TO WHOMSOEVER IT MAY CONCERN

This is to certify that the manuscript **Literature Review on Sign Language Generation**, is written by **Sayali Oak, Tanvi Shroff, Anagha Kulkarni, Rutuja Jadhav and Vedanti Donkar, Department of Information Technology**, from **MKSSS's Cummins College of Engineering for Women, Pune**. The paper is their original work and contribution. This paper is not submitted or accepted for publication anywhere else.

Sayali Oak,
Student,
Department of Information Technology,
Cummins College of Engineering for Women, Pune.
sayali.oak@cumminscollege.in

Literature Review on Sign Language Generation

Sayali Oak*[1], Tanvi Shroff*[2], Anagha Kulkarni*[3], Rutuja Jadhav*[4], Vedanti Donkar*[5]

*Department of Information Technology,
Cummins College of Engineering for Women, Pune, Maharashtra,
India-411052

[1]+91-8237883594
[2]+91-9527976425
[3]+91-7620319904
[4]+91-8793512957
[5]+91-7387158887

sayali.oak@cumminscollege.in
tanvi.shroff@cumminscollege.in
anagha.kulkarni@cumminscollege.in
rutuja.jadhav@cumminscollege.in
vedanti.donkar@cumminscollege.in

ABSTRACT

The deaf and dumb community uses sign language to communicate. Sign language is a language of signs including facial expressions and not of spoken words. It is a visual mode of communication. The position of hands, the movement of fingers and the expressions on the face play a vital role in sign language. Sign languages have a very limited set of words. The grammar is difficult to understand. On the contrary, spoken languages across the globe have a rich vocabulary. It is difficult for signers to understand a spoken language. There is a need to develop a system that establishes a link between spoken and sign languages. Translating from spoken languages to sign languages is a challenging task. This paper presents the state-of-the work that has been done in the field of translating English (a spoken language) to Indian Sign Language.

Keywords: Generative Adversarial Network (GAN), Hidden Markov Model (HMM), Indian Sign Language (ISL), Language Translation, Sign Language (SL).

1. INTRODUCTION

Communication has played an important part in our species' evolution. since the beginning of time. It is a vital part of daily human interaction. Since spoken language has always held a dominant position in our community, hard of hearing people have felt isolated and helpless.

Furthermore, the majority of information produced for the general public is disseminated by written or verbal contact. As a result, those who rely on lip-reading or other visual stimuli to understand common information in communal areas like banks, post offices, airports, restaurants, hospitals and news broadcasts have difficulty understanding general information. Rudimentary means of communication for the deaf community has always been through sign language (SL).

Two deaf people can communicate using SL if they do not have any other physical disabilities, but communication between hearing and deaf people is filled with difficulties.

As a result, the hearing person's spoken language must be translated into the SL and vice versa. This requires the presence of a human being who has knowledge of both languages. This is not always feasible and even if it is, it is expensive. Also, the presence of a translator would violate lawyer-client or doctor-patient confidentiality. As a result, a pragmatic method is necessary to meet people's daily demands.

Machine Translation and Natural Language Processing have broken this language barrier by automating the process of translation[1, 2, 3]. This development is crucial for the deaf community, as it offers a forum for connecting hearing-impaired and hearing people, as well as equal access to information for the hard of hearing community.

SL is a form of communication that uses gestures to communicate [4, 5, 6, 7, 8, 9]. It was created to assist persons who have hearing loss. They communicate with people using hand movements, facial expressions, body gestures, etc. SL is a multifaceted natural language that has a certain grammar, vocabulary, syntax, and sentence structure [10].

Every language has a predetermined vocabulary that is broken down into phonemes, whereas signs in SL are broken down into cheremes [10]. Manual and non-manual are the two types of cheremes. Hand form, placement, orientation, and movements are examples of manual cheremes, whereas non-manual cheremes include eye movements, facial expressions, and head and body positions.

There is an urgent need to translate spoken languages to SL. The next section presents the most recent research in the field of translating the English language to ISL. In section 3, the presented literature is analysed. The conclusion is presented in section 4.

2. LITERATURE SURVEY

Researchers have been researching SL generating technologies for the past couple of decades.

The researchers propose a system based on the Hidden Markov Model for translating spoken language to SL [11]. The system first recognizes the audio input to convert audio to text in the form of words. Every word has confidence associated with it to indicate the probability with which that word occurs. Using rule-based translation, the researchers convert these words to signs. The signs are converted to 3D animation using Signing Gesture Markup Language. However, manual efforts are required to prepare a database. The paper also considers only a few rules for translation. The paper also does not present much work on facial expressions.

SIGNGAN processes audio input and converts it to text using a progressive transformer [12]. The text is converted into a sequence of poses in the form of outlines. The poses in the form of outlines are processed by pose-conditioned human synthesis models to convert them into SL videos. Generative Adversarial Network (GAN) is used for generating gestures. The system has the ability to avoid blurry poses. However, the paper focuses only on hand gestures. Some work is required to handle facial expressions.

