Translator of Hindi Text to ISL and extension of ISL dictionary with WordNet

Thesis Report

submitted in partial fulfillment of the requirements

for the award of degree of

Master of Engineering

in

Computer Science and Engineering

Submitted By

Paras Vij
(801432017)

Under the supervision of:

Dr. Parteek Bhatia
Associate Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

June 2016

Certificate

I hereby certify that the work which is being presented in the thesis entitled, ***"Translator of Hindi Text to ISL and extension of ISL dictionary with WordNet"***, in partial fulfillment of the requirements for the award of degree of Master of Engineering in **Computer Science and Engineering** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Parteek Bhatia** and refers other researcher's work which are duly listed in the reference section.

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

Paras Vij
801432017
ME (CSE)

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

Dr. Parteek Bhatia
Associate Professor
Computer Science and Engineering Department
Thapar University

Countersigned by

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

(Dr. S. S. Bhatia)
Dean (Academic Affairs)
Thapar University
Patiala

i

Acknowledgement

I would like to express my gratitude to my guide Dr. Parteek Bhatia for being an amazing mentor for me during the course of thesis. Your concern and valuable advice during the entire period proved to be really helpful in completing my thesis work. Sir your patience and timely guidance has helped me during every phase of my work. You always encouraged me to work on new ideas and showed your trust upon me.

I am also thankful to Dr. Ashutosh Mishra, P.G. Coordinator, for the motivation and inspiration that triggered me for the thesis work. He has always been supportive and provided us required information on regular intervals.

I would like to acknowledge Dr. Maninder Singh for setting good standards for his students and providing all the help and facilities that were essential throughout the journey. Your encouragement time and again has helped us to achieve the set goals as well. I would like to thank Kanu Goel for supporting me in my two years of journey in Thapar University.

Above all thanks to the almighty, my family and friends for always being there for me and staying calm at times required.

Paras Vij

Abstract

Sign Language is the basic source of communication for deaf people across the world. Sign Language is used by the deaf people to communicate with each other and also by their families and friends to communicate with them. Sign Language varies from place to place. America has American Sign Language, Britain has British Sign Language and India has Indian Sign Language. Sign Languages use arms, hands, body and facial expressions for communication. Each word has a different expression. It is a major linguistic challenge that the researchers have taken to build system which could facilitate the understanding of natural language. According to World Federation of Dead there are approximately 70 million deaf and dumb people in the world and 4 million dead people in India according to All India Federation of Deaf. Due to lack of linguistically annotated and well documented data on Indian Sign Language research on Indian Sign Language Linguistics is limited.

In this thesis Hindi to Indian Sign Language with extension using Hindi WordNet system is developed. The system has been developed in Java which makes it accessible across different platforms. System takes Hindi Sentence as input and generates corresponding Indian Sign Language sentence. For converting Hindi to Indian Sign Language Hindi Dependency Parser is used which gives information about each word in sentence. After parsing each word is taken and Indian Sign Language grammar rules are applied. Words which do not pass grammar rules are removed from sentence. Words which passed grammar rules are then searched in database to find if we have signs for them. If a word has sign available it is copied into final Indian Sign Language sentence. If we do not have sign available for a word then we use WordNet to find a replacement for that word. WordNet returns synonyms for word. Each synonym is checked in our database to see if we have sign available for that synonym. If a synonym has sign available that word is replaced with synonym in final Indian Sign Language sentence. If no synonym is found which has sign available in our database then we mark word as Finger Spelled in our final sign language sentence. After processing each word final Indian Sign Language sentence is shown on the screen to user.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Sign Language

Sign Language is used as a source of communication for deaf and dumb people. Sign Language uses gestures to communicate rather than speech. Sign Language uses arms, hands, body, facial expressions for communication. It has different expression for each word. Sign Language is completely natural language and they have their own syntax and grammar. Sign language may vary from place to place. America has its own Sign Language American Sign Language (ASL), Britain has British Sign Language (BSL), and India has Indian Sign Language (ISL). Different sign languages have their own syntax and grammar. Thus a person having expertise in ASL will not be able to communicate in a better way with a person not having knowledge of ASL. According to World Federation of Deaf there are approximately 70 million deaf and dumb people in the world.

1.2 Indian Sign Language (ISL)

Indian Sign Language is a complete natural language and has its own syntax and grammar. Indian Sign Language is also used as a way of communication in Sri Lanka, Bangladesh, and Nepal and in some parts of Pakistan [12]. There are about 4 million deaf people in India according to a survey conducted by All India Federation of Deaf. Indian Sign Language is used by deaf community to communicate which includes deaf people families, their friends, school/ colleges for deaf people. Since, Indian Sign Language has its own syntax and grammar so each sentence needs to be converted from Hindi grammar to Indian sign language grammar. There is lack of linguistically annotated and well documented data on Indian Sign Language due to which research on Indian Sign Language linguistics and phonology is limited.

1.3 Need of Sign Language Systems

Sign Language is used by around 70 million people across the globe to communicate with

each other and their family, friends and with other people. In absence of interpreter it is very difficult for people who do not understand Sign Language to communicate with deaf and dumb people. Sign Language Translator provides a platform to ease the process of communication. Sign Language system has application in many domains such as railways, hospitals, educational institute and others.

1.4 Challenges in Processing of Sign Language

Sign Language being a complete natural language has its own syntax and grammar. Sign Language varies from region to region. For transformation from one language to another one needs to have detailed knowledge about both languages. There are not many experts of sign language. Machine translation of sign language is a tough task due to lack of linguistically annotated and well documented data on Indian Sign Language due to which research on Indian Sign Language linguistics and phonology is limited.

1.5 Sign Language Generation Process

Sign Language Generation Process is divided into two phases. Phase A is pre-processing phase in which Hindi Sentence is converted into ISL grammar. For the proposed system in Phase A Dependency Parser and WordNet are used in combination with ISL generator to translate Hindi sentence to ISL grammar. Phase B uses the input provided by Phase A to generate signs for the words in grammar. In Phase B we are using HamNoSys for displaying avatars for words. SIGML is used for representing HamNoSys in xml format. SIGML is played in SIGML player which shows sign using avatars. Figure 1.1 represents General Architecture of ISL generation process. The system has been divided into two phases Phase A is pre-processing and Phase B is Sign generation phase. In this thesis we have completed Phase A which converts Hindi Sentence to ISL grammar.

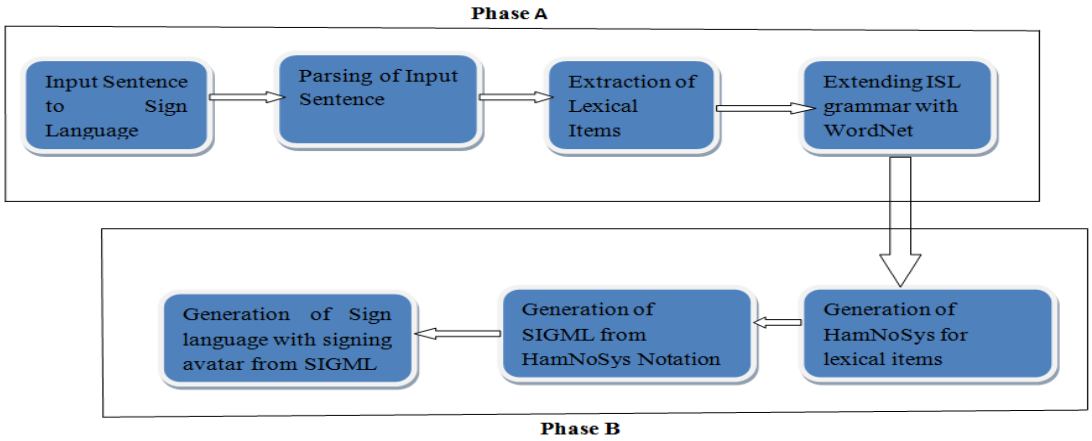

Figure 1.1 Sign Language Generation Architecture

1.6 Technologies Used in Sign Language

To generate sign language from input sense various components needs to be brought together to achieve the final output. As explained in section 1.5 Sign Language generation is a two phase process and each phase has multiple components to achieve the output. The various components used by us to generate Sign Language is given in below section.

1.6.1 Dependency Parser

Dependency Parser uses dependency graphs to represent words and their relationships to syntactic modifiers using directed edges. Dependency Parser analyzes grammatical structure of sentence establishing relationship between "head" words and words which modify those heads [14]. Hindi dependency parser by IIIT Hyderabad is used for dependency parsing. Expression (1.1) gives the information that is returned by dependency parser for each word [1].

<word id> <word> <lemma> <POS Tag><parent id> <dependency label> ...(1.1)

The description for the information tags given by (1.1) is stated in Table 1.1

Table 1.1 Description of tags given by dependency parser

Tag	Description
word id	It is the unique identification number given to a particular word.
word	The basic word in the expression is stated here.
lemma	Lemma refers to particular form that is chosen by convention to represent all the words which have same meaning.
POS Tag	The POS Tag gives the description about the Parts of Speech that the word belongs to.
parent id	Represent the identification id of word from which word has been derived.
dependency label	It represents the system generated label that determines the dependency of the word on POS tagging.

Dependency Parser has twenty one tags defined for tagging words. Different tags in dependency parser are given in figure 1.1.

SNo.	Category	Tag name
1.1	Noun	NN
1.2	NLoc	NST
2.	Proper Noun	NNP
3.1	Pronoun	PRP
3.2	Demonstrative	DEM
4	Verb-finite	VM
5	Verb Aux	VAUX
6	Adjective	JJ
7	Adverb	RB
8	Post position	PSP
9	Particles	RP
10	Conjuncts	CC
11	Question Words	WQ
12.1	Quantifiers	QF
12.2	Cardinal	QC
12.3	Ordinal	QO
12.4	Classifier	CL
13	Intensifier	INTF
14	Interjection	INJ
15	Negation	NEG
16	Sym	SYM

Figure 1.1 Parts Of Speech (POS) Tags

1.6.2 WordNet

Hindi WordNet is a system that is meant to collaborate various types of relations between the Hindi words ranging from semantic to lexical forms. It provides organized information giving a detailed overview of the meanings of the words on lexicon basis. Here, the words that have the similar meaning are grouped together which later can be interchanged in contexts demanding synonyms. There exists a synonym set, or synset, for each word that represents one lexical concept. With this ambiguity is also removed in case a word has more than one meaning. So basically 'Synsets' act as the basic building blocks of WordNet. An entry in the Hindi WordNet comprises of primarily three main components [19].

