Project Report
Computer Science
Date: 2020-10-04

ES2ISL: An Advancement in Speech to Sign Language Translation using 3D Avatar Animator

Author:
Bhavin Patel
Harshit Patel
Nidhi Patel
Manthan Khanvilkar

Supervisor:
Dr. Thangarajah Akilan

A Research study submitted in fulfillment of the requirements for the degree of Masters in Computer Science

in the

Department of Computer Science
Lakehead Univeristy,
Thunder Bay, Ontario , Canada

EXCEPTIONAL. UNCONVENTIONAL.

Acknowledgement

We would like to express our sincere gratitude to our supervisor *Dr. Thangarajah Akilan* for providing his invaluable time, guidance and suggestions throughout the course. He gave us an incredible opportunity to work on the *Speech to Sign Language Translator* project, which kept us engaged with research in the particular subject and his personal generosity helped us to enjoy our time at *Lakehead University*.

We are overwhelmed in all humbleness and gratefulness to deeply acknowledge to all the team members who have helped us achieve these ideas, well above simplicity level into something concrete. *Manthan Khanvilkar's* encouragement have been valuable, and his early insights helped greatly accomplishing the project. *Nidhi Patel* who has helped us out with her abilities and without whose enthusiasm and support, we wouldn't have been able to pursue our goals. *Bhavinkumar Patel*, *Harshit Patel* for their equal contribution, involvement in the project and good humour.

Finally, we would like to acknowledge with gratitude,the support of our family members, whose value to us grows with age.

Milestone

Task name	Duration (days)	Start date	Finish date	Lag
Initiating	20	9/2/19	9/22/19	
Identify objective	10	9/2/19	9/11/19	
Determine project feasibility				
Determine scope	4	9/12/19	9/15/19	1
Early emperical research	6	9/17/19	9/22/19	
Presentation 1	1	9/23/19	9/23/19	3
Resource Gathering	24	9/27/19	10/20/19	
Data Gathering	7	9/27/19	10/3/19	
Filtering	2	10/4/19	10/5/19	
Identify resources	3	10/6/19	10/8/19	
Study Existing system	12	10/9/19	10/20/19	
Presentation 2	1	10/21/19	10/21/19	4
Analyse existing project	16	10/26/19	11/10/19	
Presentation 3	1	11/11/19	11/11/19	
Implementation	31	11/12/19	12/12/19	
Prototype: design & specifications	9	11/12/19	11/20/19	
Formulation of algorithm	15	11/21/19	12/5/19	
Integrating System	7	12/6/19	12/12/19	1
Class Group Meeting	1	12/6/19	12/6/19	
Execution	13	12/14/19	12/26/19	
Testing system	2	12/14/19	12/15/19	
Fix issues & Modification	10	12/16/19	12/25/19	
Testing modified system	1	12/26/19	12/26/19	
Conference Paper	23	12/14/19	1/15/20	
CCECE paper first draft	3	12/14/19	12/16/19	
Improving Efficiency of the system	6	12/17/19	12/22/19	
Second draft	11	12/23/19	1/2/20	
Proofreading	1	1/3/20	1/3/20	10
Creating test samples &github link	2	1/14/20	1/15/20	
Improvement of Proposed Model	8	1/16/20	1/23/20	
Improving efficieny and performance	8	1/16/20	1/23/20	
Presentation 5	1	1/24/20	1/24/20	
Summary Report	1	1/25/20	1/25/20	
Enhancement of Proposed Model	7	1/26/20	2/09/20	
Adding more features to proposed model	15	1/26/20	2/09/20	
Presentation 6	1	2/10/20	2/10/20	1
Conference Paper	15	2/12/20	2/26/20	
CCECE paper third draft	15	2/12/20	2/26/20	
Preparation for Poster Presentation	4	2/27/20	3/1/20	
Presentation 7	1	3/2/20	3/2/20	
Poster Presentation	1	3/4/20	3/4/20	4
Conference Paper	6	3/3/20	3/8/20	
CCECE paper final draft	6	3/3/20	3/8/20	6
Increasing model's test samples	3	3/15/20	3/17/20	
Presentation 8	1	3/23/20	3/23/20	2
Draft Report	7	3/26/20	3/31/20	2
Final Report	7	4/3/20	4/9/20	

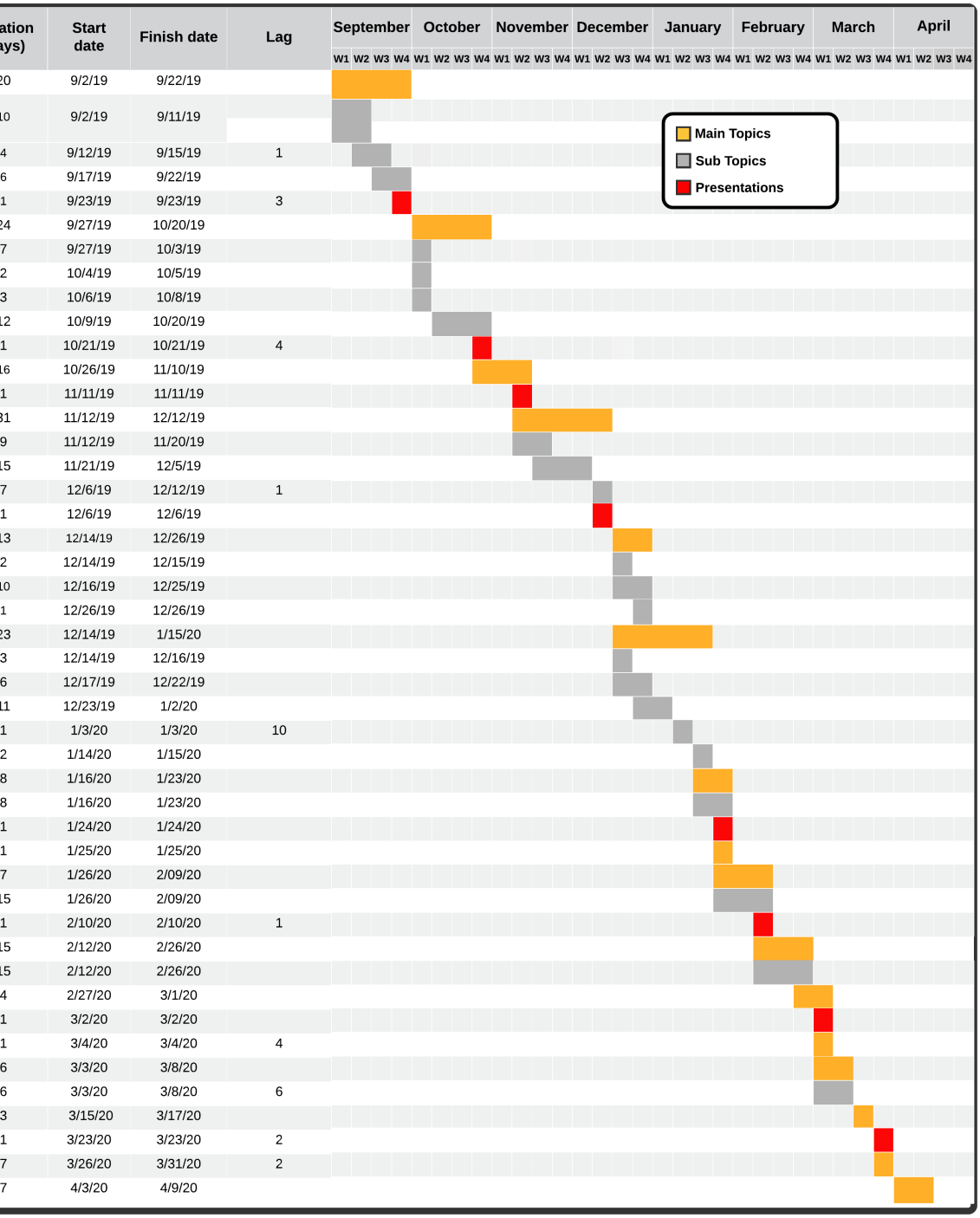


Figure 1: Gantt chart representing the milestones achieved.

