Generating Signed Language Instructions in Large-Scale Dialogue Systems

Mert İnan[1], **Katherine Atwell**[1], **Anthony Sicilia**[1], **Lorna Quandt**[2], **Malihe Alikhani**[1]
[1] Khoury College of Computer Science, Northeastern University, Boston, MA, USA
[2] Educational Neuroscience Program, Gallaudet University, Washington, D.C., USA
{inan.m, atwell.ka, sicilia.a, alikhani.m}@northeastern.edu
lorna.quandt@gallaudet.edu

Abstract

We introduce a goal-oriented conversational AI system enhanced with American Sign Language (ASL) instructions, presenting the first implementation of such a system on a worldwide multimodal conversational AI platform. Accessible through a touch-based interface, our system receives input from users and seamlessly generates ASL instructions by leveraging retrieval methods and cognitively based gloss translations. Central to our design is a sign translation module powered by Large Language Models, alongside a token-based video retrieval system for delivering instructional content from recipes and wikiHow guides. Our development process is deeply rooted in a commitment to community engagement, incorporating insights from the Deaf and Hard-of-Hearing community, as well as experts in cognitive and ASL learning sciences. The effectiveness of our signing instructions is validated by user feedback, achieving ratings on par with those of the system in its non-signing variant. Additionally, our system demonstrates exceptional performance in retrieval accuracy and text-generation quality, measured by metrics such as BERTScore. We have made our codebase and datasets publicly accessible at https://github.com/Merterm/signed-dialogue, and a demo of our signed instruction video retrieval system is available at https://huggingface.co/spaces/merterm/signed-instructions.

1 Introduction

Conversational systems have become increasingly integrated into our everyday lives, yet their accessibility to the Deaf and Hard-of-Hearing (DHH) community, who predominantly communicate through signed languages, remains limited (Glasser et al., 2017, 2020; Bragg et al., 2020). Despite growing advocacy for more inclusive interactive technologies from DHH users (Bragg et al., 2019; Blair and Abdullah, 2020; Kahlon and Singh, 2023), a comprehensive dialogue system tailored for sign language users has yet to be implemented on a global scale. In response, within the Alexa Prize TaskBot Challenge 2 framework, we developed and launched the first task-oriented, multimodal dialogue system utilizing ASL, aiming to bridge the gap between DHH users and personal voice assistants. This system translates touch-based inputs into ASL video instructions, offering a groundbreaking approach to interaction fig. This paper introduces our ASL instruction framework, marking a significant stride towards integrating conversational systems into the living spaces of sign lan-

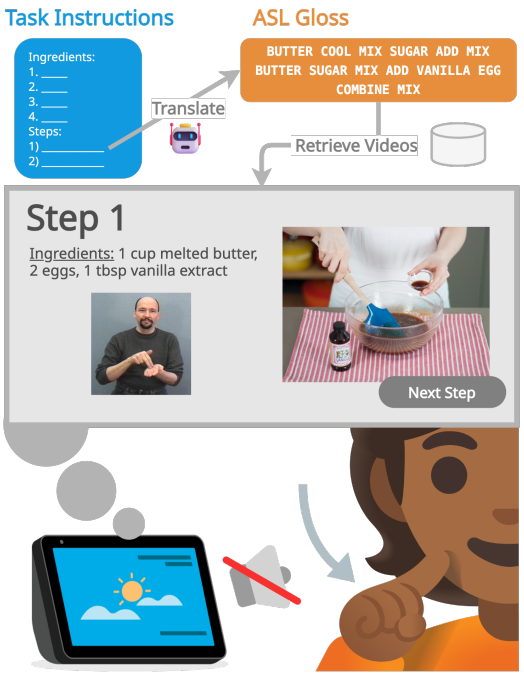

Figure 1: An overview of our multimodal dialogue system, capable of giving signed instructions to Deaf or Hard-of-Hearing users in ASL. We first translate task instructions to an intermediate textual representation called glosses using Large Language Models; then, we fetch token-level sign videos to display on the screens of Amazon Alexa Echo Show.

guage users and enhancing accessibility for the DHH community.

Many signers prefer to use ASL instead of text due to grammatical and linguistic differences between spoken and signed languages (Hariharan et al., 2018; Dangsaart et al., 2008). Yet currently, systems claiming to be accessible resort to text-based communication. As an alternative, videos or avatars of signers are options, yet these technologies are underutilized. In this paper, we show that deploying these signed systems on a large scale is, in fact, possible without much production cost and makes the system accessible to DHH users.

Further, prior linguistics research has shown that DHH community members can experience higher cognitive loads while reading compared to signing (Traxler, 2000; Kelly, 2003; Luckner and Handley, 2008). In this paper, we investigate effective strategies of multimodal information presentation for the DHH to reduce cognitive load. With repeated consultations with cognitive scientists, we design the layout of our system's user interface specifically around the cognitive load of signers (see Figure 2).

We focus on creating a framework that is applicable to a large-scale global platform (in our case, Amazon Alexa), making it impossible at this time to access camera footage. We investigate ways of receiving input with other modalities instead of voice commands and without camera access. This leads us to focus on the task of instruction generation and delivery rather than recognizing signs produced by the user. We receive input from the user via touchscreen controls of Amazon Alexa Echo Show devices so that signers can interact without using voice commands (see Figure 2 for the touch screen user interfaces where the user can interact via buttons to select tasks and navigate instructions).

To address all of the aforementioned points, in the following sections, we introduce the components of our framework. Our detailed contributions are as follows:

1. We design a multimodal task-oriented dialogue system with signed instructions and deploy it on multimodal devices.
2. We use *co-design* to build our system, actively involving community members in the design, development, and evaluation, ensuring our solutions positively impact the community.
3. We implement a novel Large Language Model (LLM)-based instruction generation technique for zero-shot text-to-sign translation. We use linguistics rules and cognitive science-based heuristics for this translation.
4. We make available a standalone library to translate instruction texts into signed instruction videos, and we release our dataset used for the top 200 signs in cooking and wikiHow domains.

