Real Time English to British Sign Language Translation for Accessible Banking

Abhishek Bharadwaj Varanasi
TCS Research
varanasi.abhishek@tcs.com

Manjira Sinha
TCS Research
sinha.manjira@tcs.com

Tirthankar Dasgupta
TCS Research
dasgupta.tirthankar@tcs.com

Charudatta Jadhav
TCS Research
charudatta.jadhav@tcs.com

***Abstract*—This paper presents a mobile-based framework for real-time English text-to-British Sign Language (BSL) translation, designed for accessible banking. The system uses a linguistically informed transformer model to translate English sentences into BSL gloss sequences, which are then converted into HamNoSys for avatar animation. The framework is optimized for handheld devices, enabling applications like bank notifications for the Deaf community. To support experimentation, an English-BSL parallel dataset was created, covering multiple domains. The proposed model achieved a 87.31% ROUGE-L score for text-to-gloss translation, demonstrating its effectiveness across domains. The system outputs were subjected to a user-based evaluation with BSL experts, providing valuable qualitative insights.**

***Index Terms*—Sign Language, Machine Translation, Avatar Animation**

I. Introduction

In today's increasingly digital economy, access to banking services is critical for financial inclusion and independence. However, individuals who are Deaf or hard of hearing, particularly those who rely on British Sign Language (BSL) as their primary mode of communication, often face significant barriers when accessing such services. BSL, like other sign languages (SL), is a non-verbal form of communication where deaf individuals use their hands, arms, and facial expressions to share thoughts and ideas. Traditional banking systems are predominantly designed around written or spoken language, creating challenges for individuals whose first language is visual and gestural. This necessitates the development of automatic means of translating spoken language texts into sign language.

A challenging aspect of sign language translation (SLT) is that SLs are multi-channeled and do not have a written form, as noted by [1]. Consequently, the recent advancements in generative AI and text-based machine translation (MT) cannot be directly applied to SL translation.

This paper introduces a mobile-based English-to-BSL translation system specifically designed to address these accessibility gaps in banking. Leveraging advancements in natural language processing (NLP), and mobile technology, the system enables real-time translation of English text into BSL through animated sign language avatars. The proposed solution aims to enhance communication, reduce reliance on human interpreters, and promote independence for BSL users. By integrating this technology into mobile banking platforms, the system aspires to make financial services more inclusive and equitable for the Deaf community. For translations from spoken languages to SLs, glosses are used to build the system in two phases: translating text to glosses, and then generating the signs from the glosses (see Figure 1). The sign generations can be made using avatar animations or auto-encoder based video generators.

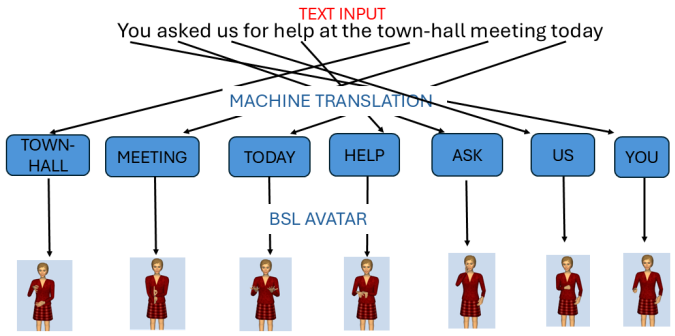

Fig. 1. Illustration of text to British Sign Language (BSL) avatar generation using glosses as intermediate step.

The proposed framework has significant potential across various industries by improving communication for Deaf and hard-of-hearing communities. By automating English-to-British Sign Language (BSL) translation, it can be integrated into apps, public notifications, and virtual meetings, enhancing accessibility. It is especially valuable in sectors like healthcare, education, telecommunications, and entertainment, helping bridge language gaps and improve access to essential information. Our work thus seeks to make significant strides in the deployment of BSL avatars on mobile devices, enhancing communication accessibility for the deaf and hard-of-hearing population (see Figure 2).

