Converting Gujarati Text and Speech into Gujarati Sign Language using HamNoSys and SigML

Abstract As deaf people use visual language, known as sign language, deaf culture is an aspect of the deaf community as a whole. Sign languages include American Sign Language (ASL), Indonesian Sign Language, and British Sign Language, among the approximately 300 individual sign languages used around the world. Not only these sign languages are different from each other, but also every individual sign language has its lexicon, meaning, and features; some use one hand, and some use two hands. Because there is no standard sign language, due to which mutual understanding is often difficult. The deaf community faces several types of communication barriers, primarily the conversion from speech to sign and from sign to speech. However, there exists an application called Handtalk, which can convert standard English into ASL, but no corresponding application has been created for Gujarati Sign Language. SignMitra is an effort to create a communication model to go from speech to Gujarati Sign Language. It aims to bridge an important gap in accessibility and inclusion by supporting the Gujarati region's deaf and dumb community by enabling them to communicate more effectively with hearing people.

Keywords Synthetic animation · Gujarati sign language · HamNoSys · Avatar · SiGML

1 Introduction

Sign languages are entirely natural, visual-gestural languages used by deaf and hard-of-hearing communities globally. Sign languages convey lexical and grammatical information through the use of both non-manual signals (facial expressions, head tilt, and body posture) and manual elements (hand shape, orientation, movement, and placement). Importantly, they do this in a simultaneous, multi-channel modality. The Deaf community in Gujarat, India, uses Gujarati Sign Language (GSL) as their native sign language. While GSL shares communicative features with many other sign languages, there are significant differences in the lexicon and grammatical schemata of GSL, as well as different cultural norms in use. For a GSL translation tool to provide accurate and culturally relevant translation, it will need to satisfactorily address certain features of language[1].

Despite having over 500,000 users, Gujarati Sign Language (GSL) is missing digital resources. There is no web-based platform for converting spoken or written Gujarati into animated signs so it can be accessible to the Deaf community. The proposed system accepts input using either a custom Gujarati keyboard or a speech-recognition plugin. The spoken text or written text is mapped to the HamNoSys transcriptions we store in a MongoDB. A Python service we created maps the HamNoSys to SiGML format and the front end receives the SiGML format to animate a 3D avatar in real time to produce the correct GSL signs.

1.1 HamNoSys Symbols for One Hand

Though 3D signs cannot be literally written, several notation systems were invented for writing signs, such as Stokoe Notation, SignWriting, and Hamburg Notation System (HamNoSys). In this study, the features handshape, direction (movement), location, and non-manual are represented by HamNoSys, which has about 200 Unicode-based symbols. Basic structure is shown in Fig.1 handshape, orientation, location and actions.

