Danish Sign Language Recognition in Virtual Reality Using Written Language Ensemble Learning

Peter Guld Leth
pleth18@student.aau.dk
Aalborg University
Aalborg, Denmark

Abstract

This work presents a multimodal ensemble Sign Language Recognition (SLR) model using an n-gram linear classifier for Natural Language Processing, a vector encoding based on euclidean distances for gesture recognition, and a fusion approach for confluencing the two. Furthermore, this work proposes a Virtual Reality (VR) User Interface (UI) based on prevailing usability heuristics. The SLR model was shown to have a mean classification accuracy of 41.5%, which is meaningfully below the state of the art, while the VR UI was found to not allow for sufficient levels of Adaptability. Still, there exists many ways in which the components of the SLR model could be improved, and it is the hope that derivative works can make use of the findings presented here for this.

Keywords

Sign Language Recognition, Gesture Recognition, Virtual Reality, Natural Language Processing, Machine Learning, Usability

1 Introduction

Sign languages serve as essential communication tools, offering individuals with hearing impairments an efficient and expressive medium to articulate their thoughts and emotions. However, the richness of these widely different languages also prevents connection between the deaf- and hearing communities, often leading to feelings of exclusion among sign language dependant minorities. Sign Language Recognition (SLR) has the potential to bridge this gap by transcribing signed language into written or spoken language, and thereby fostering effective communication [9, 18]. However, achieving a balance between the accuracy and practicality of SLR systems is a significant challenge [1, 9, 42].

Through my previous work using Virtual Reality (VR) technology, particularly the Meta Quest 2 for gesture recognition, I realized the potential of this platform for SLR due to its portability, affordability, and built-in tracking capabilities [2] compared to the state of the art. However, the biggest issue with this approach remains the lack of publicly available gesture data for model training, as well as there not existing any guidelines for how a User Interface (UI) is to be designed for a VR SLR context.

From this, a research question in two parts is given to be answered in this work:

(1) *Can a VR SLR model be developed requiring minimal training data while maintaining comparable performance to the state of the art?*
(2) *Can a set of guidelines for how to design VR UIs in the context of SLR be synthesized?*

In summary, this work set out to develop a Sign Language Recognition (SLR) system in Virtual Reality (VR) with minimal training data using Natural Language Processing (NLP) and ensemble learning in an attempt to answer (1). Secondly, this work also sought to document the efficient use of various VR usability considerations in the context of SLR, attempting to answer (2). The thinking here was that the proposed SLR system could be used for further explorations of less populous sign languages like Dansh Sign Language or "Dansk Tegnsprog" (DTS), while any usability findings could contribute towards unified usability guidelines for SLR applications in VR, making derivative works more directly comparable than they are today.

In so doing, this work presents a multimodal SLR model consisting of a Language Model (LM) using an n-gram linear classifier (3.2.1), a Gesture Model (GM) using a novel vector encoding (3.2.2), and a fusion approach using weighted-voting for confluencing the two models into one set of predictions (3.2.3). For the UI, this work presents an application flow for SLR, introduces the concept of the "HandGate" for ensuring highly regular application interactions, as well as a way of determining the ideal number of sentence samples to use (detailed in 3.1). This study also includes a usability evaluation (3.1.1), along with a thorough evaluation of the proposed SLR performance (4.2.1).

The performance and usability of the system is then evaluated with the help of two DTS professionals (4), achieving a mean classification accuracy of 41.5%, which is still meaningfully below the state of the art (4.2.1). However, it is also found that the correct classification choice was ranked in the top-three most likely choices by the model in a further 15.3% of cases, suggesting that further tweaking of decision weights and optimized training techniques may lead to significant increases in classification performance (5.1).

2 Related Works

This section contains an overview of the relevant literature, foundational concepts, and state-of-the-art techniques that inform and contextualize the research presented in this paper[1].

2.1 Sign Language

2.1.1 What is sign language? Sign language is a complex and diverse form of communication primarily used by deaf- and hard-of-hearing individuals. It encompasses a combination of hand gestures, facial expressions, and body movements to convey meaning [77]. Sign languages are not universal and have evolved independently in various regions, resulting in distinct linguistic structures and vocabularies such as American Sign Language (ASL), British Sign

[1]This chapter is structured as a sequence of questions in an attempt to convey my line of reasoning while exploring these concepts.

Language (BSL), and approximately 7,000 others [1], which continue to evolve to this day[2]. Broadly speaking, sign language consists of three parts: Manual features (gestures made with the hands), non-manual features (facial expressions and body posture), and finger spelling (spelling out words from the vocal language) [56]. The study of these different aspects have contributed valuable insights into the linguistic, cognitive, and social aspects of human communication while emphasizing the importance of making information and communication accessible to all [40, 56]. However, a significant communication gap naturally exists between the signing community and the general population [32], which often leads to limited access to education, employment, and social opportunities for deaf and hard-of-hearing individuals [52]. However, there is yet to exist any broadly available alternative to sign language.

2.1.2 Why are sign language alternatives insufficient? Cochlear implants and similar technologies have demonstrated varying degrees of success in restoring some level of hearing for deaf individuals [11, 17, 57, 80], but not all deaf individuals are eligible candidates for these interventions [28]. Furthermore, experimental approaches such as stem cell therapy show promise in addressing hearing loss, but they are not yet widely available or fully understood [26, 39]. As a result, a significant population of deaf individuals still rely on sign language as their primary mode of communication, while the less populous languages are often neglected in these studies, and risk benefitting very little from these technological developments.

2.1.3 Why focus on DTS?. DTS is the primary sign language used by the deaf community in Denmark. Like other sign languages, DTS relies on a combination of hand gestures, facial expressions, and body movements to convey meaning, providing a rich and fully functional means of communication for its users [23], and shares similarities with spoken Danish, but does also diverge from it.

Despite its domestic presence, DTS remains relatively unknown, with only about 4,000 to 5,000 native speakers, which has led to it being largely overlooked in the international sign language discourse [4, 53]. However, some elements of DTS, such as the finger spelling alphabet signs, are largely based on international standards [53], suggesting that developing recognition of DTS alphabet characters could potentially not only contribute a broader understanding of DTS sign language linguistics but also benefit works focused on other sign languages. The DTS finger spelling alphabet is illustrated in Figure 1.

2.2 Sign Language Recognition

2.2.1 What is SLR and what data is needed to perform it? SLR has emerged as a growing area of research focused on developing systems that can interpret and translate sign languages without relying on medical intervention to facilitate communication between deaf and hearing individuals [9, 18]. To accomplish this goal, SLR research mainly relies on two approaches: vision-based methods and sensor-based methods [1, 9, 42]. Vision-based methods make use of computer vision and machine learning techniques, but can be negatively impacted by factors such as occlusion, lighting conditions, and complex backgrounds. On the other hand, sensor-based

[2]https://wfdeaf.org/

Figure 1: The DTS manual finger spelling alphabet. The letters "J", "Z", "Æ", "Ø", and "Å" are performed with a motion, illustrated with black arrows, while the rest are non-moving signs. Source: https://dansktegnsprog.dk/om-sproget/temaer/tema/haandalfabetet

methods involve the use of wearable devices to capture and analyze motion and orientation data, but may be inconvenient or uncomfortable for users and might not accurately capture all nuances of sign language gestures, including the context in which signs are produced. Furthermore, both approaches typically require laboratory-like conditions and special handling to use. Finally, there is a growing need for larger-scale distributed data collection to enable more generalizable SLR models [49].

2.2.2 What are better ways of obtaining gesture data? One promising development in sensor-based SLR is the use of embedded depth sensors, such as the Leap-Motion Controller (LMC), which offers a non-invasive, accurate, and efficient way to track hand- and finger movements in real-time [10, 76, 79]. This is a compelling alternative to traditional sensor-based methods as the LMC combines the advantages of both approaches; It enables accurate tracking of hand movements without the need for wearables (except for the headset itself), making it more convenient and natural for users [79], and it overcomes some of the limitations of vision-based methods, such as occlusion and lighting conditions [38]. Multiple studies have used the LMC successfully for SLR [1, 13, 66, 76], but the sensor ultimately still requires a wired connection to an external PC. A solution to this may be the Meta Quest 2, a mass-market VR headset from Facebook. This headset offers completely on-device hand tracking functionality, with no connections to external PCs required, but otherwise works similarly to the LMC [35]. Although the specific workings are unknown, The Meta Quest 2 is thought to make use of a mixed vision- and sensor-based approach. It has also been utilized broadly for gesture recognition tasks, and has been

shown to have high reliability [2], but is yet to achieve significant momentum in the SLR literature compared to the LMC.

2.2.3 How are gestures recognized using this data? Gesture recognition plays a significant role in sign language recognition, as it focuses on the interpretation of various hand gestures, facial expressions, and body movements that constitute the linguistic components of sign languages [18, 19, 35, 42]. Within gesture recognition, two primary gesture categories can be identified: static gestures and dynamic gestures. Static gestures refer to distinct hand shapes or positions at a single point in time, while dynamic gestures involve continuous motion or changes in hand configurations over time [64]. This framework has been used to interpret letter spelling alphabets such as ASL [76], similar to that of DTS illustrated on Figure 1.

The advent of deep learning and advancements in computer vision techniques have greatly improved the performance of gesture recognition systems, enabling a more accurate interpretation of both static- and dynamic gestures in sign languages [49]. One study [76] managed 90+% recognition accuracy of select dynamic ASL finger spelling alphabet signs using the LMC and a Hidden Markov Model (HMM), which is a deep neural network model, requiring a sizable labeled dataset for training. Another study [49] achieved 87.4% mean accuracy across 510 different sign language words also using an HMM.

One promising direction in gesture recognition research, which seeks to mitigate the need for large-scale datasets, is the exploration of one-shot learning — a technique that allows models to learn and recognize new gestures or signs with minimal training examples [44, 78]. This approach has the potential to overcome some limitations of traditional deep learning methods, which often require large amounts of labeled training data to achieve optimal performance [48]. Some one-shot learning implementations of SLR have been proposed [25, 33], but they are yet to present in a form-factor that can be easily deployed at the scale that the Meta Quest 2 enables. Even so, the data is not all; it must be further processed for optimal performance.