II. Background and Related Works

Sign Languages (SL) are visual-spatial natural languages that use manual (hand shape, orientation, position, movement) and non-manual (facial expressions, eye gaze, body posture) components for communication [2]. Signs are produced within a three-dimensional space divided into 27 regions [3], [4]. SL morphology is mainly derivational, with the closed lexical class including classifier hand shapes, discourse markers, and non-manual signs [2]. Classifier hand shapes represent referent characteristics with specific hand configurations (Figure 3). British Sign Language (BSL) is known to follow a topic-

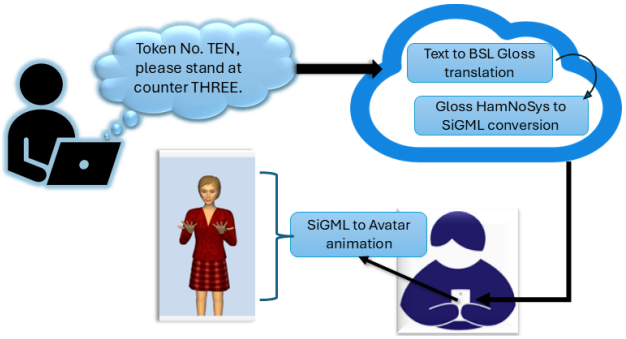

Fig. 2. Illustration of Text to BSL Avatar application flow for mobile devices

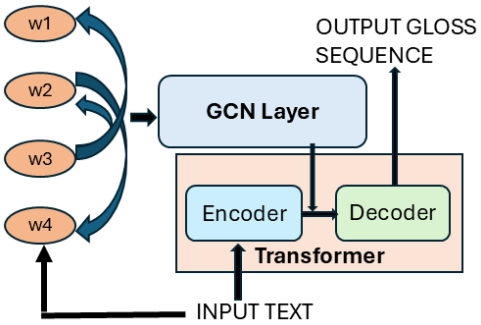

Fig. 4. Linguistically informed Transformer

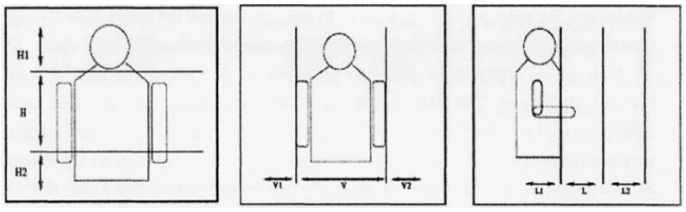

Fig. 3. Classification of signing space into horizontal, vertical, and lateral regions.

comment structure. This structure positions the main subject or theme (the topic) at the sentence's outset, followed by more specific information (the comment). By establishing context early in the sentence, BSL users efficiently convey complex ideas.

Researchers have traditionally used written representations of sign languages (SLs) to aid translation, often employing glosses—spoken language labels corresponding to SL components. Glosses serve as intermediaries in machine translation (MT) systems for SL-to-text and text-to-SL conversion, as explored by various studies ([5], [6]; [7]; [8], [9]). An earlier work by [10] made an attempt to build speech to French Sign language translator prototype using Regulus Lite platform (for machine translation) and JA Signing software (for avatar animation), focusing on its evaluation by the deaf-community. Notable approaches include [8], which combined Neural Machine Translation (NMT) with motion graphs to generate SL videos, and the current SOTA method by [11], which translates text to glosses, extracts poses, and generates videos. Earlier work by [12] used Lexical Functional Grammar for Indian Sign Language (BSL), while [13] proposed using *HamNoSys* for avatar-based text-to-SL translation.

III. The Linguistically Informed Transformer for Text to BSL Gloss

The existing NMT models excel at capturing intricate data patterns without requiring manual feature engineering, offering end-to-end solutions. However, they often overlook latent linguistic traits crucial for extracting pertinent information. To address this, we propose a transformer-based architecture that integrates word embeddings from the encoder part with diverse linguistic features inherent in text, enhancing automatic text-to-ASL gloss translation.

A. Transformer Model

The input to the model is a sentence consisting of a word sequence $x = (x_1, x_2, ..., x_T)$ representations. We then tokenize the sentence x using a wordpiece vocabulary, and then generate the input sequence $\bar{x}$ by concatenating a [CLS] token, the tokenized sentence, and a [SEP] token. Then for each token $\bar{x}_i \in \bar{x}$, we convert it into vector space by summing the token, segment, and position embeddings, thus yielding the input embeddings $h^0 \in R^{(n+2)\times h}$, where h is the hidden size and n is sequence length. Next, we use a series of L stacked Transformer blocks to project the input embeddings into a sequence of contextual vectors $h^i \in R^{(n+2)\times h}$. Here, we omit an exhaustive description of the block architecture and refer readers to [14] for more details.

