Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/88/20241639

GazeLink: A multi-language low-cost mobile eye-gesture communication system with large language models for people with amyotrophic lateral sclerosis

Xiangzhou Sun

The Webb Schools, Claremont, CA, United States

jonassunhk@gmail.com

Abstract. Amyotrophic Lateral Sclerosis (ALS) patients who have severe motor and speech impairments mostly rely on their eyes and assistive technology to communicate. However, existing high-tech products are expensive and hard to access, while low-tech products are inefficient and restrictive. To mitigate the limitations, this research proposes GazeLink, a multi-language low-cost mobile application for ALS patients to communicate efficiently with only eye movements. First, the system recognizes user eye gestures like left or up with machine learning and a template-matching algorithm. Then, it converts the eye gestures to words through a keyboard that supports English, Spanish, and Chinese. For efficiency, the system employs Large Language Models (LLMs) to generate a suitable sentence with words typed by the user and the context. Finally, the system provides text-to-speech and social media post services for both verbal and digital eye-gesture communication. Simulations conclude that sentence generation with LLMs can reduce user keystrokes by 81% while maintaining 90% of semantic similarity. Usability studies with 30 participants show that GazeLink can recognize eye gestures with 94.1% accuracy in varying lighting. After rapidly learning the user interface in under 10 attempts, first-time participants typed sentences of various lengths with their eyes at 15.1 words per minute, which is 7.2x faster than the common low-tech solution E-Tran. Experiments demonstrate GazeLink's efficiency, learnability, and accuracy in eye-gesture text entry. The system is extremely affordable (less than $0.1 a month), portable, and easily accessible online. It also supports different users, lighting, smartphones, and languages. Product testing with ALS patients and personalized LLM models will be the next step.

Keywords: Assistive Technology (AT), Eye Gesture, Amyotrophic Lateral Sclerosis (ALS), Human-Computer Interaction (HCI), Large Language Model (LLM), Computer Vision (CV).

1. Introduction

Amyotrophic lateral sclerosis (ALS), the umbrella term for Motor Neuron Disease (MND) in the United States, describes a group of neurodegenerative disorders that causes motor neurons in the brain to malfunction and degenerate prematurely, disabling muscle activity like swallowing, walking, or gripping [1]. In severe cases, the disorder evolves into an incomplete locked-in state (iLIS): the patients' limbs become immobile, and their speaking abilities worsen [2]. Patients in iLIS encounter many difficulties in their quotidian routine and require assistance from clinicians. Since oculomotor functions are generally preserved in ALS, most People with ALS (PALS) use eye movements to communicate [3].

© 2025 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/88/20241639

Current solutions include low-tech and high-tech products to aid ALS user's communication. Low-tech methods include choosing letters on a communication board with a laser pointer such as E-tran, but they are both time-consuming and limited in expression [4]. Existing companies such as Tobii Dynavox developed high-tech assistive technology with speech generation and dwell-free eye tracking [5]. However, most of their products like the TD I-Series cost around $19000, an exorbitant price for underserved populations without adequate medical care. Additionally, both low-tech and high-tech solutions have hefty physical components that are hard to carry, deliver, or fix.

Under this context, this study proposes GazeLink, an eye-gesture-based mobile text-entry system powered by LLMs to aid the communication of PALS. The specific functionalities of the GazeLink system include: 1) Calibration for accurate eye gesture recognition, 2) Accurate real-time recognition of seven eye gestures with the smartphone's front-camera input, powered by face recognition machine learning models, and a template-matching algorithm, 3) Typing keywords with the eye-gesture keyboard and a Dynamic Vocabulary Bank (DVB), 4) Converting the keywords into a suitable and grammatical sentence with fine-tuned LLM text generation, 5) communicating the text with text-to-speech (TTS) functionality or posting it to social media for digital expression, 6) A visual-aligned and modular user interface (UI) for text entry and displays for calibration, settings adjustment, and data evaluation.

GazeLink is efficient, affordable, portable, and accessible. Compared to low-tech solutions, the system is significantly faster and provides a substantially wider range of semantic expressions. The only hardware requirement of GazeLink is a standard smartphone equipped with a front-end webcam. To leverage the system's context-aware text generation capabilities, users must cover expenses associated with LLM cloud services, which are minimal compared to the costs of existing eye-typing products available on the market. Users can download the mobile application online without physical shipment and repair. The system will be scalable to tablets or laptops in the future.