- **Synset**

A set of synonymous words representing the common concept. For example, 'बगीचा'(bagicha),'बाग'(bag), 'बगगया'(baggya), 'उद्यान'(udyaan), 'वाटिका' (vatika), represents the concept of park as a public area with many plants and trees. The words in synset are arranged in order of their frequency of usage.

- **Gloss**

It consists of two parts that describe the concept.
a) Text definition: The concept denoted by Synset is explained by text definition.
b) Example Sentence: Usage of word in sentence is given by the example sentence.

- **Position in Ontology**

A hierarchical organization of concepts, in specific a categorization of entities and actions depicts ontology. It gives a detailed syntactic categorization namely verb, noun, adverb, adjective from the required word. Hindi WordNet was developed by IIT Bombay for Hindi WordNet. It is a java based API. Presently the system has 28,687 and has 63,800 unique words [4].

Each synset in Hindi WordNet is linked with other synsets through well known relationships of hypernymy , hyponymy , meronymy, troponymy etc. Synsets have semantics relations between them and words have lexical relations between them.

There are 16 relations in Hindi WordNet. Relations are described below –

- **Hyponymy and Hypernymy (is a kind of)**

Super-set hood relation between two synsets is captured by Hypernymy. Sub-set hood relation between two synsets is captured by Hyponymy. Transitive and asymmetrical properties are followed by Hyponym relations. Hypernymy is reverse of Hyponymy.

Example –बेलपत्तीis a Hyponym and पत्ता(leaf) is a hypernym.

- **Meronymy and Holonymy (Part-whole relation)**

If two synsets S1 and S2 are related to each other in such a way that S1 is one of the constituents of S2, then S1 is meronym of S2 and S2 is holonym of S1. Transitive and asymmetrical properties are followed by Meronymy relations. Holonymy are reverse of Meronymy.

Example – पत्ते(patte; leaves) is the part of वर्षृ (vriksh ; tree), meaning that पत्ते(patte; leaves) is the meronym of वर्षृ (vriksh ; tree) and वर्षृ (vriksh ; tree) is the holonym of पत्ते (patte; leaves).

- **Entailment**

Relationship two verbs is represented by Entailment. Any verb V1 entails verb V2 if truth of V2 logically follows from truth of V1. Entailment relationship is unilateral i.e. it is a one way relationship.

Example: खराणिा लेना(kharraaTaa lenaa;snore) ==> सोना (sonaa; sleep)

- **Troponymy**

Troponymy is used to denote manner relationship between two verbs. If V1 is troponym

of V2 then V1 is to V2 in some way.

- **Antonymy**

If two words W1 and W2 express opposite meanings than antonymy relation exist between W1 and W2. It is a lexical relationship and holds between two words and not two synsets.

Example – *मोटा* (moTaa;fat) ==> *पतला,* दबुला (patlaa, dublaa; thin)

- **Gradation**

Gradation represents intermediate relationship between two opposite words. It is a lexical relationship. Figure 1.3 represents gradation relationship among three words.

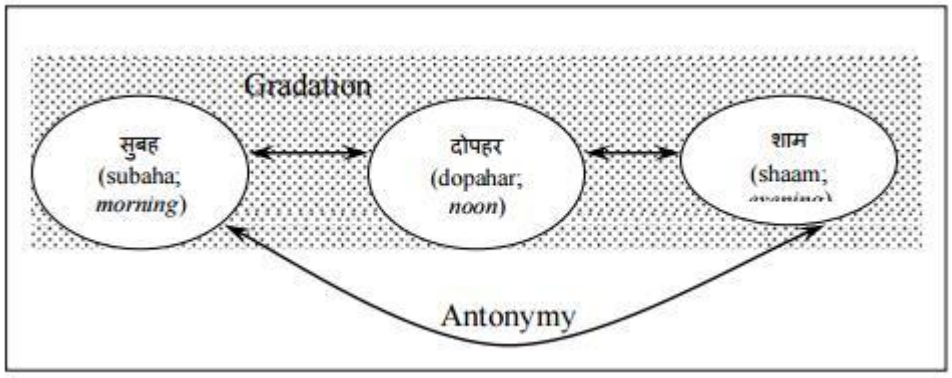

Figure 1.3 Gradation relationship among three words.

- **Causative**

By making morphological changes in base verb causation can be formed. Link between causative verb and base verb is represented by causative relations and it shows interdependency between base verb and causative verb.

Example रोना (rona; cry) ==> रुलाना (rulana; to make someone to cry) रुलाना (rulana; to make someone to cry) is a causative verb of रोना (rona; cry).

- **Ability Link**

Ability link is used to represent relationship between verbal and nominal. Ability link specifies the inherited feature of a nominal concept. Ability link is a semantic relation and specified the inherited feature of a nominal concept.

Example मछली,मछी, मतस्य , मीन, माटह(machlii, macchii, matsya, miin, maahii; fish) ==> तैरना, पैरना, पौंरना, पौरना, हेलना(tairnaa, pairnaa, pauMrnaa, paurnaa, helanaa; swim)

- **Capability Link**

Capability link is used to represent relationship between verbal and nominal. Capability link is a semantic relation and is used to specify the acquired feature of a nominal concept.

Example व्यक्तत,शख़्स,शख्स,जन (vyakti, maanas, sakhs, jan; person) ==> तैरना,पैरना,पौंरना,पौरना,हेलना(tairnaa, pairnaa, pauMrnaa, paurnaa, helanaa; swim)

- **Function Link**

Function Link is used to represent relationship between verbal and nominal. Function link specifies the function of a nominal concept. It is a semantic relation.

Example: अध्यापक,मशक्षक,आचायण,गु(adhyaapak, shikshak, aacaarya, guru; teacher) ==> पढाना,मशक्षा देना(paRhaanaa, shikshaa denaa; teach)

- **Attributes**

Attributes is used to represent a relationship between nominal and adjectives. It is used to denote properties of noun. Attributes is a semantic relations and is used to represent a relationship between nominal and adjectives.

Example: पक्षी,चचडडया,पंछी,खग,पररंदा,ववहंग,ववहंगग,पखेू,ववहग (pakshii, ciRiyaa, panchi, khag, parindaa, vihanga, vihangam, pakheru, vihaga; bird) ==> पंखदार,पंखयततु (pankhdaar, paankhdaar, pankhyukt; having wings)

- **Modifies Noun**

Modifies noun is used to represent a relationship between nominal and adjectives.Certain nouns can be modified by certain adjectives. To link such nouns and adjectives modifies noun relationship is used.

Example: सपात्र,सत्सपात्र,अच्छा पात्र (supaatra, satpaatra, acchaa paatra, eligible)

==>शख़्स,शख्स,जन,बंदा,बन्दा (vyakti, maanas, sakhs, jan; person)

- **Modifies Verb**

Modifies verb is used to represent a relationship between adverbs and verbs. Certain adverbs can only go with certain verbs. Modifies verb is used to represent such relation.

- **Derived From**

To specify the root form from which a word is derived the derived from relationship is used. This relation can go from noun to adjective or vice versa, noun to verb and adjective to verb. It is used to handle derivational morphology. It is a lexical relationship [19].

1.6.3 HamNoSys: A Notation System for Sign Language

HamNoSys is a Stokoe based notation system which is having 200 largely iconic characters These HamNoSys symbols are used to represent hand orientation, hand shape, and hand location corresponding to other parts of the body. For writing signs,these parameters should be assigned with some value. HamNoSys provides symmetryoperator for representing two handed signs non-manual components. Non-manual phonological can be represented by replacing hand graphemes with the symbols for bodypart such as head But facial movements like raised eyebrows or puffed out cheeksare complex to represent. The parameters of a sign are written in the order of symmetryoperator, non-

manual components, hand shape, hand position, hand location and hand movement as shown in figure 1.4.

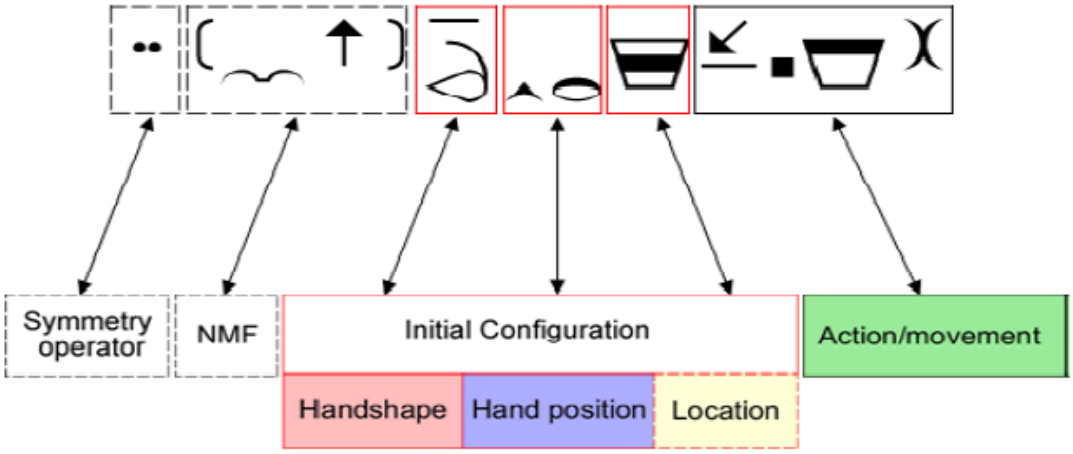

Figure 1.4 Structure of HamNoSys [22]

1.6.4 SIGML Language

SiGML is Signing Gesture Mark-up Language. It was developed for specifying signing sequences in ViSiCAST project. It defines HamNoSys symbols into XML tags form. It was developed at the University of East Anglia. It provides communication tools in form of animated figures. SIGML representation made from HamNoSys notation of sign language is readable by 3D rendering software**.**

1.7 ISL Grammar

Indian Sign Language has its own grammar. It is not dependent on the spoken English or Hindi language. Many people believe that the sign language is the manual representation of spoken English or Hindi language. But it is not true. Neither ISL depends upon spoken English or Hindi language nor on other sign languages. The unique and distinct features of ISL that makes it different from other sign languages are:

1) Number Signs: The numbers from zero to nine are formed in ISL by holding up a hand with the appropriate handshape for each number. From one to five the corresponding number of extended fingers forms the numeral sign, whereas for zero and the numbers

from six to nine special handshapes are used that derive from written numbers. Ten may either be expressed by two 5-hands or by '1+0'.