B. Syntactic Dependency Graph

Encoding the structural information directly into neural network architecture is not trivial. Marcheggiani and Titov [15] proposed a way to incorporate structural information into sequential neural networks through Graph Convolution Networks (GCN) [16], [17]. GCNs take graphs as inputs and conduct convolution on each node over their local graph neighborhoods. The syntax structure of a sentence is transferred into a syntactic dependency graph, and GCN is used to encode this graph information. This kind of architecture is already utilized to incorporate syntactic structure with BERT [18] embeddings for several NLP based tasks [19].

C. Linguistically Informed Transformer

We have incorporated a similar method for the present text-gloss translation task in this work. Here, each sentence is parsed into its syntactic dependencies graph and use GCN to consume this structural information. We use pre-trained GLOVE embeddings as our initial hidden states of vertices in GCN. The output hidden states of the GCN is combined with the context embeddings generated by the transformer model's (T5 and BART) encoder and then passed to the decoder unit (see Figure 4).

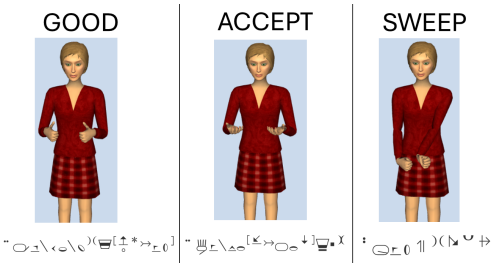

Fig. 5. HamNoSys Examples

IV. HamNoSys: A sign language notation system

HamNoSys is a standardized system for transcribing sign language gestures, focusing on hand shape, location, and movement, using a unique set of glyphs. It captures only essential features for accurate sign performance, remaining independent of the performer. While primarily documenting hand actions, it assumes natural body movements, and more recently includes non-manual elements like facial expressions and upper body movements (see Figure 6).

- **Hand Shapes:** HamNoSys represents hand shapes with symbols for various configurations, from basic forms like fists to complex combinations. It incorporates bending operators (e.g., "max.ext." for full extension) and details thumb positions and opposition. The system allows flexibility for underspecified shapes to capture diverse hand configurations.
- **Hand Orientation:** Orientation is transcribed as relative (aligned with movement paths) or absolute (fixed to a reference point). It includes subscripts for changing directions (e.g., zigzags) and allows flexibility with underspecified orientations.
- **Hand Location:** Symbols indicate body parts and spatial regions for hand placement, with diacritics for finer details (e.g., face areas like mouth or eyes). It also specifies parts of the arms, wrists, fingers, and nails, with subscripts marking general articulation zones.
- **Hand Movement:** Movements are described with symbols for paths (straight, circular, arcs), types (e.g., brushing or bouncing), and modifiers like repetition or out-of-phase motions. Wrist rotations, finger-play, and sequential finger movements are also included, enabling precise transcription of dynamic gestures.

V. BSL Avatar Animation

A computer-generated avatar for SL animation is represented by a 3D deformable surface mesh made up of small, colored, and textured polygons. Thousands of polygons are used for realism, and once the vertex coordinates are set, the mesh only needs to be transmitted once. Modern devices can render this mesh in real-time.

To adjust the avatar's posture, morphs are applied—localized distortions where specific vertices are displaced. This technique provides precise control over the mesh, especially for subtle facial animations.

In essence, by defining an avatar's skeleton, surface mesh, the attachment between the two, and its facial morphs, rendering software can generate real-time SL animations (BSL in this work) from a stream of animation parameters. These parameters include the avatar's skeleton configuration and the weights for the facial morphs in each animation frame. The result is displayed on a computer or mobile device using standard 3D rendering techniques, leveraging both software and hardware.

Each avatar also has a virtual skeleton, a structured set of virtual bones to which the mesh is attached. The position and orientation of the polygons are defined by the movement of these bones, so altering the invisible skeleton directly affects the visible mesh. This makes changing the avatar's posture more efficient, as the skeleton configuration requires far less data than transmitting the entire mesh or rendered frames. Thus, BSL animations can be transmitted over the internet by sending only skeleton configurations, with the end-user's device generating and rendering the mesh, provided it has the attachment data of the mesh to the skeleton [21].