2.2.4 What specific features and techniques are used to process gesture data? The use of a technique called feature reduction can lead to a more compact representation of gesture data, which in turn means improved computational efficiency, reduced risk of overfitting, and better overall performance for a sign recognition model [1]. Feature extraction and reduction is particularly important when working with data from a Leap Motion or similar device for gesture recognition, as these devices generate high-dimensional data with potentially redundant features.

Using various sensors, [25] has used hand position data, from which hand velocity data could be derived, and [49] used a similar principle with tracking gloves. Using the LMC in particular, [10] and [76] has found palm trajectory and the distance between finger tips and hand palm to be effective features for gesture recognition. Finally, [5] has used combined bone trajectories across a time series for better generalizability in the time domain. Still, these models largely fail to encompass the rich syntactical context inherent in sign language.

2.3 Natural Language Processing

2.3.1 What is NLP?. Natural Language Processing (NLP) is a subfield of artificial intelligence and linguistics that focuses on the development of computational methods to understand, generate, and process human languages [36]. NLP techniques have been applied to a wide range of tasks, including machine translation, sentiment analysis, information extraction, and text summarization, with the goal of enhancing the understanding of linguistic structures [50].

2.3.2 What data can be used for NLP?. One of the inherent advantages of NLP is the abundance of data available on the internet, which can be leveraged for training and evaluating models. This vast amount of data includes web pages, news articles, social media posts, and user-generated content [60]. Several publicly accessible datasets have been harnessed for NLP pursuits. These include the Penn Treebank for syntactic parsing tasks [73], the Stanford Question Answering Dataset (SQuAD) for question answering endeavors [63], and the IMDb dataset for sentiment analysis applications [82], to name a few. Tatoeba, a multilingual repository of sentences and translations, has been particularly valuable in machine translation and cross-lingual Natural Language Understanding (NLU), due to its wealth of sentence pairs in multiple languages [74]. However, verifying the soundness of such datasets can pose challenges due to potential noise, biases, or inaccuracies. In response, methods like cross-validation are utilized, wherein data is partitioned into multiple subsets, and the model is trained and tested on varying combinations of these subsets, thereby yielding a more dependable assessment of the model's performance and generalization capabilities [41]. Even so, the data cannot be passed directly into a machine learning model, it must first be vectorized through data preprocessing.

2.3.3 How is this data preprocessed to work with an NLP model? Data preprocessing and vectorization are NLP steps that involve converting raw textual data into a structured format suitable for machine learning algorithms [36]. This process typically includes tokenization, which is the process of splitting the input text into smaller units, such as words or characters [15].

One-hot encoding is a widely used technique for representing categorical data, such as alphabetical characters, as fixed-length binary vectors with a single non-zero element corresponding to the category [14, 31, 83]. One-hot encoding enables the conversion of discrete tokens into numerical representations that can be used as inputs to machine learning models. Padding is another preprocessing step that involves adding extra elements, typically zeros, to ensure that all input sequences have the same length, which is necessary for training models with fixed-size input layers, such as recurrent neural networks [29, 71]. Masking is a related technique that allows models to ignore the padded elements during training and evaluation, ensuring that they do not contribute to the loss function or influence the model's predictions [29, 55]. If these preprocessing techniques are employed, one may be able to get away with a simpler model that requires less training with a smaller runtime overhead, such as an n-gram model.

2.3.4 Why use an n-gram model for NLP tasks? N-gram models have found extensive application in a variety of NLP tasks, including

but not limited to language modeling, machine translation, and text classification [27]. An n-gram is a contiguous sequence of 'n' items derived from a particular text, where the items are usually words or characters. Within the domain of sign language recognition, n-gram models can be harnessed for classification by learning the statistical patterns of sign sequences in a dataset.

The employment of n-gram models for sign language recognition has been investigated in numerous studies. For example, [69] incorporated a HMM-based approach to recognize sequences of signs, with n-grams utilized to model the temporal dependencies between signs. Similarly, [19] leveraged n-gram models in conjunction with support vector machines to classify sign language gestures based on the positions of select hand bones across a time series (their local spatiotemporal characteristics).

In the context of NLP tasks with limited data, n-gram models present a compelling solution due to their simplicity and effectiveness, meaning they are easy to train and cross-validate. When faced with small datasets, more complex machine learning models, such as deep neural networks, may struggle to generalize and are prone to overfitting [29]. N-gram models, on the other hand, can efficiently capture local patterns and dependencies in the data, while requiring fewer parameters and less computational resources compared to deep learning models [27], making n-grams better candidates for resource-constrained devices such as the Meta Quest 2. On the other hand, these models may need external input to re-balance erroneous predictions, for example by using them in an ensemble context.

2.3.5 How can these models be used in an ensemble context? Multimodal learning represents a methodology within machine learning that integrates multiple forms of data like text, images, and audio, with the goal of boosting the performance of models [58]. Multimodal learning has been shown to increase the comprehension of linguistic interdependencies by using the results from different modalities simultaneously [3]. Conversely, ensemble learning is a strategy that aggregates several base models for superior predictive performance [24] while minimizing prediction errors [62]. Sign language recognition is a particularly relevant application for multimodal and ensemble learning techniques due to its inherent multimodal nature [9].

Recent research has also demonstrated the viability of these approaches in enhancing sign language recognition performance. For example, [42] combined deep neural networks with multimodal input features, including hand shape, motion, and body pose information, to achieve state-of-the-art results in SLR. Similarly, [34] proposed an ensemble learning approach that combined multiple convolutional neural networks, each specialized in recognizing specific aspects of sign language, such as hand shape or motion patterns. By leveraging the advantages of multimodal and ensemble learning techniques, these studies highlight the potential for significant improvements in sign language recognition performance.

This process is also known as fusion, and common methods include decision-level fusion, such as majority voting, where the final class label is determined based on the majority of votes from individual classifiers [43], and weighted voting, where classifiers are assigned weights based on their performance, giving more importance to more accurate models [62]. The downside to this approach is that it can be complex to implement and must be thoroughly evaluated, and the user must interact with the system in a highly regular manner to ensure correctness. A way to test this is to evaluate the usability of the system.

2.4 Virtual Reality Usability

2.4.1 What is usability evaluation? Usability evaluation is a critical aspect of VR application development, as it ensures that users can interact with these immersive environments effectively, efficiently, and satisfactorily [8, 12]. As VR applications become increasingly prevalent in various domains, such as entertainment, education, and training, researchers and practitioners alike have recognized the importance of assessing and optimizing their usability [7, 54].

2.4.2 What metrics and standadized procedures exist for VR usability evaluation? Traditional usability evaluation methods, such as heuristic evaluation, think-aloud protocols, and user testing, can be adapted for VR contexts [70]. Moreover, specific VR usability evaluation methods, such as the VRUSE questionnaire, have been developed to address challenges of VR experience evaluations [37]. Metrics for evaluating VR usability typically encompass performance (task completion time, error rates), user experience (user satisfaction, presence, immersion, comfort), and cognitive aspects (mental workload, memory demands, learnability) [21, 67].

2.4.3 How can designers optimize these usability metrics? Key usability heuristics specific to VR have been identified [16] and include Learnability, Cognitive Workload, Adaptability, and Ergonomics. Learnability refers to how easily users can understand and interact with the VR environment and its controls [8, 12]. Designers should ensure that the VR application provides clear guidance, tutorials, and feedback to facilitate a smooth learning curve [7]. Cognitive Workload, on the other hand, addresses the mental demands imposed on users during their interactions with the VR application. Designers should strive to minimize cognitive workload by providing intuitive interfaces, reducing visual and auditory clutter, and minimizing memory demands [70, 72].

Adaptability emphasizes the need for VR applications to accommodate users' varying preferences, skill levels, and physical abilities [61]. This can be achieved through multiple interaction techniques and accessibility options. Ergonomics involves designing VR applications that consider the user's physical comfort, safety, and well-being [46]. Ergonomic design can reduce user fatigue, discomfort, and the risk of motion sickness by considering factors such as controller design, movement mechanics, and user posture [65, 68], and to further control for user discomfort it is recommended that exposure to VR be limited to 30 minutes [68].

2.4.4 How can designers ensure that they live up to these recommendations? To best accommodate these recommendations it is crucial to test individual components in isolation at an early state. In the context of VR, Wizard-of-Oz prototyping and Semi-Structured Qualitative Studies (SSQS) have been employed to refine user interactions and gain insights into user experiences within virtual environments. Wizard-of-Oz prototyping allows developers to simulate user interactions and evaluate different input methods by utilizing a human operator to control the system's responses, thereby

enabling rapid iteration of the user experience [22, 59]. Furthermore, Wizard-of-Oz prototyping has been explored in the context of VR for gesture recognition [47, 81], with some studies [6, 75] even using the LMC. SSQSs, on the other hand, provide a flexible approach to gather user feedback on said experiences, preferences, and challenges when interacting with VR applications [20], often providing answers to the posed question while leaving room for unsuspected insights.

Additionally, quantitative technical assessments play a central role in ensuring a high-quality VR experience. One simple metric is Frames Per Second (FPS), which measures the display refresh rate in real-time applications [51]. High FPS values (ideally 90 FPS or above) are vital for creating immersive VR experiences, as low FPS can result in motion sickness, discomfort, and a reduced sense of presence [45, 65].

3 VR Prototype Development

This section details the design and development of an ensemble VR SLR prototype which can be used to answer the two-part research question presented in Section 1. This work builds on the related works presented in Section 2, with Section 3.1 detailing the VR UI, including a preliminary usability evaluation, and Section 3.2 detailing the implementation of the proposed SLR model. The VR prototype was implemented in Unity[3] using the C# programming language[4] and the Oculus Integration SDK[5].

3.1 VR User Interface Design

The overall goal of the UI was to make it clear to the user what sign to perform as well as when to perform it. Little effort is expended in the literature on the UI designs of SLR systems, but on small sample sizes with limited time such considerations may improve the quality of the gathered data. Thus, this work builds on the four usability metrics highlighted in Section 2.4.3 (Learnability, Cognitive Workload, Adaptability, and Ergonomics) in an effort to document their applicability in the domain of SLR.

