Sentence Generation for Indian Sign Language Using NLP

Dr. P. Golda Jeyasheeli  
Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, India. E-mail: pgolda@mepcoeng.ac.in

N. Indumathi  
Department of Computer Science and Engineering, Mepco Schlenk Engineering College, Sivakasi, India. E-mail: indhujegan1996@gmail.com

Received October 28, 2020; Accepted November 25, 2020  
ISSN: 1735-188X  
DOI: 10.14704/WEB/V18SI01/WEB18054

Abstract

Nowadays the interaction among deaf and mute people and normal people is difficult, because normal people scuffle to understand the sense of the gestures. The deaf and dumb people find problem in sentence formation and grammatical correction. To alleviate the issues faced by these people, an automatic sign language sentence generation approach is propounded. In this project, Natural Language Processing (NLP) based methods are used. NLP is a powerful tool for translation in the human language and also responsible for the formation of meaningful sentences from sign language symbols which is also understood by the normal person. In this system, both conventional NLP methods and Deep learning NLP methods are used for sentence generation. The efficiency of both the methods are compared. The generated sentence is displayed in the android application as an output. This system aims to connect the gap in the interaction among the deaf and dumb people and the normal people.

Keywords

Sign Language, Natural Language Processing, Bidirectional Long Short-term Memory Neural Network (Bi-LSTM).

Introduction

Sign language is the natural way of interaction for the deaf and mute people. In sign language, the meaning is conveyed through the hand movements and body gestures. A sign consists of blue collar and non-blue collar components. Blue collar components comprise palm configurations, palm movements and orientations while non-blue collar components include facial expressions and body actions. The movement of the body is the
motion itself and consists of the movement not just of the hands but of the whole body. With one hand or both, this movement may be carried out, but the main part of it will be performed by the dominant hand of the user, determined by whether he is left-handed or right-handed. Since the object of this work is real-time translation, all these components have been taken into account: the sign-to-text translation grammar rules, and the three sign-to-text part components of the signs. The Figure.1 shows some of the words in the Indian Sign Language.

![Figure 1 Few words in Indian Sign Language](image)

In order to make better communication among deaf and mute people, sign language conversion system is needed. This project makes use of text of specific words combined to interpret the sign language into text language.

The smart sign translation system designed in this project will decode the signs of the Indian sign language into English sentences. For that two methods are handled in order to compare the efficiency of the system. Initially the sign words are passed into the NLP engine for sentence conversion. Later the same sign words are passed into the Bi-LSTM
(Bidirectional Long Short-Term Memory) model for sentence conversion. By comparing the efficiency of the results, the sentence gets transmitted to the android device to display the converted sentence.

Related Work

In order to predict the manual signals made, there are two different approaches. One is vision-based approach and the other one is non-vision based approach. The vision-based approach is categorized into direct and non-direct approach.

Through the use of neural networks, a direct approach classifies the gesture. Yemane Tedla [3] et al” presents Tigrinya's work on morphological segmentation. To build a boundary detection models, separately conditional random fields (CRF) and window-based long-term memory (LSTM) neural networks are used. For the CRF and character embeddings for the LSTM networks, language-independent character and substring functions are introduced. Experiments were carried out with four models of chunk annotation scheme Begin-Inside-Outside (BIO). The first model was focused on CRFs with text, n-gram, and substring character features that were independent of language. The other two were based on deep neural architectures from LSTM and Bi-LSTM that leveraged a fixed-size window embedding character to extract the boundary features of the morpheme.

Sumeet R. Agarwal et al [4]” propounded a system interpreting the sign language for the deaf and dumb people. The sign gestures are made with the sensor glove. Each sign has its value, the values are matched with the predicted value to obtain a unique gesture. These matched words are passed to the NLP engine with its sign attribute based on the English grammar. Based on attributes, the meaningful sentences are constructed. The grammar used is the context free grammar. This grammar comprised of Non-Deterministic Finite Automata(N-DFA) with € moves. Later this N-DFA converted to DFA for the removal of ambiguity.

The indirect approach is based on the sensor glove that generates the gesture, and the input to the neural network is in the form of images. Tiago Oliveira et al [10] proposed a system that describes the Virtual Sign platform's updates and current structure, a bidirectional sign language is used as a text communication tool. It can obtain text and display a three-dimensional avatar performing a particular country's sign language, or obtain input from someone performing sign language beside the camera and present the text translation in writing.
The outcome of the system is, 3D avatar that makes the movements that suit the written content. The input words are transmitted through the specified language's grammar rules and changed accordingly with the resulting sentence being checked on the server and the gesture information is supporting the avatar. The classifiers are the same classifiers used in the software for palm select and CNN.