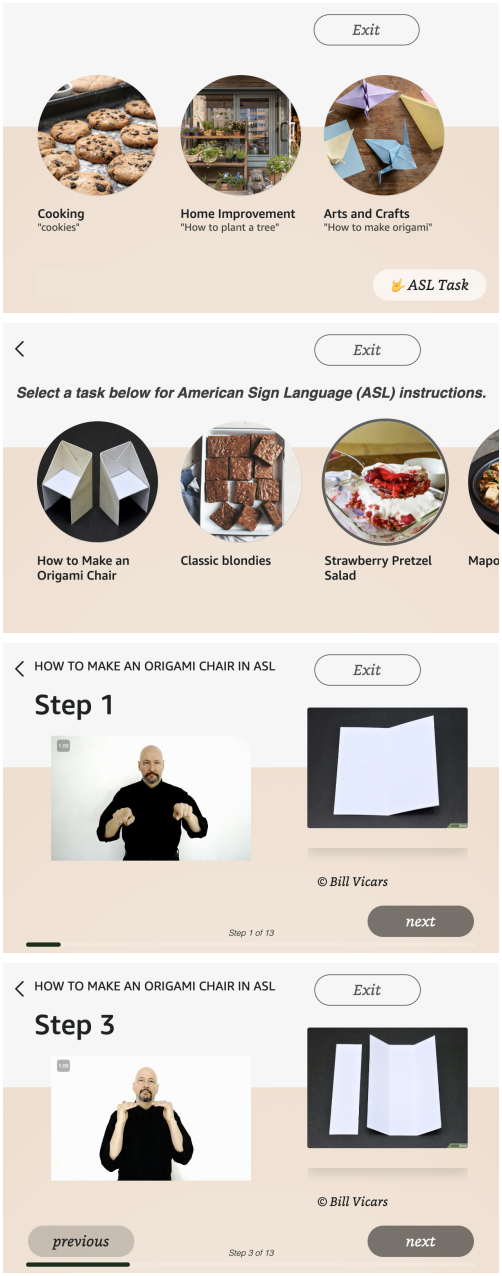

Figure 2: A storyboard of all the screens for an origami task with ASL video instructions. The first screen from the top is the landing page with an ASL Task button to enter the signed section. The second screen shows different recipes and task options. The following screens show an instruction step. Button interactions are especially important for signers as the audio is inaccessible.

We hope this effort brings more focus to the needs of signers and will be a step towards making large-scale dialogue systems more accessible to all users.

2 Related Work

With the rise of voice assistant devices, the DHH community has been mostly left behind. Yet, there have been multiple lines of work to make them more accessible. Accessibility of personal assistant devices to the Deaf and Hard of Hearing community has been assessed multiple times before by Glasser et al. (2017, 2020); Bragg et al. (2020). In addition, design approaches incorporating the DHH community have been proposed by Anindhita and Lestari (2016); Hariharan et al. (2018). We build on these in our system design.

Most of the current work in interactive system design focuses on sign recognition with the help of cameras. For instance, in Wojtanowski et al. (2020) Wizard-of-Oz studies have been done where Alexa is combined with a camera to detect signs. In SIGNS project[1], Alexa recognizes specific gestures for simple task completion (such as getting the weather forecast with a specific gesture), and Huang et al. recognized signs for a healing robot. Even though these systems provide a means for recognizing signs, they fall short in generating signs, which we focus on in this paper.

There has been some line of work by Nasihati Gilani et al. (2019) in generating avatars for 6-month-old babies to learn ASL. Also, Hrúz et al. (2011) deployed a kiosk with sign recognition and generation capabilities for Czech Sign Language. However, these have not resulted in a widely available system.

On the other hand, sign language processing has been widely studied under controlled conditions. Even though sign language generation and translation tasks are still open problems, transformer-based models in Yin and Read (2020); Yin et al. (2021); Moryossef et al. (2021); Inan et al. (2022); Müller et al. (2023); Lin et al. (2023); Viegas et al. (2023) have shown that it is possible to automate them better. As a core contribution, we present a framework to apply any of these models in large-scale interactive environments.

In order to make our system useful for signers, we need to mitigate their cognitive load interpreting instructions from multimodal devices. Models for the cognitive aptitudes and cognitive loads of sign language interpreters have been studied before by Macnamara (2012); Du Toit (2017); Tiselius (2018); Chambers (2020). These models help guide the design principles of our system, as the user will need to focus on multiple modalities simultaneously through the visual modality, which increases cognitive load.

3 A Goal-Oriented Dialogue System with Signed Instructions

We design a multimodal goal-oriented dialogue system as part of the Alexa Prize TaskBot Challenge 2 (Agichtein et al., 2023) and incorporate signed instructions. The main dialogue system that we develop follows a typical modular design: Natural Language Understanding (NLU), Dialogue Manager (DM), and Natural Language Generation (NLG). In this setting, we embed signed instructions into the multimodal NLG module (Figure 3).

Due to privacy regulations, Alexa does not allow third parties to process user gestures and videos. Hence, to increase accessibility for signers, we choose to generate signed instructions instead of recognizing signs. To support users who cannot—or prefer not to—provide voice input, our system has a scrollable touchscreen with buttons. This enables us to have a full dialogue system for signers while complying with regulations.

3.1 Task Description

We take as input a task JSON with step-by-step English text instructions, images, title, main image, and ingredients and output a JSON array of user interface screens corresponding to the gloss translations for each step and their corresponding sign videos (see Appendix A). The tasks are in the domains of cooking, home improvement, arts and crafts, and gardening. We provide our signed instruction generation as a standalone library for the camera-ready version of this paper.

3.2 Community Co-Design

To inform our system design choices, we connect with collaborators from the Deaf and Hard of Hearing (DHH) signing community at Gallaudet University (a prestigious higher education institution chartered for the DHH community). We incorporate the feedback from signers into the system's design.

The feedback incorporated into our design process includes considering the cognitive load of sign-

[1] https://projectsigns.org/

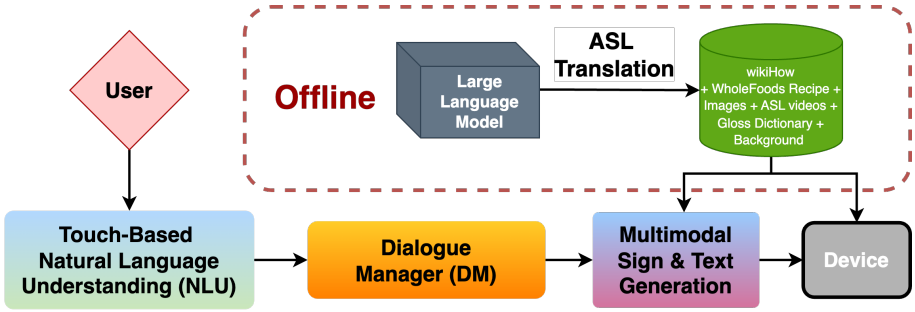

Figure 3: The overall architecture of our dialogue system with sign instructions for American Sign Language. Offline LLM translations make it easier to plug in a signing module into a traditional dialogue architecture.

ers, altering the dimensions of the text, video, and images used to communicate instructions, choosing which information to present as text versus signed videos (compare screens in Figure 2 and Appendix E for the placement of text and signed videos in the same screen), and updating the design of the interface for ASL signers.