Following the same process, signing softwares like JASigning [22] animate a pre-built avatar where the sign configuration data is fed using an XML file specific to sign language called SiGML. SiGML is a signing gesture markup language that contains information about the animation based on the HamNoSys.

Each HamNoSys symbol represents a feature such as hand shape, orientation, location, or movement, with corresponding XML tags used in SiGML for machine readability (see Figure 7). Non-manual features are often not included in HamNoSys due to animation limitations. As shown in Figure 2, on the cloud, our proposed linguistic model translates English text to BSL gloss sequence, which is then converted to SiGML by mapping HamNoSys symbols of glosses from [23] (see Figure 5 for examples) to corresponding XML tags. The mobile app receives the SiGML file, which is used by JASigning software to animate the avatar.

The path a hand follows in sign language is usually specified in the transcription, while details about accelerations and decelerations are often omitted. HamNoSys defines five movement styles: normal, fast, slow, tense, and "sudden stop at the end" (e.g., in a "punch"). There are also four types of "normal" movements: targeted, lax, hard contact, and linear [24]. "Targeted" movements are meaningful and distinct from "lax" movements, which are non-meaningful. Targeted movements have a pronounced deceleration, while "tense" movements are slow and effortful, different from just slow movements. The equation of motion of the resulting system is: $x'' + k'x' + kk'x = kk'x_t$.

This is mathematically equivalent to damped simple harmonic motion, which is critically damped when $k'/k = 4$, under-damped for smaller values of k'/k, and over-damped for larger values. These trajectories specify what proportion of the path from one posture to another the avatar should have moved after a certain time has elapsed. Based on that, the final positions are linearly interpolated between the start and the end positions [24].

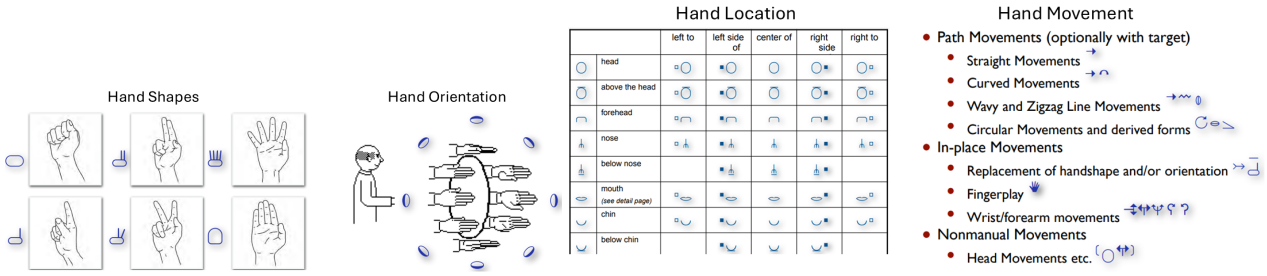

Fig. 6. Examples of HamNoSys for hand shape, orientation, location and movement [20]

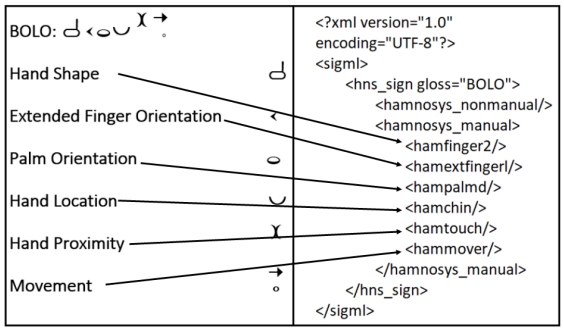

Fig. 7. HamNoSys to SiGML Example [25]

VI. Experiments

BSL-gloss parallel data: To facilitate experimentation, we have built a set of 6096 text- BSL gloss pairs with 3597 pairs corresponding to banking domain with the help of domain experts. The collected phrases are sourced from general purpose texts with a little more focus on banking-related texts and provided to British Sign Language (BSL) experts for manual translation into BSL gloss.

