

# **English Text to Indian Sign Language Translator**

COEN 296B - Natural Language Processing

By

TEAM - 8

Vardan Gupta - W1423535

Saumya Sinha - W1430155

Puneet Bhushan - W1425328

Minal Shettigar - W1421501

# Table of Contents

1. Abstract	5
2. Introduction	6
2.1 Objective	6
2.2 Problem Statement	6
2.3 Existing Approaches	6
2.4 Proposed Approach	6
2.5 Scope of Investigation	7
3. Theoretical Bases and Literature Review	8
3.1 Theoretical Background of the Problem	8
3.2 Related Research of the Problem	8
3.3 Advantages/Disadvantages of those research	10
3.4 Our Solution to the Problem	10
3.5 Where Our Solution is different	10
4. Hypothesis	11
5. Methodology	12
5.1 Data Collection	12
5.2 Algorithm design	12
5.2.1 Solution Structure	12
5.2.2 Language	16
5.2.3 Tools used	16
5.3 Output Generation	16
6. Implementation	17
6.1 Code (refer programming requirements)	17
6.2 Design document and flowchart	26
7. Data Analysis and Discussion	27
7.1 Output Generation	27
7.2 Output Analysis	28
7.3 Compare Output Against Hypothesis	29
7.4 Abnormal Case Explanation	29
8. Conclusions and Recommendations	30
8.1 Summary and Conclusions	30
8.2 Recommendations for future studies	30
9. Bibliography	31

## **List of Figures**

1. ISL Type Hierarchy	9
2. Parts of Speech Tagging	12
3. Removing Stopwords	15
4. Stemming of Words	15

## **List of Tables**

1. Media Comparison for representing sign	11
2. Comparison of Grammar of English and Indian Sign Language	13
3. Examples of Grammatical Reordering of Words of English Sentence	14

## **Acknowledgement**

We would like to thank various online websites and communities for providing the dataset for completion of the project. We would especially like to thank Mr. Ming-Hwa Wang, professor of Natural Language Processing. As our teacher and mentor, he has taught us more than I could ever give him credit for here. He has shown us at every step what is needed to be achieved with proper guidance.

## **1. Abstract**

Sign language is a natural way of communication for challenged people with speaking and hearing disabilities. There have been various mediums available to translate or to recognize sign language and convert them to text, but text to sign language conversion systems have been rarely developed, this is due to the scarcity of any sign language corpus. Our project aims at creating a translation system that consists of a parsing module which parses the input English sentence to phrase structure grammar representation on which Indian sign language grammar rules are applied. This is done by eliminating stopwords from the reordered sentence. Stemming is applied to convert the words to their root form as Indian sign language does not support for inflections of the word. All words of the sentence are then checked against the words in the dictionary containing videos representing each of the words. If the words are not found in the dictionary, its corresponding synonym is used to replace it.

The proposed system is innovative as the existing systems are limited to direct conversion of words into Indian sign language whereas our system aims to convert these sentences into Indian sign language grammar in real domain.

## **2. Introduction**

### **2.1 Objective**

The objective of the project is to convert English text language to Indian Sign Language using Natural Language Processing to enhance the communication capabilities of people with hearing disabilities.

### **2.2 Problem Statement**

Sign language is a language that uses manual communication methods such as facial expressions, hand gestures and bodily movements to convey information. This project makes use of videos for specific words combined to translate the text language into sign language.

### **2.3 Existing Approach**

Although sign language is used across the world to bridge the gap of communication for hearing or speech impaired which depend mostly on sign language for day to day communication, there are not efficient models that convert text to Indian sign language. There is a lack of proper and effective audio visual support for oral communication. While significant progress has already been made in computer recognition of sign languages of other countries but a very limited work has been done in ISL computerization.

Work done so far in this field has been much more focused on American sign language (ASL) or British sign language, but for Indian sign language, systems that have been developed are very few. The underlying architecture for most of the systems are based on:

1. Direct translation: Words from source language are directly transformed into target language words. Output may not be a desired one.
2. Statistical Machine Translation: It requires large parallel corpus, which is not readily available in case of sign language.
3. Transfer based architecture: Grammar rules are being applied so as to define a proper translation from one language system to another.

### **2.4 Proposed Approach**

Few works have been done to generate a system that is based on the above concepts listed in the existing approaches section and cater to Indian sign language. Thus we propose to develop one for Indian sign language based on transfer based translation.

The success of this translation system will depend on the conversion of English text to Indian sign language bearing its lexical and syntactic knowledge.

## **2.5 Scope of investigation**

Our project aims to encompass the domain of Indian Sign Language which roughly consists of 1500 words in its dictionary. For each of these words its corresponding video will be gathered. For words which do not fall in this dictionary, they would be replaced by their synonyms considering duplicacy of words as well as their parts of speech.

Translation of one spoken language to another spoken language is complex task if both the languages have different grammar rules. The complexity is increased many folds when source language is spoken language and the target language is sign language.

The target audience for this system is not limited to hearing impaired individuals as communication for hearing impaired people in common places like railway stations, bus stands, banks, hospitals etc is very difficult because a vocal person may not understand sign language and thus won't be able to convey any message to hearing impaired person. Thus, it is targeted to all those who wish to learn this translation, to facilitate better communication.

### **3. Theoretical Bases and Literature Review:**

#### **3.1 Theoretical Background of the Problem**

According to 2011 census of India, there are 63 million people which sums up to 6.3% of the total population, who are suffering from hearing problems. Out of these people, 76-89% of the Indian hearing challenged people have no knowledge of language either signed, spoken or written. The reason behind this low literacy rate is either the lack of sign language interpreters, unavailability of Indian Sign Language tool or lack of researches on Indian sign language.

#### **3.2 Related Research to Solve the Problem**

In spite of the modern computer system being so advanced, there is a paucity of research in developing machine translation (MT) system on sign language particularly in India. Some of the MT systems used for other sign languages are:

##### **1. Direct Translation System**

This system is based on word to word conversion. The meaning and context of sentences are not taken in consideration. Neither the grammar is changed, direct transformation into target sign language. No syntactic analysis takes place on the original text and even the order of conversion is unaffected. However while converting from English text to ISL, the word order of ISL may not be same as given text. To overcome this problem, a system is needed that has strong knowledge of both source and target language.