4 Our Signed Instruction Framework

We employ the framework shown in Figure 1 to generate signed instructions. We first retrieve instructions for a given task, and then we convert each step into gloss tokens, which are intermediary textual representations using rule-based sign language translation algorithms and LLMs. Afterward, we segment each instruction into separate gloss tokens, retrieve sign videos for each, and stitch them back-to-back to create a continuous video sequence. For each step, we display this sequence of videos and a picture of the step. The picture for each step generally shows the result of the action as described in the sign instructions. This approach is summarized in Algorithm 1.

4.1 Large Language Model Translation

For the translation of spoken English instructions to textual representations of ASL (glosses), we prompt LLMs. Multiple methods exist in implementing text-to-gloss translation: human annotation, rule-based automatic translation with heuristics (Othman and Jemni, 2012a), fine-tuned transformer-based models (Camgoz et al., 2018; Yin and Read, 2020), and prompting LLMs (Lee et al.). We make our system adaptable to all of these alternatives for text-to-gloss translation. Any one of these models can be plugged into line 4 of Algorithm 1. We choose LLM translation for our current system due to its scalability, translation understandability, and ability to adapt to out-of-domain text.

Algorithm 1 Signed Instruction Retrieval

1: $G \leftarrow \{\}$
2: $I \leftarrow$ instruction steps
3: **for** i in I **do**
4: translated $\leftarrow LLM(i)$
5: translated $\leftarrow$ PRUNE(translated)
6: **end for**
7: $S \leftarrow \{\}$
8: **for** i in translated **do**
9: **for** t_i in i **do**
10: $S[t_i] \leftarrow SIGN\_VIDEO(t_i)$
11: **end for**
12: **end for**
13: $V \leftarrow [\,]$
14: **for** i in I **do**
15: **for** t_i in i **do**
16: $V[i] \leftarrow V[i] + S[t_i]$
17: **end for**
18: **end for**
19: **return** V

We show in our system evaluation in section §5 that there is a trade-off between using LLMs or rule-based heuristics for text-to-gloss translation. Mainly, LLMs generate more diverse translations, while rule-based heuristics have higher accuracy depending on the video dataset size.

Our instructions consist of WholeFoods recipes[2] and WikiHow tasks[3]. First, we aggregate all the instruction steps of the task in a JSON construct (given in Appendix A), then using the OpenAI chat API we prompt `gpt-3.5-turbo` to "translate each step to American Sign Language gloss", and request the result in a JSON format.[4] We then ag-

[2] www.wholefoodsmarket.com/recipes
[3] www.wikihow.com
[4] Our parameters for the API call are, *temperature*=1, max *tokens*=1000, *top p*=1, *frequency penalty*=0, and *presence penalty*=0.

gregate all these steps for all recipes and tasks. For recipes, we do not translate the ingredients to glosses, as our community outreach surveys indicate that users prefer to see the ingredients written statically on the screen instead of signed versions (see Figure 1 for a reference of text-to-gloss translation steps).

After these instructions are generated, we have an additional stage of manual correction of LLM-generated glosses using rule-based heuristics for quality[5]. We also remove the punctuation in glosses, capitalize them, and concatenate the fingerspellings—in which fingers form individual letters to spell out words—if annotated using the hyphen notation (i.e. "F-I-N-G-E-R"). Here, we check that the glosses are unique across the tasks, they are all present in the available video dictionary, and they follow the general rules of ASL.

4.2 Sign Video Processing

We process the videos in four steps. First, we collect sign videos corresponding to all the glosses in our instruction set from an online platform. Then we store these videos, retrieve them on the fly while presenting instructions, and stitch them together. We give the details of these steps in the following paragraphs.

Sign Video Collection For video collection, we use widely available American Sign Language sign dictionary videos from video sharing platforms with Creative Commons licenses online [6]. We mainly use videos from Lifeprint, but if they do not contain a specific sign video, we use the ASL-Dictionary on YouTube as the backup source. If neither of these sources has a sign available, we first check if the gloss can be deconstructed into other signs or fingerspelled. If so, we check the videos for the deconstructed versions and concatenate them into a single video. If these options are not available and the gloss is crucial to the meaning of the instruction, then we search for a synonym. If it is not crucial to the meaning of the instruction, then we drop the gloss.

Video Storage We generate a dictionary for all the available sign glosses (found in Appendix Section A) and upload all the videos with their gloss as their filename to an Amazon AWS S3 bucket for storage.

Gloss-by-Gloss Sign Retrieval During a user's live use of the system for signed instructions, we retrieve videos on a token level using the video URL by cross-referencing its gloss filename. As the last step, after retrieving all the video URLs on the fly for each gloss in each instruction, we concatenate all of the URLs corresponding to the glosses together and then present them on the user interface of the app as a single stream of a video (see Figure 2).

5 System Evaluation

We evaluate our system both quantitatively and qualitatively. Because this is the first deployment of a task-oriented signed multimodal dialogue system, we chiefly compare the system with the non-signed portion of our task-oriented dialogue system. We first evaluate the performance of our LLM text-to-gloss translation and discuss the trade-offs of using an LLM for translation. Then, we evaluate our algorithm using traditional information retrieval metrics. Finally, we compare user ratings and provide detailed qualitative analyses by an expert who is fluent in ASL.

Automatic Metrics							
BLEU				ROUGE	METEOR	ChrF	WER
1	2	3	4				
9.52	1.59	0.42	0.16	0.11	0.11	23.99	2.146

	F1	Recall	Precision
BERTScore	0.80	0.81	0.79

Table 1: This table shows the automatic metric results between LLM and rule-based translations. Tasks on the web do not contain readily available ground-truth glosses. BERTScore is the best indicator of translation success.

Text-to-Gloss Translation Analysis In this section, we analyze the performance of LLM-based translations using traditional automatic text metrics (see Table 1). As also described in section § 4.1, we experiment with two translation strategies: 1) LLM translations and 2) rule-based gloss translations with heuristics. We use the rule-based heuristics strategy as ground truth in our results here because no human-annotated ASL ground truth exists for our datasets, and the accuracy of rule-based translations is high when compared to human annotations in the works of Othman and Jemni (2012b, 2019).

[5] this curation step can be omitted for the deployment of larger systems with bigger task sets, where it might be infeasible to go over each task step and glosses manually.