Abstract

Sign languages are languages which convey meaning by using the visual-manual modality. It is a language that uses signs made with hands and other gestures, including facial expressions and body postures, mainly used by people who are hearing impaired to fluidly convey their thoughts. It is very important for such people to have access to a sign language for for their social, emotional and linguistic growth.

This work proposes a model and an initial implementation of a robust system, which converts English Speech into Indian Sign Language (ES2ISL) animations. Such system may considerably enhance the lives of hearing-impaired people, especially in interaction and information exchange between concerned parties. The core purpose of the system is to bridge the communication gap between hearing-impaired people in India and others. It exploits and integrates the semantics of the Natural Language Processing (NLP), Google cloud speech recognizer API, and a predefined sign language database. The experimental results show that the proposed system outperforms existing models with an average accuracy of 77%. Hence, it overshadows the existing systems in terms of processing time by taking about $0.85s$.

Keywords: Speech recognition, sign language, NLP, ISL.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

World Health Organization (WHO) reports that there are 466M hearing-impaired people worldwide, which is about 6% of the entire world population [1]. Approximately one-third of people over the age of 65 years are affected by hearing loss. Sign language is a medium of communication used for/by hearing-impaired people, whereby arms, fingers, facial expressions, motions, and other parts of the body are used to convey messages symbolically. It is a visual-spatial language as the signer explains an occurrence using 3D space around his/her body. The early days of sign languages did not have a well-defined structure or grammar; thus, they were not or very less permissible outside their communication domain. Similar to the fully developed American Sign Language (ASL), the Indian Sign Language (ISL) has also been developed with grammar, syntax, and linguistic attributes.

According to the 2018 census, in India, about 6% of the total population, i.e., 63M, suffer from significant hearing loss [1]. From these people, most of the Indian hearing impaired have no language knowledge, either signed, spoken, or written explanation. Such a low literacy rate is caused by the following factors: (i) lack of sign language interpreters, (ii) lack of software tools, and (iii) missing ISL research. Due to communication inability, impaired people face significant setbacks in public places, such as railways, banks, and hospitals. It urges the researchers to design a system for human voice to sign language translation that to help the hearing-impaired people communicate better with the rest of the world.

Thus, this work proposes a Web-based speech to sign language translator. The proposed system, ES2ISL, performs better than the existing ones in terms of accuracy, processing time, as well as memory utilization. It gives an average accuracy of 77% concerning voice recognition, grammar parsing, and avatar action. Moreover, the system is capable of generating the output within $1s$ per conversion by minimizing the operational processing time by nearly 69% of the existing model. It also saves a significant amount of memory, unlike the existing systems,

which have different Signing Gesture Markup Language (SIGML) files for different vocabulary. However, the current system has only a single JSON file, wherein all the vocabularies, along with its content of all SIGML files are stored.

1.0.1 ISL Grammar

The Indian Sign Language has its own syntax, like other sign languages. It does not rely on the English or Hindi spoken language and has distinct manual representation. It This has some special features, such as:
i. Each number is represented by a sign with a hand gesture. Eg. The sign for number 45 will be represented with 4 followed by the 5.
ii. Signs for relationship are anticipated as' male / men' and ' female / woman '. The interrogative terms such as 'how', 'where', 'which' etc. are indicated at the end of the sentences.
iii. The ISL contains various non-manual gestures including mouth gestures, mouth pattern, facial expressions, postures of the body, head position and eye glance. ISL mainly has a word order for subject-object-verb (unlike subject-verb-object in English).

1.0.2 HamNoSys and SiGML

HamNoSys: Sign language does not have a particular written format. To define a sign, a notation system must be in place that can help to write signs. Another such system is the HamNoSys(Hamburg sign language notation system). Signing actions are translated using phonetic transcription system. HamNoSys is a syntactic representation of a sign which provides the processing of signs by machine. HamNoSys has its origins in the Stokoe notation system which implemented an alphabetic system to describe the sub-lexical parameters including hand position, hand configuration and hand movement to provide a phonological description of the signs.

SiGML Language: SiGML is an XML framework that allows the transcription of sign language gestures. SiGML is the markup language signing motions. The symbol HamNoSys is specified in the form of XML tags. It was founded in East Anglia University. It provides communication tools in the form of animated characters. SiGML representation from HamNoSys notation is input to 3D rendering software.

Chapter 2

Related Work

2.0.1 Voice to Sign Language Translators

Foong *et al.* [2] suggested a voice (English language) to sign language translation system for Malaysian hearing-impaired people using speech and image processing techniques. The system achieves a recognition rate of 80.3%. Similarly, Segundo *et al.* [3] proposed Spanish to sign language (LSE) translation system for Spanish people applying or renewing their Identity cards. The system uses the rule-based approach resulting in a 32% of Sign Error Rate (SER) and 58% Bi-Lingual Evaluation Understudy (BLEU) score. Abbas *et al.* [4] developed a framework to convert speech to Pakistani Sign Language (PSL) with bilingual subtitles. Lopez *et al.* [5] describes a new version of Spanish into LSE with new tools and features, which make the system more adaptable to any new semantic domain. The whole translation system has SER reduced to less than 10%, and BLEU score higher than 90%. For the advances in the Mexican speech-to-sign language (MSL) translator, Trujillo *et al.* [6] proposed Mexican automatic speech recognizer to convert Mexican speech to MSL with an accuracy of 97%. A mobile-based interpreter developed by Rekha *et al.* [7] for ISL in which the input voice is taken on a mobile device and then translated into a text message and stored into a cloud database, then the text message is translated to sign language.

2.0.2 Existing Models

An interesting 3D avatar technology presented by Adamo-Villani *et al.* [8]. It is an interactive virtual environment, where hearing-impaired children would communicate with a 3D virtual signers and objects to learn mathematical concepts. Cox *et al.* [9] developed a system called TESSA based on a direct translation approach. It allows communication between a hearing-impaired person and a post office clerk through British Sign Language (BSL) translation model. TESSA takes English as an input text, scans each English string word in the Text-to-Sign dictionary, blends these signs, and incorporates them into an animation.

Safar and Marshal *et al.* [10] introduced a model known as VisiCast Translator, which translates English text into BSL. It uses semantic representation level to perform BSL generation from the English text. Similarly, the Voice-Activated Network Enabled Speech to Sign Assistant (VANESSA) was implemented by Glauert *et al.* [11] to facilitate communication between assistant and their hearing impaired clients in the UK council information cents or similar environments.

SignSynth is a CGI-based articulatory sign synthesis prototype produced by Greive-Smith and Angus [12] at New Mexico University. SignSynth takes a sign language text in ASCII-Stokoe notation as its input, and converts it to an internal structure tree. This linguistic representation is then converted into a 3D animation sequence in a Virtual Reality Modelling Language(VRML or Web3D) which is automatically rendered by a Web3D browser.

Purshottam *et al* [13] presented a prototype system Indian Gestural Interaction Translator(INGIT). It is a cross-model translation system from Hindi strings to Indian Sign Language (ISL) for use in Indian Railways reservations counters. INGIT adopts a semantically mediated formulaic framework for Hindi-ISL mapping. This converts the reservation officer's input into an Indian sign language, which can then be displayed to the ISL-user. INGIT currently accepts spoken-language strings transcribed as input and generates ISL string that are translated through HamNoSys simulation to a graphical display.

Dasgupta and Basu [14] developed a machine language translation system for translating English text-to-ISL.The system helps to distribute awareness to the hearing-impaired people in India. It also presents a Sign Language dictionary tool that can create a bilingual ISL dictionary and store phonological data on ISL.