##### **2. Transfer Based Translation**

In this system, plain text is given as input and this input then undergoes syntactic and semantic transformation and then finally transformed into a sign language. In this system, source language is transformed into some intermediate text and then after applying some linguistic rules it is transformed into target language. It also called as Rule based Translation.

##### **3. Interlingua Based Translation**

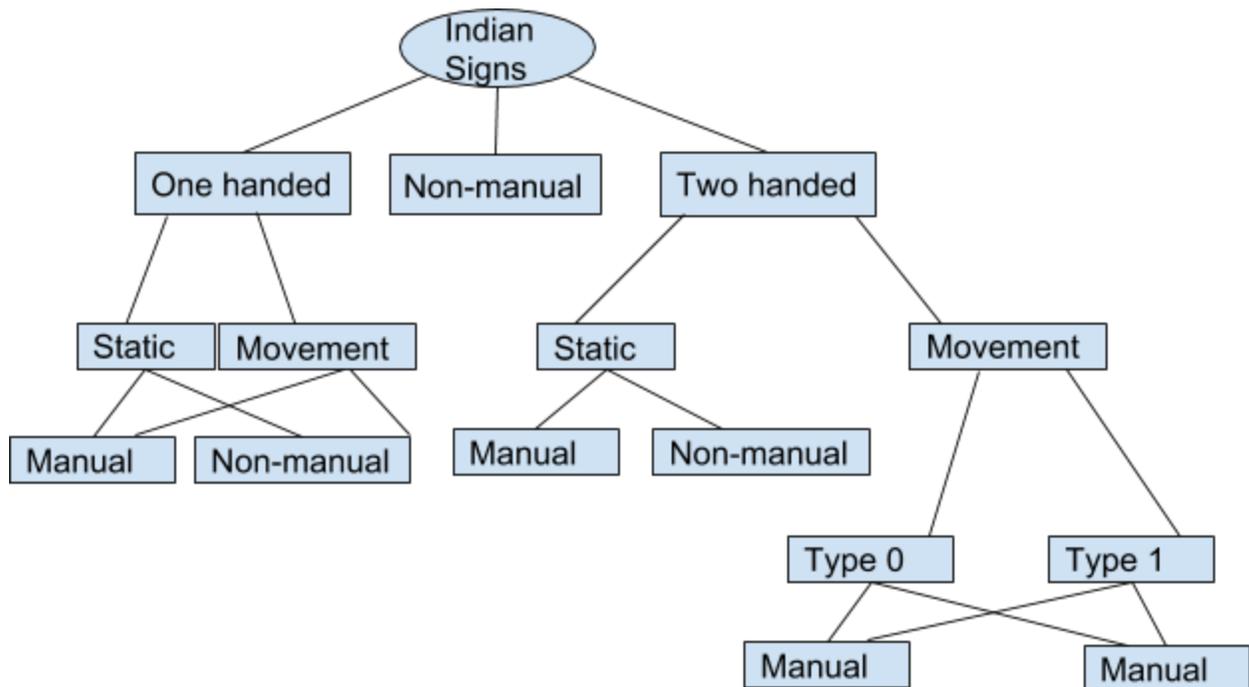
In this system, a language independent semantic structure is produced by doing only semantic analysis on the original input text. This independent structure is called Interlingua. The target language is then generated from this Interlingua. It is thus an alternative of both direct and transfer based translation.

#### **Facts about Sign Language:**

Sign languages is not just 'natural language represented by sign' or not just hand representation of the words as it is but rather it is the representation of meaning. There

are various facts associated with sign language, a natural language, which most of us are unaware. Some of them are listed below:

- NOT the same all over the world
- NOT just gestures and pantomime but do have their own grammar.
- Dictionary is smaller compared to other languages.
- Finger-spelling for unknown words.
- Adjectives are placed after the noun for most of the sign language.
- Never use suffixes
- Always sign in present tense
- Do not use articles.
- Do not uses I but uses me.
- Have no gerunds
- Use of eyebrows and non-manual expressions.
- NOT been invented by hearing people.



### **ISL Type Hierarchy**

**[Figure 1]**

The one handed signs are represented by static as well as dynamic movement of one hand. The static and dynamic movements are then classified into manual and non-manual signs.

Two handed signs use both the hands for gesturing. It can be further classified into Type 0 and Type 1.

Type 0 signs are those where the signer makes use of both the hands

Type 1 signs are those where the use of one hand (dominant) is more compared to the other hand (non-dominant)

## **Indian Sign Language Grammar**

This language has its own grammar and is not same as the manual representation of spoken English or Hindi. Certain distinct features that it has are:

1. Number representation are done with hand gestures for each hand.
2. Signs for family relationships are preceded by male or female.
3. In interrogative sentence, all the WH questions are placed in the back of the sentence.
4. It also consists of many non-manual gestures such as mouth pattern, mouth gestures, body postures, head position and eye gaze.
5. The past, present and future tense is represented by signs for before, then and after.

### **3.3 Advantages and disadvantages**

The advantage of using the video technique is that it is more effective and can convey the message in the most effective way. Whereas, the other approaches are using the images which are way less effective as compared to videos.

### **3.4 Our Solution to the Problem**

The module for translation will help hearing disabled people to understand in an efficient and easy way by providing them with a video to convey them the message of text.

### **3.5 Where our Solution is Different**

Our solution is different as we are using the technique of videos while other techniques are using the images for message conveyance. Also the stopwords have been removed so text processing efforts are also reduced.