Chung Hsein Wu et al [12] proposed system for augmentive communication between hearing impaired people. For that, a sign icon-based virtual keyboard is designed to visualize the signs for retrieving the gesture from the database. Based on the predictive sentence tree, the statistical n-gram approach is integrated with the sentence translation. To model the correspondence between signed and written Chinese, the PST tree equipped by a corpus was used. In addition, a set of rules for phrase creation, based on category trigger pairs, was developed for expansion of sentence patterns.

The training corpus for creating sentence template tree in the tokenization process is obtained from the instructional materials and everyday dialogs. The TSL vocabulary is classified as the keyword collection.

Kumud Tripathi et al [13] proposed a model for recognizing the sign gesture. The recognition is done by using gradient based key frame extortion method, by splitting the persistent gestures into the course of signs to remove unenlightening frames. Each image is pre-processed from RGB frames to HSV plane. The median filter is applied on each frame to reduce noise in the image.

After that, pre-processing is done on all the frames to extract the features by using orientation histogram approach with Principal Component Analysis (PCA) to reduce the dimensions of features. This method uses the Euclidean distance, Correlation, Manhattan distance, City block distance as classifiers.

Linqin Cai et al [14] proposed a stacked Bi-LSTM neural network model to extract the interaction between the question and answers simultaneously. A stacked Bi-LSTM neural network is used to represent the input sentence in the vector format to capture the sentence's semantics. Coattention mechanism is used to encode the sequences for obtaining the interaction between QA pair. The cosine similarity and the Euclidean distance are reconciled in order to determine the degree of correspondence between the vectors and this approach is capable of taking the vector distance and angle relationship.
Mansi Jain et al [15] propounded a system to generate the text based on the series of input data. Different series of data are the input of the system, which understands the characters and content of each series. By creating the dictionary of ideal words from input data file, it maps the words and characters with the respective index positions. Input file is divided passed to the different tensors based on the sequence length. By applying one hot encoding, the categorical features get transformed into a specified format. Each data had separate probability function by using SoftMax activation function. Based on the probability, the system shift from one word to other to get next word. By repeating the process, complete story line was generated.

Maryem Rhanoui [16] et al proposed a model based on CNN and BiLSTM neural network for document analysis in long texts. The proposed system used the doc2vec embedding for word level and sentence level document analysing. Word embedding method involves automatic natural language processing of large corpus. The CNN involves the text pre-processing using its convolutional layers for extracting the information from the corpus. The CNN output becomes the Bi-LSTM model input that preserves the sequential order between the data in both directions.

System Design

The proposed system consists of three modules. They are, sentence generation using conventional NLP method, Sentence conversion using Bi-LSTM and Text and audio output in mobile device. These modules are combined to form a smart sentence conversion system for Indian Sign language and can help deaf and speechless people to communicate with ordinary people in an easy way.

The proposed system design for our project is showed in Fig 2 in the block diagram. The Indian sign language sentences are not in the correct grammatical form and not in the meaningful format. The Aim of the proposed system is to convert those sentences into correct grammatical form. For that the specific set of sentences are chosen and converted into a correct format by applying conventional NLP method and Bidirectional Long Short-Term Memory (Bi-LSTM) method. Later by comparing the efficiency of both methods and passed into the android mobile through Bluetooth module and displayed in the android application as well as in text to speech format.
Sentence Generation using conventional NLP module involves the conventional NLP methods like POS tagging, grammar designing, parsing using LR parser. By tagging each word in the sentence by using the POS taggers. With the help of those tags the respective position of each words in the sentence get analysed. With respect to the grammar rules designed the tense of the sentence got identified. By using LR parser, the parse tree is constructed and convert the sentence into the meaningful format.

In this proposed system, sign sentences are passed into the NLP engine where each sentence gets tagged and gives some description. Basically, this tagging process is of two stages: tokenization and tagging. The sentence gets tokenized and is used to find the corresponding parts of speech of each and every token in the sentence. This tagged sentence is further analysed with the designed grammar rules in the NLP engine for arranging the tags in a meaningful position of the sentence. The proposed system utilized the Bottom-Up approach for parsing the sentences based on the defined grammar rules. Here LALR parser is one of the shifts-reduce parser which parses the sentence in an efficient way. With the help of the POS tags parse tree got constructed and structured the sentence appropriately.
Sentence Generation using Bi-LSTM module the same set of sign sentences and its respective output are passed as an input into input cell of the Bi-LSTM model. The model gets trained and learnt the features. Based on the training, the testing sentence got converted to the meaningful format.