2) Family Relationship: The signs for family relationship are preceded by the sign for 'male/man' and 'female/woman'.

E.g.: BROTHER: MAN + SIBLING

SISTER: WOMAN + SIBLING

3) The question words like WHAT, WHERE, WHICH, HOW etc. are placed at the end of interrogative sentences.

English: Where is the Bank?

ISL: BANK WHERE

4) The ISL consists of various non-manual gestures including mouth pattern, mouth gesture, facial expression, body posture, head position and eye gaze (Zeshan, 2001)

5) There is no temporal inflection in ISL. The past, present and future is depicted by using signs for before, then, and after [21].

1.8 Thesis organisation

This Thesis has been divided into 6 chapters. Chapter 1 gives a general introduction on Sign Language and Indian Sign Language. It explains Dependency Parser and WordNet which are used in the system. Chapter 2 gives an overview of already developed system. Chapter 3 includes Problem Statement, Objectives and Methodology of Proposed System. Chapter 4 discusses about implementation of developed system. Chapter 5 discusses about the results and Chapter 6 concludes the work done in this Thesis.

Chapter 2

Review of Literature

Though some work has been done on machine translation (MT) from English to American or British Sign Language (SL) (Huenerfauth, 2003), but for ISL, MT systems are still in its infancy. The Underlying architecture for most of the systems is based on three categories namely direct translation, Interlingua systems and transfer based.

2.1. Direct translation

This requires knowledge of both target and source language. Moreover, word order of the output may not be the desired one. No syntactic analysis is performed to achieve translation process. In most of the cases there is same word order in two languages however in some cases same word order may not be allowed, a perfect example of this is English to Taiwanese Sign Language. In such cases a strong hold is required in English as well as Taiwanese Sign Language [15].

2.2 Interlingua Systems

To produce a language independent semantic structure semantic analysis on source text is done. This independent structure is knows an Interlingua. A generation component is then used to produce target language from this Interlingua.

2.3 Transfer based architecture

In this architecture the system analyzes input text to some syntactic and semantic level. Linguistic to target form is done by generation component. Sign Language and machine system that are based upon the text to text machine translation systems uses transfer grammar approach. Special set of transfer rules are then employed so as read the information based upon the source language structure. Hence this produces semantic and syntactic structure in target language where a special set of transfer rules are applied [13].

2.4 Existing Systems

With advancement in technology and growing research in field of NLP there has been significant improvement in Sign Language Systems. Researchers across the globe have been developing techniques to automate the process of text/speech to sign language.

2.4.1 Tessa

Tessa is an experimental system that was developed to convert input speech to British Sign Language. Tessa was developed in the year 1997 at the University of East Anglia; Norwich. The domain of this system was Post office where the developed system was tested to convert the words uttered by a clerk to sign language so that it was understood by deaf people. Designed using 115 phrases that covered 90% of daily business transactions, this system used Avatars for displaying the generated signs. The system was built on the Entropic Speech Recognizer that required a set of acoustic models and also a network. For complete sentences an accuracy of 61% was achieved and for sign units accuracy was 81%. On asking the participants about how acceptable the phrase was, an average acceptability rating was obtained as 2.2 ranging on the scale from 1.7 to 2.8. The architecture for the Tessa system is given by the Figure 2.1 [4].

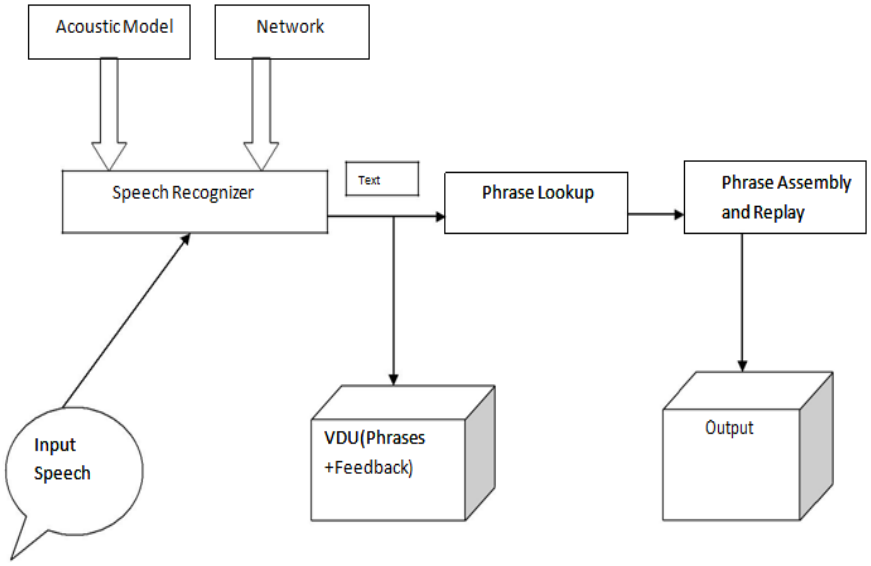

Fig 2.1 Architecture of Tessa System [4]

Tessa System has a speech recogniser which actively listens to phrases uttered by speaker. Tessa uses an entropic speech recogniser which has acoustic model that matches the input signal and a network that guides the speech recognizer during matching process. By network system is able to constraint the speech recognizer to a finite number of predefined paths using available vocabulary. System software acts as an enabler between speech recogniser module and avatar module. System displays sign using avatars.

2.4.2 Arabic Text to Arabic Sign Language

System developed to convert Arabic text to Arabic Sign Language displays the Arabic signs on the screen using the saved videos .It is web based Java application that stores the video files corresponding to the signs and is played as and when demanded. Stored entries of the database are taken into the dictionary objects rendering fast retrieval and better performance. However this system does not do any pre-processing on input text and hence appropriate grammatical structure of sign language is not followed. Also the sign videos are stored on the Internet limiting its access to the end users. Figure 2.2 Shows Flow of Arabic Text To Arabic Sign Language System [5].

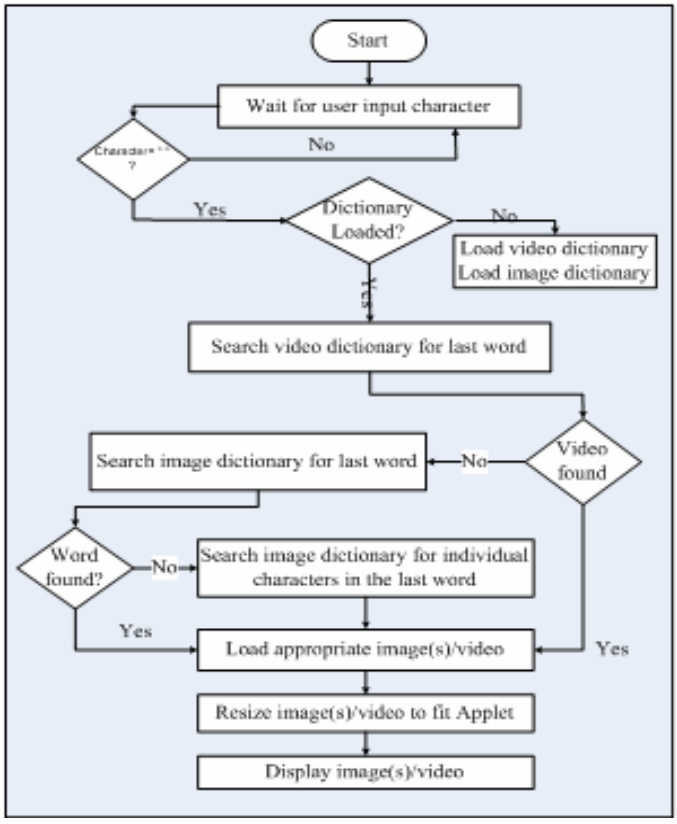

Fig 2.2 Flow of Arabic Text to Arabic Sign Language [5]

System has been developed using JSP as programming language and MS access as database. All the entries of the system are brought into dictionary object and are kept in memory to make search easy. When a word is entered and it matches an entry in memory then the location of file were word is present is accessed and subsequently is sent to browser. System works on three JSP files namely top.jsp, list.jsp and display.jsp.

2.4.3 INGIT

INGIT system is made keeping in mind Indian Railway Reservation System. INGIT is a cross modal translation system to convert Hindi String to Indian Sign Language. This uses semantically mediated formulaic framework for Hind-ISL mapping. It uses HamNoSys notation for displaying signs. The system architecture of the INGIT model is depicted in Figure 2.3. It adopts a construction grammar approach to handle formulaic inputs in terms of a construction lexicon with single constituents as well as larger phrases, with direct semantic mappings at each level [6].

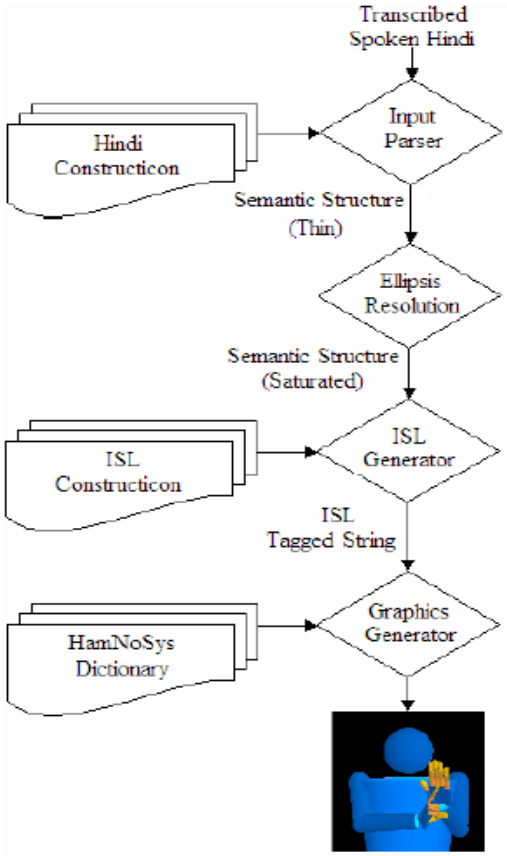

Fig 2.3 Architecture of INGIT System [6]

INGIT system works on formulaic approach which directly generates semantic structures wherever possible and compositional mode for others. It uses FCG framework for analyzing the input and generation of sign. If semantic structure of sentence is saturated it is passed to ISL generator otherwise it is passed to ellipsis resolution module. It handles elision of participants in ternary events and subjects in unary and binary events. INGIT uses HamNoSys for sign notation and avatars for displaying signs.