Divanshu Singh [15] and his team developed Text to Sign Language Translator (T2SLT), which accepts English text as an input to generate avatar-based sign language animation. T2SLT searches for the matching SIGML file to depict the action associated with it. The proposed ES2ISL translator is an advancement of the existing T2SLT [15], as described in Chapter 3.

Chapter 3

Proposed Model

3.0.1 System Architecture

The proposed system subsumes three stages with following sub-modules: Stage I {Speech to text conversion, Input parser}, Stage II {Data preprocessing, ISL generator,Word→HamNoSys→SIGML}, and Stage III {SIGML notation, Graphic generator} as depicted in Fig. 3.1.

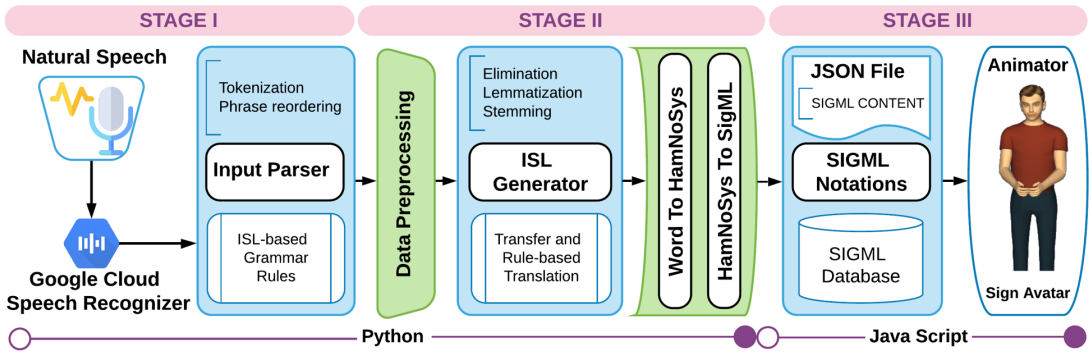


Figure 3.1: Proposed ES2ISL. A Tri-stage Operational Flow: Speech to Text Conversion, Data Preprocessing, and Word to SIGML.

3.0.2 Stage I: Speech to Text Conversion

This module uses an external or built-in microphone of any Personal Digital Assistant (PDA) to receive input from the user. It uses Google cloud speech API for converting the audio signal into text. Note that, only the Chrome and Firefox browsers support the speech-to-text API.

Verb Pattern	**Rule**	**Input Sentence**	**Parsed Sentence**	**Output Sentence**
Subject + Verb	NP V	birds fly	[NP(NNS birds)] [VP(VBP fly)]	birds fly
Verb + Object	VP NP	go school	[VP(VB go)] (NN school)	school go
Subject + Verb + Subject complement	NP V NP	his brother became a soldier	[NP(PRP$ his) (NN brother)] [VP(VBD became) (NP(DT a)) (NN soldier))]	his brother a soldier became
Subject+ Verb+ indirect object+ direct object	V NP PP	show your hands to me	[VP(VB show)] [NP(PRP$ your) (NNS hand)] [PP(TO to)] [NP(PRP me)]	your hand to me show

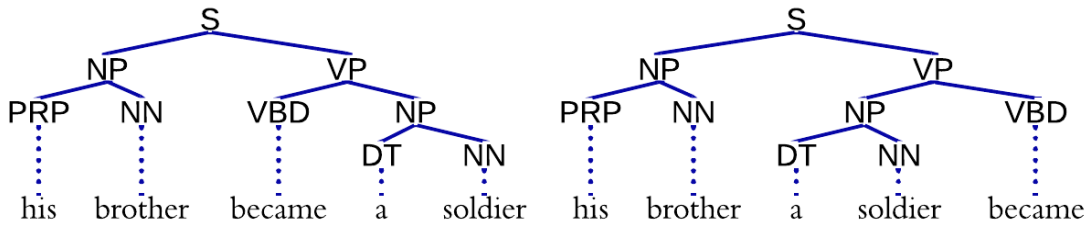
Table 3.1: Grammatical Reordering of English Language Phrases to Hindi (Indian) Language Phrases.

3.0.3 Stage I: Input Parser

This module takes the text string from the above module and splits it into sentences. Followed by this, each sentence is tokenized into words using Machine Learning (ML) and NLP tools. The output from this module is a list as the following example.
Input (*text*): His brother became a soldier.
Output (*tokens*): 'His', 'brother', 'became', 'a', 'soldier', '.'
These tokens are parsed to create a structure of phrases using Stanford parser. Stanford parser is capable of producing three different outputs, part-of-speech tagged text, context-free grammar representation of phrase tree, and type dependency representation as shown in Fig. 3.2 and Table 3.1 based on their grammatical structure.After the source language is translated to phrase tree as shown in Fig. 3.2a, the proposed ES2ISL applies the ISL grammar to modify the phrase tree such that the modified tree shown in Fig. 3.2b now represents the structure based on ISL grammar.

3.0.4 Stage I: Rule for conversion from English to ISL:

Translating the spoken language into another spoken language is a difficult task if the rules vary from one to another language. For conversion of English sentence to a sentence as per ISL grammar rules, all the verb patterns and rules are formed to convert English sentence into ISL sentence. The parsed sentence is the input to this module where the noun phrase and the prepositional phrase are fixed but if there is any verb phrase present in the sentence, it is checked recursively because the verb phrase may further be composed of noun phrase, prepositional phrase,



(a) English Language Phrase Tree (b) Hindi (Indian) Language Phrase Tree

Figure 3.2: Phrase Tree Transition.

verb phrase or even a new sentence. Table 3.1 shows some of the rule conversion.

3.0.5 Stage II: Data Preprocessing

1. Eliminator: ISL sentences are composed of main words and no linking words, suffixes are used. To comply with the ISL rules, the ES2ISL omits the linking verbs (*am, is, are, was, were*) and articles (*a, an, some, the*). Moreover, unwanted words are removed including various parts of speech such as **TO**, **MD** (modals), **POS** (possessive ending), **FW** (foreign word), **CC** (coordinating conjunction), **NNPS** (nouns plural, proper plural), **IJ** (interjection), **SYM** (symbols) and non-root verbs.
Input[*ISL sentence*] : 'his', 'brother', 'a', 'solider', 'became'
Output[*Elimininiator*]: 'his', 'brother', 'solider', 'became'

2. Stemming: The ISL does not use suffixes or words with gerunds (-ing). Every word in ISL is present in its root form. The purpose of stemming is to reduce inflectional forms and derivationally related forms of a word into a common base form. For example, a *tokenset* {*wants, wanted, wanting*} has its *root* form of *want*. This is achieved through the Natural Language Tool Kit (nltk) library. It is often useful for quick recall and selection of search queries inside information retrieval (IR) environments. Environmental documents are presented on a standard IR as a word or word vectors. Words with the same stem have the same value. An example of Stemming process is given below.
Input[*tokens*]: 'I', 'am', 'playing', 'basketball'
Output[*tokens*]: 'I', 'play', 'basketball'

3. Lemmatization: Lemmatization eliminates the inflected words to ensure that the root word is part of the language, as opposed to Stemming. This work utilizes the WordNet Lemmatizer from the NLTK library. The proposed ES2ISL lemmatizes every token wrt parts-of-speech tagging in order to improve the accuracy of a word to root(lemma) conversion. An example of lemmatization is given below.
Input[*tokens*]: 'He', 'was', 'running', 'and', 'eating', 'at', 'same', 'time'

Output[*tokens*]: 'He', 'be', 'run', 'and', 'eat', 'at', 'same', 'time'

3.0.6 Stage II: ISL Generator

This module is intended to covert tokens from the above output to Indian Sign Language as described below

1. Transfer-based conversion: The proposed system accepts tokens and analyses it syntactically and semantically. Then, the text is translated into sign language. Here, the source language is converted into an intermediate abstract form, and then certain rules and tools are applied to derive the sign-language translation.