Often, as in the NLP, in order to understand a word, we need not only the previous word but also the word to come. For that Bi-LSTM is a general architecture that use any RNN model. For finding the word we need to apply forward propagation ‘n’ times, where ‘n’ represents the no of cells. Then the activation functions are applied for combining the states and generate the text.

Given step in time \( t_t \), the mini batch data is

\[
X_{tt} \in \mathbb{R}^{s \times i}
\]

Where \( s \) - number of samples, 
\( i \) - number of data and 
\( \phi \) - hidden layer activation function.

In the bidirectional design, we undertake that the forward looking and backward-looking hidden states to this series of time step are \( \overrightarrow{H_{tt}} \in \mathbb{R}^{s \times d} \) and \( \overleftarrow{H_{tt}} \in \mathbb{R}^{s \times d} \) respectively. Here \( d \) indicates the number of hidden elements. The calculations of the forward-looking and backward-looking hidden state renovations are as follows:

\[
\overrightarrow{H_{tt}} = \emptyset (X_{tt}\overrightarrow{W_{xd}} + \overrightarrow{H_{tt-1}}\overrightarrow{W_{dd}} + \overrightarrow{b_{d}}),
\]

\[
\overleftarrow{H_{tt}} = \emptyset (X_{tt}\overleftarrow{W_{xb}} + \overleftarrow{H_{tt+1}}\overleftarrow{W_{dd}} + \overleftarrow{b_{d}}).
\]

In this, the weight specifications \( \overrightarrow{W_{xd}} \in \mathbb{R}^{i \times d} \), \( \overrightarrow{W_{dd}} \in \mathbb{R}^{d \times d} \), \( \overleftarrow{W_{xd}} \in \mathbb{R}^{i \times d} \), and \( \overleftarrow{W_{dd}} \in \mathbb{R}^{d \times d} \), and bias specifications \( \overrightarrow{b_{d}} \in \mathbb{R}^{i \times d} \) and \( \overleftarrow{b_{d}} \in \mathbb{R}^{i \times d} \) are all bidirectional model specifications.

Later we concatenate the forward-looking and backward-looking hidden states \( \overrightarrow{H_{tt}} \) and \( \overleftarrow{H_{tt}} \) to gain the hidden state \( H_{tt} \in \mathbb{R}^{s \times 2d} \) and pass it to the output layer. In deep bidirectional recurrent networks, the data is passed on as an input to the upcoming bidirectional layer. Last, the output layer enumerates the output \( O_{tt} \in \mathbb{R}^{s \times u} \) (number of outputs: \( u \)):

\[
O_{tt} = H_{tt}W_{du} + b_u.
\]
The weight specification $W_{du} \in \mathbb{R}^{2d \times u}$ and the bias parameter $b_u \in \mathbb{R}^{1 \times u}$ are the model specifications of the output layer. The two directions can have different numbers of hidden elements.

Text and audio output in mobile device module, the android device is connected to the pc through the Bluetooth. An Android application is developed to display the generated sentence along with audio output. With this text to speech functionality is included. This makes interaction much easier and comfortable for the deaf and dumb people to communicate with the normal people. Sign sentences which is used by the deaf and speechless people in their day to day life is in the gesture format that are randomly collected from the Indian sign language webpages.

Implementation

Parts of Speech (POS) tagging is the technique of NLP and it is salient method of language processing. The sentences are fed to NLP engine. The tag may represent one of the parts-of-speech tags like noun (NN), preposition (Prep), pronoun, Adjective (Adj), verb(V), adverb (Adv), conjunction and interjection.

The input is split into tokens to detect the equivalent parts of speech of each and every word in the sentence.

Algorithm for Parts of Speech Tagging Using Wordnet

WordNet Parts of Speech tagging Algorithm:

1. Take I as input and mentioned list of POS tags such as Noun, Verb, Adjective, Adverb, Preposition.
2. Find the words for mentioned tags in the training data corpus.
3. Calculate the probability of tags for every words in given data corpus are as follows:

$$P (ta | iw) = \frac{c (iw, ta)}{c (iw, t1) + \ldots + c (iw, tn)}$$

Where $iw =$ input word $M$, $ta =$ tag for input word $I$, $c (iw, ta)$ is number of times appear in corpus

4. Tag with parts of speech to the word occur with maximum probability.

After tagging each word in the sentence, set of grammar rules designed are applied on the sentences. These rules operate together for composing the viable sentences. Here the
simple and facile form of grammar called context free grammar is used in the propounded system. Because it is facile to handle and write the grammar for people.