[13] uses video representation of words according to the ISL dictionary as the dataset. The videos were cleaned which doubled the size of the dataset. The frames for each video were further extracted and their skeletal poses were obtained using OpenPose which were concatenated according to the order of the frames. This is used to train the GAN. The input English text goes through the CoreNLP parser which outputs phrase structure as indicated by the ISL language rules. The sentence is organized in a Subject-Object-Verb design subsequent to applying the ISL rules. The stop words are taken out and assuming a word is absent in the word reference then its equivalent is utilized. The words are further lemmatized using the Textblob NLP technique. To generate skeletal poses, a total of 124 joint directions are removed from the recordings utilizing OpenPose. Here, Pix2Pix GAN is used which without any noise learns the mapping of the input image with the output image. U-Net is utilized as the generator and Patch GAN is utilized as the discriminator. The Patch GAN, instead of predicting if the whole image is real or fake, predicts if a patch of the picture is genuine or counterfeit by outputting a feature map. After the text is converted into gloss, frames for the corresponding glosses are generated. This is fed as an input to the generator which outputs a realistic video. Here, OpenPose sometimes failed to detect the finger joints correctly so the output video was distorted.

The paper [14], unlike [12], translates the speech input directly into a sequence of poses using a multi-tasking transformer system coupled with a cross-modal discriminator. In addition, to aid the transformer in learning high-quality speech to SL translations, a cross-modal network is used as a discriminator. The voice segments and the matching sign pose sequences are matched using this network. This model does not focus on non-manual features like facial expressions.

In [15], the input speech/text is converted to its equivalent gloss using approaches Automatic Speech Recognition (ASR) and Neural Machine Translation literature. This gloss output is then converted into skeletons using key points. The skeletons undergo video synthesis using the human pose estimation algorithm which outputs a video of a real person performing hand gestures similar to the skeletons. This paper uses a subset of the entire dataset to test and train the model.

The ZARDOZ system architecture is explained in figure 1 below[16]. In this system [16], the input text is processed by swarming Lexperts, that use grammatical rules and heuristics to recognise complex word structures. A unification grammar is then used to parse the processed text which results in a deep syntactic/semantic representation. A deep syntactic/semantic representation is generated by decoding the processed text with a unification grammar. The unification structure is used to create a first-cut Interlingua representation; however, this portrayal is language independent because the source language's metaphoric and metonymic structures have been eliminated via schematization. The finely tuned interlingua feeds the speech tracking network, which performs anaphoric resolution before it is carried to the system's generation panels: the sign syntax agency which assigns concept-to-sign correlations to the interlingua structure’s tokens. The interlingua is converted into a stream of sign tokens by syntax and mapping agencies, and the DCL animator compiles that stream into a Doll Control Language (DCL) programme.
Using the DCL programme, an animated doll appears on the screen, articulating the suitable gesture sequence to the user.