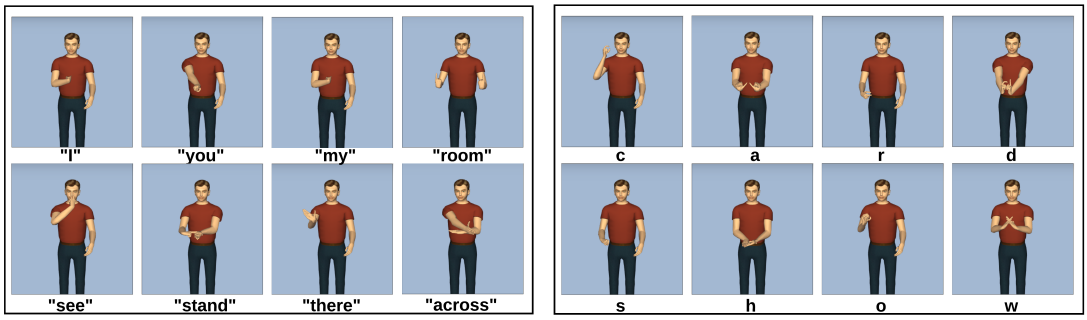
3.0.7 Stage II: Word→HamNoSys→SIGML

1. HamNoSys Generation Tool: The Hamburg Sign Language Notation System (HamNoSys) is a direct correspondence for all sign languages, not only ASL, similar to the International Phonetic Alphabet (IPA) for oral languages, between symbols and a sound transcription system. HamNoSys does not affiliate with any particular national diversified finger-spelling scheme. So, it can be implemented globally. The HamNoSys is generated using the ESIGN editor tool. The tool provides a database of signs with their respective HamNoSys notation. However, new signs can also be created if their HamNoSys notations are known. The further details of Esign Editor tool along with its usage is described in Appendix A. These notations are then stored in the database. The output of the ISL generator are the tokenized words and characters. The associated HamNoSys descriptions of the words are retrieved from the database. Further, these HamNoSys are used to look up the associated SIGML files.

3.0.8 Stage III: SIGML Notation

1. SIGML representation tool: It is necessary to convert the HamNoSys notation to its SIGML notation for the avatar to depict the sign movement of a particular word. These SIGML Notations are the textual representation in the form of XML-based encoding that enables sign language gestures to be transcribed. HamNoSys HPSG editor tool is used for this purpose. More details about HPSG tool can be found in Appendix A.

These notations are stored in SIGML files, which can later be retrieved using HamNoSys descriptions. The contents of each SIGML file are stored in one JSON file using a PHP script, which is later retrieved for the associated words or characters. Due to the dictionary limitation, the system handles the situation with the following cases:
Case 1: Having a word that matches with the SIGML file.



(a) Word-level parsing (b) Character-level parsing

Figure 3.3: Sample Hindi (Indian) Sign Language Animations Produced by the Graphic Generator as Sequence of Frames.

Case 2: Having a word that does not match with the existing SIGML dictionary. *Case* 1, is a straightforward condition since there is a corresponding SIGML file found in the database. For *Case* 2, the algorithm tokenizes the word into individual character. Then, it matches it with the corresponding character SIGML file and appends all the following SIGML files to generate the signs for that word. For example, for an input bird, the output would be generated by calling the *bird.sigml* file. Whereas, for the input cake, the module does not have an associated *cake.sigml* file. Thus, it calls the character-based individual *".sigml"* files: *c.sigml + a.sigml + k.sigml + e.sigml.*

3.0.9 Stage III: Graphic Generator

The graphic generator receives the corresponding SIGML files from the above stage and generates the correct avatar animations. Two examples of Hindi (Indian) sign language animations generated by this module for word-level parsing with an input English speech I see you standing there across my room and for character-level parsing with an input English speech show card are shown as sequence of frames in Fig. 3.3.

3.0.10 Audio to Sign Language Translator

Prior to the development of ES2ISL, the existing T2SLT [15] is improved to Audio to Sign Language Translator (A2SLT) such that the new system can accept speech as an input instead of text. Moreover, it carries out the grammar parsing of ISL. However, this model consumes more time due to the additional grammar parsing process. It also has a problem of mismatching between the performed animations and the parsed word or character levels. In response to the above issues, the ES2ISL is developed as a fully improved version of the T2SLT and A2SLT.

Chapter 4

Implementation

4.0.1 Website

Main features

Speech to Sign Language conversion.
Uses Text as an intermediate.
Tokenization text using ISL rules.
Web based interface (no installs necessary).
Client/Server Architecture.
Learning: Uses NLP tools.
Fully responsive: Everything is responsive ready so need to worry about how this site will look on mobile, tablet, and desktop.
Easy to use and user-friendly

Tools:

Front-End: SIGML url app, Sublime Text
Back-End: Pycharm, Xamp Apache Server

Languages

Front-End: HTML 5, CSS, SIGML.
Back-End: JavaScript, PHP, Python.

4.0.2 UML Diagrams:

A UML diagram is the Unified Modeling Language diagram that describes a system in a visual fashion with its key actors, roles, actions, artifacts or classes, so that details about the system can be more easily interpreted, updated, preserved and recorded.

Class Diagram:

Class Diagram in UML is a static structure diagram that describes the structure of a system with classes, attributes, operations (or methods) of the system and relationships between the objects.

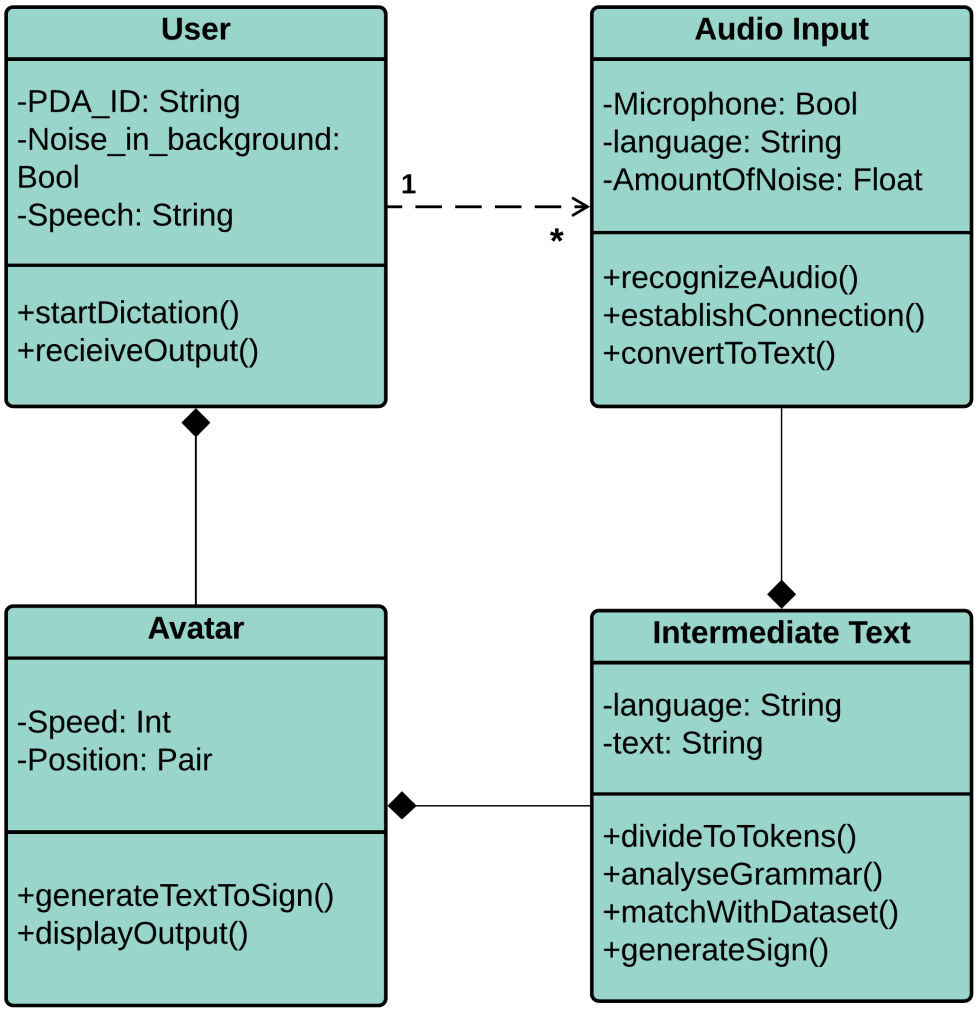


Figure 4.1: Class diagram of ES2ISL.