[6] Lifeprint.com, and the ASLDictionary channel accessible on YouTube: https://youtube.com/@smartsigndictionary

In order to generate rule-based glosses, we use the Algorithm given in Appendix B. Automatic evaluation metrics for sign translations do not yet exist. Hence, we present results using traditional automatic evaluation metrics such as BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005), ChrF (Popović, 2015), and BERTScore (Zhang et al., 2020) between LLM-generated glosses and the rule-based glosses. In this case, BERTScore is more insightful than traditional metrics because the semantic representation of tokens is more important in glossing than the specific n-gram differences.

For our system, we deploy with LLM-based translations and are able to scale from only 1-3 tasks with ASL expert manual annotations to 150 supported tasks with LLM-based translations. As shown in Figure 3, the LLM translations happen offline as all of our tasks are pre-determined. Right after the tasks are translated to ASL glosses, we have a quality control stage before they are presented to the user. So, our overall translation pipeline is a human-in-the-loop system. During the duration of our dialogue system's deployment, we observe that using LLMs reduces the time spent on the manual checking process by human annotators from 10 minutes per instruction sentence to 1 minute per sentence.

Retrieval Metrics No automatic evaluation mechanism exists for signed interactive systems; hence, in this section, we introduce two retrieval metrics—Hit Rate and Recall@1—for our Signed Instruction Retrieval Algorithm (see Algorithm 1) with the two translation modules separately. Furthermore, we also present an analysis of the changes in Hit Rate and Recall@1 in response to increases in the available video dataset size in Figure 4.

We use the following simplified definitions of Hit Rate and Recall@1:

$$\text{Hit Rate} = \frac{\#\text{ glosses w/ videos}}{\text{total }\#\text{ of glosses}} \quad (1)$$

$$\text{Recall@1} = \frac{\#\text{ glosses w/ videos}}{\#\text{ synonyms of glosses w/o videos} + \#\text{ glosses w/ videos}} \quad (2)$$

Essentially, Hit Rate measures how accurate the system is in finding videos for a given token, and Recall@1 tells how precise the system selects videos corresponding to a token among a set of

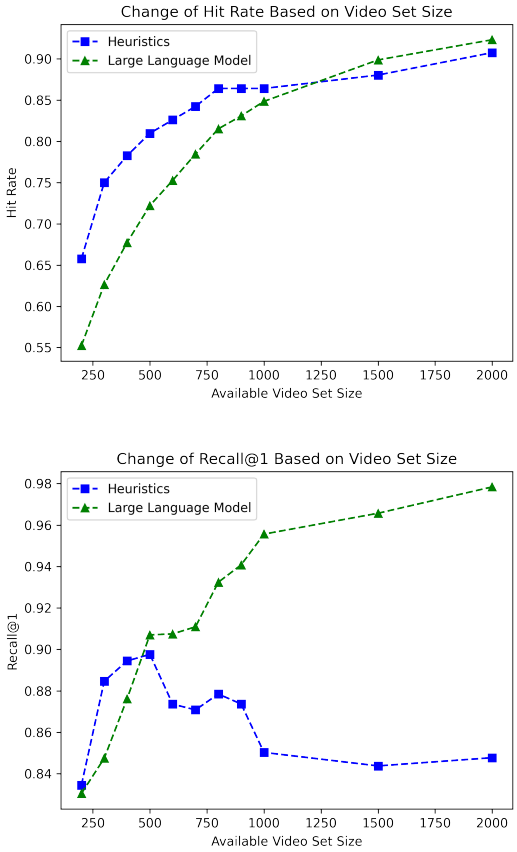

Figure 4: These plots show the changes in Hit Rate and Recall@1 for our signed instruction retrieval algorithm as the available video set increases in size. Two lines represent two methods of translation from text to gloss. In a constrained setup with limited sign video storage, these plots show how many videos are needed with different translation strategies. Overall, LLMs have more diverse translations, while rule-based heuristics provide more accurate translations changing with the video dataset size.

synonyms. For instance, for a task step consisting of glosses "CHOCOLATE CHOP ADD DOUGH MIX STIR" if the system has only videos for CHOP, ADD, COMBINE, and STIR, then the Hit Rate will be 0.5, as three out of six glosses do not have videos; and Recall@1 will be $3/4$, where the denominator also contains any synonym of a gloss that does not have a corresponding video (MIX and COMBINE are considered synonyms in this case). Hit Rate and Recall@1 are complimentary metrics where Hit Rate shows the direct presence of sign videos while Recall@1 indirectly shows how diverse the glosses and selected videos are due to the inclusion of synonyms in the denominator where multiple glosses may exist for the same video that we have in our database. We give detailed mathematical defini-

tions for both of these metrics in Appendix C.

Looking at the resulting plots in Figure 4, we can make several claims. For Hit Rate, both of the translation strategies produce similar results because our video database covers a majority of glosses present in the restricted domain of cooking and wikiHow tasks. For Recall@1, there is a dramatic difference between LLMs and heuristics. This happens because rule-based heuristics use nearly the same tokens from the text, while LLMs can generate synonymous glosses for a given token. For a more example-driven explanation, please refer to Appendix D.

Overall, the Recall@1 for our Algorithm has a minimum of around 80% and a maximum of 98%—as observed in Figure 4. This shows that our algorithm can easily be deployed as part of dialogue systems with signed instructions regardless of whether we use LLMs or rule-based heuristics translations.

User Rating Comparisons Our system interacted with a large number of public users for over a period of six months. Because this is the first task-oriented dialogue system with signed instructions, it increases our user outreach on international platforms by a large margin. However, adding this functionality could decrease overall user ratings if they do not deem the interface usable or are unsure about what ASL is. Thus, we examine the ratings before and after adding the signed instructions to our system. As shown in Appendix 7, our user ratings remain constant after adding support for this feature. Thus, we find that, besides making task-oriented systems accessible to a larger audience, adding support for signed instructions does not decrease user ratings.

Expert Qualitative Analysis One author fluent in ASL evaluated the system with special regard to the usability and clarity of the information presented. This evaluator noted *two primary strengths*: 1) the multimodal instructional support provided by having both the ASL descriptions and the instructional images available, particularly for the step-by-step tasks such as origami folding; 2) the ease of processing and attending to multiple modalities given the clear layout without overwhelming the user. To expand, giving the user the option to attend to the signed content or the referent of the images (e.g., step-by-step origami folding) allowed them to rely on each form of information to the extent they prefer. The clear layout does not overwhelm the user with too many streams of information. It also allows for sufficient processing of either sign videos, images, or both without distracting the user.