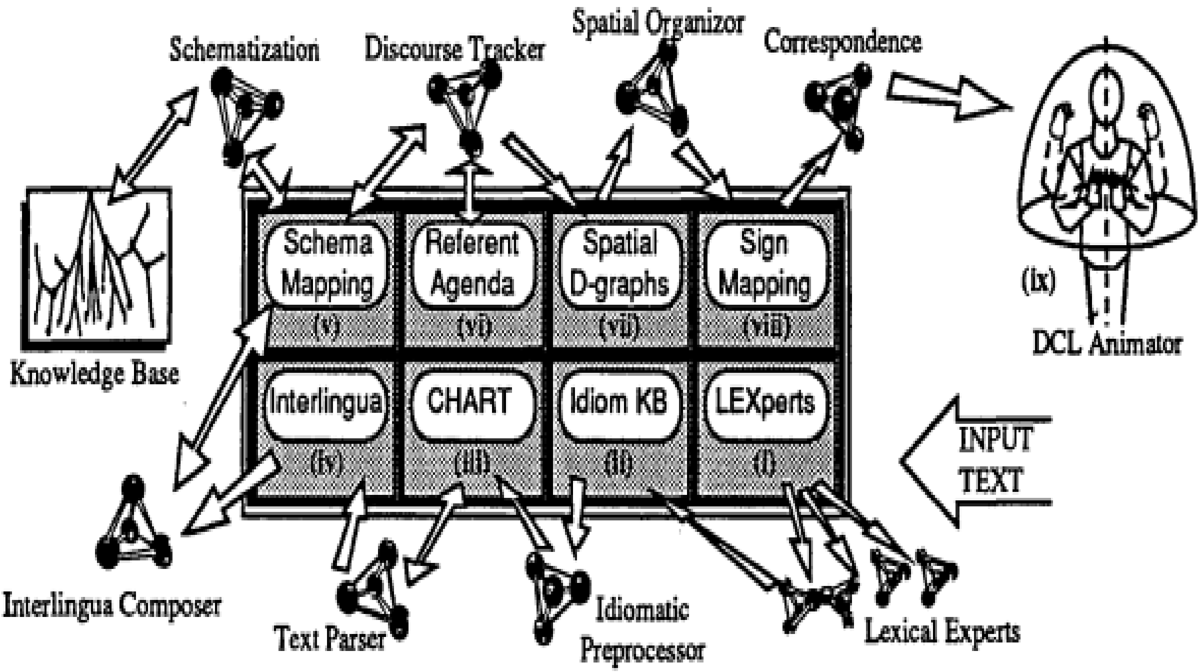


Figure 1: ZARDOZ architecture

The model proposed in [17], takes input speech from the user which is changed over into text utilizing google cloud speech API. This text is then tokenized using Natural Language Processing (NLP) and Machine Learning (ML) tools and then is converted into a parse tree using Stanford parser. This parse tree is then transformed to match the ISL grammar rules. This output then goes through data preprocessing where the stop words are eliminated and the words are stemmed and lemmatized using the NLTK library. These output tokens, as shown in figure 2 below, are then translated into SL for which the corresponding Hamburg Notation System (HamNoSys) is retrieved from the database. Then the corresponding Signing Gesture Markup Language (SiGML)files are retrieved if present, otherwise the words are fingerspelled letter-by-letter. A graphic generator then generates the avatar animation using the SiGML file. Here, only Chrome and Firefox browsers can be used since none of the others support speech-to-text API. The hand movements of the 3D avatar are not as clear as compared to the hand movements of realistic video output.

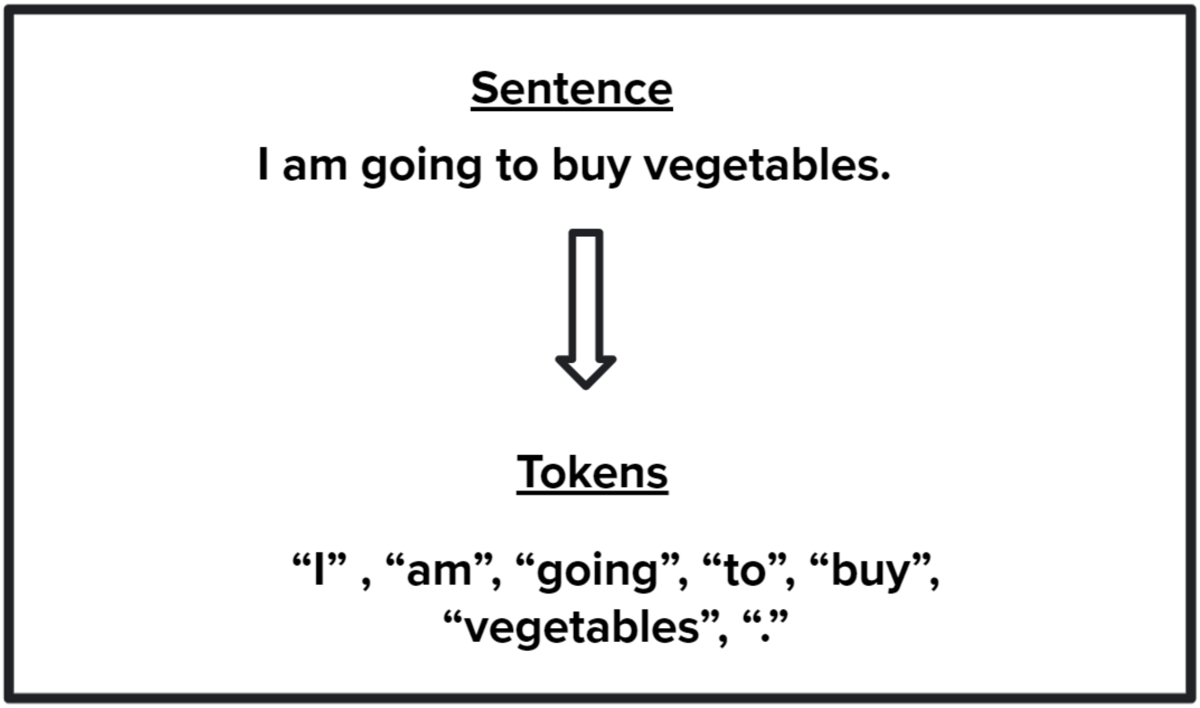


Figure 2: Tokenization

In [18], the youtube videos are downloaded as the input along with the subtitles. The discourse input is changed over to message and converged with the captions. The text is restructured according to ISL rules and the keywords are extracted. These words are then mapped to 3D avatar gestures stored in the database. Although if the word is not there in the database of gestures then what action is taken is not mentioned. This paper also uses gestures recorded based on a 6th-grade solar system video so the domain of this is extremely restricted.

The system proposed in [19] supports both English and Hindi texts. Numbers, words, and sentences can all be used as input. There are two options for Hindi input. The user can either use transliteration or directly enter Hindi words or sentences into the interface. The next step is to parse the input using UDPipe (a natural language processing toolkit). As root words, labels, and tense data, the handled results are coordinated into an information structure. This data structure is used to process words according to ISL grammar rules for tense, aspect, and modality. Subject–Object–Verb is the order of processing. A data set of the ISL Dictionary is grown physically through the administrator board by the SL specialists. The framework extricates the HamNoSys from the information base for each root word in the ISL sentence. Following that, the framework produces SiGML documents from its HamNoSys. Each SiGML document represents a word's sign. For further processing, the SiGML document generated by HamNoSys is utilized as information. For the purpose of generating animation frames, the animation server uses it. Moreover, these casings are utilized to characterize a symbol's posture. A 3D avatar animation is used to play the sign for the user through these frames. The advantage here is that this system uses HamNoSys in real-time to render gestures rather than pre-recorded videos, allowing for greater symbol generation flexibility. The dictionary, on the other hand, is made physically, which is a tedious cycle. Furthermore, the system searches for the HamNoSys of each word, which is a time-consuming process.