Fine-tuning: For text to BSL gloss translation task, we used 4877 sentences for training, 609 sentences for validation and 610 sentences for testing. We fine-tuned the three transformer models T5-small, T5-base and BART-base on one A100 GPU (refer Table I for model hyperparameters).

The linguistic embeddings which are GCN's output hidden states are combined with the last hidden state of the encoder part as described in section IV.C during both the fine-tuning processes. In that way, the GCN is trained along with the transformer model.

Evaluation: Apart from using the standard MT evaluation parameters like, ROUGE-L [26] and BLUE [27] scores we also advocate using a modified BERTScore [28] as performance metrics. As the BERT models are trained on natural English text, we cannot rely on the sentence embeddings it gives for the BSL gloss sequences for the reasons explained in introduction. Hence, we proposed to get the word embeddings of each gloss present in the BSL gloss sequence and aggregate them to get the sentence embedding of the BSL gloss sequence which can be further used to calculate the cosine similarity score (modified BERTScore).

TABLE I
Hyperparameters for the model.

Hyperparameter	Value
Training epochs	50
Maximum learning rate	1×10^{-4}
Weight Decay	1×10^{-5}
Warm-up Epochs	10
Batch size	4
Gradient accumulation steps	4
Attention Dropout	0.1
Optimizer	Adam [29]

VII. Results

A. Text to BSL gloss models evaluation

The results are reported by comparing the model performance upon fine-tuning on our text-to-BSL gloss parallel dataset between our models of choice T5-small, T5-base [30] and BART-base [31] (Table II).

The T5-base model is the top performer for text-to-BSL translation, but several challenges persist. These include differences in word order between the topic and comment parts of predicted and gold texts, as well as inconsistent placement of the **wh** word in **wh**-questions. Additionally, helping verbs and articles are sometimes not removed, though they should be filtered using SpaCy's part-of-speech tagging. Finally, some gloss translations replace words with synonyms, which doesn't affect signing but lowers ROUGE-L and BLEU scores.

TABLE II
Test scores of T5-small & -base and BART-base model upon fine-tuning on our BSL gloss parallel dataset

Model	T5-small	T5-base	BART-base
ROUGE-L	86.57	**87.31**	83.50
BLEU-1	84.97	**85.67**	84.83
BLEU-2	73.48	**74.71**	70.41
BLEU-3	66.87	**67.43**	65.67
BLEU-4	60.47	**61.27**	59.50
Modified BERTScore	**88.38**	89.03	88.97

B. Subjective Evaluation

Apart from the automatic evaluation, we also chose to perform a user based evaluation technique for the proposed MT engine.

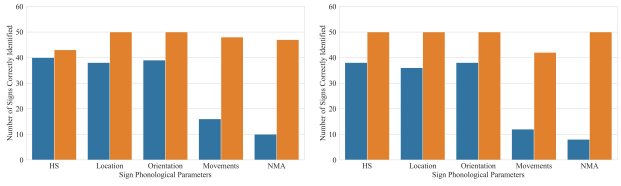

Fig. 8. Comparing the sign understandability between avatar based and video signs for both one-handed and two-handed signs.

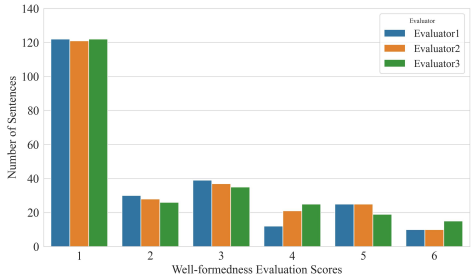

Fig. 9. Well-formedness scores of the output generated sentences.

The generated outputs of the system are shown to a group of BSL experts. The evaluators were asked to rate each of the output BSL sentences in terms of certain parameters. The overall performance of the system has been evaluated based on the following criteria:

1) Sign understandability
2) Well-formedness of the output BSL sentence

We perform a two level of evaluation. In the first level, we identify the performance of the system in terms of sign representation and sign understandability. In the second level of evaluation, we compute the systems accuracy in terms of well-formedness of the BSL sentence.

1) Sign understandability test: The goal is to identify whether the avatar based representation conveys correct information to the BSL user. This is the prime motivation behind performing the sign understandability test. Accordingly, we randomly collected 100 BSL signs both in the form of avatar based animations as well as in video format. We then classified them into two classes: a) Single handed signs and b) Double handed signs. Each class is a set of 50 signs.