2.4.4 Zardoz

Zardoz is a multilingual sign translation system was designed to translate textual or spoken language into various graphical animated sign-languages that include particularly Irish Sign Language (ISL), Japanese Sign Language (JSL) and American Sign Language (ASL). Zardoz was designed in the year 1991 at Trinity College, Dublin, Ireland. Being a form-driven approach, it works by processing the English input serially by using the morphological analysis, parsing by a unification grammar and idiomatic reduction. It considers the issues of broad syntactic and semantic categories of sign language, and employs spatial-dependency graphs to specify the output syntax for the sign languages in a robust and flexible way [4]. The systems described so far work on domain specific information and on a limited set of words. Figure 2.4 depicts architecture of Zardoz system [7].

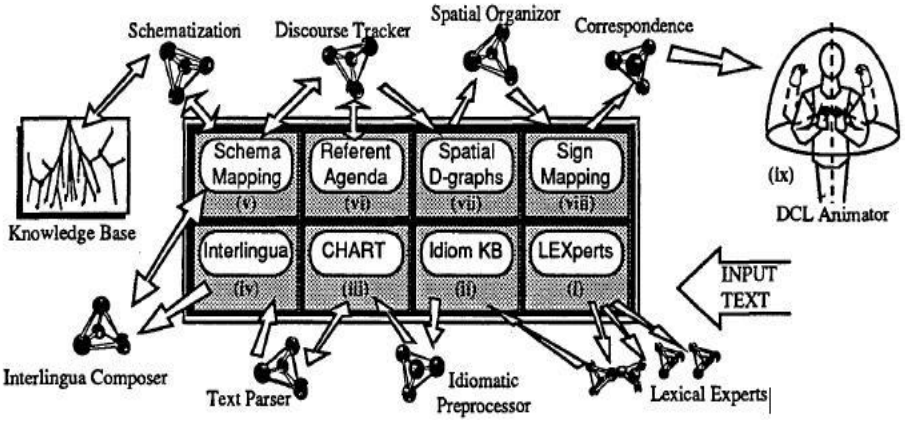

Fig 2.4 Architecture of Zardoz System [7]

It uses syntactic framework of spatial dependency graphs. SD-graph represents sytatic context and is collection of constraints for ordering the elements of Interlingua frame structure. Whenever source and target language uses different anaphoric discrimination

systems anaphoric resolution system is required for translation process. Fluid articulation of utterance is required to display an animated sequence.

2.4.5 Visicast Translator

Visicast is English to British Sign Language Translator. Visicast has semi-automatic mode of sign sequence preparation which involves human intervention wherever necessary. Visicast uses Natural Language processing techniques to construct a DRS semantic representation of text. This semantic representation will be translated to capture morphological rules and semantic conventions.

Example sentence given in 2.1 outlines how a sentence will be represented in British Sign Language as stated in 2.2.

English Sentence: The boy bought the book. …(2.1)

British Sign Language: BOY — BOUGHT — WHAT — BOOK ... (2.2)

Visicast provides an interface where user can enter sentence, here user can make necessary changes to sentence if required. After this at syntactic level input is parsed with Carnegie Mellon University (CMU) link grammar parser. After this DRS semantic representation of text is generated from parsed text. Head Driven Phrase Structure grammar is used to define the morphology and syntax of sign generation. SIGML is then used to represent sign notations in form of XML. SIGML is easily understood by three-dimensional rendering software to display signs in form of animations [8]. Figure 2.5 shows architecture of Visicast translator.

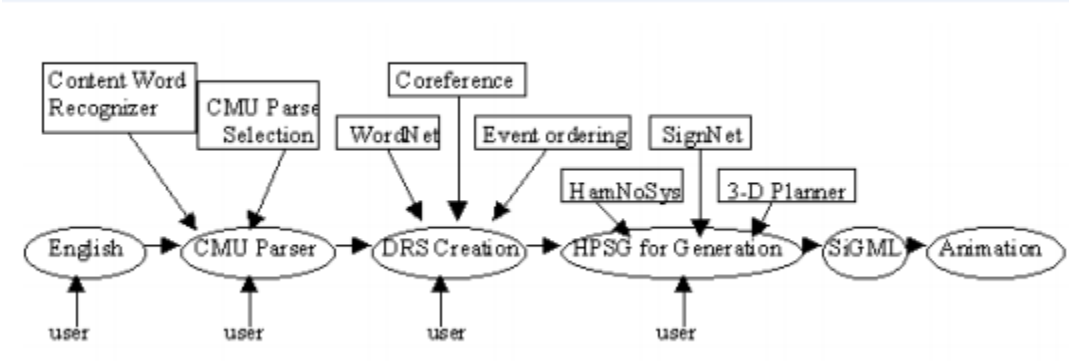

Fig 2.5 Architecture of Visicast System [8]

Visicast translator adopts a semi-automatic approach for generation for sign sequence. It supports human intervention in generation of sign sequences wherever necessary. NLP techniques are used to construct a DRS semantic representation of English Text. This representation is then translated to a form which captures morphological rules of sign construction and semantic convention used in BSL. It uses Gesture Markup Language (GML) for representation of signs.

2.4.6 ASL Workbench

ASL workbench is a text to American Sign Language System. It uses LFG style f-structure to analyze the input text. ASL workbench has specified rules to convert f-structure representation of English text to American Sign Language. Translation from English f-structure to ASL f-structure is done using structural correspondence and performing lexical selections. For noun ASL workbench does finger spelling. c-structure and p-structure for sentence is generated by generation module [9].

An f-structure is a finite set of attribute–value pairs, such that an attribute is a symbol and its value is namely symbol (e.g., SINGULAR or +), a semantic form (a potentially complex symbol in single quotes),a set; or an f-structure. C-structure models the surface exponence of syntactic information, such as word order and constituency. Architecture of ASL Workbench is given in figure 2.6.

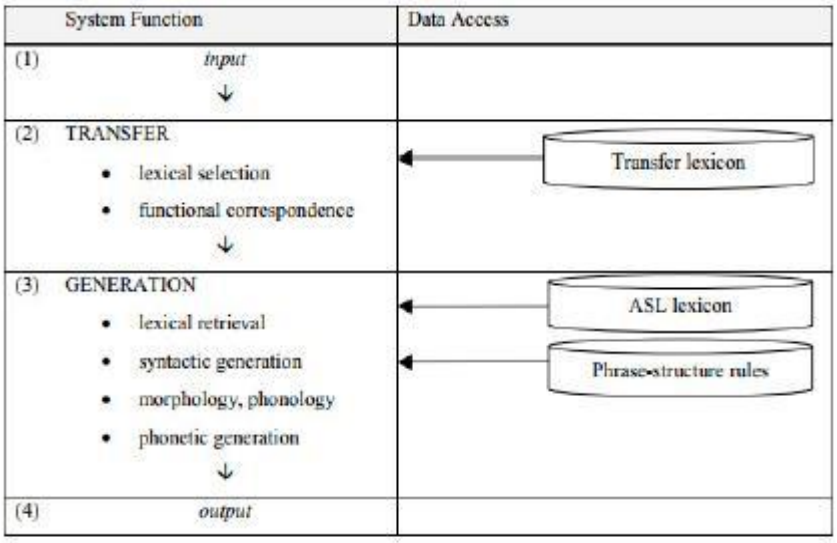

Figure 2.6 Architecture of ASL Workbench [9]

2.4.7 Machine Translation from Text to Indian Sign Language

A novel machine translation system was designed for converting English text to Indian Sign Language. System takes simple text as input here simple means sentence having one verb only. The input sentence is then parsed using Minipar parser. Dependency structure is constructed from parse tree. Before parsing is done pre-processing is done to remove frozen phrases and temporal expressions [10]. Architecture of Text to ISL MT system is given in figure 2.7.

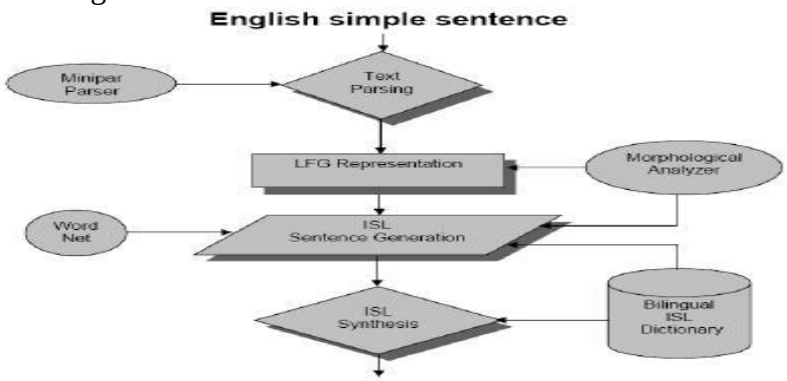

Figure 2.7Architecture of Text to ISL MT [10]

The f-structure encodes grammatical relation like (tense, object and subject) it also includes higher syntactic and functional information of a sentence. English f-structure is converted to ISL f-structure by applying proper grammar rules. During generation two main operations are performed namely Lexical Selection and Word order correspondence [16].

Table 2.1 gives an overview of systems discussed.

Table 2.1 Literature Survey Conclusion

Authors	**Input Language**	**Output form**	**Name of system**	**Application/Domain Area**	**Approach/Description**
[4] Cox *et al*. (2002)	English speech	British Sign Language (BSL)	TESSA	post office operations(Accuracy: 61% for complete phrases and 81% for sign units.)	phrase lookup approach
[5] M. Mohandes (2006)	Arabic Text	Arabic Sign Language	Arabic Text To Arabic	Not Domain Specific	Java based cross-platform application

			Sign Language		
[6]P. Kar *et al.*(2007)	Hindi	Indian Sign Language(ISL)	INGIT	Indian Railways reservation counters.	hybrid-formulaic approach and semantically mediated formulaic framework for Hindi-ISL mapping
[7] T. Veale *et al.*(1998)	English text/spoken	fluid sign language.	Zardoz	NA	Form-driven approache and knowledge-intensive interlingua approach
[8] J.A. Bangham *et al.* (2000)	English text	British Sign Language(BSL)	ViSiCAST	mainstream applications outside the context of deaf signing.	Used Gesture Markup Language for language and notation and the virtual human signing
[10] Dasgupta *et al.* (2008)	English text	Indian Sign Language	NA	NA	Converts English Text to ISL with proper grammar rules.
[17] Suszczanska , *et. al* (2002)	English Text	American Sign Language	ASL Workbench	NA	used LFG styled f-structure to analyze input text

Sign language systems are getting better with time with development of various NLP techniques. Tessa, Arabic Text to Arabic Sign Language, INGIT, Zardoz, Visicast, ASL project, Machine Translation from Text to Indian Sign Language system takes English sentence and convert it into respective Sign Language grammar. The system discussed so far uses different approaches for translating text to sign language grammar.