In this proposed system [20], the info text is first gone through the preprocessing unit, which attempts to recognize frozen phrases (phrases composed of idioms and metaphors) and temporal expressions (related to day, time) that the syntactic parser can't distinguish. A parse tree is used to create a dependency structure after the text is parsed with the Minipar parser. The dependency structure generated by Minipar is more similar to the functional structure of Lexical Functional Grammar (LFG) (fstructure). The f-structure encodes the input sentence's grammatical relations (such as subject, object, and tense). During the generation stage, the English f-structure is changed over to the ISL f-structure using appropriate transfer syntax rules. During the generation stage, 2 major tasks are carried out: a) lexical selection and b) word request correspondence. As ISL lacks articles and conjunctions, they are omitted from the generated output. Limitations of this system are that it takes input as simple sentences only(with one main verb)

This system was developed for the Indian Railway reservation counter [21]. This system, named INGIT meaning “signed” in Sanskrit [21], utilizes a conventional methodology that creates the semantic construction straightforwardly where conceivable and falls back to a compositional model for the rest. Input Parser, Ellipsis Resolution Module, ISL Generator (counting ISL vocabulary with HamNoSys phonetic portrayals) are the framework's fundamental modules.

INGIT accepts interpreted communication in language strings that can be labelled for sound examples as info. It only supports one of these tags, the "?" tag. For inputs that match a unit development (or expressions in creation that match such a development), the immediate semantics map for the information will be produced promptly, or then again if a sub-structure, it will be passed to the proper design. If the semantic structure is swamped, it is passed to the ISL generator; otherwise, it is passed to the ellipsis resolution module, which fills in any missing parameters. Once the ISL gloss is ready, it is given as input to the ISL generator to produce signs. This system uses the HamNoSys Notation. The corpus for validating this system was small. It constituted 230 utterances (many repeated) and a vocabulary of only 90 words.

The proposed system here consists of three parts [22]. 1.) an interface where the user can enter English text, 2.) a language processing system that translates English text to ISL format as well as a virtual avatar that serves as a translator at the interface. To translate the input English text into ISL various steps are followed such as Part of Speech tagging followed by an Eliminator which removes the unwanted tokens from the input text followed by Stemming which converts the verb in the input text into its corresponding root form.

Figure 3 below shows Part of Speech tagging and Stemming. Here the sentence “I am buying vegetables” is broken down into parts of speech where “I” is the pronoun, “am” is the auxiliary verb, “buying” is a verb and “vegetables is a noun. Then in the stemming part, the words are shortened to convert them into the root form. The word “buying” is converted into “buy” and “vegetables” is converted to “vegetable”. Unwanted words like “am” are removed and the words you get after this are “I”, “buy”, “vegetable”.