## **4 Hypothesis**

Translation from source language to target language requires bilingual dictionary. Thus, we will be creating one such dictionary that contains the English word and its equivalent Indian sign. The counterpart for the English word could be in the format of images, videos or coded sign language text (gloss). All the approaches have their own pros and cons but the video approach is well suited as it has an edge over other systems. A comparison of all the media has been given in the table as shown below:

Kind of media	Pros	Cons
Video Signs	<ul style="list-style-type: none"> <li>● Realistic</li> <li>● Easy to create</li> </ul>	<ul style="list-style-type: none"> <li>● Time consuming to create</li> <li>● High memory consumption</li> <li>● Not supported by translation system</li> </ul>
Pictures	<ul style="list-style-type: none"> <li>● Very less memory consumption</li> </ul>	<ul style="list-style-type: none"> <li>● Time consuming to create</li> <li>● Not realistic as compared to videos</li> <li>● Not supported by translation system</li> </ul>
Code Sign Language Text	<ul style="list-style-type: none"> <li>● Minimal Memory Consumption</li> <li>● Supported by translation system as it is the written form and can be processed very easily</li> </ul>	<ul style="list-style-type: none"> <li>● Very difficult to read and understand</li> <li>● Required to be learnt</li> </ul>

**Media Comparison for representing sign**

**[Table-1]**

Because videos provide much realistic content as compared to image and coded text so we preferred using videos.

If time permits we wish to develop a system based on synthetic animations as it requires less memory consumption, can easily be reproduced, is supported by translation system and follows an avatar based approach which can be easily customized.

## **5 Methodology**

### **5.1 Data Collection**

We will be using the <http://www.indiansignlanguage.org/> to download the video clips of each and every word. We will manually label each of the video and remove videos that we will find irrelevant. We will like to maintain an unfiltered input that covers a wide range of words.

### **5.2 Algorithm Design**

The system consists of 5 modules:

1. English parser for parsing the English text
2. Sentences reordering module based on ISL grammar rules
3. Eliminator for eliminating stopwords
4. Stemming for getting the root words of each word and synonym replacement for words not in dictionary.
5. Video conversion module.

The input to the system is a written English text which is parsed to create a phrase structure based on its grammar representation. Then reordering is done to meet ISL grammar needs since, English text follows Subject-Verb-Object structure whereas ISL follows Subject-Object-Verb structure along with variation of negative and interrogative sentences. After which unwanted words are removed, as ISL will only those words which have meanings and all helping words like linking verbs, articles etc are not used. The output of which is sent to lemmatization module which reduces each of the words to its root form. The words not present in dictionary are replaced with their synonyms.

#### **5.2.1 Solution Structure**

##### **1. Parsing of the Input English Text**

To carry out rule based conversion of one language to another, grammatical structure of both the source and target language must be known. Parsing is the answer to acquiring this grammatical structure. Stanford parser is capable to produce three different outputs, part-of-speech tagged text, context free grammar representation of phrase structure and type dependency representation. The parser uses Penn tree tags for parsing the English sentence.



**Parts of Speech Tagging**

**[Figure 2]**

**2. Grammar rules for conversion from English to ISL**

Since both spoken language and sign language have different grammar rules. The complexity to translate them is increased many folds. Comparison between ISL and English grammar is listed below:

English Grammar	Indian Sign Language Grammar
English grammar is well structured and a lot of research work has been carried out to define the rules for it. English grammar follows the subject-verb-object order.	ISL is invented by deaf and a little work has been done to study the grammar of this language. The structure of sentences of ISL follows the subject-object-verb order[13].
English language uses various forms of verbs and adjectives depending upon the type of the sentence. Also, a lot of inflections of the words are used in English sentences.	ISL does not use any inflections ( gerund, suffixes, or other forms ), it uses the root form of the word.
English language has much larger dictionary	Indian sign language has a very limited dictionary, approximately 1800 words[10].
Question word in interrogative sentences is at the start in English	In Indian sign language, the question word is always sentence final
A lot of helping verbs, articles, and conjunctions are used in the sentences of English	In Indian sign language, no conjunctions, articles or linking verbs are used

**Comparison of Grammar of English and Indian Sign Language**

**[Table-2]**

Therefore ISL grammar rules, require all the verb patterns (20 patterns) being shifted after the corresponding noun occurrence. Some of the rule conversion are given in the following table

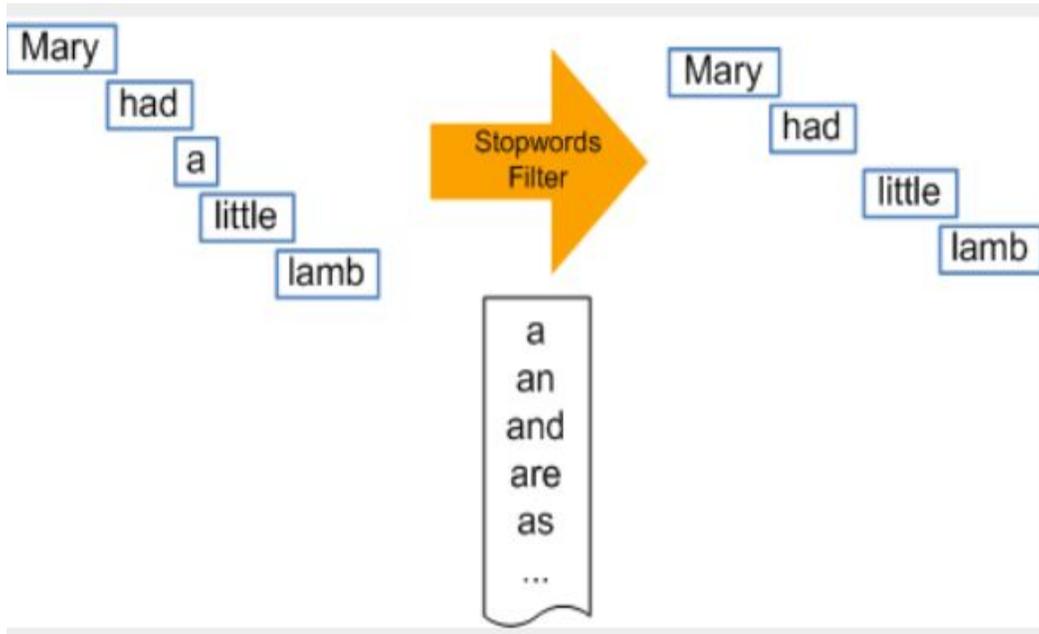
Verb Pattern	Rule	Input Sentence	Parsed Sentence	Output Sentence
verb+object	VP TO NP	Go to school	(VP (VB Go ) (TO to) (NP (NN school ) ) )	School to go
Subject + verb	NP V	Birds fly	(NP (NNS birds ) ) (VP (VBP fly ) )	Birds fly
subject + verb + subject complement	NP V NP	His brother became a soldier	(NP (PRP\$ his ) (NN brother ) ) (VP (VBD became ) (NP (DT a ) (NN soldier ) ) )	His brother a soldier became
subject + verb + direct object + preposition +prepositional object	NP V NP PP	She made coffee for all of us	(NP (PRP She ) ) (VP (VBD made ) ) (NP (NN coffee ) ) (PP (IN for ) (NP (NP (DT all ) ) (PP (IN of ) (NP (PRP us ) ) ) ) ) )	she coffee for all of us made