The *primary limitation* of the current system lies in the segmented nature of the ASL videos. Currently, there is a lack of smooth transitions between signs, and different signers present each sign within one instruction. The flow of the signs appears disjointed, consequently impeding clear understanding. The absence of step-by-step visuals in certain tasks necessitates increased reliance on signing. The disjointed nature of the current signing videos rendered some tasks less comprehensible.

Overall, the multimodal presentation of signing alongside informative images enhances accessibility and suggests that a dynamic display of signed content will greatly enhance future task-oriented dialogue systems. For future iterations of our system, we plan to incorporate either human models signing the entire content or synthesized avatars (Quandt, 2020; Quandt et al., 2022).

6 Conclusion

In this work, we discussed a multimodal, task-oriented dialogue system designed to generate ASL instructions on a platform with global reach. Emphasizing the critical importance of Deaf and Hard-of-Hearing (DHH) community engagement throughout the development cycle, our approach integrates extensive feedback from both the signing community and experts in the field. Our system not only marks a significant technological milestone but also enriches the dialogue on how video-based ASL instruction delivery can be effectively scaled internationally. We observed a nuanced preference among signers for avatar-based instructions—a finding underscored by our expert analysis. Our system has improved the landscape of conversational AI, making it accessible and responsive to the unique needs of the DHH community.

We make the code available for our pipeline and encourage future researchers to incorporate it into their own task-oriented systems to increase accessibility. We hope that this system is a step towards developing dialogue systems that can understand *and* generate signs for all signed languages. We encourage everybody to interact with signed tasks by visiting `https://huggingface.co/spaces/merterm/signed-instructions`.

7 Acknowledgement

This project was completed as part of and received funding from the Alexa Prize TaskBot Challenge 2. We would like to thank the Alexa Prize team, especially Lavina Vaz and Michael Johnston, for supporting us throughout the competition and for giving us the resources to develop and deploy our system to a large audience. We would also like to thank our team members: Yuya Asano, Qi Cheng, Dipunj Gupta, Sabit Hassan, Jennifer Nwogu, and Paras Sharma.

References

Eugene Agichtein, Michael Johnston, Anna Gottardi, Cris Flagg, Lavina Vaz, Hangjie Shi, Desheng Zhang, Leslie Ball, Shaohua Liu, Luke Dai, Daniel Pressel, Prasoon Goyal, Lucy Hu, Osman Ipek, Sattvik Sahai, Yao Lu, Yang Liu, Dilek Hakkani-Tür, Shui Hu, Heather Rocker, James Jeun, Akshaya Iyengar, Arindam Mandal, Saar Kuzi, Nikhita Vedula, Oleg Rokhlenko, Giuseppe Castellucci, Jason Ingyu Choi, Kate Bland, , Yoelle Maarek, and Reza Ghanadan. 2023. Alexa, let's work together: Introducing the second alexa prize taskbot challenge. In *Alexa Prize TaskBot Challenge 2 Proceedings*.

Vidia Anindhita and Dessi Puji Lestari. 2016. Designing interaction for deaf youths by using user-centered design approach. In *2016 International Conference On Advanced Informatics: Concepts, Theory And Application (ICAICTA)*, pages 1–6. IEEE.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Johnna Blair and Saeed Abdullah. 2020. It Didn't Sound Good with My Cochlear Implants: Understanding the Challenges of Using Smart Assistants for Deaf and Hard of Hearing Users. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4):1–27.

Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, Christian Vogler, and Meredith Ringel Morris. 2019. Sign Language Recognition, Generation, and Translation: An Interdisciplinary Perspective. In *ASSETS '19: Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*, pages 16–31. Association for Computing Machinery, New York, NY, USA.

Danielle Bragg, Meredith Ringel Morris, Christian Vogler, Raja Kushalnagar, Matt Huenerfauth, and Hernisa Kacorri. 2020. Sign Language Interfaces: Discussing the Field's Biggest Challenges. In *CHI EA '20: Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, pages 1–5. Association for Computing Machinery, New York, NY, USA.

Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. 2018. Neural Sign Language Translation. [Online; accessed 9. Oct. 2023].

Cindy Chambers. 2020. Mindfulness and Interpreter Cognitive Load. *Digital Commons@WOU*.

Srisavakon Dangsaart, Kanlaya Naruedomkul, Nick Cercone, and Booncharoen Sirinaovakul. 2008. Intelligent Thai text – Thai sign translation for language learning. *Computers & Education*, 51(3):1125–1141.

P. T. Petri Du Toit. 2017. Mitigating the cognitive load of South African Sign Language interpreters on national television. [Online; accessed 20. Jul. 2023].

Abraham Glasser, Kesavan Kushalnagar, and Raja Kushalnagar. 2017. Deaf, Hard of Hearing, and Hearing Perspectives on Using Automatic Speech Recognition in Conversation. In *ASSETS '17: Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, pages 427–432. Association for Computing Machinery, New York, NY, USA.

Abraham Glasser, Vaishnavi Mande, and Matt Huenerfauth. 2020. Accessibility for Deaf and Hard of Hearing Users: Sign Language Conversational User Interfaces. In *CUI '20: Proceedings of the 2nd Conference on Conversational User Interfaces*, pages 1–3. Association for Computing Machinery, New York, NY, USA.

Dhananjai Hariharan, Sedeeq Al-khazraji, and Matt Huenerfauth. 2018. Evaluation of an English Word Look-Up Tool for Web-Browsing with Sign Language Video for Deaf Readers. In *Universal Access in Human-Computer Interaction. Methods, Technologies, and Users*, pages 205–215. Springer, Cham, Switzerland.

Marek Hrúz, Pavel Campr, Zdenek Krňoul, Milos Železný, Oya Aran, and Pinar Santemiz. 2011. Multimodal dialogue system with sign language capabilities. In *ASSETS '11: The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility*, pages 265–266. Association for Computing Machinery, New York, NY, USA.

Xuan Huang, Bo Wu, and Hiroyuki Kameda. Development of a Sign Language Dialogue System for a Healing Dialogue Robot. In *2021 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on*

Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), pages 25–28. IEEE.

Mert Inan, Yang Zhong, Sabit Hassan, Lorna Quandt, and Malihe Alikhani. 2022. Modeling intensification for sign language generation: A computational approach. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2897–2911, Dublin, Ireland. Association for Computational Linguistics.

Navroz Kaur Kahlon and Williamjeet Singh. 2023. Machine translation from text to sign language: a systematic review. *Univ. Access Inf. Soc.*, 22(1):1–35.

Leonard P Kelly. 2003. Considerations for designing practice for deaf readers. *Journal of deaf studies and deaf education*, 8(2):171–186.