Fig. 4.1 represents the class diagram of ES2ISL model. Here single user can give multiple voice input to the system which is represented as 1 to *. The (-) sign before the attributes represent that the attributes are private whereas (+) sign before the functions/methods represent the public scope of those functions. The class User is dependent on class Audio Input for taking the voice as an input. Further, each of these classes such as Audio Input, Intermediate Text and Avatar are composite of each other meaning if any of these classes fail to exist, the other class adds no meaning to the system or cannot function.

Activity Diagram:

Activity Diagram is a behavioral diagram in UML diagram to illustrate the complex aspects of the system. It is an improved version flow chart that models the flow from one activity into another.

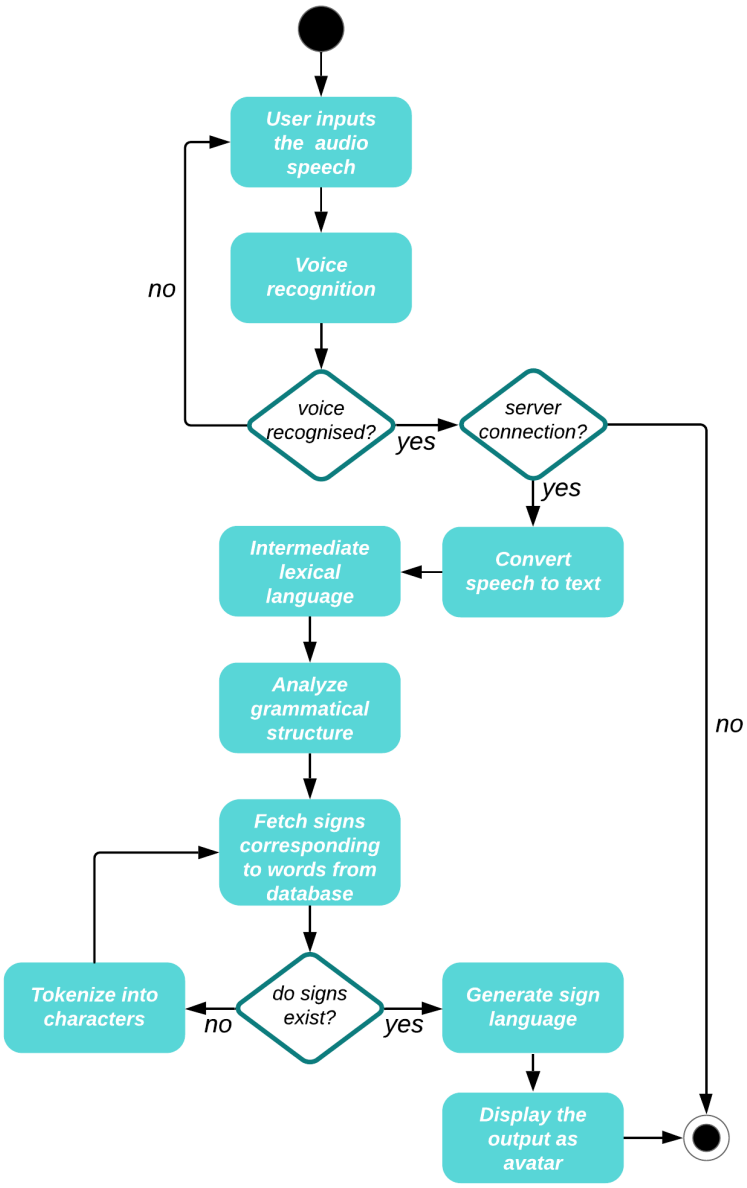


Figure 4.2: Activity diagram of ES2ISL.

The process flow of the ES2ISL model is depicted in Fig. 4.2. It also represents certain conditions or decisions, which decides if the system is allowed to proceed further or if it is terminated. For example, if the system is unable to connect to the server, it will suspend automatically or if the system is unable to recognize

the voice, it will loop back to the voice input. Similarly, the process continues from one activity to another till system completes all of its activities.

Use Case Diagram:

A case diagram at its simplest is an illustration of the user's interaction with the system that illustrates the user's relationship with the different use cases affecting the user.

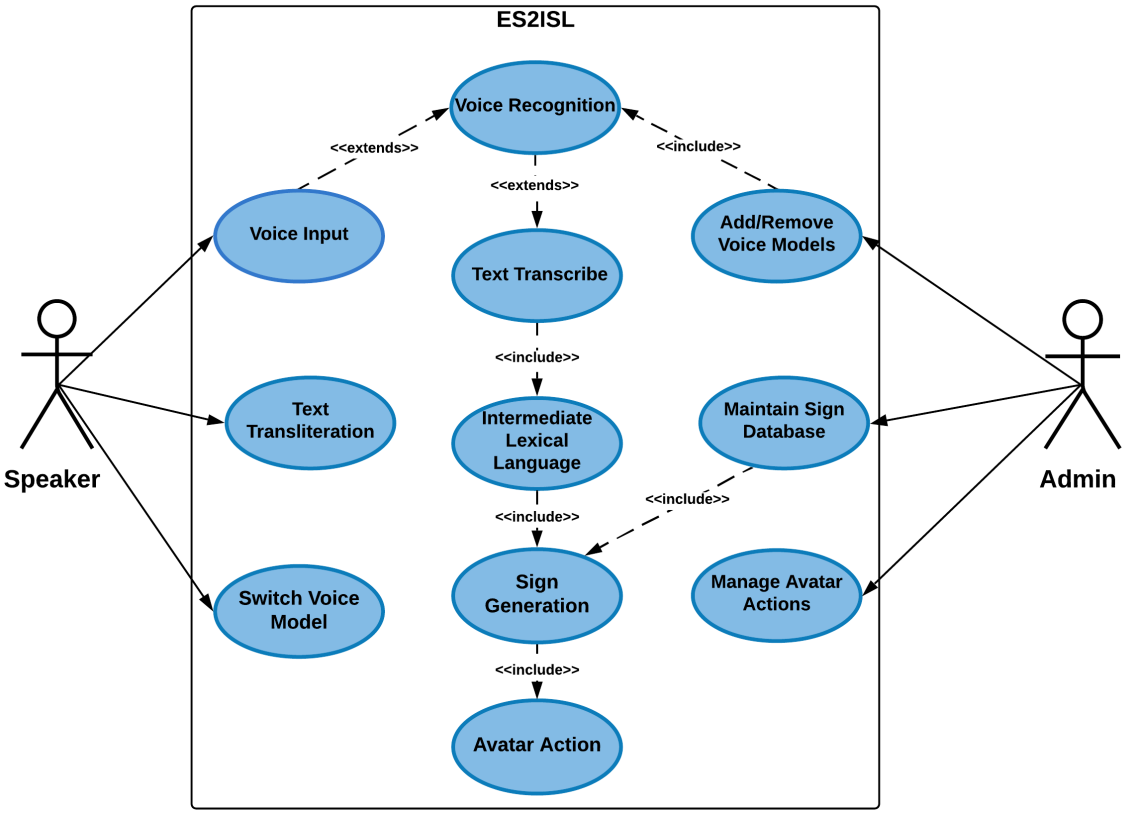


Figure 4.3: Use Case diagram of ES2ISL.