To ensure learnability and minimal cognitive workload, the prototype was designed to support only two user interactions; raising both hands in the beginning to start the application, and raising their preferred hand to indicate to the recognition system which hand to run recognition on. This was named the "HandGate" system and meant that the rest of the UI operated on a timer system where transitions between different states would happen automatically in almost all cases without input (See Figure 2). The prototype was also designed with minimal visual clutter, and the UI display in-app contained only a handful of elements at a time. Finally, the application contained a video instructor, showing how each sign is to be performed above the UI in the virtual scene. An example of the in-app UI is displayed on Figure 3, and the initial design sketch is available in Appendix E.

To facilitate adaptability and good ergonomics, the system was designed to support input from both hands, in addition to supporting both standing and seated modes of operation seamlessly. The application was also designed with no movement controls and without the need for users to look in different directions, minimizing

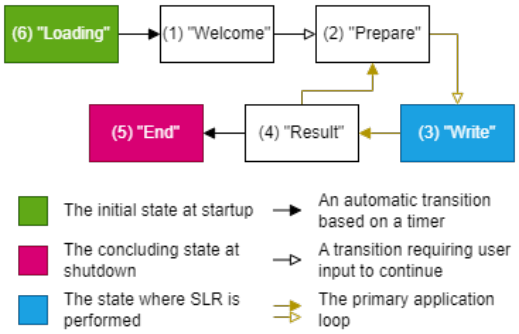

Figure 2: A flow-chart of the different states that make up the SLR scenario in the prototype application. The application starts in the "Loading" state, before moving to the "Welcome" state. After the user has signalled that they wish to continue, they enter the primary input-response loop where a letter input is provided by the system in the "Prepare" state, after which the user attempts to replicate that sign in the "Write" state, and finally is given feedback on that input in the "Result" state. When there are no more letters, the application exits in the "End" state.

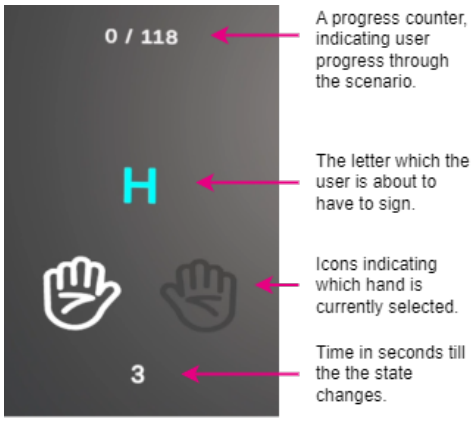

Figure 3: A screenshot from the "Prepare" (2) state screen of the UI in the application. This state is intended to allow the user to prepare for the next letter to sign, in this case an "H". Furthermore, the system communicates to the user that the left hand is selected (with the two hand icons) and that the hand must be raised for three more seconds to pass the "HandGate".

the chance of fatigue. Finally, the UI text was designed to be large enough to be readable for people with poor eyesight and the entire experience was designed to fit into a 30 minute exposure window[6].

3.1.1 Usability Evalution. A preliminary Usability Evaluation was performed during the development of the UI and provided significant insights ahead of the final evaluation (2.4.1). A Wizard-of-Oz approach (2.4.4) was used to examine the UI independently from the recognition system. Four participants were convenience-sampled *pro bono*, and were asked to "think-aloud" while engaging with the

[3] https://unity.com/
[4] https://dotnet.microsoft.com/en-us/
[5] https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022

[6] See Appendix A.1.1 for how this was achieved.

application after having answered a pre-experience questionnaire to assess their level of familiarity with VR and sign language. Quantitative data from a post-experience SSQS revealed median sign input times, informing the number of letters and sentences used in the final evaluation. Moreover, runtime performance was assessed, showing near-optimal FPS. Qualitative findings highlighted issues with gesture timing, hand icon interpretation, perceived testing pressure, and hand positioning. To address these, a hands-on tutorial was introduced before the final evaluation, intended to further improve system comprehension and user experience. These refinements were implemented in the final evaluation described in 4. Further details on the usability evaluation is available in Appendix A.1

3.2 SLR Model Implementation

The proposed ensemble SLR model consists of two-parts; an NLP n-gram model and a gesture recognition model. These two models are then confluenced using a weighted-voting fusion approach.

3.2.1 Language Model. The proposed Language Model (LM) is an n-gram model (2.3.4) and was developed in Python using PyTorch[7] and trained on a copy of the Tatoeba dataset for Danish language sentences[8] (2.3.2). The model was then imported into Unity by using the ONNX file format[9] and was executed using the Unity Barracuda package[10]. The following sentences were excluded, leaving $\sim 17,000$ sentences:

(1) Sentences that were <12 letters- or >36 letters long.
(2) Sentences that contained special characters aside from ",", ".", "?", or "!".
(3) Sentences that contained arabic numerals ("1", "2", "3", etc.).
(4) Sentences that were duplicates of other sentences.

Each letter was then tokenized (2.3.3) to an unsigned eight-bit integer representing the index of that letter in the alphabet. An extra value was reserved for denominating the whitespace character, and a value of 0 was appended to each sentence to signify the start of the next sentence. This format resulted in a compression of $\sim 20\%$ of the data. Finally, the sentence order was randomized and split into a training set (80%) and validation set (20%).

The n-gram model was defined as a simple linear classifier that uses a "sliding window" to analyze a given sentence and thereby guess the next letter in the sequence. The training set was one-hot encoded (2.3.3) into feature vectors with the same number of dimensions as letters in the alphabet. The LM was then trained on different permutations of the 80/20 data segmentation, as well as different n-gram (window) sizes. The results of these training sessions are graphed on Figure 4.

Although the user is never asked to input a whitespace character[11], the LM was still trained with this token included in the data to improve its predictions. This was done by designing the LM to exclude training cases where a whitespace character was the target of the training using Masking (2.3.3), and by implementing a custom loss function that ignored whitespace character weights by setting

[7]https://pytorch.org/
[8]The dataset was retrieved the 22nd of February 2023, totalling 56,076 sentences.
[9]https://onnx.ai/
[10]https://docs.unity3d.com/Packages/com.unity.barracuda@1.0/manual/index.html
[11]There is no DTS finger spelling alphabet sign that signifies a whitespace.

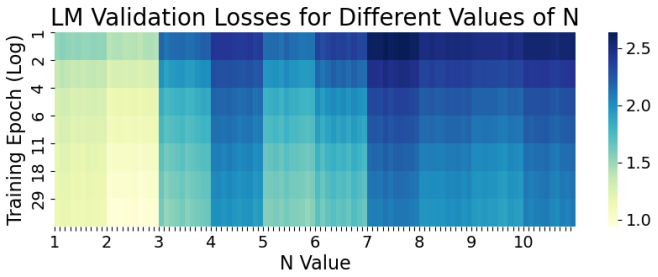

Figure 4: A heatmap of LM validation loss for different values of N, each tested on ten different permutations of the training data (x-axis), against the number of epochs trained (y-axis). Note that the y-axis is scaled logarithmically as the gains from successive epochs are quick to diminish.

the associated logit to -infinity during gradient descent. The LM was trained using Cross Entropy Loss and the SGD optimizer using a learning rate of 0.1 and a batch size of 32 for 30 epochs. As shown on Figure 4, the LM showed no significant deviation during cross-validation, and thus a permutation trained on $N = 2$ performed best and was selected with a validation accuracy of $\sim 54\%$.

3.2.2 Gesture Model. The proposed Gesture Model (GM) works by comparing the euclidean distances between vector representations of the users hands and reference targets. The Meta Quest 2 was used for data gathering as it was deemed the most advantaged platform (2.2.2). These reference targets are one-shot encoded (2.2.3) vector representations performed by the author, and were encoded with specific features (2.2.4) to help the GM generalize between different signers and across a time series. These features were deduced for each of the 24 individually tracked bones supplied by the Meta Quest 2.

In short, these feature vectors encoded the distance between their starting point at the start of the inference window, and the mean position of the bones at the end of the window. Furthermore, the length of this vector was scaled according to how much distance was covered by each bone during the inference window. In this way, the model is given the ability to discern between bones which are moved in a straight line and bones which are moved in a curve. This feature is intended to encode more information than simply velocity (2.2.4), and thus aid in the classification of dynamic gestures specifically (2.2.3) while reducing the feature dimensionality to a single vector for the entire time series.

3.2.3 Fusion Method. The model implementation uses a novel, although simple, fusion approach. The goal of this approach is to dynamically weight the LM and GM depending on their classification confidence using weighted-voting (2.3.5). This is done by calculating the difference between the first- and second choice of the GM and, depending on how small that difference is, attribute more weight to the LM. The thinking here is that if the difference between the two top predictions from the GM are very large then the model is very confident in its choice and the LM should thus not be considered in the prediction, whereas if the two values are very close then the LM is given the final say. In the fusion implementation, a constant of $c = 12$ was defined through trial and error, and the factor by which to weight the language results k are then

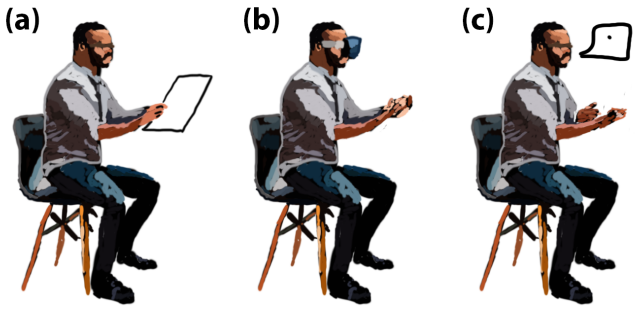

Figure 5: A storyboard realized as a hybrid sketch of the evaluation procedure. (a) shows the participant reading and filling out the questionnaire (4.1.1), (b) shows the participant engaging with the VR prototype (4.1.2), and (c) shows the participant during the post-evaluation interview (4.1.3).

given by $k = g_1/g_2 * c$ where g represents a sorted list of rankings from the GM in ascending order[12]. The highest rated c elements in the language result are then weighted by multiplying them with k before the results (votes) of both models are summed and ranked from highest to lowest, with the highest value representing the final model prediction.