**Examples of Grammatical Reordering of Words of English Sentence**

**[Table-3]**

### 3. Elimination of Stop Words

Since ISL deals with words associated with some meaning, unwanted words are removed these include various parts of speech such as TO, POS(possessive ending), MD(Modals), FW(Foreign word), CC(coordinating conjunction), some DT(determiners like a, an, the), JJR, JJS(adjectives, comparative and superlative), NNS, NNPS(nouns plural, proper plural), RP(particles), SYM(symbols), Interjections, non-root verbs

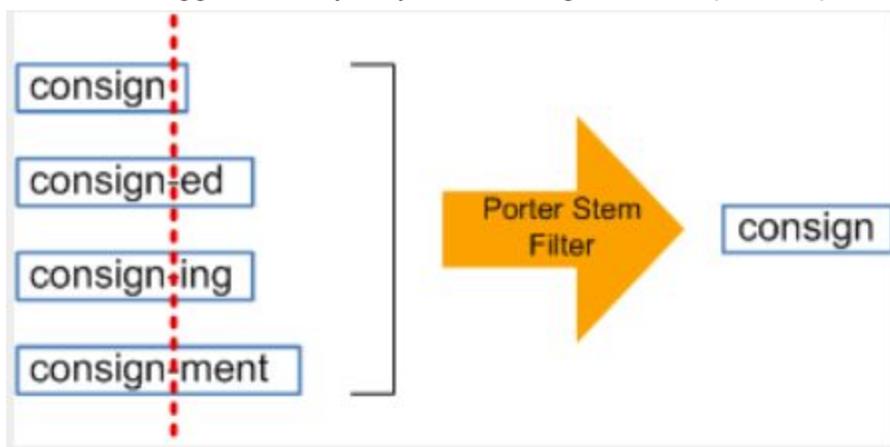


**Removing Stopwords**

**[Figure 3]**

### 4. Lemmatization and Synonym replacement

Indian sign language uses root words in their sentences. So we convert them to root form using Porter Stemmer rules. Along with this each word is checked in bilingual dictionary, if word does not exist, it is tagged to its synonym containing the same part of speech.



**Stemming of Words**

**[Figure 4]**

## **5. Video Conversion Stage**

After the completion of above stages we get the ISL transformed text , the program will then find matches from the dataset available for each of the word. This will be based on the basic string matching algorithm between the processed input text and labels of videos. Finally a display of set of videos as a sequence one after the other can be seen on the screen.

### **5.2.2 Language**

We will be using Python(version 3) for the development of the project. Also, we will be using many libraries of python for implementation of the project.

### **5.2.3 Tools/Libraries Used**

- 1- Numpy
- 2- JASigning - Sigml file conversion into animated format
- 3- movies.py
- 4- nltk

## **5.3 Output Generation**

The output of this module will be a movie clip of ISL translated words. The database will be having video for each and every separate words and the resultant video will be a merged video of such words.

## **6. Implementation**

### **6.1 Code**

#### **English text to Isl text conversion code:**

```
import sys
import argparse
from nltk.parse.stanford import StanfordParser
from nltk.tag.stanford import StanfordPOSTagger, StanfordNERTagger
from nltk.tokenize.stanford import StanfordTokenizer
from nltk.tree import *
from nltk import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from nltk.stem import PorterStemmer
import nltk

inputString = " "
import os
java_path = "C:\\Program Files\\Java\\jdk-9.0.4\\bin\\java.exe"
os.environ['JAVAHOME'] = java_path

for each in range(1,len(sys.argv)):
    inputString += sys.argv[each]
    inputString += " "

# inputString = raw_input("Enter the String to convert to ISL: ")

parser=StanfordParser(model_path='D:/stanford-parser-full-2018-02-27/edu/stanford/nlp/models
/lexparser/englishPCFG.ser.gz')

# o=parser.parse(s.split())

englishtree=[tree for tree in parser.parse(inputString.split())]
parsetree=englishtree[0]

dict={}

# "*****subtrees*****"

parenttree= ParentedTree.convert(parsetree)
for sub in parenttree.subtrees():
    dict[sub.treeposition()]=0
```

```

#"-----"

isltree=Tree('ROOT',[])
i=0
for sub in parenttree.subtrees():
    if(sub.label()=="NP" and dict[sub.treeposition()]==0 and dict[sub.parent().treeposition()]==0):
        dict[sub.treeposition()]=1
        isltree.insert(i,sub)
        i=i+1

    if(sub.label()=="VP" or sub.label()=="PRP"):
        for sub2 in sub.subtrees():
            if((sub2.label()=="NP" or sub2.label()=="PRP')and dict[sub2.treeposition()]==0 and
dict[sub2.parent().treeposition()]==0):
                dict[sub2.treeposition()]=1
                isltree.insert(i,sub2)
                i=i+1

for sub in parenttree.subtrees():
    for sub2 in sub.subtrees():
        # print sub2
        # print len(sub2.leaves())
        # print dict[sub2.treeposition()]
        if(len(sub2.leaves())==1 and dict[sub2.treeposition()]==0 and
dict[sub2.parent().treeposition()]==0):
            dict[sub2.treeposition()]=1
            isltree.insert(i,sub2)
            i=i+1

parsed_sent=isltree.leaves()

words=parsed_sent

stop_words=set(stopwords.words("english"))
# print stop_words

lemmatizer = WordNetLemmatizer()
ps = PorterStemmer()
lemmatized_words=[]

for w in parsed_sent:
    # w = ps.stem(w)