Chapter 3

Problem Statement

The Sign Language is primary way of communication for deaf and dumb people. It helps them to communicate by gestures rather than speaking. If a vocal person is unaware of sign language the communication between hearing impaired people and vocal person is achieved through a sign interpreter. It is always not possible to have an interpreter when a person is unaware of sign language. So it becomes highly important to develop a system which can be generates signs for a particular word. According to All Indian Federation of the deaf a survey reports that an estimated number of nearly 4 million are deaf people and greater than 10 million people are reported to be hard of hearing in India. With advancements in computers and technology it has become possible to develop an automatic interpreter which can be used to convert Text to Sign Language Gestures and can be accessed across the globe.

3.1 Objectives

The main objectives of this research is as follows

- To understand ISL and its grammar.
- To understand the usage of Hindi Dependency Parser API and its usage.
- To develop translator for conversion of Hindi sentence to ISL grammar.
- To extend the capability of ISL dictionary using the Hindi WordNet API.
- To test and validate the proposed system.

3.2 Methodology

In this thesis a system has been developed which takes Hindi sentence as input parses it using Hindi Dependency parser. Each sentence is saved in a unique file generated. Parser is called on the saved file to process the information on each word in sentence. Parser is called from Java program using Process API of java. Make function is used to call the parser for execution. Parser generates the POS tagging information for each word in sentence and save the output in the output file which has same name as the input file. Once

the parser has completed parsing the ISL generator module is called to process information provided by Parser. Now grammar rules are applied on the words based on the POS tagging information provided by Dependency Parser and rephrased Indian Sign Language Sentence is generated. Now database is searched for each word to check if sign is available. If a word does not have sign database then WordNet module is called on that word. WordNet module has a Java API which is provided in JAR file. Jar file is based on zip file format which is used to pack classes related to a common task into one [18].WordNet Module returns synonyms for input word. Now for each synonym database is searched to see if sign is present. If sign is present than word is replaced with the synonym. In case no synonym is found who has sign in database than that word is marked to be finger spelled.

Chapter 4

Architecture and Implementation of Proposed system

4.1 Architecture

The architecture of the developed system is given here, discussing each of its modules briefly. Also Fig 4.1 depicts various modules of the system architecture. The architecture has been developed keeping mind the requirement of an integrated and reliable system.

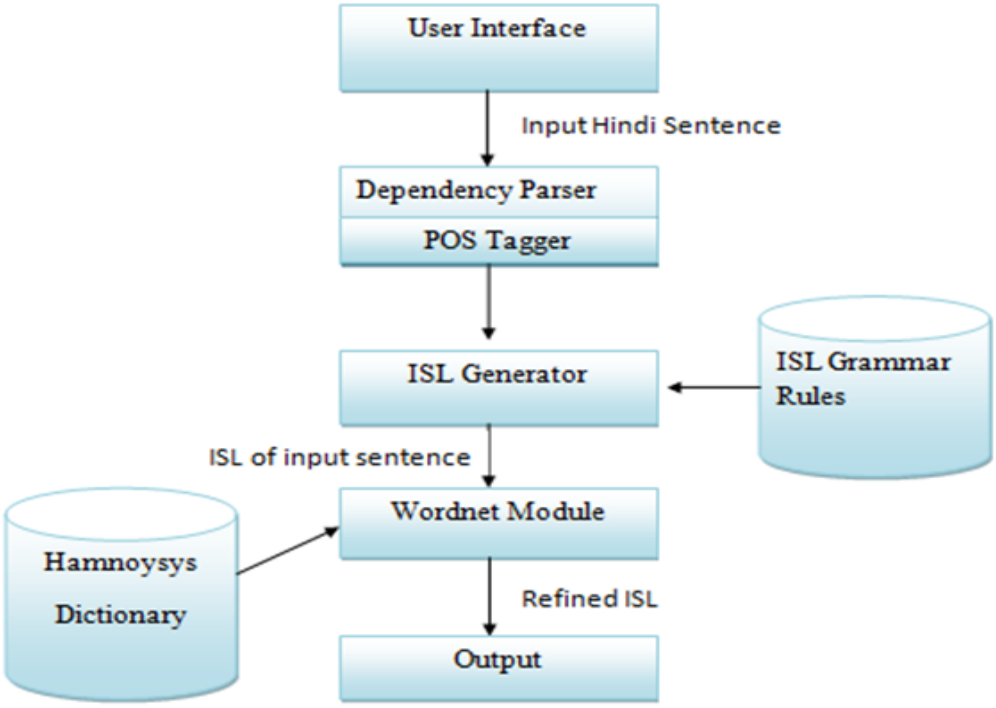

Fig 4.1 Architecture of Developed System

4.1.1 User Interface

User interface is the front face of application. User sees only the screen which is provided to him for giving input. User is not aware of what is happening at the backend and sees the output at an interface. The System User Interface has been developed using Java Swings which is used for developing user friendly Java Desktop Application. System has only one screen where user enters Hindi input sentence and on the same screen translated ISL is

shown. System has used swings user interface components given in table 4.1.

Table 4.1 Java Swings Components

Component	**Class**	**Usage**
Label	Javax.swing.JLabel	Display text or image or both.
Panel	Javax.swing.JPanel	Organize different components.
Text Field	Javax.swing.JTextField	To insert or display text.
Button	Javax.swing.JButton	To perform action on different events.

4.1.2 Dependency Parser

To convert the input sentence in accordance with ISL parser is required to process the sentence and extract information. Hindi Dependency Parser developed by IIIT Hyderabad has been used. It gives POS tag information along with other information for every word. Dependency parser has been written in Java and Python. To execute dependency parser we need to call **make** command of Process Builder Class in Java with path of parser. The execution of dependency parser using make command is described in table 4.2.

Table 4.2 Dependency Parser Execution Script

```
List<Strings> command=new ArrayList();
command.add(make);
command.add(filename.output);
ProcessBuilder pb=new ProcessBuilder();
pb.directory(/etc/parser);
Process process=pb.start();
```

ArrayList() is used to store the list of commands that will be executes by ProcessBuilder class to call the parser. Command is the object of ArrayList class which contains the command required for execution. In this code a system process gets started to execute the parser from our Java code.

Once the parser has finished execution it signals the Process Builder Class. Process Builder has waitFor() function which checks whether the process has finished execution or not. The

value of waitFor() function becomes zero when process has finished execution [20]. When parser completes execution its results are save in the output file. The output file has same name as the input file which has the input sentence for parsing. The extension of output file is .output.txt. This .output.txt file has all the information about the parsed sentence. Table 4.3 shows parser output for sentence कौए की चोंच पानी तक आराम से पहुँच गई (kauve ki chaunch paani tak aaram se pahunch gayi).

Table 4.3 Parsed output for Hindi sentence

Input Word	**Parsed Output**
कौए	<1> <कौए> <कौआ> <NN> <3> <r6>
की	<2> <की> < का ><PSP:का > <1> <lwg_rsp>
चोंच	<3> <चोंच> < चोंच > < NN > <8> <k1>
पानी (paani)	<4>< पानी >< पानी >< NN > <8> <k1>
तक (tak)	<5> <तक> <तक> <RP> <4> <lwg_rp>
आराम (aaram)	<6> <आराम > <आराम> <NN> <8> <adv>
से (se)	<7> <से> <से> <PSP: से> <6> <lwg>
पहुंच (pahunch)	<8> <पहुंच> <पहुंच> <VM> <0> <main>
गई (gayi)	<9> <गई> <गई> <VAUX> <8> <lwg_aux>

4.1.3 ISL Generator

Indian Sign Language generator takes output file given by Dependency Parser as input and applies ISL Grammar rules to map input sentence into ISL Sentence. Grammar rules are applied on the bases of POS tagging information. Also reframing of sentence is done in this phase. Example text from Hindi to ISL is given in (4.2). In the example '*ने*' is removed from the input sentence to map with ISL grammar. ISL generator has rules set which are applied on the parsed input text .

Input Sentence

कौआ बड़े ध्यान से सोचने लगा

Output ISL

कौआ बड़े ध्यान सोच लगा ... (4.2)

ISL generator has rules. The rules a rea pplied on the POS tagging information of ISL. Different POS tags and their usage has been explained in subsequent section.

i. **NN (Noun)**

The tag NN meant for the nouns has been taken from the Penn tags as it is. In contrast to Penn tags which makes a distinction between noun plural (NNS) and the noun singular (NN) in Indian Language tagging scheme some tags that were based on grammatical information have been avoided. Information that can be accessed from any of the other resources has not been incorporated.

Plurality associated with a word can always be obtained from the morph analyzer. So it will not be included in POS tagging scheme. However, as stated earlier, that in case particular information is found to be crucial at level of POS tagging, it can be later incorporated taking the help of linguistic and heuristics rules. The approach is meant to bring number of POS tags down, achieving consistency, simplicity, and a sense of better machine learning with a small corpus. So in current tagging scheme only NN tag has been incorporated and NNS has been avoided.

ii. **NST (Noun denoting temporal and spatial expressions)**

The NST tag has been adopted for covering an important aspect of the Indian languages. Certain expressions are used for the denotation of the temporal and spatial information.

For instance the ‘Upara’ (:up/above), 'pahale' (for expressing :before), 'Age' (for expressing :front) ,'*nIce*' (for expressing :below), are the content words that represent the features such as space and time. The above expressions can used be in various ways. For instance,

a)

Phrase: tum andar betho

ISL:'you' 'inside' 'sit'

English meaning: “You sit inside”.

b)

Phrase: *vaha* neeche so rahA thA .