Fig. 4.3 represents the use case diagram for ES2ISL model. It depicts the actions performed by user(speaker) as well as admin. Actions such as giving the voice input, transliteration of the text as well as changing the voice model can be performed by user whereas adding or removing the voice models, managing the sign database as well as avatar actions can be performed by admin. Here some of the process are inherited or extended by other process. Process such as transcribing the text from audio is extended from audio output after the voice is recognized. Similarly, the sign generation is inherited from sign database. Overall, the use case summarizes some of the relationships between use cases, actors, and systems.

Chapter 5

Experimental Study

5.0.1 Dataset

The data samples are collected from an existing project - Text to Sign Language Translator [15]. The original number of samples in the database of T2SLT is limited to 1113 English language words. However, the proposed system is updated with an extended corpus of 3268 English language words. Further, there is a look up table (LUT)-like JSON script is written to store all the words and their respective SIGML files for an efficient retrieval.

The experimental study takes 101 natural speech inputs spoken by a human in real-time. The speaker utters some short and long speeches, like: 'How are you?' , 'I am your assistant.', 'The stars look fascinating at night.', and 'The bag of ice melted quickly than expected.' evaluate the proposed model.

5.0.2 Performance Evaluation

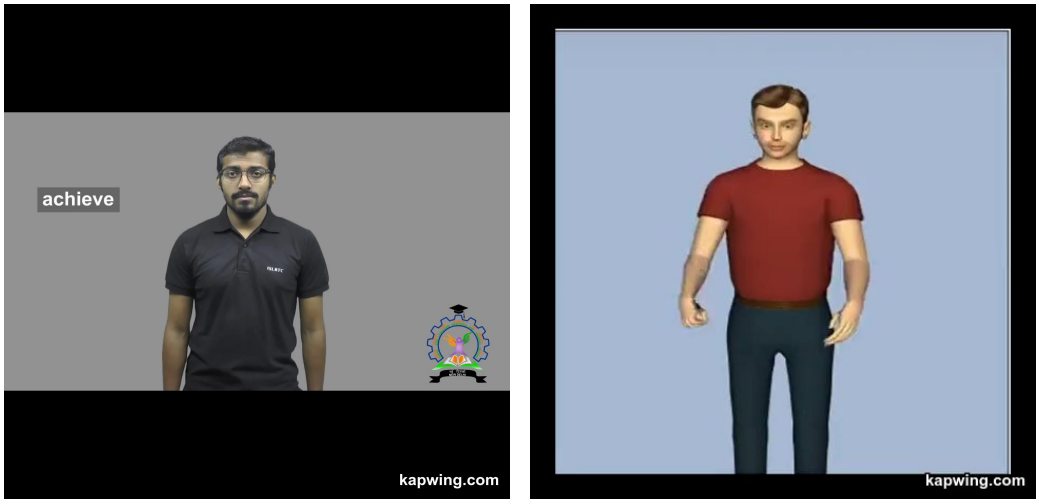
The performance evaluation is carried out through performance comparison of the performance of the proposed ES2ISL with the existing system, T2SLT [15], and the A2SLT. The performance is measured based on the following metrics: accuracy of the system in terms of voice recognition (V_a), grammar parsing (P_a), and action performed (A_a) by the avatar defined as

$$V_a = (\alpha \times 100)/N, \tag{5.1a}$$

$$P_a = (\beta \times 100)/N, \tag{5.1b}$$

$$A_a = (\gamma \times 100)/N, \tag{5.1c}$$

where α, β, γ, and N stands for the correctly recognized voice, parsed grammar of sentence and performed action by the avatar, and the total number of test samples respectively. The correctness of voice recognition is determined by the spoken word/sentence and input taken by the model. Both of the words or sentences are



(a) Human depicting action in real time. (b) Avatar depicting action in real time.

Figure 5.1: Video showing comparison of actions performed by human and avatar for word "achieve" in real time.

then compared to find the similarity between them. The spoken sentence/word is also determined by its confidence level. Similarly the grammar parsing is determined by the structure of the sentence in ISL. Since the transliteration is also available, both the parsed sentence and its transliteration are compared to find the correctness of parsed grammar. The action performed by the avatar is determined by comparing the avatar action with the source. The source is taken from Indian Sign Language Research and Training Centre(ISLRTC) webiste [16]. As described in Fig. 5.1, the comparison of the action for word "achieve" is done to find the authenticity of the action performed by the avatar in real time. The sequence of frames for the the video is given in Appendix B. Since the action matches with the verified source, it is considered to be the action correctly performed. Similarly, the failed test cases have also been taken into consideration to find the accuracy of the system.

5.0.3 Processing Time Analysis

This analysis is carried out under a real-time environment with a network speed of 12 $Mbps$, 2.4 GHz network bandwidth, 8 GB RAM, and 1.60 GHz $i5-8250U$ CPU on a machine. The time complexity is measured for two conditions: (i). time taken after voice recognition to perform an action (T_S) and (ii). time taken between two words/characters to perform an action (T_W).

Chapter 6

Results

6.0.1 Performance Evaluation.

Evaluation		T2SLT [15]	A2SLT	ES2ISL
Accuracy	V_a (%)	NA	60	75
	P_a (%)	NA	72	72
	A_a (%)	50	65	85
Time Complexity	T_s (s)	5.35	5.20	1.00
	T_w (s)	2.35	2.20	0.70

Table 6.1: Performance Analysis: V_a, P_a, and A_a stand voice recognition, grammar parsing, and avatar animation accuracy respectively. T_S, and T_W stand for time taken for performing an action after speech recognition and an action between two words/characters respectively.

It can be seen from the comparisons given in Table 6.1 and Fig. 6.1 that the overall performance of the proposed ES2ISL is quite remarkable. The T2SLT is based on text input, so it lacks the feature of voice recognition as well as parsing grammar. Further, the A2SLT recognized 61 sentences correctly out of 101, which gives an accuracy of 60%, whereas the ES2ISL recognizes 76 sentences correctly, improving the accuracy to 75%. In terms of grammar parsing, the A2SLT functions well. It does 73 out of 101 correct sentence grammar parsings with an accuracy of 72%. Similarly, the ES2ISL achieve the same accuracy of 72% in grammar parsing. In terms of actions performed, the T2SLT has a limited corpus of SIGML files that bottleneck brings down its accuracy to 50%. However, the A2SLT has a big corpus. Due to that, it achieves a better accuracy of 65%. When compared to the other two models, the ES2ISL leads the race with 85% in terms of actions performed.

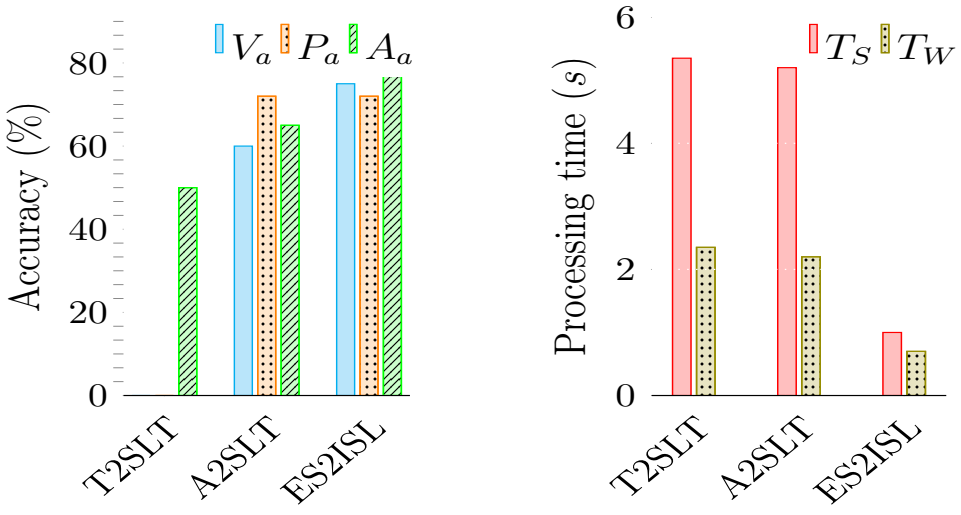


Figure 6.1: Performance Analysis: V_a, P_a, and A_a stand voice recognition, grammar parsing, and avatar animation accuracy respectively. T_S, and T_W stand for time taken for performing an action after speech recognition and an action between two words/characters respectively.