4 Prototype Evaluation

This section describes the data collection procedures and analytical methods employed to evaluate the prototype proposed in Section 3. The evaluation procedure is introduced in Section 4.1, while the results are presented in Section 4.2

4.1 Procedure

The goal of the final evaluation was to document the performance of the system on a sample of the DTS-literate population. To achieve this, two DTS professionals were recruited *pro bono*, one male and one female aged 62 and 60, with both being employed as sign language interpreters in the medical sector. Both participants were right-handed, used glasses, and rated their VR experience-level 1/5. For the purpose of the evaluation, the input provided by these individuals to the system were considered ground truths, and the recognition performance of the proposed system would then be given by the degree to which the system is able to detect the provided sign gestures. The evaluation procedure consisted of three overall parts; introduction, scenario, and interview. A storyboard of the procedure is presented on Figure 5.

4.1.1 Pre-Evaluation Introduction. This initial part of the experiment was intended to inform the participant about the goals of the experiment and introduce them to the functionality of the VR prototype. First, after reading and signing a consent form, the participant was introduced to the study, where particular care was taken to inform the participant that it was the system that was being evaluated and not them. Next, the participant answered a pre-evaluation questionnaire, the same as for the preliminary evaluation described in 3.1.1, providing insights about their familiarity with VR technology. Finally, the participant was introduced to the workings of the prototype through a special debug build running on a Windows 11 PC, allowing the experiment conductor to see what the participant was seeing in the headset, thus allowing them to answer any questions and provide greater context if needed. Here, similarly to the preliminary evaluation, the participant would be encouraged to "think-aloud" in the hopes that any questions regarding the prototype could be clarified ahead of the next step in the procedure.

4.1.2 Evaluation Scenario. In this main part of the evaluation the participant would go through the pre-planned prototype scenario. Here, the participant would use the prototype build running natively in the Meta Quest 2, and the effort was to limit interactions between the conductor and the participant during this phase. The participant would be prompted by the system to spell out eight Danish sentences, one letter at a time, using the DTS manual finger spelling alphabet as described in 2.1.3. The number of sentences was determined based on the average time taken to perform one input from the usability evaluation (3.1.1), extrapolated into a 30 minute upper-bound window to avoid participant discomfort from prolonged VR exposure (2.4.3), and the sentences were sampled from the validation set of sentences as described in Appendix A.2. This means that rarer characters such as "C", "Q", "W", and "X" were not tested. At the point of sampling, the sentences were already in random order, and the order was kept constant between participants[13].
While the participant engaged with the scenario, the system would record telemetry about the performance of the prototype, such as FPS and the results of both the language- and gesture models individually, as well as the post-fusion results, as defined in Section 3.2.3. The system would also log task completion time and error rate (2.4.2). In this case, the completion time is considered the time it takes the user to engage the inference window, more specifically the time spent in the "prepare" step, as documented in Section 3.1, while error rate was defined as the number of failed inputs over total inputs.

4.1.3 Post-Evaluation Interview. In this final step of the evaluation, immediately after the participant had gone through the inference scenario, they were asked to fill out a modified VRUSE questionnaire (2.4.2). Specifically, only select questions pertaining to the usability heuristics found particularly applicable to VR were presented to the participants, totalling 19 questions[14] — each translated from English to Danish and rated on a Likert Scale in the range 1 to 5, with 1 being "strongly disagree" and 5 being "strongly agree". Finally, the evaluation was concluded with a post-experiment Semi-Structured Interview. This interview took the form of a loose discussion between the participant and the conductor, where key remarks brought up by the participant during the evaluation could be discussed further, in addition to anything noted down by the conductor during any preceding steps in the procedure. The results of the evaluation are presented in Section 4.2 and discussed in Section 5.

[12]Bare in mind that a lower rank is better in the case of the GM because the rankings are defined in terms of euclidean distances from target vectors.

[13]The sentences used are available in Appendix B.

[14]The selected questions are available in Appendix C, along with a legend of categories.

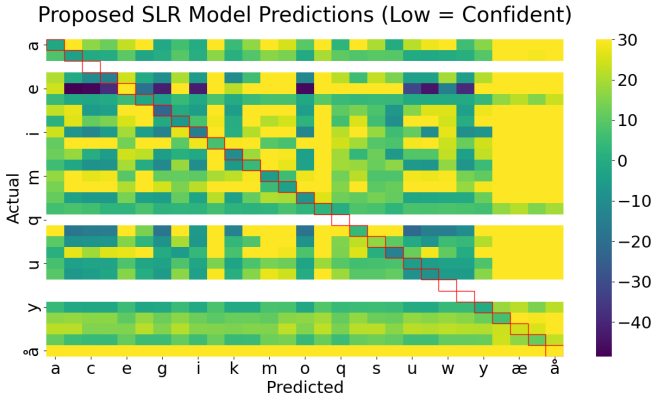

Figure 6: A confusion matrix of the classifications made by the proposed SLR model. The matrix plots the predicted alphabet letter (x-axis) onto the ground truth target letter from the input sentences (y-axis), totalling 236 samples.

4.2 Results

This section presents the findings from the prototype evaluation outlined in 4 which can be used to answer the research question presented in 1. Section 4.2.1 provides quantitative data for assessing the performance of the proposed SLR model, while Sections 4.2.2 and 4.2.3 provide feedback on the VR UI.

4.2.1 Classification Performance. The classification performance of the proposed SLR model was analyzed using application telemetry data recorded using a custom logging system, including a custom logging format and data converter for the Pandas[15] library for Python[16]. The results of this analysis are graphed on Figure 6. This confusion matrix shows the predicted letter for every sign input against the target letter, and one would ideally see a pronounced "staircase" going down and to the left where the indices of both axis converge on the same letter (red outlines). Although that is visible, the plot also reveals significant amounts of noisy (wrong) predictions. The application had a mean accuracy of ~ 41.5% across the two participants, totalling 236 predictions. The data logs also show that the correct prediction was ranked in the top three most likely signs in a further 15.3% of cases.

Furthermore, the data indicates that dynamic gestures ("J, "Z", "Æ", "Ø", and "Å" signs) had slightly higher classification accuracy (44.4%) versus static gestures (41.3%), although the sample size was very small, with dynamic gestures only making up 8% of the data (18 samples). Specifically the letters "E" and "R" had really low accuracy, having a large influence in the final accuracy metric as they were the most common letters in the sentence data (23.6% of letters, see Appendix A.2.1). Finally, the data shows that the LM influenced the GM 30 times (12.7% of predictions), of which 12 of those times lead to the correct classification (5% of total predictions, 40% LM influence accuracy). The classification accuracy for each letter is shown on Figure 7.

[15]https://pandas.pydata.org/
[16]https://www.python.org/

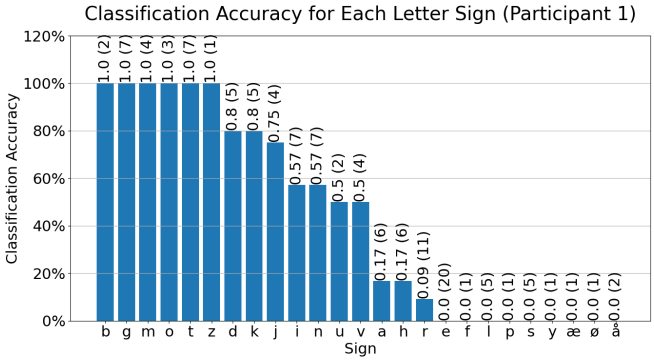

Figure 7: A bar chart of classification performance for each letter sign, sorted from best to worst. Note that the letters not included in the sentence data (4.1.2) were omitted. Each bar is annotated with the exact accuracy as well as the number of occurrences of this letter in the sentence data (in parentheses).

4.2.2 Questionnaire Results. The results of the VRUSE questionnaires are available on Table 1. The results reveal that the participants scored Appropriateness and Ease of Use favorably, while Learnability and Intuitiveness was rated more poorly. Highest rated was the "Comfort" category, which encompassed three questions regarding eye-strain, sickness, and disorientation. The application FPS data was also reviewed and found to be in line with the findings in the usability evaluation (See Appendix A.1.1).

Questionnaire Results		
Category	**P1 (mean)**	**P2 (mean)**
Appropriateness (2)	4.5	3.0
Ease of Use (7)	3.8	3.5
Learnability (3)	4.0	2.5
Intuitiveness (3)	2.7	2.0
Comfort* (3)	4.7	5.0
Functionality (Overall)	5.0	3.0
Usability (Overall)	4.0	3.0

Table 1: A table of the results from the questionnaire. Answers were converted into a favorability-scale and then averaged. (*) This category was not explicitly defined in VRUSE (2.4.2). The category numbers indicate the number of questions asked pertaining to that category.

4.2.3 Interview Results. The interview made it clear that both participants harboured similar sentiments about certain parts of the VR prototype, while their experiences also deviated substantially. Most notably, participant 2 reported that the signs for "R", "D", "S", and "T", demonstrated in the app, were not the signs they usually used. This participant later remarked, during a review of both old- and new DTS alphabets that they most likely were using a mixture of the two normally.

both participants reported having understood the UI early on, within the first 20 signs, which stands in contrast to some of the answers given in the questionnaire (4.2.2) They also very quickly both

reported strain in the arm and hand used for signing. Participant 1, although they were seemingly able to keep up with the speed of state changes in the app, reported it went too fast to be able to discern all the UI text, specifically the sentence being written by them, while participant 2 remarked that they wished the app would speed up. Participant 2 also remarked that the Meta Quest 2 headset was uncomfortably heavy, while participant 1 reported it was comfortable to wear.

5 Discussion

This section seeks to answer the research questions presented in Section 1 using the results presented in Section 4.2 — the performance of the proposed VR SLR system is discussed (5.1), as well as the results regarding the UI (5.2). To recap, this is the two-part research question which it is the goal of this work to answer:

(1) *Can a VR SLR model be developed requiring minimal training data while maintaining comparable performance to the state of the art?*
(2) *Can a set of guidelines for how to design VR UIs in the context of SLR be synthesized?*

5.1 Proposed SLR model viability

The proposed SLR model performed substantially below the state of the art of other models (2.2.3). Although the model is very computationally efficient and light-weight compared to the state of the art, both in terms of run- and training time, this level of performance is still too poor to be useful. Even if further tweaking could result in higher confidence, as suggested with the 15.3% near-correct predictions, the model would still not be competitive. Although this is most certainly primarily attributable to the design of the model, it does also present issues in terms of how NLP-augmented models can be compared with others; as seen from the data, low accuracy of a couple common letters, namely "E" and "R" alone decreased the mean accuracy of the model by $\sim$ 23% of the total due to the need for realistic testing sentences, something which non-NLP models do not need to account for. This same need for data of a certain distribution meant that not all classes could be validated, and four letters were omitted from testing. A greater number of sample sentences could eliminate this, but it would be at the cost of either increased VR exposure, above the recommended amount, or alternatively would require a drastically larger number of participants, testing different sentence pairs. Thus, how exactly validation data should be selected to invite greater comparability between models with different modalities has potential for further investigation.