ISL: 'he' 'downstairs' 'sleep' 'PROG' 'was'

English meaning: “He was sleeping downstairs”.

iii. **NNP (Proper Nouns)**

NNP tags are included to cover the Proper Nouns in Indian Language. Unlike Penn tags no distinction has been made for proper noun singular and proper noun plural. In PENN tags proper noun plural is represented using NNPS and proper noun singular is represented using NNP. In current tagging system NNP tag is used for representing singular proper noun and plural proper noun.

iv. **PRP (Pronoun)**

In current tagging scheme in contrast to Penn Tags no distinction has been made between possessive pronouns and personal pronouns. In case of POS tagging the pronouns are tagged as PRP category. In the Indian languages all the pronouns used instead of the nouns are used as inflections for all the cases such as the possessive, accusative etc. But in case there is a separate made tag made for the possessive pronoun then new tags will have to be designed for such cases as well. Deciding that the tagging for the pronouns will prove to be helpful for the anaphora, thus resolution tasks should be retained.

v. **DEM (Demonstratives)**

The tag 'DEM' was included for marking the demonstratives. The necessity of including the DEM tag for the purpose of demonstratives was encouraged to cover the difference between demonstrative and the pronoun. For instance take the case of following two sentences:

Sentence 1:

vaha ladaki meri behan hain (demnostrative)

'That' 'girl' 'my' 'sister' 'is'

Sentence 2:

vaha meri behan hain (pronoun)

'She' 'my' 'sister' 'is'

Sentence 1 and Sentence 2 show the use of demonstrative and pronoun POS tagging.

vi. **VM (Verb Main)**

In a language verbal construction can include more than one word form of sequence VM tag is used to depict the verbal constructions. Main verb and one more auxiliaries(VAUX AUX …) are part of verb group sequence. Supporting verbs are also tagged as VAUX in current tagging scheme. So this group can be categorized as non-finite or finite verb schemes. For the purpose of finiteness main verb may not be marked.

vii. **VAUX (Verb Auxiliary)**

VAUX will be used to represent the auxiliary verbs .This tag was adopted as such from the Penn tags.

viii. **JJ (Adjective)**

In contrast to PENN tags which make a difference between superlative and comparative adjective no such distinction is made in current tagging scheme. JJ is being used to depict the adjectives. However in the current scheme for the Indian languages, the JJ tag has included the 'tama' (superlative) forms and the 'tara' (comparative) forms of the adjectives as well. For instance in the Hindi language the sarvottama meaning ‘the best’ and the adhikatara meaning the ‘more times’ has been also be marked as the POS JJ.

ix. **RB (Adverb)**

For the purpose of describing the adverbs the tag ‘RB’ was taken from the Penn tags.

Similar to adjectives POS tagging, Penn tags are there which hereby make a difference between superlative and the comparative adverbs. This difference is not made in the given

tagging system. This has been in accordance with the concept of coarseness in linguistic analysis of the document.

x. **PSP (Postposition)**

For expressing the Postpositions the POS tag used is PSP. All the existing Indian languages have concept of postpositions which are the used to express the grammatical functions for instance the case etc.

For example:
mohana khet meM beej daal raha tha

English form: 'Mohan' 'field' 'in' 'seed' 'put ' 'PROG' 'was'
Here the ***'meM'*** in the above example is a postposition that will be tagged as PSP.

xi. **RP (Particle)**

Expressions such as *hI,nA,sA,jI, sA,bhI* etc in Hindi shall be marked as RP.

Here the '*nA* 'in the given list is distinct from negative *nA.* Hindi and some other Indian languages have ambiguity in 'nA' that can be used both for negation (NEG) and for reaffirmation (RP).

xii. **CC (Conjuncts (co-ordinating and subordinating)**

Subordinating and co-coordinating conjucts has been tagged using CC in current tagging scheme. Subordinating conjuncts and prepositions has been tagged The Penn tag set.

xiii. **WQ (Question Words)**

In Penn Tag set there is a distinction between different form of usage of 'wh-'words and has marked them accordingly for WRB, WQ, WDT etc however in current tagging scheme there is no distinction. The 'wh-' words in the English language can act in form of the questions, as determiners and the relative pronouns also.

xiv. **QF (Quantifiers)**

All quantifiers which specify about the quantity of the subject are classified as the QF tagged words. For instance the jyada meaning more, kama meaningless, bahuwa for the lots are all being marked as QF.

xv. **QC (Cardinals)**

QC tag is used to represent the cardinals. Any word that denotes the cardinal number will now have the Penn tag set CD for cardinal numbers and they have not talked about the ordinals.

xvi. **QO (Ordinals)**

QO is used to denote the expressions that denote the Ordinals.

xvii. **CL (Classifiers)**

The tag CL has been included to mark classifiers. Many Indian languages have a rich classifier system. Basically the classifier represents the morpheme or the word that has been used in many languages for the purpose of classifying the noun according to its meaning or semantics.

xviii. **INTF (Intensifier)**

This Intensifier tag is the one that is not present in Penn tag set as such. Rather the words like 'kama','bahuta' etc. Are the ones where intensifying adverbs or the adjectives shall be annotated as INTF.

xix. **INJ (Interjection)**

Here the INJ is used to mark the interjections. Also apart from the normal interjections, also the affirmatives for instance the Hindi 'HAz' meaning 'yes' shall also be tagged as INJ. For the state of purpose this is the going to be the only example of such word, where it has been clubbed under the Interjections category.

xx. **NEG (Negative)**

This special POS tag is used to mark the negatives .For instance the Hindi words like

‘nahIM’ meaning ‘not’, 'nA' meaning ‘no’, ‘not’ etc. Will also be marked as NEG POS here.

xxi. **SYM (Special Symbol)**

In the linguistics all those words that cannot be classified under any of the other tags will

now be tagged as the SYM POS category. Here this tag is almost similar to the Penn tag 'SYM'. Including the special symbols like %,$, etc in the SYM category. To note as the frequency of the occurrence of many such symbols is quite less in Indian languages so there is no separate tag that can be used for such symbols [1].

4.1.4 WordNet

The Hindi WordNet is a system that is meant to collaborate various types of relations between the Hindi words ranging from semantic to lexical forms. It provides organized information giving a detailed overview of the meanings of the words on lexicon basis. Here the words that have the similar meaning are grouped together which later can be interchanged in contexts demanding synonyms. There exists a synonym set, or synset, for each word that represents one lexical concept. With this ambiguity is also removed in case a word has more than one meaning. So basically 'Synsets' act as the basic building blocks of WordNet. An entry in the Hindi WordNet comprises of primarily three main component [19].

Hindi WordNet developed by IIT Bombay has been integrated in the system to increase the overall efficiency of the system. IIT Bombay has created an API in java to use WordNet.
To integrate Hindi WordNet JHWNL jar needs to be added as external jar in Java Project. Jar included in out project has been shown in figure 4.2.

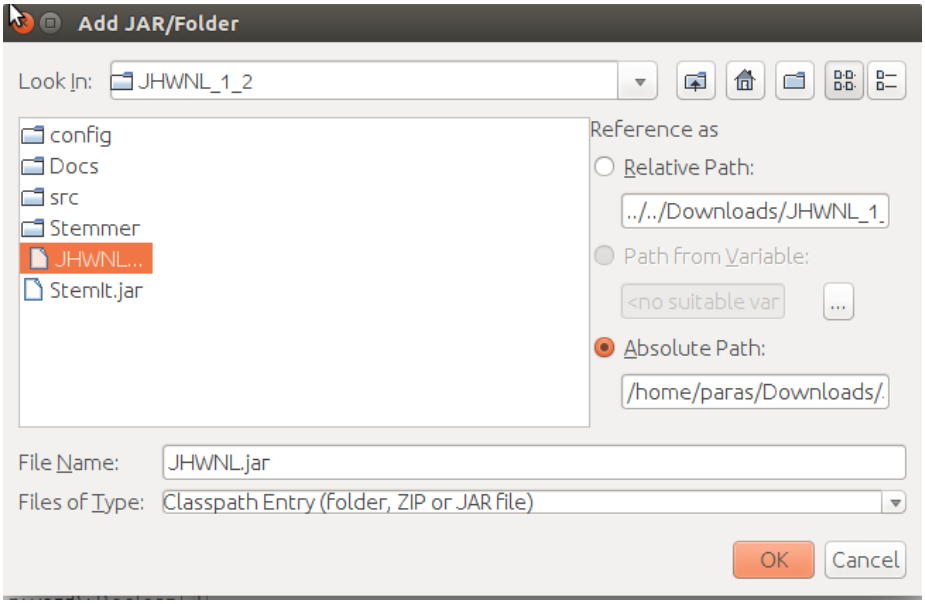

Figure 4.2 Adding JHWNL Jar in project.

Once jar is included all the classes of WordNet API can be accessed. To find synonyms different classes that needs to be included in project is given in table 4.4.

Table 4.4 JHWNL classes

```
import in.ac.iitb.cfilt.jhwnl.JHWNL;
import in.ac.iitb.cfilt.jhwnl.JHWNLException;
import in.ac.iitb.cfilt.jhwnl.data.IndexWord;
import in.ac.iitb.cfilt.jhwnl.data.IndexWordSet;
import in.ac.iitb.cfilt.jhwnl.data.Synset;
import in.ac.iitb.cfilt.jhwnl.data.Word;
import in.ac.iitb.cfilt.jhwnl.dictionary.Dictionary;
```

All the classes are accessible only if jar is included. Hindi WordNet API is dependent on three folders namely config, MorphIN and database. These there folders should be added in root of project otherwise invalid configuration file exception will be thrown.
Hindi WordNet API comes with rich set of functions for processing information.

To initialize WordNet static function initialize of JHWNL should be called. Figure 4.3 represents the technique for getting all the words which are similar to input word from WordNet. demoIWSet is object of IndexWordSet class which is available in API. demoIndexWord is object of IndexWord[] class which is used to get size of words which are similar to given words. demoIndexWord is then initialized to base index array which points to input word..

```
//  Look up the word for all POS tags
IndexWordSet demoIWSet = Dictionary.getInstance().lookupAllIndexWords(word);
//  Note: Use lookupAllMorphedIndexWords() to look up morphed form of the input
IndexWord[] demoIndexWord = new IndexWord[demoIWSet.size()];
demoIndexWord = demoIWSet.getIndexWordArray();
```

Figure 4.3 Technique for getting words similar to input word

To find senses for different words we need to find synonyms for all the words which are similar to input word. To find synonyms we need to fetch the Senses of words. Technique

for getting Sense is given is figure 4.4.

```
int size = demoIndexWord[i].getSenseCount();

//System.out.println("Sense Count is " + size);
synsetOffsets = demoIndexWord[i].getSynsetOffsets();


Synset[] synsetArray = demoIndexWord[i].getSenses();
```

Figure 4.4 Technique for getting Senses

Table 4.5 represents different senses for word ध्यान. It explains the usage of a single word in many senses that can be possible in different scenarios.