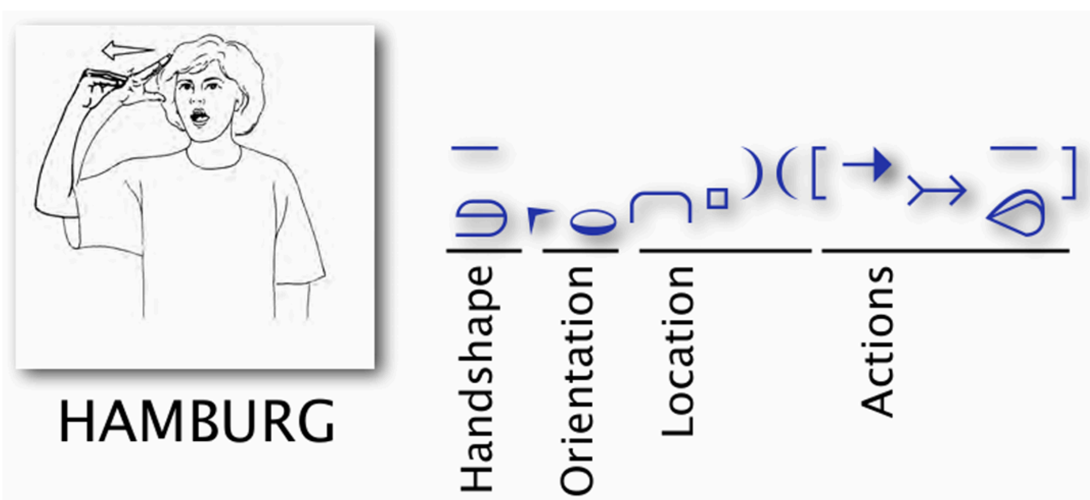
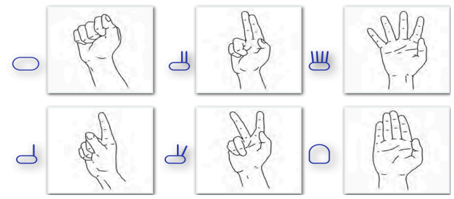
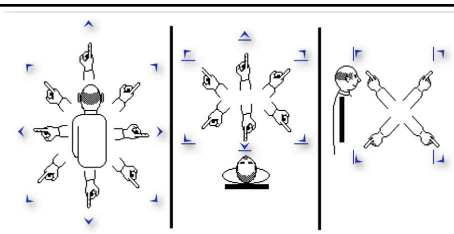
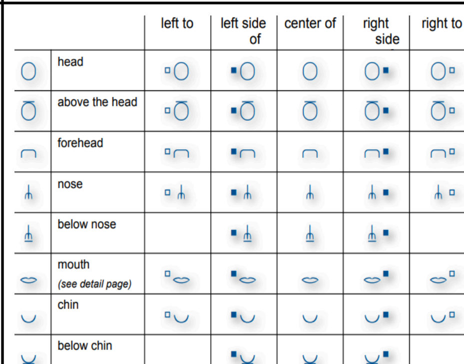
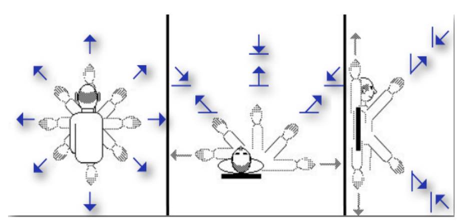


Fig. 1 HamNoSys notation[2]

Table 1 HamNoSys parameters

Parameter	Description	Figure
Handshape	HamNoSys depicts handshape with combinations of basic fingers and thumb position, fingers bending, as well as, variants. The handshape of an individual finger can be specified by itself along with initial ones when needed. For example, the symbol for open handshape is shown in Fig. 2.	 Fig 2 Some basic Handshapes
Hand Orientation	Hand orientation is defined by two parameters (a) Extended Finger Direction - three perspectives (signer's view, bird's view, and the right view (see Fig. 3)), and (b) Palm Orientation - defined using eight symbols each defined in relation to the extended finger direction.	 Fig 3 Extended Finger Direction
Location	The hand's location in 3D space is defined using the location parameters. The location is specified by the coordinates x and y in the frontal plane, while the z-coordinate specifies depth. For example, the symbol for head location is shown in Fig. 4.	 Fig 4 Head Locations
Actions	In HamNoSys, actions refer to non-manual, in-place, and path movements (changes of hand location). For example, the movement of the hand can be straight, curved, zig-zag, or circular and can also be a combination of these movements and occur simultaneously (Figs. 5).	 Fig 5 Straight Movements

1.2 HamNoSys Symbols for Two hands

Two-handed movements in HamNoSys follow symmetry rules, indicating just the dominant hand and deriving the non-dominant hand via the mirrored handshape, orientation, movement and location. Palm orientation is laterally mirrored and finger orientations flipped as needed. The locations will be same as that of the one-handed notation, but additional descriptors are added for hand relationships (side-by-side, contact). Contact types such as touching, crossing or interlocking are explicitly encoded.