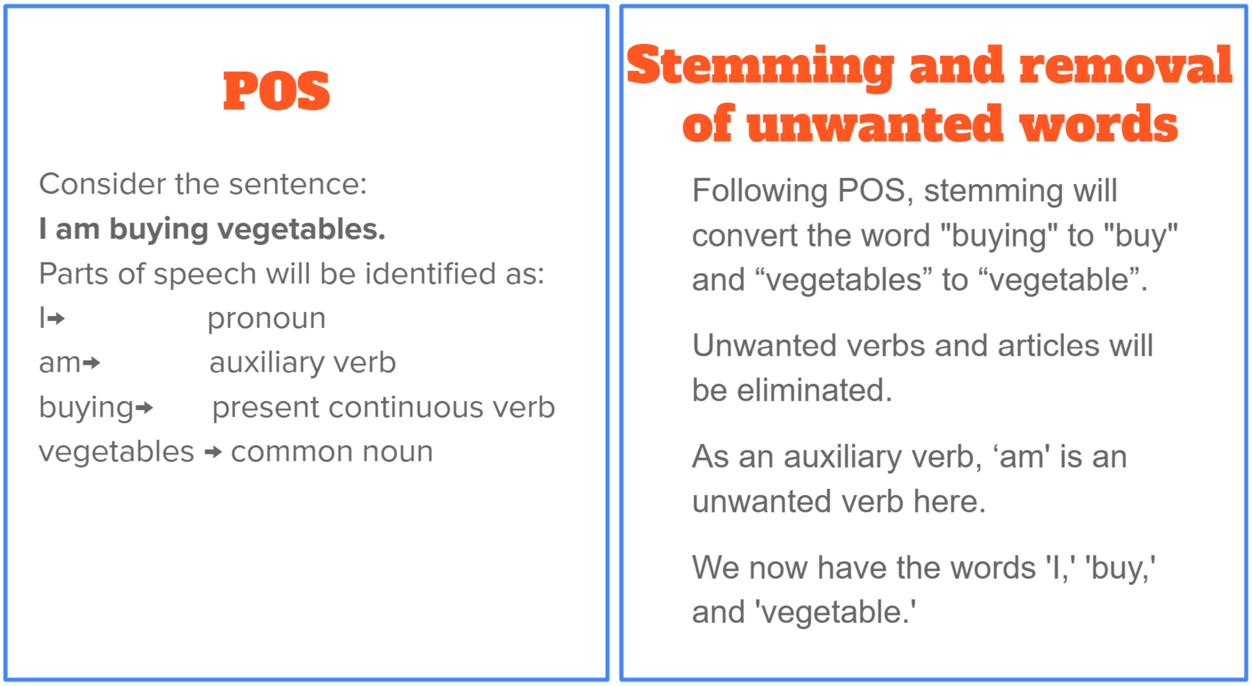


Figure 3: POS and Stemming

Lastly, Phrase Reordering is done where the sentence structure is changed based on the ISL grammar regulations. ISL essentially imitates a Subject-Object-Verb Agreement. Once the input text is translated into ISL it will be queried in the database which has the notation format for every word. This is LOTS Notation which represents characteristics such as Hand location(L), Hand Orientation (O) in the 3D space, Hand Trajectory movement (T), Hand shapes (S) and non-manual elements. The Loader Module queues the sequence of notations based on the order of occurrence. The Animator produces the animation of the input sentence. In this system, the speed of the animation is increased as the inter sign transition gesture is inserted to reduce gesture jitters.

This paper specifically focuses on the representation of SL [23]. This system is a Java Avatar Signing (JASigning) synthetic SL animation system. The avatar software in this system is implemented in Java. SiGML, an XML application closely based on the HamNoSys, is the input for a JASigning application [24]. The signing avatar in this system is formulated on conventional 3D computer animation methods. These techniques are supplementary data files included in the system that define the avatar qualities that are required to increase the system's SL performance. This system uses AnimGen software which is used to generate a sequence of animation frames. JASigning System is particularly like SiGMLSigning System as far as its capacities, however, the vital contrast between SiGMLSigning and JASigning is that the SiGML Signing System could run uniquely on Windows PC frameworks, while JASigning System can be conveyed on numerous platforms.

The system proposed in [26] takes the railway announcements as the input and produces a virtual depiction of the corresponding ASL signs using animation. Figure 4 shown below takes input and uses rule-based and phrase-based algorithms to convert it into the required format for ASL generation [26].

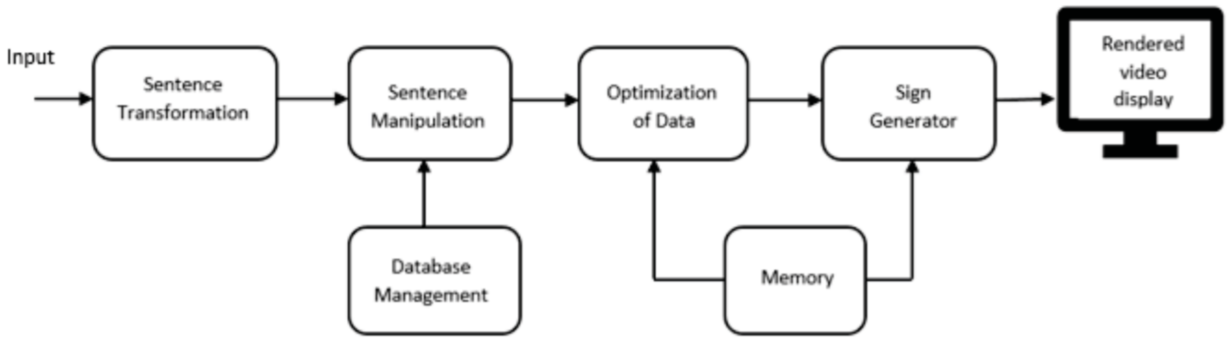


Figure 4: Block Diagram

The database management stores the data about the ASL gloss. To augment the productivity of the system, the glosses which do not exist are also explicitly mentioned in the database. The next step is optimizing the data. This is used to transform the data into the required frames for the SL after removing the redundant words. In the memory, the data related to the corpus is saved which gives information about the parameters of the active bones which are found after manually trying it on the avatar. This data is given as an input to the 'Sign Generator' and the 'Optimization of Data' block. The sign generator produces a video file for every word input and saves it in the required format. Blender software, which runs python scripts, is used for the generation of animation videos. This video of the animated avatar is then displayed on the user interface as the output.

In [27] a mobile application called "Wadhan" is created to aid communication between the hard of hearing and hearing people during emergency situations. This application transforms the gestures into speech and vice versa. Speech Recognition is done using the library created by CMUSphinx 4 toolkit. This library has a limited vocabulary since it is only meant for emergencies. The input training dataset was created manually by recording different sentences that are used in emergency situations. Figure 5 below shows the block diagram of the system proposed. The input audio is first cleaned to remove any noise and then it is transformed into text by using the created library. This text output is then taken as the input by the animation generator which splits the sentences into words using NLP. The animation for the respective words is taken and compressed into one video using the Pillow library in python and the GZip algorithm. If the sign for a particular word is not present then it is fingerspelled. This video is then displayed on the app's user interface.