Furthermore, it was observed that the LM model in isolation performed worse (40% accuracy) than during training ($\sim$ 54%). However, measuring LM performance as a function of observed influence over the GM model is inherently flawed, as each input to the GM results in different classification probabilities, and thus different values when the LM influence is applied. Actually, this may suggest that the LM was not given enough influence over the final predictions. Derivative works could seek to document the proposed fusion approach in greater detail to illuminate this further. It was also found that the model was able to generalize slightly better across dynamic gestures (44.4%) than static ones (41.3%). However, again considering the sample size, especially for dynamic gestures, this is within the margin of error. Further work is needed to document the proposed vector encodings and their ability to generalize over dynamic gestures in the time domain.

Overall, the proposed SLR model is not performant enough during classification to be favorable over incumbent techniques such as HMMs (2.3.4). The proposed model is also highly reliant on every gesture input being performed with equal durations, which cannot be enforced during natural sign language speech. However, the proposed fusion approach was capable of increasing classification performance by 5%, even with the clear bias towards under-weighing the LM predictions. Because of this, it would still be interesting to see how the same fusion approach would perform given a more sophisticated LM and GM. Such an LM could be using word embeddings with a more sophisticated neural network like a Long-Short Term Network, and the GM could be upgraded to a HMM approach in turn, as has already been extensively documented in the literature (2.3.4). The ensemble approach still shows tremendous potential, especially if the LM-part can transition to encode for a syntactical understanding of sign language itself, instead of using the written language as a proxy for this, since that approach will most likely not scale beyond the alphabet signs. This could potentially be achieved using a single model of sufficient complexity, but the amount of training data needed to facilitate this is still the major hindrance. Because of this, work on cataloguing and developing datasets for sign language signs still has tremendous potential.

5.2 Guidelines for SLR application UIs in VR

Indications of the proper implementation of the highlighted usability heuristic (2.4.3) was observed, and the use of the VRUSE questionnaire (2.4.2) proved valuable to the study, even though the value of the questionnaire in itself was diminished. Firstly, no meaningful statistical analysis, beyond deriving mean scores, could be carried out due to the limited sample size. This also meant the data from the questionnaires could not be statistically correlated with any observations or application telemetry. This in turn means the questionnaire results currently only serve as an indication of user sentiment concerning the different usability heuristics (2.4.3). Subtracting further from the validity of this approach is that only a subset of the questions were used, in the interest of time. The original VRUSE methodology describes 100 questions in distinct categories, where the user assess different aspects of the application in a specific order, which was not replicated here. The VRUSE questions were also translated from English to Danish, which means the original meaning of the questions may have been lost to some degree. However, specifically the category concerning Comfort was answered very favorably, suggesting that the general VR principles for reducing fatigue and discomfort (2.4.4) are directly applicable to applications in the field of SLR. As for the other categories, only Intuitiveness was rated below 2.5/5 on average, suggesting that the application, despite its flaws, was still relatively well received. Of cause in the case of Intuitiveness, it is important to note that a poor score should not be taken to mean that this principle does not apply to this domain, but instead that the principles were not implemented correctly in the prototype. Regrettably, no attempts were made to assess Cognitive Workload, despite the users at times clearly exhibiting signs of cognitive stress when interacting with

the application. Future works should make an effort about designing the study more carefully to allow for cognitive assessments.

What proved to be the primary utility of the questionnaire questions were as conversation starters for the post-scenario interviews. Firstly, the interviews echoed the mediocre level of satisfaction indicated by the questionnaire answers themselves. Although, specifically the ergonomics of the application will need revising, as both participants got tired in their arms and shoulders very quickly after beginning to use the application, with both having to take short breaks to finish the scenario. It is clear that the system must be able to support a more natural signing position of the arm/hand. This was predictable, but was avoided initially to ensure the signs were performed in a way that allowed optimal conditions for the Meta Quest 2, however it is clear now that no compromises can be made in this regard. Both participants confirmed that performing these signs in close proximity to the upper chest would be optimal and far more natural, and should thus be the benchmark for SLR using gesture recognition in VR going forward.

Another interesting result that emerged was that specifically one participant reported finding it more natural to take instructions from the video recording of each letter sign being performed, rather than having the letter displayed to them, while the other participant found the opposite to be the case. Although the videos and UI letters only exist as a means of elicitation and a comparison of the two was not the subject of study in this work, it was still unexpected that a preference between one or the other emerged. Further research into techniques for gesture elicitation in an SLR context may further illuminate this. Regarding the "HandGate", one participant specifically remarked that it was frustratingly slow. This suggests that the "HandGate" did not support sufficient levels of Adaptability (2.4.3) and thus needs further development, as increasing the speed with witch signs can be performed could in turn increase the sample size of future experiments. Interestingly, it was the same participant that remarked on the displayed gestures for letters "R", "D", "S", "T" deviating from the gestures they normally used. The fact that such variation in vocabulary exists on such a small sample size indicates that future studies should consider introducing controls for this during participant recruitment, especially when studying languages like DTS where multiple different finger spelling alphabets exists.

In total, further work is still needed to document how the highlighted usability heuristics (2.4.3) for VR applications in the SLR domain are to be implemented optimally, and potential future SLR prototypes will remain largely incomparable until a standard emerges here.

6 Conclusion

Unfortunately, neither research question could be definitively answered on the basis of the presented results. The LM and GM were not sophisticated enough to achieve comparable performance to the state of the art, but if new iterations of those models in isolation could be developed, the fusion model could probably be carried over, thus not refuting the use of an ensemble approach in this context. Secondly, further guidance is needed for how to compare NLP- and non-NLP augmented SLR models.

In respect to the second part of the research question, this work failed to draw definitive conclusions about the utility of the different measures taken to accommodate the various usability heuristics presented. This could potentially be improved with a full implementation of VRUSE, a means of assessing Cognitive Workload, as well as techniques for increasing participant throughput. Finally, the application became strenuous to use very quickly; the sensors used for SLR must be able to support the user performing the sign gesture in a natural position to them, and must be able to support combinations of different vocabularies interchangeably if they exist.

Still, it is the hope that future works that seek input on how to design usability-considerate SLR systems, as well as works that seek inspiration on how gesture models can be augmented using NLP for enhanced classification performance, can find use of the findings presented in this work.

References

[1] IA Adeyanju, OO Bello, and MA Adegboye. 2021. Machine learning methods for sign language recognition: A critical review and analysis. *Intelligent Systems with Applications* 12 (2021), 200056.

[2] Naveed Ahmed, Mohammed Lataifeh, and Imran Junejo. 2021. Visual Pseudo Haptics for a Dynamic Squeeze/Grab Gesture in Immersive Virtual Reality. In *2021 IEEE 2nd International Conference on Human-Machine Systems (ICHMS)*. IEEE, 1–4.

[3] Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. 2018. Multimodal machine learning: A survey and taxonomy. *IEEE transactions on pattern analysis and machine intelligence* 41, 2 (2018), 423–443.

[4] Brita Bergman and Elisabeth Engberg-Pedersen. 2010. Transmission of sign languages in the Nordic countries. (2010).

[5] Jordan J Bird, Anikó Ekárt, and Diego R Faria. 2020. British sign language recognition via late fusion of computer vision and leap motion with transfer learning to american sign language. *Sensors* 20, 18 (2020), 5151.

[6] Andreea Dalia Blaga, Maite Frutos-Pascual, Chris Creed, and Ian Williams. 2021. Freehand grasping: An analysis of grasping for docking tasks in virtual reality. In *2021 IEEE Virtual Reality and 3D User Interfaces (VR)*. IEEE, 749–758.

[7] Doug A Bowman, Ernst Kruijff, Joseph J LaViola, and Ivan Poupyrev. 2001. An introduction to 3-D user interface design. *Presence* 10, 1 (2001), 96–108.

[8] Jennifer Brade, Mario Lorenz, Marc Busch, Niels Hammer, Manfred Tscheligi, and Philipp Klimant. 2017. Being there again–Presence in real and virtual environments and its relation to usability and user experience using a mobile navigation task. *International Journal of Human-Computer Studies* 101 (2017), 76–87.

[9] Danielle Bragg, Oscar Koller, Mary Bellard, Larwan Berke, Patrick Boudreault, Annelies Braffort, Naomi Caselli, Matt Huenerfauth, Hernisa Kacorri, Tessa Verhoef, et al. 2019. Sign language recognition, generation, and translation: An interdisciplinary perspective. In *Proceedings of the 21st International ACM SIGACCESS Conference on Computers and Accessibility*. 16–31.

[10] Ariel Caputo, Andrea Giachetti, Simone Soso, Deborah Pintani, Andrea D'Eusanio, Stefano Pini, Guido Borghi, Alessandro Simoni, Roberto Vezzani, Rita Cucchiara, et al. 2021. SHREC 2021: Skeleton-based hand gesture recognition in the wild. *Computers & Graphics* 99 (2021), 201–211.

[11] Peder U Carlsson and Bo EV Håkansson. 1997. The bone-anchored hearing aid: reference quantities and functional gain. *Ear and hearing* 18, 1 (1997), 34–41.

[12] Ananth N Ramaseri Chandra, Fatima El Jamiy, and Hassan Reza. 2019. A review on usability and performance evaluation in virtual reality systems. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 1107–1114.

[13] Hong Cheng, Lu Yang, and Zicheng Liu. 2015. Survey on 3D hand gesture recognition. *IEEE transactions on circuits and systems for video technology* 26, 9 (2015), 1659–1673.

[14] Francois Chollet. 2021. *Deep learning with Python*. Simon and Schuster.

[15] Sanghyun Choo and Wonjoon Kim. 2023. A study on the evaluation of tokenizer performance in natural language processing. *Applied Artificial Intelligence* 37, 1 (2023), 2175112.