The BSL expert rate each sign based on the recognition of the following features:

1) Recognizing the hand shapes (HS)
2) Finger and palm orientation (Orientation)
3) Hand location (Location)
4) Hand movements (Movements)
5) Non-manual components (NMA)

Each of the signs was classified as valid or invalid according to their understandability and quality. Figure 8 summarizes the comparative study between avatar and video based one handed and two handed signs. The X-axis specifies the different phonological parameters of a sign and the Y-axis shows the number of signs correctly recognized by an evaluator. We can observe that representing signs with pre-recorded video performs better than avatar based signs where there exists information loss as well as ambiguity in understanding, particularly for signs having complex hand-shapes, movement and non-manual components. However, there are certain cases where video based signs also fail to provide correct information like, in case of directional signs, where the movement depends upon the location of subject or object avatar based dynamic representations performs much better.

2) Well-formedness test: The second level of evaluation is primarily concerned with identifying the performance in terms of the Well-formedness of the generated sentence. We define the well-formedness of a sentence in terms of its grammatical structure like, syntax, word ordering, tense, correct usage of lexical items and morphological attachments. In other words we can say that a sentence is well-formed if the syntax, word ordering, and morphological attachments of the sentence are correct and proper lexical items are used to establish the meaning of the sentence.

The parameters that identifies wellformedness of the output BSL sentence are defined as follows:

1) Grammar, word usage and style are all appropriate no rewriting is needed
2) Minor correction needed
3) Minor word order errors
4) Attachment tense and number errors
5) Phrase and clauses missing
6) Subject and predicate missing

Figure 9 summarizes the well-formedness scores assigned by each of the experts. The X-axis shows the evaluation score range(1-6) for wellformedness and the Y-axis shows the number of sentences that received the score. The graph in Figure 9 shows that the score range of above 4 is given to the least number of sentences while most sentences scored below the range of 4, which is desirable.

C. Analyzing memory requirements and rendering time

We compare the memory requirements of two applications: text-to-BSL video and text-to-BSL avatar. The text-to-BSL video application maps glosses to video clips from a dictionary, whereas the text-to-BSL avatar uses an animated avatar. The text-to-BSL video application requires about 7 GB of disk space, and this will increase as the gloss-video dictionary grows. While videos can be stored in the cloud, this introduces higher latency, dependency on a strong internet connection, and additional storage costs. In contrast, the text-to-BSL avatar application only requires 910 MB, making it more lightweight and easier to deploy on mobile devices.

We compare the rendering times for text-to-BSL video and text-to-BSL avatar applications. Rendering time includes both text-to-gloss translation and gloss-to-video mapping/avatar generation.

As illustrated in the plot (Figure 10), the rendering time of text-to-BSL video app is more than text-to-BSL avatar app. The rendering times of both the applications increases almost linearly with number of words in the input text. Also, the gap between

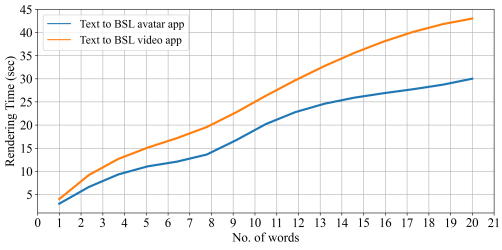

Fig. 10. Sign rendering time (in sec) vs Number of words in the input text.

rendering time of text-to-BSL video app and text-to-BSL avatar app increases with number of words in the input text.

VIII. Conclusion

This paper presents a linguistically informed transformer model for real-time translation of English text into British Sign Language (BSL), addressing challenges in traditional sign language translation. By incorporating linguistic features and using Graph Convolution Networks (GCN), the model achieves an 87.31% ROUGE-L score for text-to-gloss translation. The system converts English text into BSL glosses, which are then used to animate avatars for accessible communication on mobile devices. We developed a lightweight mobile application that allows Deaf users to receive real-time banking notifications in BSL via animated avatars. This framework enhances accessibility in sectors like banking, healthcare, and education, offering a more inclusive solution for the Deaf community.