2 Literature Review

Year & Author	Title	Description	Methodology	Limitations
2020 – Goyal et al; [3]	Automatic Translation of Complex English Sentences to Indian Sign Language Synthetic Video Animations	English text → ISL via HamNoSys → SiGML → avatar; handles compound/complex sentences.	Rule-based syntactic simplification → lexical lookup → HamNoSys encoding → SiGML generation → JA-Signing avatar playback.	Coverage limited by lexicon/grammar; non-manuals mostly templated; domain-specific sentences.
2021 – Ahmed et al;(Sign4PSL) [4]	A Real-Time Automatic Translation of Text to Sign Language	English text → Pakistan SL (PSL) with HamNoSys → SiGML converter and avatar ("Sign4PSL").	Text normalization → lexical module → HamNoSys authoring → automatic HamNoSys → SiGML conversion → avatar rendering.	Manual HamNoSys authoring bottleneck; limited vocabulary & weak non-manual modeling.
2022 – Walsh et al; [5]	Examining Language Representation for Neural Sign Language Production	Proposes Text-to-HamNoSys (T2H) as a phonetic intermediate; shows advantages vs. gloss for SLP.	NLP encoders predict HamNoSys strings; use HamNoSys-derived handshape as auxiliary supervision.	Focused on representation, not a full avatar pipeline; limited non-manual coverage.
2022 – Singh et al; [6]	Machine Translation of Multiple Languages into ISL	MT to "Text for ISL" with a note that rendering can be via HamNoSys/SiGML or stitched videos.	NMT to ISL-style text; discusses HamNoSys/SiGML as a standard rendering option.	The paper emphasizes MT; no full HamNoSys rendering implementation is reported.
2023 – Arkushin et al; [7]	Ham2Pose: Animating Sign Language Notation into Pose Sequences	The first method to animate HamNoSys text is 2D/3D pose sequences, which are language-agnostic.	Transformer encoders map HamNoSys tokens to pose, weak supervision; DTW-MJE metrics.	Uses pose, not full avatar; relies on high-quality HamNoSys input; limited non-manuals.
2023 – Rahman et al; [8]	Automatic 3D Animated Bangla Sign Language Gestures from Bangla Text and Voice	Bangla text/voice → SigML-driven 3D animations (numerals demo).	ASR + text normalization → mapping → SigML generation → avatar playback.	Narrow domain (numerals); HamNoSys details implicit via SigML; small evaluation.
2023 – PakParse team	PakParse: Machine Translation from Text to Pakistan	English/Urdu text → PSL using HamNoSys & SiGML with 3D	MT + lexical resources → HamNoSys transcription → SiGML → 3D avatar.	Manual HamNoSys for many signs; limited objective

[9]	Sign Language	avatar.		metrics.
2023 – García-Martínez et al; [10]	Sign Language Dataset for Automatic Motion Generation (LSE)	LSE dataset with HamNoSys phonemes + videos + landmarks to support SL production.	Curated 754 signs with HamNoSys; provides multi-modal alignments for training production models.	Not an end-to-end translator; Spanish SL only; limited vocabulary.
2024 – Uchida et al; [11]	HamNoSys-based Motion Editing Method for Sign Language	JSL pipeline with Japanese text→gloss→HamNoSys-guided motion editing to refine avatar output.	Transformer text→gloss; then edit avatar motion using HamNoSys constraints for accuracy/fluency.	Requires expert HamNoSys; editing is still partly manual/heuristic; JSL-specific.
2024 – Balayn et al; [12]	State of the Art of Automation in Sign Language: A Systematic Review	Broad review referencing HamNoSys/SiGML use in production systems and SL pipelines.	Systematizes pipelines: text/ASR → MT → HamNoSys/SiGML → avatar; catalogs tools.	Survey; not a system; scattered metrics.
2024 – Moryossef et al; [13]	A Data-Driven Representation for Sign Language Production	Discusses HamNoSys vs. gloss; positions HamNoSys as an avatar-ready intermediate.	Comparative analysis; motivates phonetic/parameterized representations for production models.	Conceptual; no full translator implementation.
2024 – Rina Damdoo et al; [14]	An Integrative Survey on Indian Sign Language Recognition and Translation	ISL survey highlighting pipelines using HamNoSys/SiGML for synthesis/ translation.	Reviews text/ASR→ISL approaches; summarizes tools and gaps relevant to HamNoSys pipelines.	Secondary source; limited new evaluation.
2024 – Ishara Kotha et al; [15]	A Comprehensive Avatar-Based Bangla Sign Language Corpus	HamNoSys-based Bangla corpus (~3.8k words) + text→avatar system using HamNoSys/SiGML.	Lexicon in HamNoSys; generation of avatar in real time; interpreter-based evaluation.	Focus on vocabulary/isolated signs; limited sentence-level prosody/non-manuals.
2025 – Neves & Coheur [16]	HamNoSys2SiGML: Translating HamNoSys Into SiGML	Open tool for conversion. HamNoSys → SiGML, used by recent translators.	Stand-alone converter enabling avatar playback; decouples from proprietary tools.	Depends on accurate HamNoSys inputs; non-manual coverage varies.
2025 – S2S Translator team [17]	Sign Language to Sign Language Translator	Uses HamNoSys2SiGML in a sign↔sign pipeline; demonstrates HamNoSys as a stable intermediate.	Keypoint models → HamNoSys sequence → SiGML → avatar for target SL.	S2S focus; not full text/speech input; intermediate still relevant but indirect.
2025 – Bouguerra et al. [18]	3DZSignDB: 3D Avatar SiGML Data for Algerian Sign Language	Dataset to support text→ALSL via SigML (HamNoSys-based methods referenced).	Curates SigML/3D avatar data for ALSL; intended for translation/production systems.	Dataset only; no translator implementation.
2023 –Verma, A. [19]	Converting Voice Signal to Visual Indian Sign Language	Speech (ASR) → ISL with synchronized body/hand motions; cites HamNoSys/SiGML avatar approach.	ASR + NLP → mapping to sign units → HamNoSys/SiGML avatar synthesis.	Concept/prototype; limited empirical results; HamNoSys stage is partly future work.