[16] Ngip Khean Chuan, Ashok Sivaji, and Wan Fatimah Wan Ahmad. 2015. Usability heuristics for heuristic evaluation of gestural interaction in HCI. In *Design, User Experience, and Usability: Design Discourse: 4th International Conference, DUXU 2015, Held as Part of HCI International 2015, Los Angeles, CA, USA, August 2–7, 2015, Proceedings, Part I*. Springer, 138–148.

[17] Vittorio Colletti, Robert V Shannon, Marco Carner, Sheila Veronese, and Liliana Colletti. 2009. Progress in restoration of hearing with the auditory brainstem implant. *Progress in brain research* 175 (2009), 333–345.

[18] Helen Cooper and Richard Bowden. 2007. Sign language recognition using boosted volumetric features. In *Proceedings of the IAPR Conference on Machine Vision Applications*. 359–362.

[19] HM Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. 2012. Sign language recognition using sub-units. *Journal of Machine Learning Research* 13 (2012), 2205–2231.

[20] John W Creswell and Cheryl N Poth. 2016. *Qualitative inquiry and research design: Choosing among five approaches*. Sage publications.

[21] James J Cummings and Jeremy N Bailenson. 2016. How immersive is enough? A meta-analysis of the effect of immersive technology on user presence. *Media psychology* 19, 2 (2016), 272–309.

[22] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. 1993. Wizard of Oz studies: why and how. In *Proceedings of the 1st international conference on Intelligent user interfaces*. 193–200.

[23] Maartje De Meulder and Joseph J Murray. 2017. Buttering their bread on both sides? The recognition of sign languages and the aspirations of deaf communities. *Language Problems and Language Planning* 41, 2 (2017), 136–158.

[24] Thomas G Dietterich. 2000. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings 1*. Springer, 1–15.

[25] Philippe Dreuw and Hermann Ney. 2008. Visual modeling and feature adaptation in sign language recognition. In *ITG Conference on Voice Communication [8. ITG-Fachtagung]*. VDE, 1–4.

[26] Albert SB Edge and Zheng-Yi Chen. 2008. Hair cell regeneration. *Current opinion in neurobiology* 18, 4 (2008), 377–382.

[27] Jacob Eisenstein. 2019. *Introduction to natural language processing*. MIT press.

[28] René H Gifford, Michael F Dorman, Henryk Skarzynski, Artur Lorens, Marek Polak, Colin LW Driscoll, Peter Roland, and Craig A Buchman. 2013. Cochlear implantation with hearing preservation yields significant benefit for speech recognition in complex listening environments. *Ear and hearing* 34, 4 (2013), 413.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[30] Gintautas Grigas and Anita Juškevičienė. 2018. Letter frequency analysis of languages using latin alphabet. *International Linguistics Research* 1, 1 (2018), p18–p18.

[31] Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with NumPy. *Nature* 585, 7825 (2020), 357–362.

[32] Saeid Hassanzadeh. 2012. Outcomes of cochlear implantation in deaf children of deaf parents: comparative study. *The Journal of Laryngology & Otology* 126, 10 (2012), 989–994.

[33] Seyed Ramezan Hosseini, Alireza Taheri, Minoo Alemi, and Ali Meghdari. 2021. One-shot learning from demonstration approach toward a reciprocal sign language-based HRI. *International Journal of Social Robotics* (2021), 1–13.

[34] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. 2015. Sign language recognition using 3d convolutional neural networks. In *2015 IEEE international conference on multimedia and expo (ICME)*. IEEE, 1–6.

[35] Lin Jiang, Xiaoyang Yu, and Lijun Wang. 2020. A brief analysis of gesture recognition in VR. In *SID Symposium Digest of Technical Papers*, Vol. 51. Wiley Online Library, 190–195.

[36] Dan Jurafsky and James H Martin. 2019. Speech and language processing (3rd (draft) ed.).

[37] Roy S Kalawsky. 1999. VRUSE—a computerised diagnostic tool: for usability evaluation of virtual/synthetic environment systems. *Applied ergonomics* 30, 1 (1999), 11–25.

[38] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2013. Real time hand pose estimation using depth sensors. *Consumer Depth Cameras for Computer Vision: Research Topics and Applications* (2013), 119–137.

[39] Sonja Kleinlogel, Christian Vogl, Marcus Jeschke, Jakob Neef, and Tobias Moser. 2020. Emerging approaches for restoration of hearing and vision. *Physiological reviews* 100, 4 (2020), 1467–1525.

[40] Edward S Klima and Ursula Bellugi. 1979. *The signs of language*. Harvard University Press.

[41] Ron Kohavi et al. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, Vol. 14. Montreal, Canada, 1137–1145.

[42] Oscar Koller, Hermann Ney, and Richard Bowden. 2016. Deep hand: How to train a cnn on 1 million hand images when your data is continuous and weakly labelled. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3793–3802.

[43] Ludmila I Kuncheva. 2014. *Combining pattern classifiers: methods and algorithms*. John Wiley & Sons.

[44] Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science* 350, 6266 (2015), 1332–1338.

[45] Steven M LaValle, Anna Yershova, Max Katsev, and Michael Antonov. 2014. Head tracking for the Oculus Rift. In *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 187–194.

[46] Joseph J LaViola Jr. 2000. A discussion of cybersickness in virtual environments. *ACM Sigchi Bulletin* 32, 1 (2000), 47–56.

[47] Myungho Lee, Kangsoo Kim, Salam Daher, Andrew Raij, Ryan Schubert, Jeremy Bailenson, and Greg Welch. 2016. The wobbly table: Increased social presence via subtle incidental movement of a real-virtual table. In *2016 IEEE virtual reality (VR)*. IEEE, 11–17.

[48] Hugo Leon-Garza, Hani Hagras, Anasol Peña-Rios, Ozkan Bahceci, and Anthony Conway. 2022. A Hand-Gesture Recognition Based Interpretable Type-2 Fuzzy Rule-based System for Extended Reality. In *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2894–2899.

[49] Kehuang Li, Zhengyu Zhou, and Chin-Hui Lee. 2016. Sign transition modeling and a scalable solution to continuous sign language recognition for real-world applications. *ACM Transactions on Accessible Computing (TACCESS)* 8, 2 (2016), 1–23.

[50] Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028* (2002).

[51] Katerina Mania and Alan Chalmers. 2001. The effects of levels of immersion on memory and presence in virtual environments: A reality centered approach. *Cyberpsychology & behavior* 4, 2 (2001), 247–264.

[52] Marc Marschark, Patricia Sapere, Carol Convertino, and Rosemarie Seewagen. 2005. Access to postsecondary education through sign language interpreting. *Journal of Deaf Studies and deaf education* 10, 1 (2005), 38–50.

[53] William B McGregor, Janne Boye Niemelä, and Julie Bakken Jepsen. 2015. Danish sign language. *Sign languages of the world: a comparative handbook* (2015), 195–233.

[54] Ryan P McMahan, Doug A Bowman, David J Zielinski, and Rachael B Brady. 2012. Evaluating display fidelity and interaction fidelity in a virtual reality game. *IEEE transactions on visualization and computer graphics* 18, 4 (2012), 626–633.

[55] Stephen Merity, Nitish Shirish Keskar, and Richard Socher. 2017. Regularizing and optimizing LSTM language models. *arXiv preprint arXiv:1708.02182* (2017).

[56] Thomas B Moeslund, Adrian Hilton, Volker Krüger, and Leonid Sigal. 2011. *Visual analysis of humans*. Springer.

[57] Albert Mudry and Mara Mills. 2013. The early history of the cochlear implant: a retrospective. *JAMA Otolaryngology–Head & Neck Surgery* 139, 5 (2013), 446–453.

[58] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y Ng. 2011. Multimodal deep learning. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 689–696.

[59] Michael Nielsen, Moritz Störring, Thomas B Moeslund, and Erik Granum. 2004. A procedure for developing intuitive and ergonomic gesture interfaces for HCI. In *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, GW 2003, Genova, Italy, April 15-17, 2003, Selected Revised Papers 5*. Springer, 409–420.

[60] Sinno Jialin Pan and Qiang Yang. 2010. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering* 22, 10 (2010), 1345–1359.

[61] Randy Pausch, Jon Snoddy, Robert Taylor, Scott Watson, and Eric Haseltine. 1996. Disney's Aladdin: first steps toward storytelling in virtual reality. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. 193–203.

[62] Lior Rokach. 2010. Ensemble-based classifiers. *Artificial intelligence review* 33 (2010), 1–39.

[63] Marc-Antoine Rondeau and Timothy J Hazen. 2018. Systematic error analysis of the Stanford question answering dataset. In *Proceedings of the Workshop on Machine Reading for Question Answering*. 12–20.

[64] Ashok K Sahoo, Gouri Sankar Mishra, and Kiran Kumar Ravulakollu. 2014. Sign language recognition: State of the art. *ARPN Journal of Engineering and Applied Sciences* 9, 2 (2014), 116–134.

[65] Dimitrios Saredakis, Ancret Szpak, Brandon Birckhead, Hannah AD Keage, Albert Rizzo, and Tobias Loetscher. 2020. Factors associated with virtual reality sickness in head-mounted displays: a systematic review and meta-analysis. *Frontiers in human neuroscience* 14 (2020), 96.

[66] Jacob Schioppo, Zachary Meyer, Diego Fabiano, and Shaun Canavan. 2019. Sign language recognition: Learning american sign language in a virtual environment. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–6.

[67] Mel Slater, Martin Usoh, and Anthony Steed. 1995. Taking steps: the influence of a walking technique on presence in virtual reality. *ACM Transactions on Computer-Human Interaction (TOCHI)* 2, 3 (1995), 201–219.

[68] Kay M Stanney, Ronald R Mourant, and Robert S Kennedy. 1998. Human factors issues in virtual environments: A review of the literature. *Presence* 7, 4 (1998), 327–351.

[69] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time american sign language recognition using desk and wearable computer based video. *IEEE Transactions on pattern analysis and machine intelligence* 20, 12 (1998), 1371–1375.

[70] Alistair Sutcliffe and Brian Gault. 2004. Heuristic evaluation of virtual reality applications. *Interacting with computers* 16, 4 (2004), 831–849.