Huije Lee, Jung-Ho Kim, Eui Jun Hwang, Jaewoo Kim, and Jong C. Park. Leveraging Large Language Models With Vocabulary Sharing For Sign Language Translation. In *2023 IEEE International Conference on Acoustics, Speech, and Signal Processing Workshops (ICASSPW)*, pages 04–10. IEEE.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Kezhou Lin, Xiaohan Wang, Linchao Zhu, Ke Sun, Bang Zhang, and Yi Yang. 2023. Gloss-free end-to-end sign language translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12904–12916, Toronto, Canada. Association for Computational Linguistics.

John L Luckner and C Michele Handley. 2008. A summary of the reading comprehension research undertaken with students who are deaf or hard of hearing. *American annals of the deaf*, 153(1):6–36.

Brooke Macnamara. 2012. Interpreter Cognitive Aptitudes. *Journal of Interpretation*, 19(1):1.

Amit Moryossef, Kayo Yin, Graham Neubig, and Yoav Goldberg. 2021. Data augmentation for sign language gloss translation. In *Proceedings of the 1st International Workshop on Automatic Translation for Signed and Spoken Languages (AT4SSL)*, pages 1–11, Virtual. Association for Machine Translation in the Americas.

Mathias Müller, Zifan Jiang, Amit Moryossef, Annette Rios, and Sarah Ebling. 2023. Considerations for meaningful sign language machine translation based on glosses. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 682–693, Toronto, Canada. Association for Computational Linguistics.

Setareh Nasihati Gilani, David Traum, Rachel Sortino, Grady Gallagher, Kailyn Aaron-Lozano, Cryss Padilla, Ari Shapiro, Jason Lamberton, and Laura-Ann Petitto. 2019. Can a Signing Virtual Human Engage a Baby's Attention? In *IVA '19: Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, pages 162–169. Association for Computing Machinery, New York, NY, USA.

Achraf Othman and M. Jemni. 2012a. English-ASL Gloss Parallel Corpus 2012: ASLG-PC12. [Online; accessed 20. Jul. 2023].

Achraf Othman and Mohamed Jemni. 2012b. English-asl gloss parallel corpus 2012: Aslg-pc12.

Achraf Othman and Mohamed Jemni. 2019. Designing High Accuracy Statistical Machine Translation for Sign Language Using Parallel Corpus: Case Study English and American Sign Language. *J. Inf. Technol. Res.*, 12(2):134–158.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Lorna Quandt. 2020. Teaching ASL Signs using Signing Avatars and Immersive Learning in Virtual Reality. In *ASSETS '20: Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility*, pages 1–4. Association for Computing Machinery, New York, NY, USA.

Lorna C. Quandt, Athena Willis, Melody Schwenk, Kaitlyn Weeks, and Ruthie Ferster. 2022. Attitudes Toward Signing Avatars Vary Depending on Hearing Status, Age of Signed Language Acquisition, and Avatar Type. *Front. Psychol.*, 13:730917.

Elisabet Tiselius. 2018. Exploring Cognitive Aspects of Competence in Sign Language Interpreting of Dialogues: First Impressions. *HJLCB*, (57):49–61.

Carol Bloomquist Traxler. 2000. The Stanford Achievement Test, 9th Edition: National Norming and Performance Standards for Deaf and Hard-of-Hearing Students. *J. Deaf Stud. Deaf Educ.*, 5(4):337–348.

Carla Viegas, Mert Inan, Lorna Quandt, and Malihe Alikhani. 2023. Including facial expressions in contextual embeddings for sign language generation. In *Proceedings of the 12th Joint Conference on Lexical and Computational Semantics (*SEM 2023)*, pages 1–10, Toronto, Canada. Association for Computational Linguistics.

Gabriella Wojtanowski, Colleen Gilmore, Barbra Seravalli, Kristen Fargas, Christian Vogler, and Raja Kushalnagar. 2020. "Alexa, Can You See Me?" Making Individual Personal Assistants for the Home Accessible to Deaf Consumers. *California State University, Northridge.*

Kayo Yin, Amit Moryossef, Julie Hochgesang, Yoav Goldberg, and Malihe Alikhani. 2021. Including signed languages in natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7347–7360, Online. Association for Computational Linguistics.

Kayo Yin and Jesse Read. 2020. Better sign language translation with STMC-transformer. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5975–5989, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert.

A Input Constructs

Here we show the JSON format of the tasks:


```
{
        "title": "Classic blondies",
        "main_image": "496287a8ff0ecd2875af4c.jpg",
        "ingredients": [
            "1\u00bd sticks butter, plus more for greasing",
            "1\u2153 cups brown Muscovado sugar",
            "2 eggs",
            "1 tbsp vanilla extract",
            "1\u00bd cups all-purpose flour",
            "1 tsp salt",
            "6 oz semisweet chocolate (chopped) or chocolate chips"
        ],
        "task_images": [
            "00052039c78b0c243540f.jpg",
            "08b46fd0067b1f931863e.jpg",
            "12f87efcb8f5623aaf9ab.jpg",
            "a5a679a26a4e5f26ce57c.jpg",
            "58e3947ac34dbb60f47a7.jpg"
        ],
        "task_texts": [
            "Preheat oven to 175F. Line a square baking pan with aluminum foil,
            letting some hang over the sides. Grease the foil with a pat of butter.
            Set pan aside. Melt remaining butter in a small saucepan over
            medium-low heat until it starts to brown and smell nutty, swirling
            it around the pan from time to time. Transfer to a large mixing bowl
            and allow to cool completely.",
            "Add sugar to cooled butter and mix until emulsified. Then, add eggs
            and vanilla to the butter and sugar mixture, and beat until combined.",
            "Whisk together flour and salt in a small bowl. Stir into butter
            mixture and beat until combined.",
            "Chop the chocolate and add to the batter, or simply use chocolate
            chips. Stir until incorporated.",
            "Transfer batter to prepared baking dish and bake in preheated oven
            at 175F for approx. 20 \u2013 30 min. until golden brown. Blondies
            should not be too soft in the middle and just starting to crack on
            top. Cool completely, and then use the  ALUMINUM-FOIL to help
            transfer the blondies out of the pan. Cut into squares and enjoy!"
        ],
        "task_glosses": [
            "OVEN HEAT PAN SQUARE BAKE LINE ALUMINUM-FOIL BUTTER SPREAD
            PAN PUT-ASIDE BUTTER REMAINING MELT PAN TRANSFER BOWL LARGE
            COOL COMPLETELY",
            "BUTTER COOL MIX SUGAR ADD MIX BUTTER SUGAR MIX ADD VANILLA
            EGG COMBINE MIX",
            "FLOUR SALT WHISK BOWL SMALL TOGETHER MIX BUTTER ADD STIR
            COMBINE MIX",
            "CHOCOLATE CHOP ADD DOUGH MIX STIR",
            "DOUGH TRANSFER DISH BAKE OVEN HEAT APPROXIMATELY 20 30
            MINUTE BROWN MIDDLE SOFT COMPLETELY COOL ALUMINUM-FOIL
            TRANSFER PAN SQUARE CUT ENJOY"
        ]
}
```