Table 4.5 Different Senses for word ध्यान

Sense Index Id	**Words in a sense**
98	[याद, ध्यान, ख़याल, खयाल, ख़्याल, ख्याल, स्मतत, सध, सगध, तसव्वर, तसव्वर, तसौवर, अभभज्ञान
3073	[ध्यान, खयाल, ख़याल, ख़्याल, ख्याल, परवाह, भलहाज, भलहाज़, तवज्जोह, तवज्जो, तवज्जह, मलाहज़ा, मलाटहज़ा, मलाहजा, मलाटहजा]
4999	[ध्यान, गौर, ग़ौर, मनोयोग, मनोन्न्नयोग, अभभतनवेश,प्रहाण, फोकस]
6332	[ध्यानयोग, ध्यान, योग, जोग]
34662	[ध्यान, ख़याल, खयाल, ख़्याल, ख्याल, स्मतत, याद, सध, सगध, तसव्वर, तसव्वर, तसौवर, नज़र, नजर]

4.2 Implementation of Proposed System

In proposed system user is provided with an interface where he can put Hindi Sentence which he wants to convert into ISL grammar. When user enters sentence an input file is generated which has a random unique file. Java file handling API is used to save the Hindi input text in file. Java IO class is used to handle the file handling. Figure 4.5 shows various steps needed to save unique file with input sentence. File is saved with .input.txt file. Figure 4.6 represents an input file generated when user enters a sentence.

```
File fileDir = new File("/home/paras/Downloads/parser/" + fileName+".input.txt")
Writer out = new BufferedWriter(new OutputStreamWriter(
        new FileOutputStream(fileDir), "UTF8"));
```

Figure 4.5 Generate an input file for storing input sentence

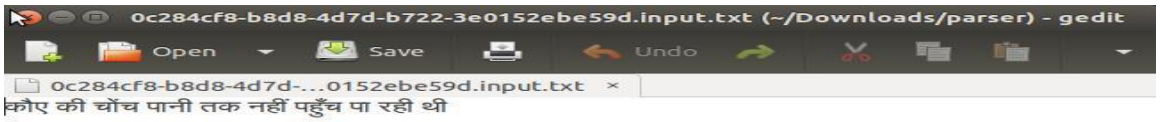

Figure 4.6 .input.txt file for parser

Once input file is generated parser is executes as explained earlier. The output of parser is saved in filename.output.txt file. Figure 4.7 represents an output file generated from parsed input. The output file will have the same name as the input file however the extension will be .output.

Figure 4.7 .output.txt File for Parser

POS tag and other information for each word in input sentence is present in output.txt file. The formation of output file completes first phase of pre-processing.

Now .output file needs to be read using Java File Handling API and process each word. Code for reading the output is given in figure 4.8. Format of output files has already been explained.

```
Scanner scan = new Scanner(new java.io.File("/home/paras/Downloads/parser/" +
        fileName + ".output"));

while (scan.hasNext()) {

            String line = scan.nextLine();
            String[] elements = line.split("\t");
```

Figure 4.8 Code for reading input file

For each word information is separated by a '\t' and each word is separated by a '\n'. line.split("\t") splits the information by tab-space and stores it in String array. scan.nextLine() function is used to read next in output file which has information for next word. On splitting POS tagging information will be stored in 3rd index of string array.

List data structure is used to store the root words. If word passed the rules for ISL grammar its lemma is stored in roots List. Each word is added to back of the list using add function of List Data Structure.

Once all the words have been processed database needs to be checked to see if sign is stored in database for the root word. To increase efficiency of system WordNet is used which will help in giving alternate words for words which are not available in dictionary. When a synonym for word is found in dictionary the word will be replaced with that synonym. However if no alternative is found than that word will be marked as finger spelled which is represented as F.S. in proposed system.

Comma Separated Values (CSV) format is used to store words for which signs are available in a database. CSV file is used as it is cross platform and there is no need to install additional software for accessing CSV files. CSV allows data to be stored in table structured format. Each column in CSV file is separated by comma and each row by

newline [18].

OpenCsv library is used to read and write in csv file. OpenCsv library is an API which is used to read and write CSV files in Java. To use OpenCSV opencsv.jar needs to be included in the project. OpenCSV comes with rich set of functions which can be used to create and read csv files. In current system we are using OpenCSV to read our dictionary of sign language which has been stored in csv format. Figure 4.9 represents code for reading signs database file in csv format.

```
CSVReader csvReader = new CSVReader(new FileReader("words.csv"));

String[] nextLine;

while ((nextLine = csvReader.readNext()) != null) {
    //sSystem.out.println(nextLine[0]);

    if (nextLine[0].equals(words.toString())) {
        return nextLine[0];
    }
}
```

Figure 4.9 Reading csv file using OpenCSV

To read CSV file we need to create an object of CSVReeader class and give path of csv file. csvReader.readNext() function is used to read next row in csv file it returns array of string. Each column is stored as an index in the array returned.

Table 4.6 outlines algorithm to find the root words once ISL generator has removed words which are not required in Indian Sign Language.

Table 4.6 Algorithm to find root words

```
Procedure finalRootWords(list : root words)
    for each word in list:
        if word in database:
            save word in rootWordFile
        else:
            synonym=findSynonym(word)
```

```
            if(synonym==Null):
                mark word as fingerspelled and save in file
            else:
                replace word with synonym and save synonym in rootWord File
            end if
        endif
        end for
end Procedure
Procedure findSynonym(word):
 Synset[] synsets=all senses of word
        for i=0;i<synsets.size;i++:
            Words words=synsets[i].getWords()
            for each word in words:
            if word in database:
                return word
    endfor
     endfor
endprocedure
```

For each root word given by ISL generator for sentence sign language database is searched to check if sign available. Procedure finalRootWords() is used to iterate through all the root words given by ISL generator. If sign is available it is saved in rootWordFile. If sign is not available then findSynonym procedure is called which generate all the synonyms for word. It check if sign is available for synonym and if sign is available for synonym it is returned back to finalRootWords() procedure which replaces the word with synonym and save it in rootWordFile. If no synonym for word has sign available the word is marked Finger Spelled (F.S.) and save in rootWordFile.

Suppose user enters '**कौआ बडे ध्यान से सोचने लगा**' (kauaa badhe dhyaan se sochne laga) .This text will be saved in a file. Suppose random file name generated is 06712asdfc123. So text will be saved in 06712asdfc123.input.txt file. Now parser needs to

be executed on this file. To execute parser make 06712asdfc123.output file is executed. The parser executes and generates 06712asdfc123.output.The format of output file generated is given in Table 4.7. Now grammar rules will be applied on this sentence. After applying grammar rules ISL mapped sentence is as shown in 4.3.

कौआ ध्यान सोच लगा … (4.3)

Here '*सोच*' is not in database of words. Now WordNet module is executed to find all the synonyms for '*सोच*'. After getting all the synonyms for '*सोच*' database is checked to see if anyone of the synonyms that are found are available in dictionary. Synonyms that are found using WordNet has '*गचंता*' available in dictionary. Now '*सोच*' is replaced with '*गचंता*'.

The final sentence is given in 4.4.

कौआ ध्यान गचंता लगा ... (4.4)

The final sentence is saved in '06712asdfc123-rootwords.txt' shown in figure 4.10. Also final sentence is displayed in front of ISL sentence in application which is shown in figure 4.11.

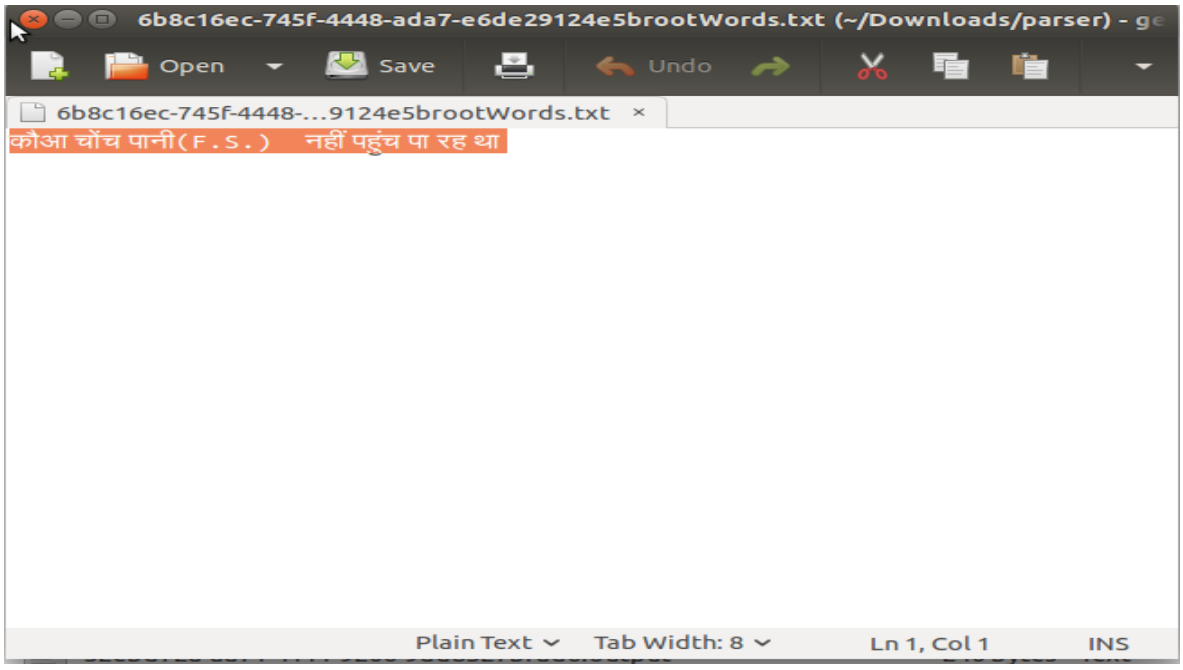

Figure 4.10 Root Words File

Figure 4.11 ISL mapped sentence shown on Screen

Table 4.7 Dependency Parser Output File

Word id	**Word**	**Lemma**	**POS Tag**	**Parent id**	**Dependency label**
1	कौआ	कौआ	NN	3	mod
2	बडे	बडे	JJ	3	Nmod_adj
3	ध्यान	ध्यान	NN	5	adv
4	से	से	PSP:से	3	lwg_psp
5	सोचने	सोच	VM	0	Main
6	लगा	लगा	VAUX	5	lwg_vaux

Chapter 5
Results and Discussion

The proposed system has been tested on Hindi sentences taken from Internet. There are 7 categories of Hindi sentences namely विधिवाचक वाक्य (Affirmative Sentence), निषेधवाचक वाक्य (Negative Sentence), आज्ञावाचक वाक्य (Imperative Sentence), प्रश्नवाचक वाक्य (Interrogative Sentence), विस्मयादिबोधक वाक्य (Exclamatory Sentence), संदेहवाचक वाक्य (Skeptical Sentence), इच्छावाचक वाक्य (Wishful Sentence) on which system has been tested**.** 10 sentences from each category was picked and tested making a corpus of 70 sentences. 70 sentences were then converted to ISL grammar sentences and were shown to ISL experts. Total of 416 Hindi words were present in corpus. In 22 sentences WordNet replaced a word whose sign was not available with a synonym for which sign is available. Table 5.1 list observation of experiment. By using WordNet we were able to decrease the percentage of words with no sign from 29.2% to 21.4%. As size of Sign Corpus would increase this number will go down further. Table 5.2 list Hindi to ISL Grammar sentence. Table 5.3 list Hindi to ISL Grammar sentence with WordNet.