[71] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems* 27 (2014).

[72] John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science* 12, 2 (1988), 257–285.
[73] Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The Penn treebank: an overview. *Treebanks: Building and using parsed corpora* (2003), 5–22.
[74] Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS.. In *Lrec*, Vol. 2012. Citeseer, 2214–2218.
[75] Nhan Tran, Josh Rands, and Tom Williams. 2018. A hands-free virtual-reality teleoperation interface for wizard-of-oz control. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*.
[76] Aurelijus Vaitkevičius, Mantas Taroza, Tomas Blažauskas, Robertas Damaševičius, Rytis Maskeliūnas, and Marcin Woźniak. 2019. Recognition of American sign language gestures in a virtual reality using leap motion. *Applied Sciences* 9, 3 (2019), 445.
[77] Clayton Valli and Ceil Lucas. 2000. *Linguistics of American sign language: An introduction*. Gallaudet University Press.
[78] Tianyi Wang, Xun Qian, Fengming He, Xiyun Hu, Yuanzhi Cao, and Karthik Ramani. 2021. Gesturar: An authoring system for creating freehand interactive augmented reality applications. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. 552–567.
[79] Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the accuracy and robustness of the leap motion controller. *Sensors* 13, 5 (2013), 6380–6393.
[80] Blake S Wilson and Michael F Dorman. 2008. Cochlear implants: a remarkable past and a brilliant future. *Hearing research* 242, 1-2 (2008), 3–21.
[81] Huiyue Wu and Jianmin Wang. 2013. Design of bare-hand gestures for object manipulation in virtual reality. *JOURNAL OF INFORMATION &COMPUTATIONAL SCIENCE* 10, 13 (2013), 4157–4166.
[82] Alec Yenter and Abhishek Verma. 2017. Deep CNN-LSTM with combined kernels from multiple branches for IMDb review sentiment analysis. In *2017 IEEE 8th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON)*. IEEE, 540–546.
[83] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems* 28 (2015).

A Work Sheets

This appendix serves to document the work which contributed meaningfully to the findings of this paper, but which could not be included in the main document. In short, A.1 goes into greater detail about the usability evaluation (3.1.1), while A.2 details how sentences were chosen for the final prototype evaluation (4).

A.1 Usability Evaluation

The goals of the usability evaluation were twofold; first, it was intended to function as a pilot for the final evaluation (4), and secondly also served to assess the UI in isolation from the recognition system, so that any issues found here could be corrected (2.4.1). This was necessary both to ensure insufficiencies in the UI design did not influence the SLR performance data, as well as to gather data on how many samples could be expected to be gathered within the 30 minutes upper-bound exposure window advised in 2.4.3.

However, to evaluate the UI, a means of providing feedback to the user in correspondence to the input they provided was necessary. For this, a Wizard-of-Oz approach was utilized to great effect (2.4.4), meaning the system could be evaluated before a working SLR implementation was available, by allowing for the experiment conductor to administer feedback to the user through the UI in accordance with the observed gesture being performed by them. For this evaluation, four 4th semester students were convenience-sampled *pro bono* and tasked to go through the experiment procedure one-at-a-time. The participants were first introduced to the study and asked to sign a consent form with a small questionnaire[17]. The participants rated their experience-level with VR 4.3/5, and their experience-level with sign language 1.8/5. The participants were all males aged 21 to 24, and were all right-handed. Furthermore, two participants reported to be using glasses or contacts every day, and all participants rated their experience with sign language 3 or lower on a scale of 1 to 5. Finally, three participants rated their experience with VR a 5 out of 5, with one rating it at a 3.

They then engaged with the application scenario, where only the finger spelling letters "A" and "B" were available as input. The scenario consisted of 16 sign prompts in the order "AABBABABBBAABABA" with "-" denoting a sentence change. This meant that the participants' progress would reset half-way through the scenario to mimic a sentence change in the intended final scenario. This allowed every function of the system to be tested with only these two input options. This was important to ensure the Cognitive Workload associated with performing the correct sign was kept to a minimum, so that stress related to this did not influence the participants ability to discern the UI. During the scenario, the participants were encouraged to "think-aloud" (2.4), and the experience was periodically halted by the conductor when a comment made by a participant invited further discussions. In total, the participants spent an average of 5 minutes and 49 seconds in the application, and the application was halted 10 times, an average of 2.5 times or 51 seconds per participant (14.6% of time spent in-app), during which the participants kept on the headset.

[17] The consent form and pre-scenario questionnaire is available in Appendix D.

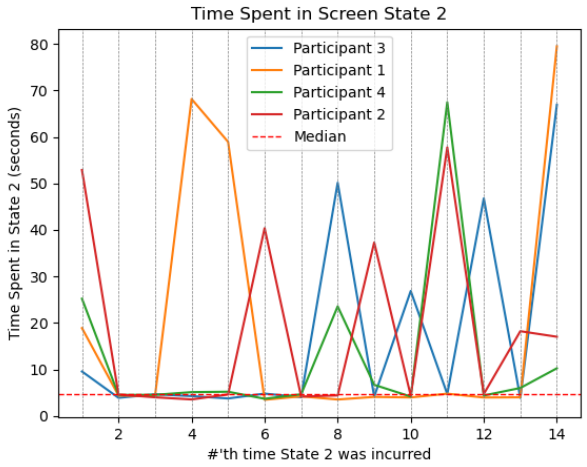

Figure 8: A plot of participants time spent in the "Prepare" (2) screen state (3.1). Marked with a dotted red line is the median used for deriving the number of sentences to use for the final evaluation. The outlier y-values represent the participants taking time to express a thought or answer a question from the conductor.

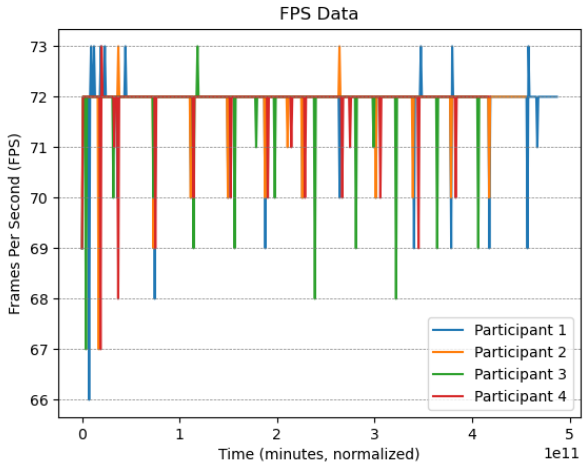

Figure 9: A plot of FPS samples from all participants. Only participant 1 experienced a dip to 66 FPS (0.0564% of total samples) but this drop was likely still too small to be noticeable.

A.1.1 Quantitative Results. This high-fidelity test of the system provided numerous insights which influenced how the final evaluation was to be conducted. Firstly, it was crucial to gain an understanding of the time taken by participants to perform one sign input so that the number of letters appropriate for the final evaluation could be estimated, ensuring that a broad representation of letters were tested within the upper-bound comfort threshold of 30 minutes (2.4.2). Since some screen states contained interactions, and were thus of variable duration, this value needed to be measured. The application telemetry revealed that the participants spent an average of 4 seconds in the "Prepare" (2) state (3.1) — the only repeating state with an interaction. Adding five seconds from the other repeating state durations (3) and (4) gives 9 seconds per sign input, which, if we allow for a three minute (10%) buffer, yields the number of letters possible within the evaluation time frame $num_L = 27/(9/60) = 180\ letters$. Finally, the number of sentences to use for the final evaluation is then given by $num_S = 180/24 = 7.5 \sim 8\, sentences$ since sentences can be between 12- and 36 letters long, as documented in 3.2.1. The measurements are plotted on Figure 8.

The usability evaluation also provided insights into the runtime performance of the application, more specifically the FPS, plotted on Figure 9. From the figure it is clear that the participants spent the vast majority of their time at the system-imposed limit of 72 FPS, with a total mean fps of 71.93 and a median of 72. Each participant experienced a dip below 72 FPS between 12 and 15 times across a total of 1772 samples. This means between 3.09% and 3.59% of samples were below the benchmark[18]. Finally, the times at which FPS dropped were not able to be correlated with any event in the app, meaning the most likely explanation is a lower-level hiccup in Unity or the Oculus Android runtime.

A.1.2 Qualitative Results. The usability evaluation also yielded a number of qualitative results in the form of the post-scenario interview. Here, issues or comments raised by the participant during their interaction with the prototype were further explored, and included things like the degree to which different ui elements were observed by the participant, as well as what their impression of their own performance was during the scenario. These discussions could be quite lengthy and helped identify the following issues:

(1) Some people would hold the gesture during the countdown, while others only per form the gesture when the blue border lights up as they are supposed to.
(2) The hand icons were confusing — some users interpreted them to depict the intended gesture, while others correctly read them, although it would often take them familiarizing themselves with the system first.
(3) The system was perceived as a test of the user by some participants, despite it being a test of the system. This may be due to the impression by one user that the progress had a "combo" mechanism even though it didn't.
(4) Multiple users had trouble finding the "sweet spot" for the hand raise gate in the beginning. Some users would raise their hands very high to activate it.
(5) Some users began performing the gesture before the inference window triggered, which would make them inconsistent with what the system expects.
(6) One user remarked interpreting the sequence and progress counter as a combo system. Other users may have perceived it similarly, but did not mention it.

(2) and (3) were fixed with minor UI changes and re-wording of interface texts. (1), (4), (5), and (6) were all solved by introducing a hands-on introduction before the experiment scenario where

[18]Here it is important to note that this does not translate to ≈ 3% of time spent in app to be below threshold, as every data point is sampled across a one second time frame.

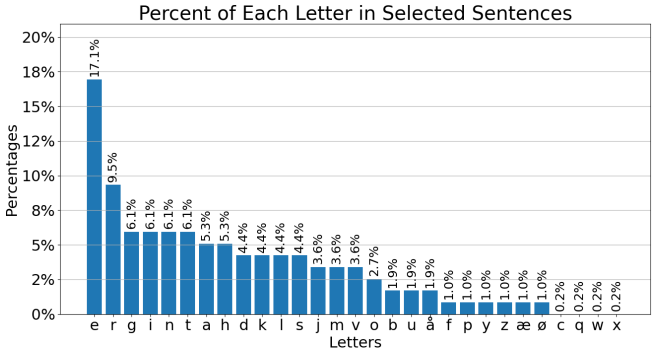

Figure 10: A bar chart of the distribution of letters in the selected sentence pairs.

the problematic elements of the UI were explained to the participant to avoid future confusion. All these changes and fixes were implemented ahead of the final evaluation, presented in 4.