```

```

lemmatized_words.append(lemmatizer.lemmatize(w))

islsentence = ""
print(lemmatized_words)
for w in lemmatized_words:
    if w not in stop_words:
        islsentence+=w
        islsentence+=" "

print(islsentence)

```

**Video Generation Code:**

```

from nltk.parse.stanford import StanfordDependencyParser
import numpy as np
import cv2
import imageio
imageio.plugins.ffmpeg.download()
from moviepy.editor import VideoFileClip, concatenate_videoclips
from moviepy.editor import VideoFileClip, concatenate_videoclips

import nltk
import os
import sys

try:
    os.remove("my_concatenation.mp4")
except:
    pass
print(sys.path)
name=""

for each in range(1,len(sys.argv)):
    name+=sys.argv[each]
    name+=" "

input_text=name

text = nltk.word_tokenize(input_text)

result=nltk.pos_tag(text)

for each in result:
    print(each)

```

```

dict={}
dict["NN"]="noun"
arg_array=[]

for text in result:
    arg_array.append(VideoFileClip(text[0]+".mp4"))
    print(text[0]+".mp4")
print(arg_array[0])

final_clip = concatenate_videoclips(arg_array)
final_clip.write_videofile("my_concatenation.mp4")

```

### **Display code for video generation**

```

<?php
if(isset($_POST['username']))
{
    $filename= $_POST['username'];
    $pyscript_video = 'video_convert.py';
    $pyscript_convert = 'convert2isl.py';
    $python='C:\\Users\\varda\\Miniconda3\\python.exe';
    $cmd = "$python $pyscript_convert $filename";
    $isl = exec("$cmd");
    echo("isl");
    $cmd = "$python $pyscript_video $isl";
    exec("$cmd");
    session_start();
}
?>
<!DOCTYPE html>
<html>
<head>

    <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
    <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <link rel="stylesheet" href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">
    <script defer src="https://code.getmdl.io/1.3.0/material.min.js"></script>
    <link rel='stylesheet' href='../English2ISLGenerator-master/css/one.css'>
</head>
<body>
    <div id="toolbar" style="height:100px">

    <div id="title" style="font-size:30px;text-align:center;margin-left:450px">

```

## Text to Indian Sign Language Conversion

```
</div>
  <form>
    <button class="mdl-button mdl-js-button mdl-button--icon mdl-button--colored"
style="margin-left:1450px;margin-top:50px" formaction="main.php" >
    <i class="material-icons">home</i>
    </button>

  </form>
</div>
<div id="content" data-0="padding-top: 192px;" data-192="padding-top: 190px;"
style="padding-bottom:50px">
</div>
  <?php
  if (isset($filename)){
  echo '<center style="padding-top:30px">';
  echo '<video width="1080" height="480" controls autoplay>';
  echo '<source src="my_concatenation.mp4" type="video/mp4">';
  echo 'Your browser does not support the video tag.';
  echo '</source></video>';
  echo '</center>';
  $vars = array_keys(get_defined_vars());
  foreach($vars as $var) {
  unset("${$var}");
}
  }    ?>
  <form action="" method="post" name="myform" id="myform">
  <div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label"
style="padding-left:20px;margin-left:550px">
  <INPUT class="mdl-textfield__input" TYPE = "Text" NAME = "username" >
  <label class = "mdl-textfield__label" for="inputText"
style="font-size:20px;padding-left:20px">English text: <?php
if(!empty($_POST['username'])){echo $isl;}?> </label>
  <button type="submit"class="mdl-button mdl-js-button mdl-button--fab
mdl-js-ripple-effect mdl-button--colored " id="Video"
style="margin-left:300px;background-color:#01579B;padding:10px">
  <i class="material-icons" >arrow_forward</i>
  </button>
</div>
<br>
```

```
</form>

</body>
</head>
```

### **Display code for animation generation:**

```
<!DOCTYPE html>
<?php
if(isset($_POST['englishtext']))
{

    $isl = "";
    $pyscript = 'convert2isl.py';
    $python='C:\\Users\\varda\\Miniconda3\\python.exe';
    $englishinput = $_POST['englishtext'];
    $cmd = "$python $pyscript $englishinput";

    $isl = exec("$cmd");

}

?>
<html>
<head>
    <?php require_once("include.php"); ?>
    <title>ISL : Avatar Page</title>
    <meta http-equiv="Access-Control-Allow-Origin" content="*">
    <meta http-equiv="Access-Control-Allow-Methods" content="GET">
    <script src="//code.jquery.com/jquery-1.11.1.min.js"></script>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/icon?family=Material+Icons">
    <link rel="stylesheet"
href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">
    <script defer src="https://code.getmdl.io/1.3.0/material.min.js"></script>
    <link rel='stylesheet' href='../English2ISLGenerator-master/css/one.css'>
    <link rel="stylesheet" href="css/cwasa.css" />
    <script type="text/javascript" src="js/allcsa.js"></script>
    <script language="javascript">
        // Initial configuration
        var initCfg = {
            "avsbsl" : ["luna", "siggi", "anna", "marc", "francoise"],
            "avSettings" : { "avList": "avsbsl", "initAv": "marc" }
        };
```

```

// global variable to store the signal list
var sigmlList = null;

// global variable to tell if avatar is ready or not
var tuavatarLoaded = false;
</script>
</head>
<body onload="CWASA.init(initCfg);" style="margin-top:0!important;">
<div id="toolbar" style="height:100px">

<div id="title" style="font-size:30px;text-align:center;margin-left:450px">
Text to Indian Sign Language Conversion

</div>
<form>
<button class="mdl-button mdl-js-button mdl-button--icon mdl-button--colored"
style="margin-left:1450px;margin-top:50px" formaction="main.php" >
<i class="material-icons">home</i>
</button>
</form>
</div>
<div id="content" data-0="padding-top: 192px;" data-192="padding-top: 190px;"
style="padding-bottom:50px">
</div>
<div style="padding-top:80px">
<div id="loading" class="container"><div class="row text-center"><span
style="background-color:#ebf8a4; padding: 8px 20px;">Loading ... Please
wait...</div></div></div>
<!-- left side division starts here -->
<div style="width:40%; padding:15px; float:left; margin-left:14%;">

<form action="" method="post" name="myform" id="myform">
<div class="mdl-textfield mdl-js-textfield mdl-textfield--floating-label" >
<input class="mdl-textfield__input" type = "Text" name = "englishtext"
style="margin-bottom:5px">
<label class ="mdl-textfield__label" for="inputText" style="font-size:20px;">English text: </label>

<button type="submit" class="mdl-button mdl-js-button mdl-button--fab mdl-js-ripple-effect
mdl-button--colored " id="parseisl"
style="margin-left:300px;background-color:#01579B;padding:10px">
<i class="material-icons" >arrow_forward</i>