Bottom-Up parsing is precisely known as parsing algorithm and it is needed in the shift-reduce parsers. The parser begins with the input words in this parsing and attempts to construct trees from the words in an upward direction by applying rules from the built grammar. In the lexicon, the parser starts by looking up each word and creating a tree with the part of speech for each word.

Tactile sign language is not composed of words such as is, am, are and article. In order to create a coherent sentence parse tree, terms and articles are placed in the appropriate positions.

The Bi-LSTM model were configured as follows:

1. The maximum no of sentences was 500
2. There was a Bi-LSTM layer with LSTM units
3. There was a fully connected layer
4. There was a SoftMax activation function
5. The loss function was “Categorical-Cross entropy”.
6. The optimizer was “Adam”

Working for Bi-LSTM Model

**Input:** Tactile sign sentences are passed in to the model

**Output:** Generated sentences in a meaningful format

1. The input is shredded between -1 and 1, with the tanh activation function. Could this be represented as

   \[ iw = \tanh (b^a + g^U^a + h_{t-1}V^a) \]

   Where \( U^a \) and \( V^a \) are the previous weights of cell output and inputs. Parallelly, \( b^a \) is executing as an input bias.

   Note, only input weights are considered by the exponents(a).

   \[ O = \sigma(b^o + y^U^o + h_{t-1}V^o) \]

   The LSTM segment output is

   \[ iw \odot O \]
Where \( \circ \) is element-wise multiplication.

2. Forget state gate and loop: the forgotten gate output expression is

\[
fr = \sigma (b^{fr} + k_t Ur^{fr} + h_{t-1} V^{fr})
\]

The performance of the product shows the location of the forgotten gate and prior state.

For this calculation the equation is

\[
J_{t-1} \circ fr
\]

In another technique the performance of forgotten loop is determined. For distinct time frames

\[
J_t = J_{t-1} \circ fr + iw \circ O
\]

3. Output gate is evaluated as

\[
P = \sigma (b \circ X_t U \circ + h_{t-1} V \circ)
\]

The final outcome of the cell can be expressed as tanh shredding

\[
ith_t = tanh(J_t) \circ P
\]

**Results and Discussion**

The result of executing POS tagging algorithm for few sign sentences are shown in table 1.

The proposed system uses the Bottom-Up technique and it uses the shift-reduce parsers. In this LALR parser as a Bottom-Up parser. It is a simpler LR parser version for separating and evaluating the sentence. The approach can be explained with the help of parse tree.

The same sign sentences are passed into the Bi-LSTM model along with its meaningful format to train the model. There are totally 500 sentences are passed as an input in to the model. The model learned the features of the sentences based on training, thereby the new sentences get generated into a meaningful format. The 70 per cent of the data is taken as a training set and the balance data is taken as the model's test set.
Table 1 Result of POS tagging

<table>
<thead>
<tr>
<th>Input Sentences</th>
<th>Generated Tags for the sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>I see you all</td>
<td>[('I', 'PRP'), ('SEE', 'VBP'), ('YOU', 'PRP'), ('ALL', 'DT')]</td>
</tr>
<tr>
<td>Come here today</td>
<td>[('COME', 'NNP'), ('HERE', 'NNP'), ('TODAY', 'NNP')],</td>
</tr>
<tr>
<td>Now see me talk Indian sign language</td>
<td>('NOW', 'NNP'), ('SEE', 'NNP'), ('ME', 'NNP'), ('TALK', 'NNP'), ('INDIAN', 'NNP'), ('SIGN', 'NNP'), ('LANGUAGE', 'NNP')</td>
</tr>
<tr>
<td>Have good time, where you go?</td>
<td>('HAVE', 'NNP'), ('GOOD', 'NNP'), ('TIME', 'NNP'), (',', '.'), ('WHERE', 'NNP'), ('YOU', 'NNP'), ('GO', 'NNP'), (',', '.'))</td>
</tr>
</tbody>
</table>

The input sentences are passed into the model via the input gate of the Bi-LSTM model. The process done in both forward and backward direction. The possible output vectors are computed by the LSTM cell of the model.

Meanwhile, the LSTM cell is composed of three separate gates: Forget gate, Output gate and Input gate for learning the features of the sentences in to a vector state.