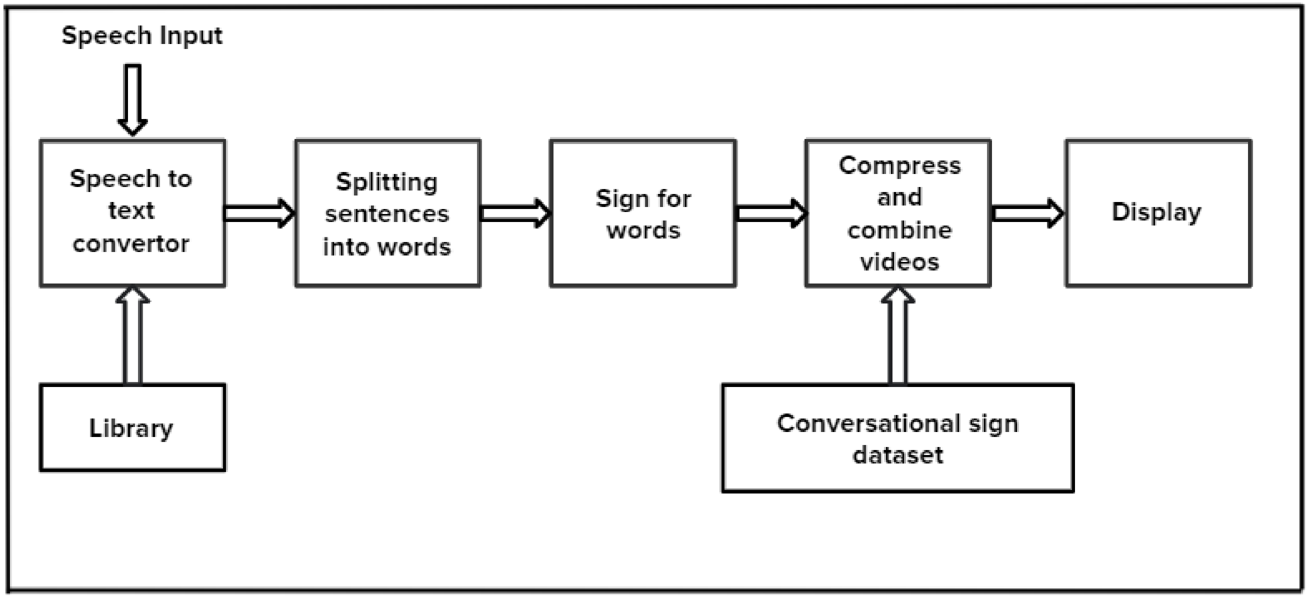


Figure 5: Block Diagram of “Wadhan”

3. DISCUSSION

In this section, we compare the different approaches discussed in section 2.

Methods used in [12], [13] use a video dataset to train the generator to produce realistic videos. Realistic video means that the video has a life-like person signing as opposed to a 3D animated avatar. So when the text input is given to the GAN, the output generated by it is a realistic video of the corresponding sign. In [11], [17], [19] the HamNoSys is retrieved for the corresponding text. The HamNoSys is then converted into a SiGML file. This is further converted into a 3D animation. Figure 6 below [25] shows the process of converting the word “time” to its corresponding HamNoSys which is further converted into a SiGML file. This file is used to produce the corresponding 3D animation.

If HamNoSys or the SiGML file is not present then it needs to be created. This takes some time and effort as the HamNoSys for each word has to be prepared manually. In GANs, adding non-manual features is easily possible and more comprehensible because it can be trained with realistic videos. On the other hand, there is not much research on HamNoSys notations for non-manual features. But retrieving the HamNoSys of a word and the HamNoSys of its corresponding non-manual feature would be a hassle.

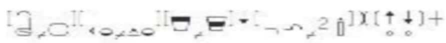
Word	HamNoSys	SiGML	Avatar Animation
Time		<sigml> <hns_sign gloss="time"> <hamnosys_nonmanual> </hamnosys_nonmanual> <hamnosys_manual> <hamparbegin/> <hamfinger2/> <hamthumbacrossmod/> <hamfingerbendmod/> <hamplus/> <hamfist/> <hamparend/> <hamparbegin/> <hamextfingerl/> <hampalmd/> <hamplus/> <hamextfingero/> <hampalmd/> <hamparend/> <hamparbegin/> <hamshoulders/> <hamplus/> <hamchest/> <hamparend/> <hammoved/> <hamparbegin/> <hamwristback/> <hamhandback/> <hamplus/> <hamindexfinger/> <hamfingertip/> <hamparend/> <hamtouch/> <hamseqbegin/> <hammoveu/> <hamsmallmod/> <hammoved/> <hamsmallmod/> <hamseqend/> <hamrepeatfromstart/> </hamnosys_manual> </hns_sign> </sigml>	

Figure 6: Conversion of English text to SL

In [16], a method called Doll Control Language (DCL) is used. Here, the refined interlingua (language independent) is passed to the system's generation panels. The language-independent structure is converted to a horizontal output sequence of tokens using the mapping and syntax agencies. This is further assembled by the Doll Control Language (DCL) animator into a DCL programme.

On the other hand, systems in [17], [19], [21], [23] use the HamNoSys method. Unlike in DCL, this is based on the source language itself and not an intermediary language. SL can be represented in any region using HamNoSys as it is a universal transcription.
The signing order is defined using SiGML, which is used to depict each symbol of HamNoSys in XML format. The avatar reads the SiGML input and produces the animation frames.