Table 5.1 Observation Table

Total number of Hindi Sentences Taken for Testing	70
Hindi Sentences Categories Covered	7
Total Words in Hindi Sentence	350
Total Words in ISL grammar	280
Number of words with No sign available	60
Number of words whose Synonym with sign was found	22
Percentage of words with no Sign without WordNet	29.2%
Percentage of words with no Sign with Wordnet	21.4%

Table 5.2 Hindi Sentences and their ISL mapped sentences.

Hindi Sentence	ISL Sentence	Type
लड़का घूमने जा रहा था	लड़का घूम जा रह था	विधिवाचक वाक्य
कौआ लगातार पत्थर डालता गया	कौआ लगातार(F.S.)पत्थर डाल गया	विधिवाचक वाक्य
धीरे धीरे पानी ऊपर आने लगा	धीरे(F.S.) पानी ऊपर आ लगा	विधिवाचक वाक्य
लड़का बोल नहीं सकता है	लडका बोल नहीं सक(F.S.)	निषेधवाचक वाक्य
लड़का काम पर नही गया था	लडका काम(F.S.) नही गया था	निषेधवाचक वाक्य
लड़की पानी नहीं पीती	लडकी पानी नहीं पी	निषेधवाचक वाक्य
लड़के अब पढ़ो	लडका पढ़ो	आज्ञावाचक वाक्य
बैठ जाइये	बैठ जा	आज्ञावाचक वाक्य
नीचे जाओ	नीचा जा	आज्ञावाचक वाक्य
लड़का कहाँ जा रहा है	लड़का कहाँ जा रहा	प्रश्नवाचक वाक्य

क्या घड़ा भरा है	क्या घडा भर	प्रश्नवाचक वाक्य
क्या पानी है	क्या पानी	प्रश्नवाचक वाक्य
वाह कितना सुंदर फूल है	वाह कितना(F.S.) सुंदर फूल	विस्मयादिबोधक वाक्य
भारत जीत गयी	भारत(F.S.) जीत गया	विस्मयादिबोधक वाक्य
लड़का मान गया	लडका मान(F.S.) गया	विस्मयादिबोधक वाक्य
शायद लड़का दिल्ली जाए	शायद लडका दिल्ली(F.S.) जा	संदेहवाचक वाक्य
शायद लड़के को प्यास लगी हैं	शायद लडका प्यास	संदेहवाचक वाक्य
शायद रेलगाड़ी देर से आएगी	शायद रेलगाडी(F.S.) देर(F.S.) आ	संदेहवाचक वाक्य
भगवान की लड़के पर कृपा हो	भगवान(F.S.) लडका कृपा	इच्छावाचक वाक्य
भगवान से शांति की दुआ कर	भगवान(F.S.) शांत दुआ	इच्छावाचक वाक्य

Table 5.3 ISL Grammar Sentence with WordNet

Hindi Sentence	ISL Sentence
लडकी खूबसूरत है	लडकी सुंदर
लडका लम्बा है	लडका ऊंचा
कौआ बहुत प्यासा था	कौआ बहुत पिपासु था
लड़का क्या चाहता है	लड़का क्या चाहता है
शायद लड़की सो गयी	शायद लडकी सोना गया

Figure 5.1 to 5.5 shows output of Hindi Sentences on our proposed system screen.

Figure 5.1 Hindi to ISL mapping of Thirsty crow sentence 1

Figure 5.2 Hindi to ISL mapping of Thirsty crow sentence 2

Figure 5.3 Hindi to ISL mapping of Thirsty crow sentence 3

Figure 5.4 Hindi to ISL mapping of Thirsty crow sentence 4

Figure 5.5 Hindi to ISL mapping of Thirsty crow sentence 5

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Sign language is first language for communication for people who are deaf or take birth in dead families. Sign language is preferred language for communication for them and written language acts as secondary way of communication. As sign language is a complete natural language it becomes very important to build a system which can map text written in Hindi language to Indian Sign Language grammar so that it becomes ready for generation of signs. To convert Hindi text to Indian Sign Language Hindi Dependency Parser is used to provide us information about each word in sentence. Grammar rules are applied on the basis of information provided by dependency parser on each word. Parser has been integrated into Java Swings application and is being called using Process Builder class of java. Dependency Parser provides root word for each word. Words which do not pass the ISL grammar rules are removed from the word list of sentence. Each root word is searched in database to check if sign is available for the word. If sign is not available Hindi WordNet is called to generate synonyms for word. Each synonym is then checked in database to see if sign is available. If sign is available for a synonym that synonym replaces the original word in final ISL sentence. In case no synonym is found for which sign is available the word is marked as Finger Spelled. WordNet has been integrated to increase the efficiency of the system. The system has been tested on Hindi short story Thirsty Crow.

6.2 Limitations and Future Scope

- Mapping Hindi Text to Indian Sign Language system is a desktop based application.
- A mobile and web based version of the application will increase the reach to more people.
- The system needs to run on Linux based machine which further reduces the

References

[1]."Universal Networking Language(UNL):Relations" [Online] Available : http://www.unlweb.net/wiki/index.php?title=UniversalRelations [Accessed 10 June 2016]

[2]."Hindi Dependency Parser" [Online] Available: http://sivareddy.in/downloads [Accessed November 2015]

[3]."Dependency Parser " [Online] Available : http:// nlp.stanford. edu/software/ nndep.shtml [Accessed June 2016]

[4].Cox *et al*., "Tessa, a system to aid communication with deaf people" Proceedings of the fifth international ACM conference on Assistive technologies. ACM, 2002

[5].M. Mohandes, "Automatic translation of Arabic text to Arabic sign language" AIML Journal 6.4, pp: 15-19, 2006

[6].P. Kar *et al*., "Ingit: Limited domain formulaic translation from hindi strings to indian sign language", ICON, 2007

[7].T. Veale *et a*l., "The challenges of cross-modal translation: English-to-Sign-Language translation in the Zardoz system.", Machine Translation 13.1 pp: 81-106, 1998

[8].J.A. Bangham *et al.*, "Virtual signing: Capture, animation, storage and transmission-an overview of the visicast project", Speech and Language Processing for Disabled and Elderly People (Ref. No. 2000/025), IEE Seminar on. IET, 2000

[9].M. Huenerfauth, "Generating American Sign Language classifier predicates for English-to-ASL machine translation", Diss. University of Pennsylvania, 2006

[10].T. Dasgupta and A. Basu, "Prototype machine translation system from text-to-Indian sign language", Proceedings of the 13th international conference on Intelligent user interfaces. ACM, 2008

[11]."Thirsty Crow" [Online] Available: http://www.learning-hindi.com/post/17096412668 lesson-127-प-य-स-क-आ-the-thirsty-crow [Accessed March 2016]

[12].K. Dixit and A.S. Jalal, "Automatic Indian sign language recognition system", Advance Computing Conference (IACC)IEEE 3rd International.IEEE, 2013

[13].R. Kaur and P. Kumar, "HamNoSys generation system for sign language", International Conference on Advances in Computing, Communications and Informatics.IEEE, 2014

[14].Sign Languages of different countries [online] available at https://en.wikipedia.org/wiki/Sign_language [Accessed March 2016]

[15].What is a csv file [Online] available a https://support.bigcommerce.com/articles/Public/What-is-a-CSV-file-and-how-do-I-save-my-spreadsheet-as-one csv file wala

[16].Butt, Miriam, and Tracy Holloway King. *Lexical Semantics in LFG*. Center for the Study of Language and Inf, 2006. F-structre wala

[17].N. Suszczanska, P. Szmal, and J. Francik., "Translating Polish Texts into Sign language in the TGT System", the 20th IASTED International Multi-Conference on Applied Informatics, Innsbruck, Austria, pp. 282-287, 2002.

[18]. What is a jar file [Online] available at https://docs.oracle.com/javase/8/docs/technotes/guides/jar/jarGuide.html [Accessed March 2016].

[19]."IIT Bombay Hindi WordNet "[Online] available at , http://www.cfilt.iitb.ac.in/wordnet/webhwn/other/hwn_docs_2.pdf, [Accessed January, 2016]

[20].Process Builder Class [Online] available at, https://docs.oracle.com/javase/7/docs/api/java/lang/ProcessBuilder.html [Accessed February 2016]

[21].Zeshan, Ulrike. Sign language in Indo-Pakistan: A description of a signed language. John Benjamins Publishing, 2000

[22].Robert, S., 2010. "HamNoSys4.0 For Irish Sign language", Workshop Workshop Hand Book, Centre for Next Generation Localization, Version 3.0, Draft 2.0, Dublin City University, Iceland, pp 1-65

List of Publications

Research Paper

- P. Vij and P. Kumar, " Mapping Hindi Text to Indian Sign Language with extension using Wordnet" in International Conference on Advances in Information Communication Technology & Computing, AICTC, Bikaner, Jaipur, Rajasthan, India, ACM, 2016.

  [Accepted]

Video Presentation

Video for the thesis presentation can be seen at link:

https://www.youtube.com/dashboard?o=U