References

[1] G. Langer, S. König, and S. Matthes, "Compiling a basic vocabulary for german sign language (dgs)–lexicographic issues with a focus on word senses," in *Proceedings of the XVI EURALEX International Congress: The User in Focus*, 2014, pp. 767–786.

[2] U. Zeshan, "Indo-pakistani sign language grammar: a typological outline," *Sign Language Studies*, pp. 157–212, 2003.

[3] S. Sinha, "A skeletal grammar of indian sign language," *Unpublished master's diss., Jawaharlal Nehru University, New Delhi, India*, 2003.

[4] ——, "A grammar of indian sign language," Ph.D. dissertation, PhD dissertation, Jawaharlal Nehru University, New Delhi, India, 2009.

[5] N. Cihan Camgoz, S. Hadfield, O. Koller, and R. Bowden, "Subunets: End-to-end hand shape and continuous sign language recognition," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3056–3065.

[6] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7784–7793.

[7] Y. Chen, R. Zuo, F. Wei, Y. Wu, S. Liu, and B. Mak, "Two-stream network for sign language recognition and translation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 17 043–17 056, 2022.

[8] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, "Text2sign: towards sign language production using neural machine translation and generative adversarial networks," *International Journal of Computer Vision*, vol. 128, no. 4, pp. 891–908, 2020.

[9] B. Saunders, N. C. Camgoz, and R. Bowden, "Adversarial training for multi-channel sign language production," *arXiv preprint arXiv:2008.12405*, 2020.

[10] B. David and P. Bouillon, "Prototype of automatic translation to the sign language of french-speaking belgium evaluation by the deaf community," *Modelling, Measurement and Control C*, vol. 79, no. 4, pp. 162–167, 2018.

[11] A. Moryossef, M. Müller, A. Göhring, Z. Jiang, Y. Goldberg, and S. Ebling, "An open-source gloss-based baseline for spoken to signed language translation," *arXiv preprint arXiv:2305.17714*, 2023.

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

[13] Sugandhi, P. Kumar, and S. Kaur, "Sign language generation system based on indian sign language grammar," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 19, no. 4, pp. 1–26, 2020.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[15] D. Marcheggiani and I. Titov, "Encoding sentences with graph convolutional networks for semantic role labeling," *arXiv preprint arXiv:1703.04826*, 2017.

[16] K. Webster, M. R. Costa-Jussà, C. Hardmeier, and W. Radford, "Gendered ambiguous pronoun (gap) shared task at the gender bias in nlp workshop 2019," in *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, 2019, pp. 1–7.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[18] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[19] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," *Advances in neural information processing systems*, vol. 28, 2015.

[20] T. Hanke, "Hamnosys–hamburg notation system for sign languages," *Institute of German Sign Language, Accessed in*, vol. 7, 2010.

[21] M. Lazor, D. B. Gajić, D. Dragan, and A. Duta, "Automation of the avatar animation process in fbx file format." *FME Transactions*, vol. 47, no. 2, 2019.

[22] R. Elliott, J. R. Glauert, J. Kennaway, I. Marshall, and E. Safar, "Linguistic modelling and language-processing technologies for avatar-based sign language presentation," *Universal access in the information society*, vol. 6, pp. 375–391, 2008.

[23] E. Efthimiou, S.-E. Fontinea, T. Hanke, J. Glauert, R. Bowden, A. Braffort, C. Collet, P. Maragos, and F. Goudenove, "Dicta-sign–sign language recognition, generation and modelling: a research effort with applications in deaf communication," in *Proceedings of the 4th Workshop on the Representation and Processing of Sign Languages: Corpora and Sign Language Technologies*, 2010, pp. 80–83.

[24] R. Kennaway, "Avatar-independent scripting for real-time gesture animation," *arXiv preprint arXiv:1502.02961*, 2015.

[25] C. Neves, L. Coheur, and H. Nicolau, "Hamnosys2sigml: translating hamnosys into sigml," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 6035–6039.

[26] C.-Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text summarization branches out*, 2004, pp. 74–81.

[27] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 2002, pp. 311–318.

[28] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," *arXiv preprint arXiv:1904.09675*, 2019.

[29] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*. Ieee, 2018, pp. 1–2.

[30] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[31] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," *arXiv preprint arXiv:1910.13461*, 2019.