The latest version of the previously manufactured synthetic 3D avatar signing model, SiGMLSigning, is JASigning. The framework utilizes SiGML as its feedback input. The AnimGen software receives the last Gestural SiGML sign sequence, which generates movement information for the avatar. JASigning System is particularly like SiGMLSigning System as far as its capacities, however, the vital contrast between SiGMLSigning and JASigning is that the SiGML Signing System runs uniquely on Windows PC frameworks, while JASigning System can be conveyed on numerous platforms. The JASigning system can work with both web and desktop applications that permits the clients to have a virtual human, or avatar, play out an SL arrangement of a sentence. None of these papers has much focus on the non-manual features like facial expressions, body gestures, etc. which are also a crucial part of SL. Also, they have not proposed a system or method to handle data securely. The SiGML files for different words are uploaded to the server. If someone gets unauthorized access to this server, then this will lead to a loss of confidentiality. For example, the conversation between a doctor and patient is private and hence must be protected against any unauthorized access.

Paper	**Advantages**	**Limitation**
[11]	The recognizer of the system has been trained with many speakers so it is speaker-independent.	Creating a HamNoSys for every word which is not present takes up a lot of time.
[12]	The grade of hand image synthesis is improved with a new key point-based loss. Training on big and diverse datasets is possible because of controllable video production.	Does not take facial expressions into consideration.
[13]	The output video is realistic as opposed to a 3D avatar.	In several cases, OpenPose was unable to recognise the skeletal joints of the person's fingers in the image.
[14]	Speech input is directly translated into a series of poses.	No focus was given to non-manual features
[15]	The study's findings show that generated films outperform skeletal representations.	The model is incapable of producing high-quality hand images.
[16]	The entire infrastructure has been implemented, including parsing, interlingua creation, and an animation component.	There hasn't been a thorough grammar produced for all languages.
[17]	Unlike present systems, which have distinct SIGML files for a different language, it also saves a substantial amount of memory.	Because none of the other browsers implements the speech-to-text API, only Chrome and Firefox can be used. When compared to the hand movements of actual video output, the 3D avatar's hand movements are not as clear.

[18]	While creating a video database is retrieving the contents of the video. This system uses youtube videos along with subtitles and hence creating a database is an easier process.	Vocabulary range limited.
[19]	Supports both English and Hindi texts. This system uses HamNoSys in real-time to render gestures rather than pre-recorded videos, allowing for greater symbol generation flexibility.	The dictionary is created manually, which is a time-consuming process.
[20]	Preprocessing unit is designed to identify frozen phrases and temporal expressions.	Input here can be only simple sentences. The system cannot handle complex sentences.
[23]	Handles data securely. The JASigning System is available on a variety of platforms.	The avatar has violent elbow movements.
[26]	Glosses that do not exist are also explicitly mentioned in the database to augment the productivity of the system.	Animated avatar makes output difficult to understand. Limited vocabulary.
[27]	Model accuracy is 68.21%. User-friendly app and reasonable processing time.	Limited vocabulary.

4. CONCLUSION

This paper presents a literature survey on SL generation. There are numerous sign languages across the globe. This paper focuses on the research work based on generating Indian Sign Language from English. Primarily, two steps are involved in sign language generation - converting English sentences to Indian Sign Language vocabulary, which is limited, and animating these Indian Sign Language words. Various methodologies used by researchers for the above-mentioned steps are discussed in the paper. The authors have identified some limitations in the implementations.

The authors propose to implement a system based on the literature described in this study and overcome some of the limitations discussed.

References

1. Barve P., Mutha N., Kulkarni A., Nigudkar Y., Robert Y. (2021) Application of Deep Learning Techniques on Sign Language Recognition—A Survey. In: Sharma N., Chakrabarti A., Balas V.E., Bruckstein A.M. (eds) Data Management, Analytics and Innovation. Lecture Notes on Data Engineering and Communications Technologies, vol 70. Springer, Singapore. https://doi.org/10.1007/978-981-16-2934-1_14
2. Patil A., Kulkarni A., Yesane H., Sadani M., Satav P. (2021) Literature Survey: Sign Language Recognition Using Gesture Recognition and Natural Language Processing. In: Sharma N., Chakrabarti A., Balas V.E., Bruckstein A.M. (eds) Data Management, Analytics and Innovation. Lecture Notes on Data Engineering and Communications Technologies, vol 70. Springer, Singapore. https://doi.org/10.1007/978-981-16-2934-1_13
3. Robert Y., Nigudkar Y., Kulkarni A., Mutha N., Barve P. (2021) Literature Survey: Application of Machine Learning Techniques on Static Sign Language Recognition. In: Abraham A., Sasaki H., Rios R., Gandhi N., Singh U., Ma K. (eds) Innovations in Bio-Inspired Computing and Applications. IBICA 2020. Advances in Intelligent Systems and Computing, vol 1372. Springer, Cham. https://doi.org/10.1007/978-3-030-73603-3_16
4. M. Taskiran, M. Killioglu and N. Kahraman, "A Real-Time System for Recognition of American Sign Language by using Deep Learning," 2018 41st International Conference on Telecommunications and Signal Processing (TSP), 2018, pp. 1-5, doi: 10.1109/TSP.2018.8441304.
5. M. R. Islam, U. K. Mitu, R. A. Bhuiyan and J. Shin, "Hand Gesture Feature Extraction Using Deep Convolutional Neural Network for Recognizing American Sign Language," 2018 4th International Conference on Frontiers of Signal Processing (ICFSP), 2018, pp. 115-119, doi: 10.1109/ICFSP.2018.8552044.
6. Stoll, S., Camgoz, N.C., Hadfield, S. and Bowden, R., 2020. Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks. International Journal of Computer Vision, 128(4), pp.891-908.
7. R. D. RAJ and A. JASUJA, "British Sign Language Recognition using HOG," 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), 2018, pp. 1-4, doi: 10.1109/SCEECS.2018.8546967.
8. J. Zamora-Mora and M. Chacón-Rivas, "Real-Time Hand Detection using Convolutional Neural Networks for Costa Rican Sign Language Recognition," 2019 International Conference on Inclusive Technologies and Education (CONTIE), 2019, pp. 180-1806, doi: 10.1109/CONTIE49246.2019.00042.
9. C. O. Sosa-Jiménez, H. V. Ríos-Figueroa, E. J. Rechy-Ramírez, A. Marin-Hernandez and A. L. S. González-Cosío, "Real-time Mexican Sign Language recognition," 2017 IEEE