3 3 3

Architecture

3.1. System Overview

The proposed system is a web-based Gujarati Sign Language (GSL) translator designed to convert Gujarati speech or text input into animated sign output. Its architecture comprises two primary components: a frontend web interface, responsible for capturing and processing user input, and a backend engine, which retrieves, processes, and renders the corresponding signs using HamNoSys notation and SiGML. This modular design ensures a clear separation of concerns, enabling efficient data handling, accurate linguistic translation, and seamless user interaction.

3.2. Backend Components

3.2.1 HamNoSys Annotation and Dataset Development

Gesture data was recorded through observation and video documentation at multiple regional schools for the hearing impaired. The gestures were manually annotated into HamNoSys symbol strings using a trial-and-error visual validation approach. The annotations were iteratively refined using open-access tools for HamNoSys input[20] (Fig 6) and playback. This manually created linguistic dataset forms the basis of the sign mapping layer.

Fig 6 HamNoSys input

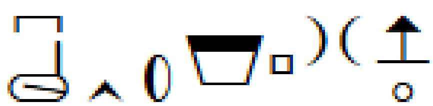
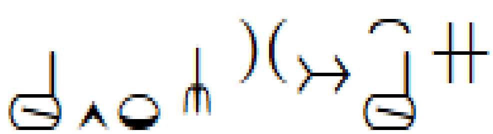
3.2.2 Conversion to SiGML

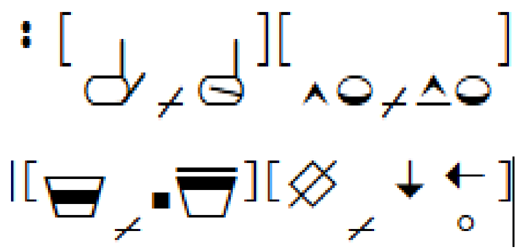
Each HamNoSys string was translated into SiGML using a Python code(Fig 7) adapted from open-source resources (**https://github.com/carolNeves/HamNoSys2SiGML?tab=readme-ov-file**). The resulting SiGML code has the structure needed for avatar-based sign animations. The conversion process ensured compatibility with the animation renderer and preserved gesture integrity. These SigML strings are then put in the MongoDB database.

```
PS D:\HamNoSys2SiGML-master (2)\HamNoSys2SiGML-master\Original> python HamNoSys2SIGML.py "□□□□□□□□□□/" "ગાય"
<?xml version="1.0" encoding="UTF-8"?>
<sigml>
        <hns_sign gloss="ગાય">
                <hamnosys_nonmanual/>
                <hamnosys_manual>
                        <hamdoublebent/>
                        <hamlrat/>
                        <hamlrat/>
                        <hamfinger2/>
                        <hamthumboutmod/>
                        <hamextfingeru/>
                        <hampalmd/>
                        <hamhead/>
                        <hamlrbeside/>
                        <hammoveu/>
                        <hamarcr/>
                </hamnosys_manual>
        </hns_sign>
</sigml>
```