```

```

</button>
</div>

</form>
<label class="mdl-textfield__input" for="inputText">The text to animate: <?php
if(!empty($_POST['englishtext'])){ echo $isl;}?> </label><br>
<button type="button" class="btn btn-primary" onclick="yahoo();" id="a">Generate Play
Sequence</button>
<button type="button" id="btnClear" class="btn btn-default">Clear</button>

</div>
<div id="dom-target" style="display: none;">
  <?php
    //$output = "42"; //Again, do some operation, get the output.
    echo htmlspecialchars($isl); /* You have to escape because the result
                                will not be valid HTML otherwise. */
  ?>
</div>

<div id="menu2">
<br>
Words will be displayed here
</div>

<div id="menu3">
<br>
Alphabets will be displayed here
</div>

<div id="menu4">
<br>
Number will be displayed here
</div>

</div> <!-- left side division ends here -->
<script language="javascript" src="js/animationPlayer.js"></script>
  <?php
    // This is the main player where the animation happens
    include_once("animationPlayer.php");
  ?>

```

```

<script type="text/javascript" src="js/player.js"></script>
<script type="text/javascript">
/*
    Load json file for sigml available for easy searching
*/
$.getJSON("js/sigmlFiles.json", function(json){
    sigmlList = json;
});

// code for clear button in input box for words
$("#btnClear").click(function() {
    $("#inputText").val("");
    $("#debugger").html("");
});

// code to check if avatar has been loaded or not and hide the loading sign
var loadingTout = setInterval(function() {
    if(tuavatarLoaded) {
        $("#loading").hide();
        clearInterval(loadingTout);
        console.log("Avatar loaded successfully !");
    }
}, 1000);

// code to animate tabs

alltabhead = ["menu1-h", "menu2-h", "menu3-h", "menu4-h"];
alltabbody = ["menu1", "menu2", "menu3", "menu4"];

function activateTab(tabheadid, tabbodyid)
{
    for(x = 0; x < alltabhead.length; x++)
        $("#"+alltabhead[x]).css("background-color", "white");
    $("#"+tabheadid).css("background-color", "#d5d5d5");
    for(x = 0; x < alltabbody.length; x++)
        $("#"+alltabbody[x]).hide();
    $("#"+tabbodyid).show();
}

activateTab("menu1-h", "menu1"); // activate first menu by default

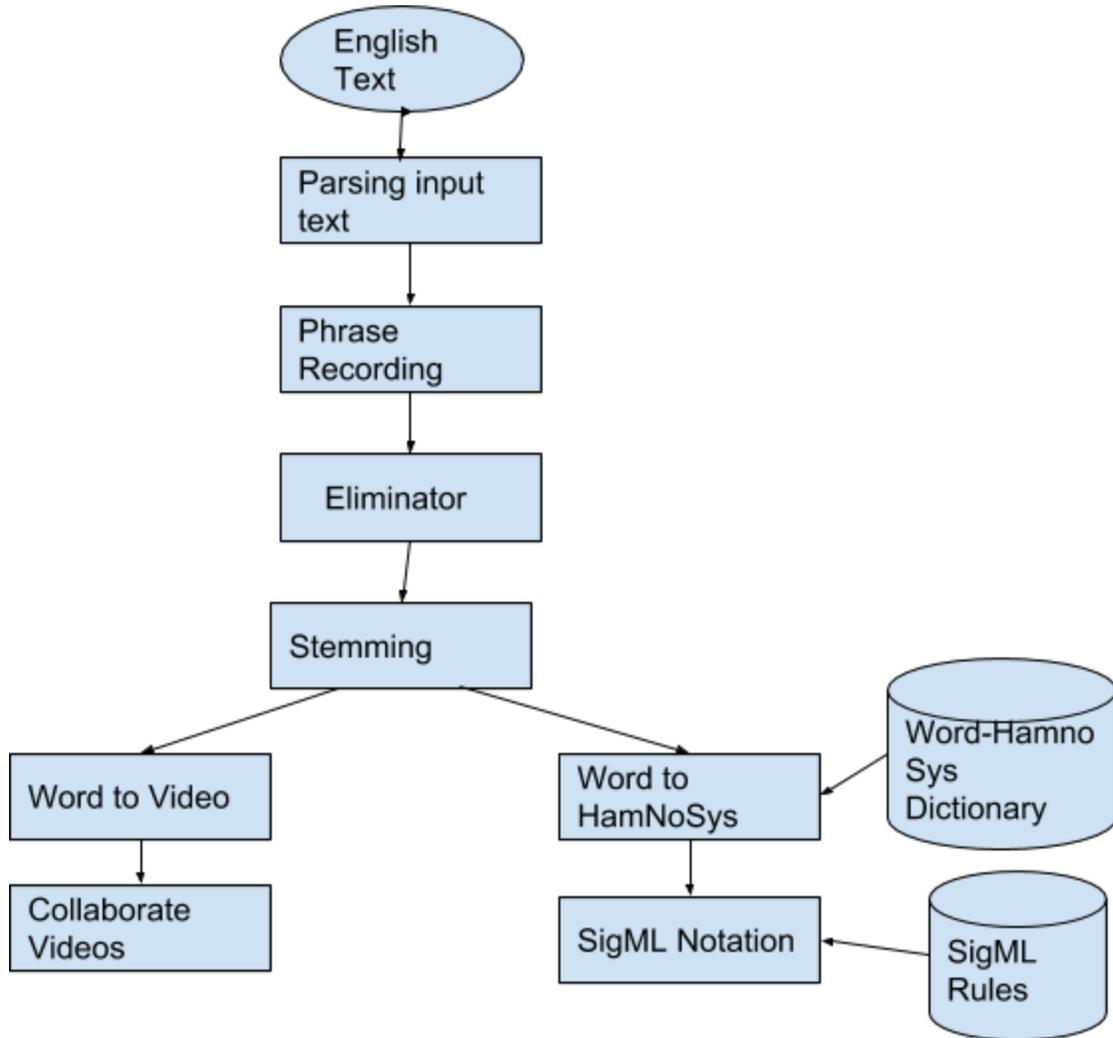
```