International Autumn Meeting on Power, Electronics and Computing (ROPEC), 2017, pp. 1-6, doi: 10.1109/ROPEC.2017.8261606.

10. Sugandhi, P. Kumar and S. Kaur, "Indian Sign Language Generation System," in Computer, vol. 54, no. 3, pp. 37-46, March 2021, doi: 10.1109/MC.2020.2992237.
11. San-Segundo, R., Barra, R., Córdoba, R., D'Haro, L.F., Fernández, F., Ferreiros, J., Lucas, J.M., Macías-Guarasa, J., Montero, J.M. and Pardo, J.M., 2008. Speech to sign language translation system for Spanish. Speech Communication, 50(11-12), pp.1009-1020.
12. Saunders, B., Camgoz, N.C. and Bowden, R., 2020. Everybody sign now: Translating spoken language to photorealistic sign language video. arXiv preprint arXiv:2011.09846.
13. N. Vasani, P. Autee, S. Kalyani and R. Karani, "Generation of Indian sign language by sentence processing and generative adversarial networks," 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 1250-1255, doi: 10.1109/ICISS49785.2020.9315979.
14. Kapoor, P., Mukhopadhyay, R., Hegde, S.B., Namboodiri, V. and Jawahar, C.V., 2021. Towards Automatic Speech to Sign Language Generation. arXiv preprint arXiv:2106.12790.
15. Duarte, A.C., 2019, October. Cross-modal neural sign language translation. In Proceedings of the 27th ACM International Conference on Multimedia (pp. 1650-1654)
16. Veale, T., Conway, A.: Cross modal comprehension in ZARDOZ, an English to sign-language translation system. In: 4th International Workshop on Natural Language Generation, pp. 67–72 (1994).
17. Patel, B.D., Patel, H.B., Khanvilkar, M.A., Patel, N.R. and Akilan, T., 2020, August. ES2ISL: an advancement in speech to sign language translation using 3D avatar animator. In 2020 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-5). IEEE.
18. Mehta, N., Pai, S. and Singh, S., 2020. Automated 3D sign language caption generation for video. Universal Access in the Information Society, 19(4), pp.725-738.
19. Sugandhi, Parteek Kumar, and Sanmeet Kaur. 2020. Sign Language Generation System Based on Indian Sign Language Grammar. ACM Trans. Asian Low-Resour. Lang. Inf. Process. 19, 4, Article 54 (July 2020), 26 pages. DOI:https://doi.org/10.1145/3384202
20. Dasgupta, T., Basu, A.: Prototype machine translation system from text-to-Indian sign language. In: Proceedings of the 13th International conference on Intelligent User Interfaces, pp. 313– 316 (2008).
21. Kar, P. et al.: INGIT: Limited domain formulaic translation from Hindi strings to Indian sign language. In: International Conference on Natural Language Processing (2007)
22. Raghavan, R.J., Prasad, K.A., Muraleedharan, R. and Geetha, M., 2013, October. Animation system for Indian Sign Language communication using LOTS notation. In 2013 International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA) (pp. 1-7). IEEE.
23. Elliott, R., Bueno, J., Kennaway, R. and Glauert, J., 2010, May. Towards the integration of synthetic sl animation with avatars into corpus annotation tools. In 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies, Valletta, Malta (p. 29).
24. M. Varghese and S. K. Nambiar, "English To SiGML Conversion For Sign Language Generation," 2018 International Conference on Circuits and Systems in Digital Enterprise

Technology (ICCSDET), 2018, pp. 1-6, doi: 10.1109/ICCSDET.2018.8821212.
25. Sugandhi, P.K. and Kaur, S., 2018. Online multilingual dictionary using Hamburg notation for avatar-based Indian sign language generation system. International Journal of Cognitive and Language Sciences, 12(8), pp.120-127.
26. F. Shaikh, S. Darunde, N. Wahie and S. Mali, "Sign Language Translation System for Railway Station Announcements," 2019 IEEE Bombay Section Signature Conference (IBSSC), 2019, pp. 1-6, doi: 10.1109/IBSSC47189.2019.8973041.
27. D. Dewasurendra, A. Kumar, I. Perera, D. Jayasena and S. Thelijjagoda, "Emergency Communication Application for Speech and Hearing-Impaired Citizens," 2020 From Innovation to Impact (FITI), 2020, pp. 1-6, doi: 10.1109/FITI52050.2020.9424899.