी निंदे हे हे डवी डवी डवी हे पूढी व्हिम्वर वि मविभा डु डिंडाच्हिंड लेखन दे। |
| 15. | बहुत सुंदर प्रयास के लिए बधाई। एक ब्लॉग पोस्ट इस पर कल लिखता हूँ। | बहुत मूंटच वेम्बिम रूढी डप्यो। डिंच घरणा थेमट डिम खट बेल्ड हिम्बट टडं। |