Fig. 7 Python Script to convert HamNoSys to SiGML

Table 2 Some HamNoSys Notations

No.	Gujarati words	HamNoSys Notations	Avatar
1.	સાત (saat)		
2.	રડવું (raḍ-vuṁ)		
3.	લીલો (lee-lo)		

4.	ફૂટ **(phoot)**		
5.	ગાય **(gaay)**		

3.2.3

MongoDB-Based Sign Repository

The MongoDB database storing the SiGML data contains a collection named *sigml*, organized into subcategories such as *barakhadi*, *family*, *food*, and others. Each document in the collection stores a Gujarati word or phrase along with its corresponding SiGML markup string. This structure allows efficient data retrieval and guarantees scalability for future expansion. (Fig. 8).

Fig. 8 Database with collection ‘sigml’ and data stored in it

3.3 Frontend Components

3.3.1 Input Interface

The frontend of the system is developed using HTML, CSS, and JavaScript and supports two modes of input. The text input mode is implemented through a custom Gujarati keyboard with Barakhadi support, while the speech input mode leverages the JavaScript inbuilt webkitSpeechRecognition function to convert Gujarati audio into text (Fig. 9).

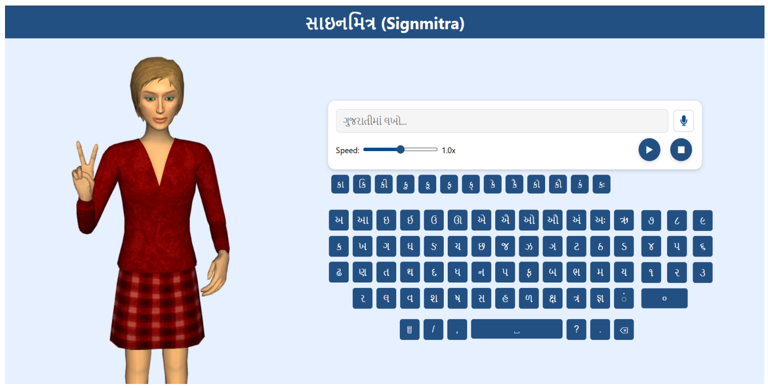


Fig. 9 Interface of our Web App

3.3.2 Communication and Query Layer

Once the frontend has accepted the user's input, it sends a request to the backend server for the specific SiGML string. The exact key that was used for input is then used by the backend to search the SigML collection that exists within MongoDB Atlas. If the SiGML is found, it is then returned to the frontend.

3.3.3 Animation Rendering Engine

The frontend implements an open-source SiGML animation avatar model(**https://vhg.cmp.uea.ac.uk/tech/jas/vhg2021/SiGML-Player-gui.html**)[21], which takes the signal code and animates a 3D avatar accordingly. The avatar model parses positional and motion data from the SiGML string that originated from the manually generated HamNoSys notation.

4 Conclusion

We have presented a web-based translator that allowed the input of Gujarati speech or text to GSL using HamNoSys symbol strings. The notation system has a flexible architecture and highlights accessibility and use of data that was assigned by a human manual coding. Therefore, the potential is there to utilize this translation system and develop it into a much higher degree of interaction in developing the use of grammar rules, and NLP on this web service. In conclusion, we are stimulated by the use of a standardized gesture notation system has shown that there is a potential for these types of tools to develop inclusive communication systems for people in regional communities.

5 References

[1]N. Aasofwala, S. Verma and K. Patel, "A Novel Speech to Sign Communication Model for Gujarati Language," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICIRCA51532.2021.9544635.