In these three gates, the features are learned and by applying the activation function the output sequence are produced.

The table 2 represents the comparison of generated sentences by using conventional NLP model as well as Bi-LSTM model.

Table 2 Comparison of Generated sentences by using the two models

<table>
<thead>
<tr>
<th>S. No</th>
<th>Input sequence</th>
<th>Actual Output</th>
<th>Generated Output Sequence of Conventional model</th>
<th>Generated Output Sequence of Bi-LSTM model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Long-time see not</td>
<td>I have not seen you for a long time</td>
<td>Long-time no see</td>
<td>I have not seen you for a long time</td>
</tr>
<tr>
<td>2</td>
<td>I chop wood</td>
<td>I chop wood</td>
<td>I chopped wood</td>
<td>I chop wood</td>
</tr>
<tr>
<td>3</td>
<td>I go walk</td>
<td>I go for a walk</td>
<td>I am going for a walk</td>
<td>I go for a walk</td>
</tr>
<tr>
<td>4</td>
<td>I fond swim.</td>
<td>I like to swim</td>
<td>I like swim.</td>
<td>I like to swim.</td>
</tr>
<tr>
<td>5</td>
<td>I watch all</td>
<td>I watched you all</td>
<td>I will watch</td>
<td>I watched you all</td>
</tr>
</tbody>
</table>

From the above table it shows that the Bi-LSTM model will produce the appropriate meaningful sentences. The figure 3 represents the accuracy graph for the Bi-LSTM model. The generated sentences from the proposed system are transmitted through
Bluetooth. By comparing the efficiency, the Bi-LSTM model is better than the conventional model. So, the output generated by the Bi-LSTM model is connected to the android device via Bluetooth and pass the output. The generated sentence is displayed in the text field of the android application.

![Figure 3 Bi-LSTM accuracy and loss](image)

The application contains the text-to-speech feature to produce audio output of the sentence displayed in the application with the in-built speaker of the mobile device. The following figure 4 represents the UI design of the android application.

![Figure 4 Android Application UI Design](image)

The Android application will help to display the generated meaningful sentences with the audio output. The figure 5 represents the generated output sentence.
The table 3 describes the evaluation by comparing the two methods. With elevated training data, accuracy can be better.

Table 3 Evaluation comparison of both conventional and Bi-LSTM methods

<table>
<thead>
<tr>
<th>S.No</th>
<th>Method Used</th>
<th>Accuracy</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Conventional NLP</td>
<td>0.906</td>
<td>0.933</td>
<td>0.953</td>
</tr>
<tr>
<td>2</td>
<td>Bi-LSTM model</td>
<td>0.956</td>
<td>0.975</td>
<td>0.964</td>
</tr>
</tbody>
</table>

In contrast with the existing systems, the proposed system has achieved more precision. The comparison of the accuracy of the propounded system with the current system is listed in table 4.

Table 4 Comparison of propounded system accuracy with other current existing systems

<table>
<thead>
<tr>
<th>Author of the system</th>
<th>Methodology used</th>
<th>Accuracy achieved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampada S. Wazalwar</td>
<td>Conventional NLP</td>
<td>89.5%</td>
</tr>
<tr>
<td>Urmila Shrawankar</td>
<td>Sensor glove, NLP</td>
<td>92.9%</td>
</tr>
<tr>
<td>Sumeet R. Agarwal</td>
<td>Image processing, NLP</td>
<td>84.3%</td>
</tr>
<tr>
<td>Sayli Dixit</td>
<td>Sensor gloves, NLP</td>
<td>91.5%</td>
</tr>
<tr>
<td>Kumud Tripathi</td>
<td>OpenCV, NLP</td>
<td>91.2%</td>
</tr>
<tr>
<td>Proposed system</td>
<td>Conventional NLP</td>
<td>90.6%</td>
</tr>
<tr>
<td></td>
<td>Bi-LSTM model</td>
<td>95.8%</td>
</tr>
</tbody>
</table>
This paper explains, a smart sentence generation system for Indian Sign Language which is designed for interpreting the sign words of Indian sign language into a meaningful sentence. This system helps to generate the sentence in a correct grammatical format by using NLP. The sentence is processed by both conventional and deep learning-based NLP methods. In conventional way, it generated the sentence by parsing the sentence based on the POS tagging method. The sentence created by the use of the bidirectional long short-term memory recurrent neural network in deep learning. Later the comparison is done measuring the efficiency of the above two methods. With the help of the android application the sentence gets displayed in text as well as audio.

References