A.2 Sentence Selection

Since the proposed SLR model uses an NLP model in its weights, we must ensure that we provide realistic input sentences to the model. A study [30] has determined the commonality of Danish alphabet letters using publicly available data from the internet. Thus, to ensure the proposed model is tested on data as resembling of a realistic scenario as possible, we must select the sentences from the dataset which best fulfill the same letter orders as was found in [30]. However, selecting the optimal combination of sentences proved surprisingly complex.

If we were to brute-force this problem by checking every possible combination of sentences, the size of the search space would be the combination of n sentences taken k at a time, where n is the total number of sentences (approximately 17,000) and k is the number of sentences we want to choose, which in this case is eight (4.1.2). This would result in a combinatorial explosion, making the search space incredibly large and the problem intractable using brute-force.

A.2.1 Genetic Selection. A genetic algorithm serves as an efficient tool for optimization problems, especially when dealing with extensive and intricate search spaces. This approach is inspired by natural evolution and involves a population of potential solutions, called "individuals" (in this case eight-way sentence pairs, sampled at random).

The fitness of each individual, indicating how closely the order of letter frequencies aligns with a predefined target ordering, is assessed. After evaluation, a selection process takes place based on a fitness score. Selected individuals then undergo the processes of crossover (recombination) and mutation (random replacement), forming a new "generation".

This cycle of selection, crossover, and mutation repeats over a fixed number of iterations (in this case 5000 generations). This implementation enables efficient exploration of the large search space, finding near-optimal solutions in a relatively short time span. However, doing this, it is still difficult to achieve total parity with the target ordering due to only sampling eight sentences, but the final result is still meaningfully more representative than a random sample of sentence pairs. The resulting letter distribution is shown on Figure 10.

B - Selected Sentences

1. Hvilken vej skal jeg gå?
2. Det har de ikke gjort.
3. Jeg har influenza.
4. Tom er elev.
5. Her er min tegnebog.
6. Vi sympatiserer med dig.
7. Hun er stærk.
8. Hans tå bløder.

C - Selected VRUSE Questions and Legend

VRUSE has been designed according to established statistical principles. A five-point Likert attitude scale was used as follows:

5	4	3	2	1
Strongly agree	Agree	Undecided	Disagree	Strongly disagree

1. [Appropriateness] The level of functionality (control) provided by the system was appropriate for the task.
2. [Ease of use] I found it easy to access all the functionality (control) of the system.
3. [Learnability] It was difficult to remember all the functions available.
4. [Learnability] I understood the meaning of the control interface.
5. [Intuitiveness] I kept making mistakes while interacting with the system.
6. [Ease of use] Visual feedback relating to the interface was inadequate.
7. [Ease of use] My eyes felt uncomfortable after using the system.
8. [System performance] I had difficulty getting used to the display.
9. [Simulator sickness] I felt nauseous when using the system.
10. [Appropriateness] When menus were displayed, I fully understood their meaning.
11. [Ease of use] I did not need any further help.
12. [Learnability] It was difficult to understand the operation of the interface.
13. [Ease of use] The user can tailor the system to suit their needs.
14. [Disorientation] I felt disorientated in the virtual environment.
15. [Ease of use] There was no means of ‘undoing’ an operation.
16. [Intuitiveness] The system did not work as expected.
17. [Intuitiveness] I can see a real benefit in this style of man–machine interface.

18. Overall, I would rate the VR system in terms of functionality as: very satisfactory, satisfactory, neutral, unsatisfactory, or very unsatisfactory.
19. Overall, I would rate the system usability as: very satisfactory, satisfactory, neutral, unsatisfactory, or very unsatisfactory.

Spørgeskema

1. Mit niveau af kontrol over systemet var tilstrækkeligt til den givne opgave.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

2. Jeg fandt det nemt at tilgå alle systemets funktioner.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

3. Det var svært at huske all de tilgængelige funktioner.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

4. Jeg forstod meningen med kontrolgrænsefladen.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

5. Jeg blev ved med at lave fejl mens jeg brugte systemet.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

6. Det visuelle feedback fra grænsefladen var utilstrækkeligt.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

7. Mine øjne føltes ubehagelige efter at have brugt systemet.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

8. Jeg havde svært ved at vænne mig til skærmen.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

9. Jeg fik kvalme af at bruge systemet.

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

10. Jeg forstod fuldstændig meningen med hvad grænsefladen viste.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

11. Jeg havde ikke brug for yderligere hjælp.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

12. Det var svært at forstå hvordan grænsefladen fungerede.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

13. Brugeren kan skræddersy systemet som det passer dem.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

14. Jeg følte mig desorienteret i de virtuelle omgivelser.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

15. Der var ikke nogen måde at "fortryde" en handling.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

16. Systemet fungerede ikke som forventet.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

17. Jeg kan se en virkelig fordel med denne type grænseflade.

    Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget enig

18. Overordnet vil jeg bedømme funktionaliteten som:

    Meget utilfredsstillende 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget tilfredsstillende

19. Overordnet vil jeg bedømme brugervenligheden som:

    Meget utilfredsstillende 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget tilfredsstillende

Deltager nummer _____ (udfyldes af dataansvarlig)

D – Consent Form and Pre-Evaluation Questionnaire

Samtykke-tekst

Dette er en anmodning om dit samtykke til at behandle dine personoplysninger. Formålet med behandlingen er evaluering af en Virtual Reality (VR) prototype i forbindelse med min kandidatafhandling.

Du giver samtykke til at behandle følgende oplysninger om dig: Navn, køn, alder, brug af briller eller kontaktlinser (ja/nej), foretrukne skrivehånd (venstre/højre), erfarenhed med VR (skala fra 1 til 5), erfarenhed med Dansk Tegnsprog (skala fra 1 til 5).

Jeg, Peter Guld Leth, er personligt dataansvarlig for dine oplysninger.

Dine oplysninger bliver opbevaret sikkert, og jeg benytter dem udelukkende til ovenstående formål.

Du har altid ret til at trække dit samtykke tilbage. Ønsker du senere at trække dit samtykke tilbage, kan du skrive til mig på pleth18@student.aau.dk.

Databeskyttelsesforordningen giver dig ret til at få en række oplysninger, som du finder i dette dokument.

☐ Jeg giver hermed samtykke til, at Peter Guld Leth må behandle mine oplysninger i henhold til ovenstående formål og oplysninger.

Dato:

Navn:

_____

Underskrift

Sådan behandler jeg dine data

Dataansvarlig

Peter Guld Leth
Egholmsgade 2, 3. th.
9000, Aalborg

Formålet med at behandle dine oplysninger

Jeg er ved at udvikle et tegnsprogsgenkendelsessystem i Virtual Reality (VR) som min kandidatafhandling. Systemet er en VR-app bestående af en simpel Unity-scene med en brugergrænseflade (UI) samt en 3D-repræsentation af brugerens hænder som spores af Oculus-sensorerne.

Mit mål er at evaluere systemet på tegnsprogsprofessionelle, enten konsulenter eller studerende. Før dette skal jeg dog få en idé om systemets evne til at kommunikere information til brugeren, herunder om systemet er i stand til at gøre brugeren opmærksom på, hvad de skal gøre på ethvert givent tidspunkt, og om de kan fortolke de oplysninger, der leveres af UI. Det er altså systemet, der evalueres, og ikke de deltagende forsøgspersoner!

Jeg behandler disse personoplysninger

☐ Almindelige personoplysninger (jf. art. 6, stk. 1, litra a)
(Fx navn, adresse, e-mail, alder, selvoffentliggjorte data mv.)

I form af navn, køn og alder.

☐ Følsomme personoplysninger (jf. art. 9, stk. 2, litra a)
(Fx helbredsoplysninger, race, politisk overbevisning mv.)

I form af brug af briller eller kontaktlinser (ja/nej), foretrukne skrivehånd (venstre/højre), erfarenhed med VR (skala fra 1 til 5), erfarenhed med Dansk Tegnsprog (skala fra 1 til 5).

Sådan opbevarer jeg dine oplysninger

Jeg opbevarer dine personoplysninger, så længe det er nødvendigt i forhold til formålet med at indhente dit samtykke og i henhold til gældende lovgivning. Herefter sletter jeg dine personoplysninger.

Dine rettigheder

Når jeg behandler dine personoplysninger, har du ifølge databeskyttelsesforordningen flere rettigheder. Det betyder bl.a., at du har ret til sletning og dataportabilitet.

I visse tilfælde har du ret til indsigt, berigtigelse, begrænsning og til at gøre indsigelse mod vores behandling af de omfattede personoplysninger.

Vær opmærksom på, at du ikke kan trække dit samtykke tilbage med tilbagevirkende kraft.

Vil du klage?

Mener du ikke, at jeg lever op til mit ansvar, eller jeg ikke behandler dine oplysninger efter reglerne, kan du klage til Datatilsynet på dt@datatilsynet.dk.

Jeg opfordrer dig dog til også at kontakte mig først, da jeg vil gøre, hvad jeg kan, for at imødekomme din klage.

Overdragelse til tredjepart

Dine data (eller dele af dine data) kan blive overdraget til AAU i forbindelse med bedømmelsen af mit speciale. Efter bedømmelsen er sket, og efter klagefristen er udløbet, vil dit data blive slettet.

Spørgeskema

1. Mit køn: Mand ___ Kvinde ___ Andet ___
2. Min alder: _____ År
3. Jeg bruger briller eller kontaktlinser til hverdag: Ja ___ Nej ___
4. Jeg foretrækker at skrive med: Venstre hånd ___ Højre hånd ___
5. Jeg er *meget erfaren* med brug af Virtual Reality udstyr:

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget Enig
6. Jeg er *meget erfaren* med Dansk Tegnsprog:

   Meget uenig 1 ___ 2 ___ 3 ___ 4 ___ 5 ___ Meget Enig

Deltager nummer _____ (udfyldes af dataansvarlig)

E - Initial UI Design Sketch