Here is the dictionary of all the available glosses that have corresponding videos on the system.

```
[
    "1", "2", "3", "4", "5", "6", "7", "8", "9", "10", "15",
    "20", "25", "30", "35", "40", "45", "50", "55", "60", "65", "70",
    "75", "80", "85", "90", "95", "100", "3-MINUTES",
    "5-MINUTES", "ADD", "ALL", "AND", "ALUMINUM-FOIL",
    "APPROXIMATELY", "BACON", "BAKE", "BAKING-POWDER", "BAKING-SODA",
    "BERRY", "BLACK", "BLENDER", "BOIL", "BOTTOM", "BOWL",
    "BREAD", "BRING", "BROWN", "BUTTER", "BUTTERMILK",
    "CAKE-PAN", "CAKE", "CAREFUL", "CENTER", "CHAIR",
    "CHICKEN", "CHOCOLATE", "CHEESE", "CHOP", "CINNAMON",
    "COAT", "COMBINE", "COMPLETELY", "CONNECT", "CONTINUE",
    "COOK", "COOL", "CORNER", "COVER", "CREAM", "CREAM-CHEESE",
    "CRUSH", "CUP", "CUT", "DEGREE", "DISH", "DIVIDE",
    "DRAIN", "DOUGH", "DOWN", "EACH", "EDGE", "EGG",
    "ENJOY", "EQUAL", "FAHRENHEIT", "FINISH", "FIRST",
    "FLAP", "FLIP-OVER", "FLOUR", "FOLD", "FORM", "FOUR",
    "FRY", "GARLIC", "GOLDEN", "GREEN", "GROUND", "HALF",
    "HEAT", "HOUR", "HOT", "IF", "IN", "INCH", "LARGE",
    "LAST", "LEFT", "LINE", "LOW", "MAKE", "MEAT",
    "MEDIUM", "MEET", "MELT", "MIDDLE", "MINUTE",
    "MIX", "MORE", "NOT", "OIL", "OLIVE", "ON", "ONION",
    "OR", "OTHER", "OVEN", "OVER", "PAN", "PANCAKE",
    "PAPER", "PART", "PEPPER", "PLACE-INTO", "PORK",
    "POT", "POTATO", "POUR", "PRESS", "PRETZEL",
    "PUT", "PUT-ASIDE", "QUARTER", "RED", "REDUCE",
    "REMAINING", "REMOVE", "REPEAT", "RICE", "RIGHT",
    "ROLL", "SALT", "SAUCE", "SAUTEE", "SEASON", "SEPARATE",
    "SERVE", "SIDE", "SIMMER", "SIX", "SLICE", "SMALL",
    "SMOOTH", "SOFT", "SOUP", "SPREAD", "SPRINKLE", "SQUARE",
    "SQUASH", "START", "STIR", "STRONG", "SUGAR", "SYRUP",
    "TABLESPOON", "TEASPOON", "THEN", "THREE-QUARTERS",
    "THROUGH", "TO", "TOMATO", "TOP", "TOGETHER", "TOSS",
    "TRANSFER", "UNTIL", "UP", "USE", "VANILLA", "VERTICAL",
    "WATER", "WAY", "WELL", "WHISK", "WITH"
]
```

B Rule-based Gloss Translation Algorithm

We give the pseudocode for the rule-based heuristics algorithm as follows:

Algorithm 2 Rule-based Heuristic Glosses

```
heuristic_glosses ← []
for sentence in task['task_texts'] do
    sentence ← UPPERCASE(sentence)
    text ← TOKENIZE(sentence)
    pos_tagged ← POSTAGGING(text)
    for token in pos_tagged do
        if IsNotDesiredPOS(token[1]) then
            REMOVETOKEN(pos_tagged, token)
        end if
    end for
    for i in range(LENGTH(pos_tagged)) do
        pos_tagged[i] ← (LEMMATIZE(pos_tagged[i][0]), pos_tagged[i][1])
    end for
    sentence ← ""
    for token in pos_tagged do
        sentence ← sentence + token[0] + ""
    end for
    sentence ← STRIP(sentence)
    heuristic_glosses.APPEND(sentence)
end for
return heuristic_glosses
```

C Detailed Mathematical Definitions for Retrieval Metrics

To define Hit Rate and Recall@1 more precisely, we first introduce some requisite definitions:

- D: set of glosses in our dictionary
- n: total number of task instructions
- $I = \{i_0, i_1, ..., i_n\}$: set of all task instructions
- m_k: total number of glosses in instruction k
- $i_k \in I =< g_{k0}, g_{k1}, ..., g_{km_k} >$
- $g_{kl} \in i_k$: gloss in instruction i_k (ordered)
- $syn(g)$: the set of synonyms found for gloss g using `wordnet.synsets`

We formalize our simplified definitions of Hit Rate and Recall@1 below, using our notation. Note that because we take into account repeated glosses in our instruction set, the sets below are *multisets* and thus contain repeated elements that are factored into the cardinality of the set.

$$\text{Hit Rate} = \frac{|g_{kl} : g_{kl} \in D, i_k \in I, g_{kl} \in i_k|}{|g_{kl} : i_k \in I, g_{kl} \in i_k|} \quad (3)$$

$$\text{Recall@1} = \frac{|g_{kl} : g_{kl} \in D, i_k \in I, g_{kl} \in i_k|}{|g_{kl} : g_{kl} \in D, i_k \in I, g_{kl} \in i_k| + |g_{kl} : g_{kl} \notin D, i_k \in I, g_{kl} \in i_k|} \quad (4)$$

D Detailed Examples for Retrieval Metrics

For example, for the instruction, *"Chop chocolate and add to batter. Stir until incorporated."*, the LLM generates, "`CHOCOLATE CHOP ADD DOUGH MIX STIR`", while heuristics generates "`CHOP CHOCOLATE ADD BATTER STIR UNTIL INCORPORATE`". Here, it can be seen that LLM produces `DOUGH` (a synonym of "batter" for our purposes), while heuristics directly uses the same wording. This adds diversity to the generated glosses, and as the number of videos increases, it positively affects the score of LLMs. For the heuristics algorithm, as the tokens are never changed into synonyms, even after a lot of videos are added to the set, the algorithm cannot retrieve videos and gets lower Recall@1 scores.