[2]HamNoSys–Hamburg notation system for sign languages. (n.d.). https://www.sign-lang.uni-hamburg.de/dgs-korpus/files/inhalt_pdf/HamNoSys_2018.pdf

[3]Shabina, B., Sony, S., & Rajeev, R. R. (2024, November). An approach for Malayalam sentence to Indian sign language generation using synthetic animations. In *AIP Conference Proceedings* (Vol. 3134, No. 1, p. 020002). AIP Publishing LLC.

[4]Sanaullah, M., Ahmad, B., Kashif, M., Safdar, T., Hassan, M., Hasan, M. H., & Aziz, N. (2022). A real-time automatic translation of text to sign language. *Computers, Materials & Continua*, *70*(2).

[5]Walsh, H., Saunders, B., & Bowden, R. (2022). Changing the representation: Examining language representation for neural sign language production. *arXiv preprint arXiv:2210.06312*.

[6]Dhanjal, A. S., & Singh, W. (2022). An automatic machine translation system for multi-lingual speech to Indian sign language. *multimedia Tools and Applications*, *81*(3), 4283-4321.

[7]Arkushin, R. S., Moryossef, A., & Fried, O. (2023). Ham2pose: Animating sign language notation into pose sequences. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 21046-21056).

[8]Islam, M. A., Karim, A., Rahaman, M. A., Rahman, M., Hossain, M. P., & Kabir, S. A. (2022, December). Automatic 3d animated bangla sign language gestures generation from bangla text and voice. In *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)* (pp. 1-6). IEEE.

[9]Mahmood, Rana & Ahmad, Haseeb & Ann, Qurat-ul & Ashraf, Muhammad. (2023). PakParse: Machine Translation from Text to Pakistan Sign Language for Effective Communication with Deaf People. Applied and Computational Engineering. 2. 482-490. 10.54254/2755-2721/2/20220568.

[10]Villa-Monedero, M., Gil-Martín, M., Sáez-Trigueros, D., Pomirski, A., & San-Segundo, R. (2023). Sign language dataset for automatic motion generation. *Journal of Imaging*, *9*(12), 262.

[11]Uchida, T., Miyazaki, T., & Kaneko, H. (2024, May). HamNoSys-based Motion Editing Method for Sign Language. In *Proceedings of the LREC-COLING 2024 11th Workshop on the Representation and Processing of Sign Languages: Evaluation of Sign Language Resources* (pp. 376-385).

[12]Kumar Attar, R., Goyal, V., & Goyal, L. (2023). State of the art of automation in Sign Language: a systematic review. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(4), 1-80.

[13]Walsh, H., Ravanshad, A., Rahmani, M., & Bowden, R. (2024, May). A data-driven representation for sign language production. In *2024 IEEE 18th International Conference on Automatic Face and Gesture Recognition (FG)* (pp. 1-10). IEEE.

[14]Damdoo, R., & Kumar, P. (2025). An integrative survey on Indian sign language recognition and translation. *IET Image Processing*, *19*(1), e70000.

[15]Rahman, M. S., Islam, M. D., Dewan, P., & Islam, M. F. Isharakotha: A Comprehensive Avatar-Based Bangla Sign Language Corpus. *Prato and Islam, Md Fuadul, Isharakotha: A Comprehensive Avatar-Based Bangla Sign Language Corpus*.

[16]Neves, C., Coheur, L., & Nicolau, H. (2020, May). Hamnosys2sigml: translating hamnosys into sigml. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 6035-6039).

[17]Sharath S R, Suraj S, Abishek Kumar G M, P Siddharth, Nalinadevi K,Sign Language To Sign Language Translator,Procedia Computer Science,Volume 260,2025

[18]Zerrouki, T., Slimani, M. F., Mami, A., & Mazari, R. (2025). 3DZSignDB: 3D avatar SigML data for Algerian sign language. *Data in Brief*, 111568.

[19]Verma, A., & Nunes, J. (2023). Converting Voice Signal to Visual Indian Sign Language.

[20]https://www.sign-lang.uni-hamburg.de/hamnosys/input/

[21]*SIGML Signing App New GUI: VHG.2021*. (n.d.).
https://vhg.cmp.uea.ac.uk/tech/jas/vhg2021/SiGML-Player-gui.html