</script>

</body>

</html>

## 6.2 Design Document and Flowchart



## 7. Data Analysis and Discussion

### 7.1 Output Generation

For a given english text the system aims at generating its equivalent sign language depiction. Our system generates these outputs in the following two ways:

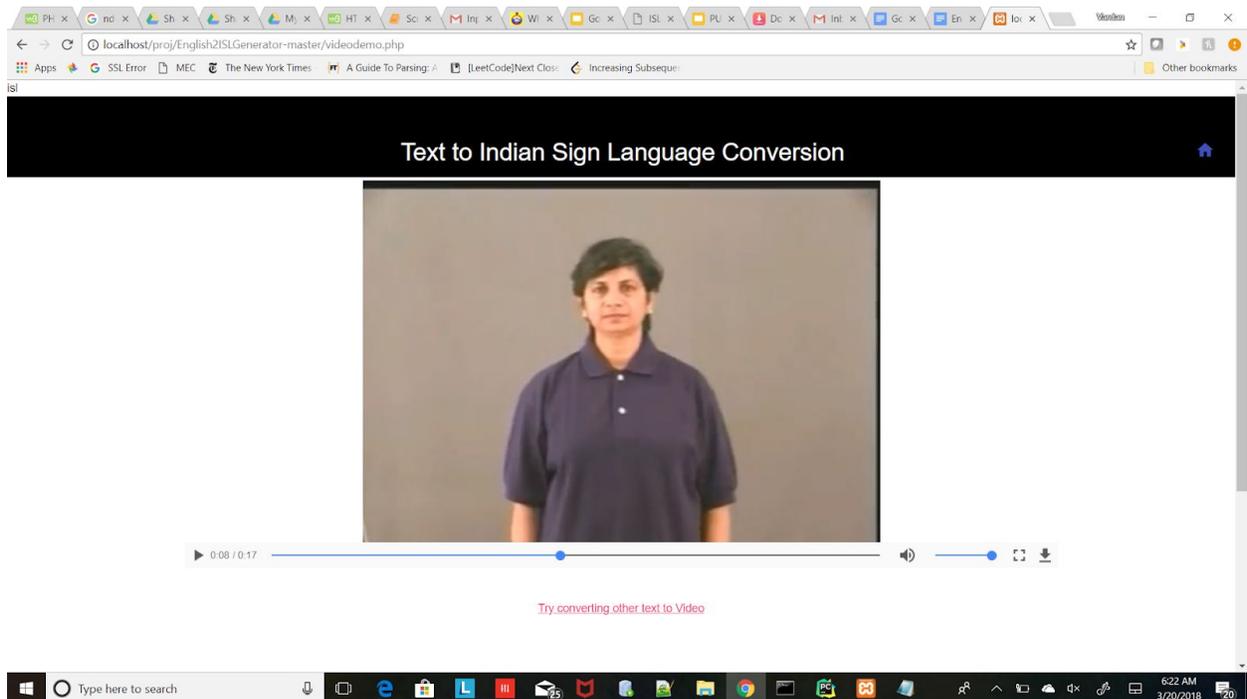
1. Video generation - Output from the ISL conversion phase of input sentence is passed to video generation phase wherein for each of the words in the sentence are looked up in the database for its corresponding video file and then these files are all concatenated to produce a more structured, informative and easy to understand visual depiction of Indian Sign Language.
2. Synthetic Animation generation - In this approach the ISL converted text is checked for its corresponding SigML file wherein these files are generated by a process called Hamnosys to SigML conversion to generate a markup language that is compatible with JASigning tool to generate equivalent synthetic animations.

Consider the following input sentence to which the generated ISL, video output and animation output are as follows:

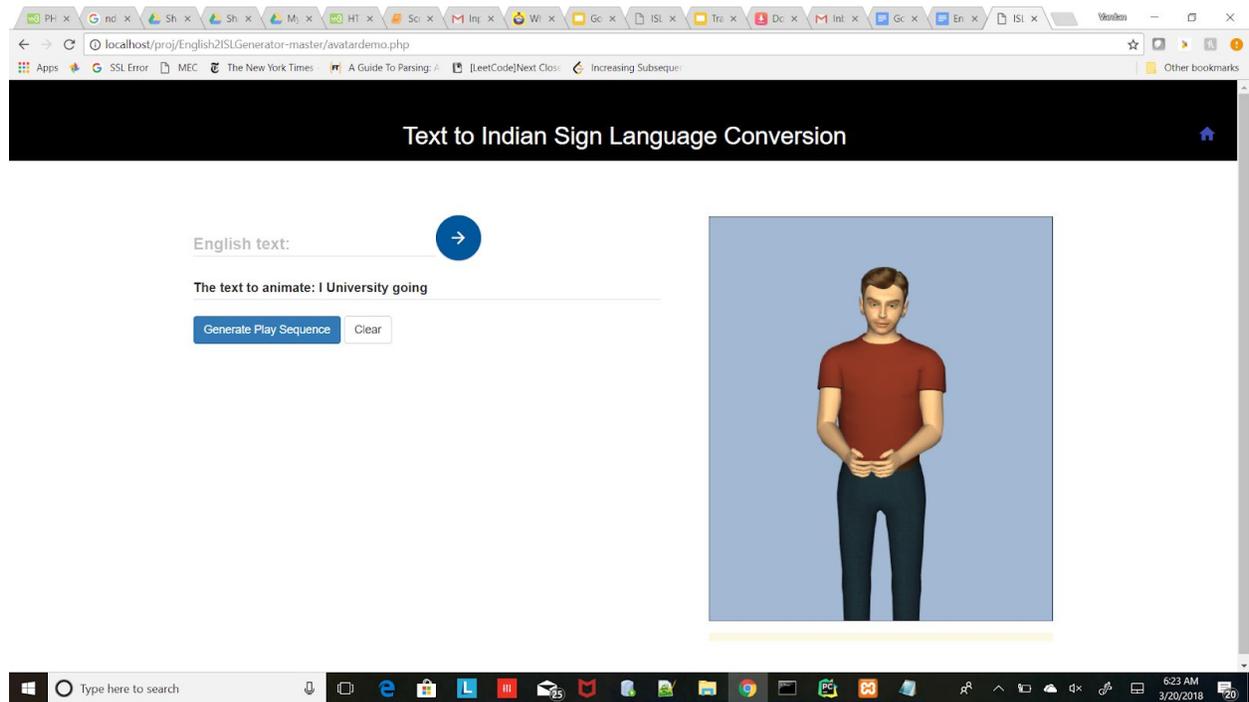
**English input text:** I am going to the University.