As it can be seen from the comparisons tabulated in Table 6.1 and Fig. 6.1, the proposed system ES2ISL has a quicker response compared to the other models. For performing an action after voice recognition, the systems: T2SLT, A2SLT, and ES2ISL take around $5.35s$, $5.20s$, and $1s$ respectively to generate the output. Thus, the proposed system saves 84% of valuable processing time. On the other hand the T2SLT and A2SLT consumes $2.30s$ to perform an action between two words or characters, whereas ES2ISL is capable of generating the output within $70ms$ achieving a saving of 69% of valuable computational time. In general, the proposed ES2ISL is a highly optimized and quicker responsive model for hearing-impaired people.

Chapter 7

Conclusions and Future Work

This paper presents an interactive system for hearing impaired people for impactful communication. It transforms English speech into a 3D avatar animation that portrays signs of Hindi(Indian) language rather than GIFs, pictures, or videos for handling the memory effectively. It generates a realistic and vibrant appeal of animations.

The future work lays in the direction of improving the current model with a custom speech recognition system than Google API. Moreover, the model will have support for various sign languages such as ASL and BSL. It will also be implemented using different parsing techniques as well as support for two way communication using server-client architecture will also be implemented.

References

[1] S. Davey, C. Maheshwari, S. K. Raghav, N. Singh, K. Muzammil, and P. Pandey, "Impact of indian public health standards for rural health care facilities on national programme for control of deafness in india: The results of a cohort study," *Journal of family medicine and primary care*, vol. 7, no. 4, p. 780, 2018.

[2] O. M. Foong, T. J. Low, and W. W. La, "V2s: Voice to sign language translation system for malaysian deaf people," in *International Visual Informatics Conference*. Springer, 2009, pp. 868–876.

[3] R. San-Segundo, R. Barra, R. Córdoba, L. D'Haro, F. Fernández, J. Ferreiros, J. M. Lucas, J. Macías-Guarasa, J. M. Montero, and J. M. Pardo, "Speech to sign language translation system for spanish," *Speech Communication*, vol. 50, no. 11-12, pp. 1009–1020, 2008.

[4] A. Abbas and S. Sarfraz, "Developing a prototype to translate text and speech to pakistan sign language with bilingual subtitles: A framework," *Journal of Educational Technology Systems*, vol. 47, no. 2, pp. 248–266, 2018.

[5] V. López-Ludeña, R. San-Segundo, C. G. Morcillo, J. C. López, and J. M. P. Muñoz, "Increasing adaptability of a speech into sign language translation system," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1312–1322, 2013.

[6] F. Trujillo-Romero and S.-O. Caballero-Morales, "Towards the development of a mexican speech-to-sign-language translator for the deaf community," *Acta Universitaria*, vol. 22, pp. 83–89, 2012.

[7] K. Rekha and B. Latha, "Mobile translation system from speech language to hand motion language," in *2014 International Conference on Intelligent Computing Applications*. IEEE, 2014, pp. 411–415.

[8] N. Adamo-Villani, E. Carpenter, and L. Arns, "3d sign language mathematics in immersive environment," *Proc. of ASM*, pp. 2006–2015, 2006.

[9] S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M. Wells, M. Tutt, and S. Abbott, "Tessa, a system to aid communication with deaf people," in *Proceedings of the fifth international ACM conference on Assistive technologies*. ACM, 2002, pp. 205–212.

[10] T. Hanke, I. Marshall, E. Safar, C. Schmaling, G. Langer, and C. Metzger, "Interface definitions," *ViSiCAST Deliverable D5-1*, vol. 14, p. 2006, 2001.

[11] J. Tryggvason, "Vanessa: A system for council information centre assistants to communicate using sign language," *School of Computing Science, University of East Anglia*, 2004.

[12] A. B. Grieve-Smith, "Signsynth: A sign language synthesis application using web3d and perl," in *International Gesture Workshop*. Springer, 2001, pp. 134–145.

[13] P. Kar, M. Reddy, A. Mukherjee, and A. M. Raina, "Ingit: Limited domain formulaic translation from hindi strings to indian sign language," *ICON*, vol. 52, pp. 53–54, 2007.

[14] T. Dasgupta and A. Basu, "Prototype machine translation system from text-to-indian sign language," in *Proceedings of the 13th international conference on Intelligent user interfaces*, 2008, pp. 313–316.

[15] D. Singh, "Text to sign language," *Thapar University*, 2016.

[16] ISLRTC, "http://islrtc.nic.in/," 2015.

Appendices

Appendix A

Esign Editor and HPSG tool

A.0.1 About Esign editor

The Esign editor tool allows users to write the signed text to be performed by the Esign avatar. It provides the user with an economic approach by choosing sign sequences from the lexicon and, if necessary, altering them with the aid of specialist editors (which concentrate on various aspects of the sign's phonetics morphology). The user can instantly verify the content by sending the text to the Avatar while "writing" the signed text. The tool is supported in three languages (German, English, German and Dutch) for Windows 32/64 bit and MacOS platforms.

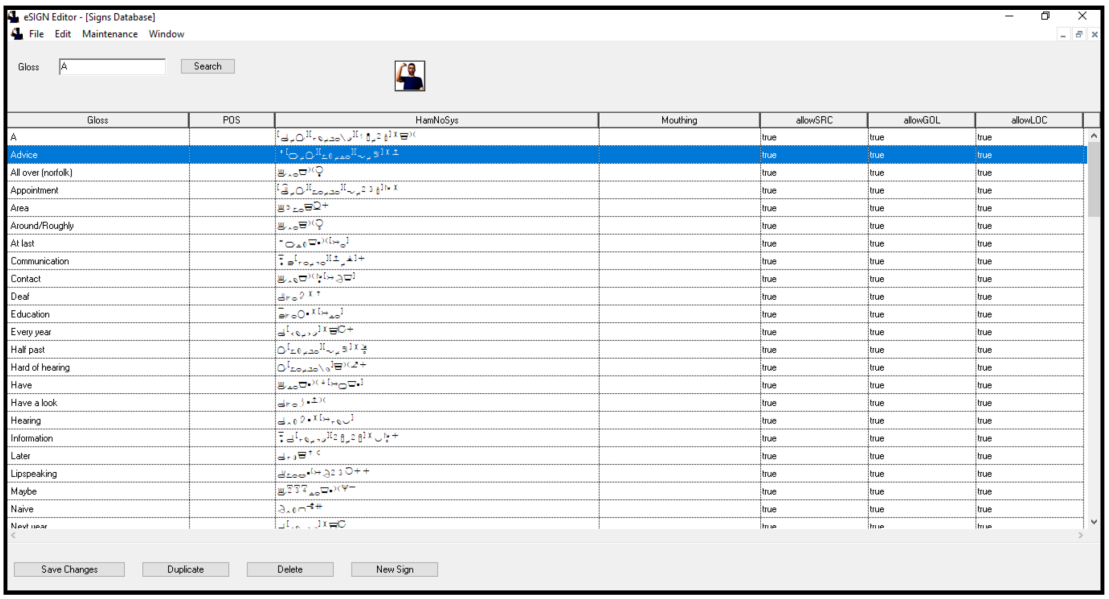


Figure A.1: Interface of Esign editor tool containing sign database representing the words with their HamNosys.