E Interface Details

We show more screenshots of details in the interface in Figures 5, and 6.

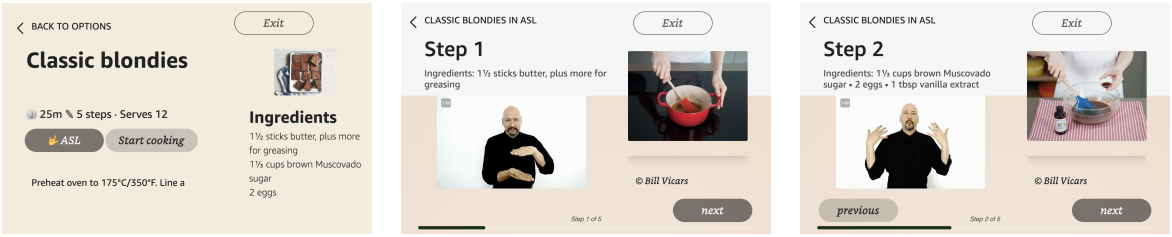

Figure 5: These are the screens for an alternative task of a classic blondies recipe. The main difference for recipes is that at each step, relevant ingredients are shown in addition to the signed instruction video. This is to ensure less cognitive load on the user. Also, the first panel shows the ASL button that exists in supported recipes.

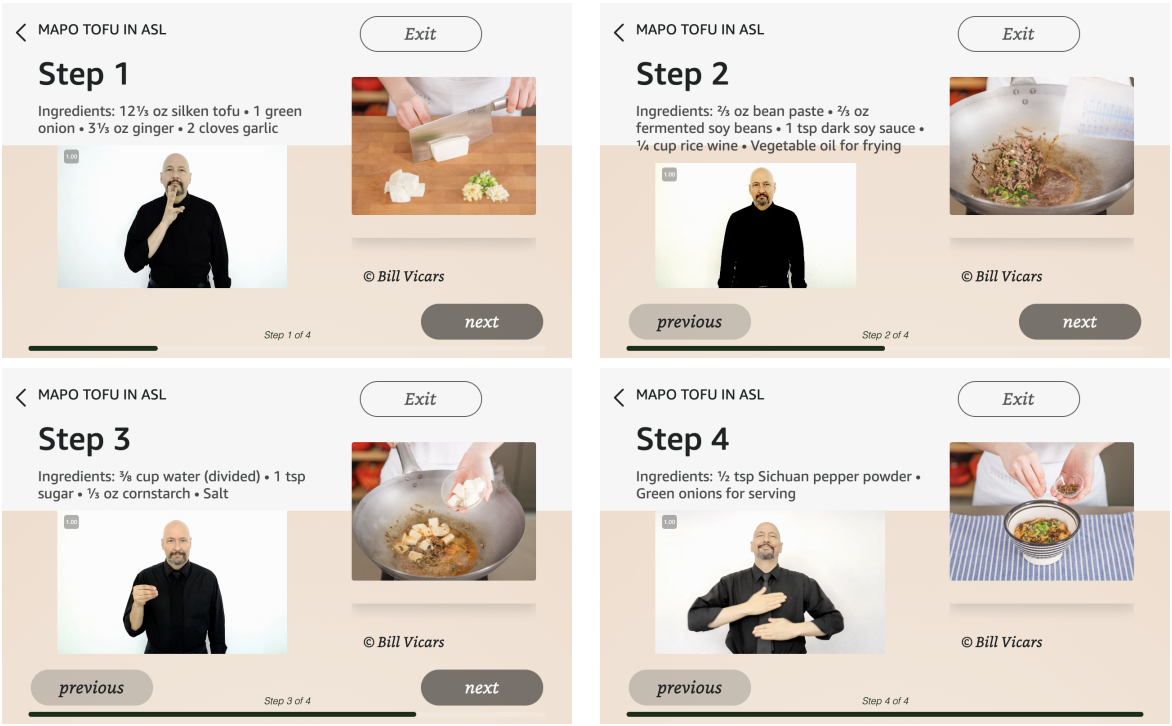

Figure 6: This figure demonstrates the screenshots of our signed multimodal dialogue bot for the recipe of Mapo Tofu. This example is chosen to stress the fact that certain international recipes that have terms that may not exist in ASL are also supported in the bot. In these cases, the ingredients are written on the screen and the instructions are signed without the specific terminologies, like "tofu", and images are shown to aid with grounding the referred ingredient.

F User Rating Analysis

We show a plot of 7-day averages of user ratings before and after adding support for signed instructions in Figure 7.

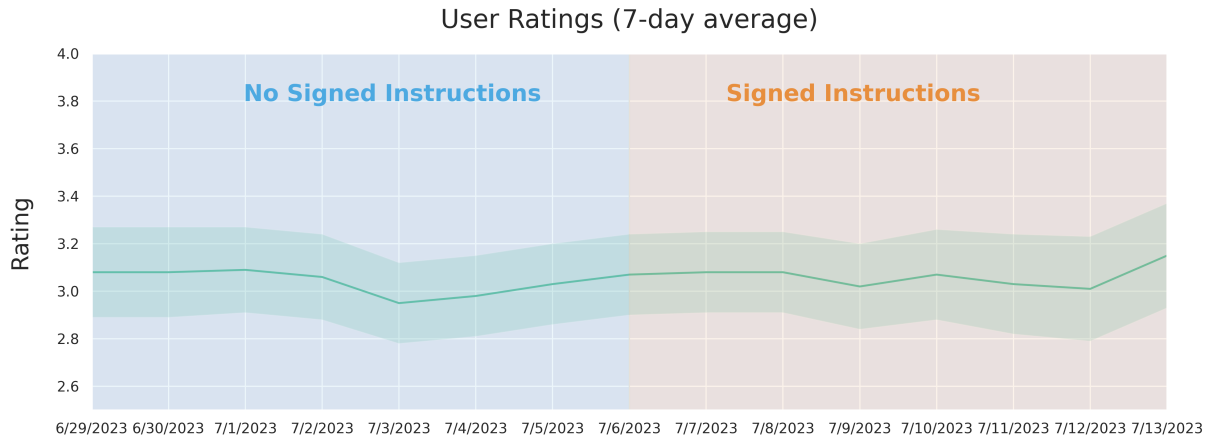

Figure 7: User ratings of our system before and after adding support for instructions in ASL. Here, we show the week before and after adding signed instructions. Reaching out to real users and communities that use signed languages is the main goal of our system. Adding ASL support allows our system to engage with a larger audience without decreasing overall user ratings.