**ISL parsed text:** I University going

**Video Generated Output:**



## Animation Generated Output:



## 7.2 Output Analysis

The given input text undergoes parsing using the StanfordParser to identify the subject, verb and object in the given sentences and which according to Indian Sign language grammar requires its verb and object positions to be swapped. This process fairly returns proper results and after this we require the system to lemmatize each of the words as sign language does not follow tense, however lemmatization follows a corpus and we chose wordnet corpus which not necessarily stem each of the verb forms, due to which there is a requirement to modify video and sigml files to support this issue. The generated output is then checked for stopwords and they are eliminated if present. After which the ISL sentence is then passed through two methods to generate sign language

Each of these methods were being analysed and we came up with the following analysis for each of them:

1. Video
  - The process of finding the relevant videos and creating a collaboration of these videos is a time taking task.
  - The video output couldn't ensure consistency due to the lack of research going on this field.
  - All words not necessarily find their equivalent word file, thus such words are skipped.
2. Synthetic Animation

- The process of mapping the sigml notations to generate animation is comparatively less time consuming
- This ensures consistency as the entire sentence is enacted by the same character throughout.
- However, these animations do not provide a realistic feel as is provided by videos.
- Finger spelling are used for those words which are not present in the database.

### **7.3 Compare Output Against Hypothesis**

Our approach with the use of animation and light weight videos improves the overall performance by pre-processing the input text using NLP techniques and making use of the relevant term from the input sentence. Furthermore, we have used visual depiction for the entire input sentence and not for a particular word that is depicted in many of the previous projects. This ensures that our system is more robust and useful during live conversation. Videos and synthetic animations do provide a realistic appeal to the output. This system can be used readily and requires less prior knowledge about sign language.

### **7.4 Abnormal Case Explanation**

Since much research hasn't been done on Indian Sign Language grammar, there are few words and types of sentences which still haven't found their respective translation.

Exclamation words like Oh! Alas! Or Hurray! don't have a direct translation in the Indian sign language, however these are words that are based on facial cues and expressions and thus need to be handled.

Imperative sentences such as "Book the flight", however they can be converted into ISL grammar format resulting in "Flight Book", they cannot be expressed in the imperative format owing to the sentence structure.

Also words having more than one parts of speech, example book has both verb and noun form cannot be handled well by the conversion system, since the original input sentence after being parsed and processed is converted into ISL grammar format which does not have a parsing structure, which indeed becomes difficult to identify the nouns and verbs from a given sentence, eventually causing the wrong form of the sentence being selected.

However, since less research has been carried out in this field, there are many pitfalls that needs to be worked on and some scenarios which needs to be thought of.

## **8 Conclusions and Recommendations**

## **8.1 Summary and Conclusions**

In this project, we have tried to develop a system that would be helpful for the disabled people having communication difficulties by developing a system that would help them in expressing themselves clearly and easily.

Our model successfully converts the entire input sentence into a single visual rather than depicting different words through a GIF/ picture giving the model a much realistic and lively appeal. Much more development on this track can be done as the ISL dictionary is still small and needs to grow eventually.

## **8.2 Recommendations for future studies**

As the ISL is new and not much development has been done on this topic, many new videos for different words can be added to the dictionary to widen its scope and help people communicate better using this language. Moreover, text to speech integration can be performed in the project to enable better communication strategies which will allow users to convert text to Indian Sign Language with manually entering the input sentence.

## **9 Bibliography**

- Goyal, L., & Goyal, V. (n.d.). Automatic Translation of English Text to Indian Sign Language <http://www.aclweb.org/anthology/W16-6319>

- Zouhour Tmar, Achraf Othman & Mohamed Jemni: A rule-based approach for building an artificial English-ASL corpus  
<http://ieeexplore.ieee.org/document/6578458/>
- Teranai Vichyaloetsiri & Pongpisit Wuttidittachotti: Web Service framework to translate text into sign language: <http://ieeexplore.ieee.org/document/8035336/>
- P. Kar, M. Reddy, A. Mukherjee, A. M. Raina. 2017. INGIT: Limited Domain Formulaic Translation from Hindi Strings to Indian Sign Language. ICON.
- Dictionary | Indian Sign Language. (n.d.). Retrieved July 15, 2016, from <http://indiansignlanguage.org/dictionary>
- Zeshan, U., Vasishta, M. N., & Sethna, M. (2015). Implementation of Indian Sign Language in educational settings. Asia Pacific Disability Rehabilitation Journal, 16(1), 16- 40
- S. Cox, M. Lincoln, J. Tryggvason, M. Nakisa, M . Wells, M. Tutt, S. Abbott. 2002. Tessa, a system to aid communication with deaf people. Fifth international ACM conference on Assistive technologies.
- M. Vasishta, J. Woodward and S. DeSantis. 2011. An Introduction to Indian Sign Language. All India Federation of the Deaf (Third Edition).