Usage of Esign editor

Fig. A.1 represents the interface of Esign editor tool which contains a large sign database. This database incorporates various words along with its HamNoSys representation. For example, the HamNoSys for the word "advice" is represented in HamNoSys column of Esign editor tool which is shown in Fig. A.1. This HamNoSys representation can be further used to generate Sigml representation as shown in HamNosys section.

A.0.2 About HPSG tool

The Head-driven phrase structure grammar (HPSG) tool facilitates working with HamNoSys in the context of HPSG environments that normally simply ignore fonts and therefore cannot display feature values appropriately. Therefore, this utility can convert back and forth between HamNoSys strings and ASCII representations. The ASCII representation of a HamNoSys symbol is its name as defined in the SiGML context.

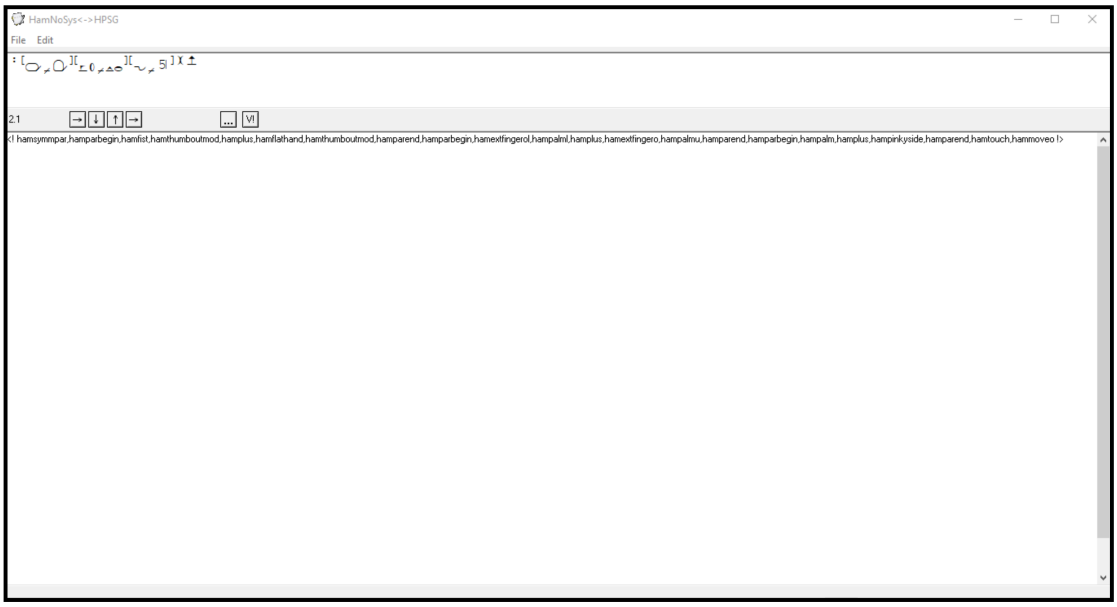


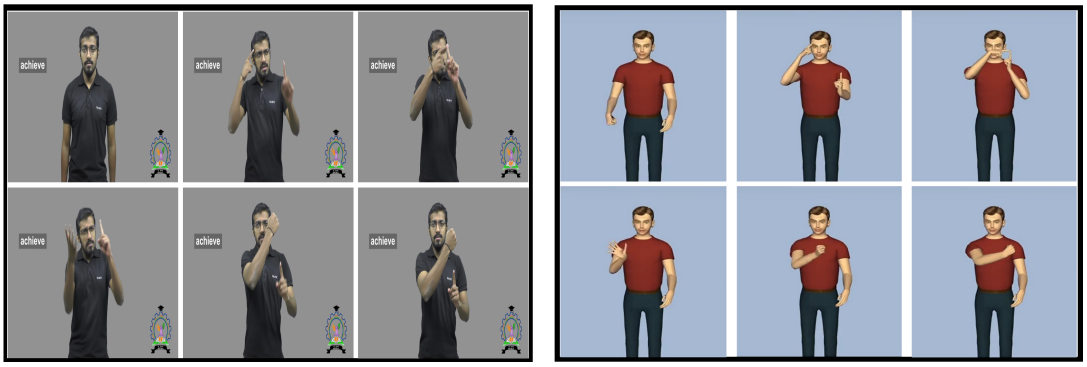
Figure A.2: Interface of Ham2HPSG tool that converts the HamNoSys to Sigml representation.

Usage of HPSG tool

Fig. A.2 describes the interface of Ham2HPSG tool where the HamNoSys for word "advice" is pasted in the textbox and the down-arrow button is clicked to get its Sigml form in the below textbox. These Sigml are stored in XML form encoding in the database and then they are fed to the avatar for depicting the sign movements.

Appendix B

Qualitative Analysis



(a) Human generating sequence of frames. (b) Avatar generating sequence of frames.

Figure B.1: Sequence of Frames Produced by Human and Graphic Generator for word "achieve".

As described in Fig. 5.1, the comparison of actions performed by human and avatar for word "achieve" in real time is represented using video. But if the video is not supported or failed to play, the comparison is depicted using sequence of frames in Fig. B.1.

Appendix C

IEEE Permission to Reprint

In reference to the IEEE copyrighted material which is used with permission in this project report, the IEEE does not endorse any of Lakehead University products or services. Internal or personal use of this material is permitted. If interested in reprint/republishing IEEE copyrighted material for advertisng or promotional purposes or for creating new collective works for resale or redistribution, please go to **https://www.ieee.org/publications_standards/publications/rights/permissions_faq.pdf** to learn how to obtain a License from RightsLink.

Appendix D

Declaration of Co-Authorship / Previous Publication

D.0.1 Co-Authorship Declaration

We hereby declare that this project report incorporates material that is result of joint research, as follows: This report also incorporates the outcome of a research under the supervision of professor Thangarajah Akilan. The key ideas, primary contributions, experimental designs, data analysis, interpretation, and writing were performed by the author, and the contribution of co-authors was primarily through the provision of proof reading and reviewing the research papers regarding the technical content.

We are aware of the Lakehead University Senate Policy on Authorship and we certify that we have properly acknowledged the contribution of other researchers to our report, and have obtained written permission from each of the co-author(s) to include the above material(s) in our report.

We certify that, with the above qualification, this project report, and the research to which it refers, is the product of our own work.

D.0.2 Previous Publication

This project report includes two original papers that have been previously published/submitted for publication in peer reviewed journals and conferences, as follows:

Thesis Chapter	Publication Title/full citation	Publication status
Chapter 1	D.P. Bhavinkumar, B.P. Harshit, A.K. Manthan, R.P. Nidhi, T. Akilan, "ES2ISL: An Advancement in Speech to Sign Language Translation using 3D Avatar Animator.",CCECE 2020 International Conference on. IEEE, ©2016 IEEE	Accepted
Chapter 2	A.K.Manthan, R.P. Nidhi, T. Akilan, "S2SA : Speech to Sign Language Animation using Google Speech Recognition and IBM Watson.", SMC 2020 International Conference on IEEE, ©2016 IEEE"	Submitted

We certify that we have obtained a written permission from the copyright owner(s) to include the above published material(s) in our report. We certify that the above material describes work completed during our registration as a graduate student at the Lakehead University.

D.0.3 General

We declare that, to the best of our knowledge, our project report does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in our report, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that we have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, we certify that we have obtained a written permission from the copyright owner(s) to include such material(s) in our report. We declare that this is a true copy of our report, including any final revisions, as approved by our report committee and the Graduate Studies office, and that this report has not been submitted for a higher degree to any other University or Institution.