To evaluate GazeLink comprehensively, this research divides the system into key components. Computation evaluations assess LLMs in terms of semantic similarity and keystroke savings, and results conclude that the integration of text generation significantly increases the text-entry rate while preserving accuracy. Usability studies with first-time participants demonstrate the system's robustness in recognizing eye gestures and the learnability of our eye-gesture keyboard. Finally, comprehensive system testing shows that most users can swiftly type an accurate sentence with GazeLink after a training session of less than 20 minutes. A questionnaire collected after the experiments also concludes that most users consider Gazelink to be easy to use and mentally effortless.

The main contributions of GazeLink include 1) adaptable and robust eye gesture recognition on mobile devices with machine learning and template-matching, 2) a user-centered eye-gesture keyboard to type words in multiple languages, 3) multi-language sentence generation from keywords and context with LLMs for faster text entry, 4) a new easy-to-learn eye-gesture keyboard layout that allows users to independently type a complete sentence swiftly.

2. Background

2.1. Related Works

A dwell-free gaze-tracking system named EyeK designed by Sarcar et al. reduces visual search time and dwell time by testing novel keyboard layouts [6]. Despite being low-cost, the external hardware is hard for PALS to access, and production difficulties may hinder the globalization of the system.

A study by Fan et al. examines the recognition of eyelid gestures for people with motor impairments on mobile devices [7]. The accuracy may be insufficient for a highly efficient text-entry system, but 12 gestures offer many possibilities.

Cecotti et al. present a multimodal virtual keyboard that employs both eye and hand gestures for text entry [8]. However, this system is not suitable for most PALS, who cannot perform hand gestures due to their condition.

A portable, low-cost system named GazeSpeak developed by Zhang et al. allowed PALS to communicate words with gesture-based eye typing on iOS smartphones [9]. They extracted eye frames

DOI: 10.54254/2755-2721/88/20241639

from the smartphone's camera and stored calibration files for template matching. Then, they interpreted the eye gestures in real time and converted the inputs into words. However, there are several limitations to the system. First, a trade-off of an ambiguous keyboard is a constricted vocabulary bank due to the time complexity of word prediction. The system is also incapable of typing grammatical sentences as the vocabulary bank lacks verbs of all tenses or plural nouns. Additionally, the system did not capitalize on the conversational context.

Another system named KWickChat proposed by Shen et al. generates grammatical and semantic sentences with keywords and context information to accelerate the text-entry rate for ALS patients [10]. The context they collected includes previous dialogues, user inputs, and personal information. Using a Natural Language Processing (NLP) model named BERT, they shortened sentences into keywords, and, with the extracted context, trained a GPT-2 model evaluated on semantic and grammatical accuracy. However, the system's suitability is not applied and examined with an interactable text entry system. Moreover, the up-to-date GPT-3.5 model that may improve the results of sentence generations is not adopted.

2.2. Large Language Models

Transformer is a combination of encoders and decoders introduced in 2017 by Google [11]. Encoders translate the input text into a continuous representation, while decoders take the continuous representation as input to generate an output sequence. The key mechanism of transformers is self-attention, which allows the model to identify dependencies between different parts of the input sequence and the most relevant words. Self-attention improves the model's understanding of context and its ability to manage long-range dependencies in sequences of various lengths.

Adapting a transformer architecture, the Generative Pre-trained Transformer (GPT) is pre-trained with a large dataset and analyzes a prompt given by the user to predict responses. The GPT-3.5 model is a GPT developed by OpenAI and used for a variety of Natural Language Processing (NLP) tasks. With significantly more parameters, GPT-4 is more robust than GPT-3.5 and multimodal but slower due to its complexity [12].

Fine-tuning, a supervised learning process, further improves GPT performance for specific tasks. It requires a labelled dataset that the LLM will use to update its weights. For OpenAI GPT models, A relatively small dataset of around 100 examples is sufficient in significantly boosting performance for certain tasks.

3. System design

GazeLink's primary functionality is independent text entry for PALS, which follows a four-step process: 1) the system extracts the user's eye region in real-time and recognizes the eye gesture with a template-matching algorithm, 2) an eye-gesture keyboard converts the eye gestures to keywords, 3) on-cloud LLMs use the keyword and conversation context to generate a suitable sentence, 4) the user communicates the sentence via phone speaker or social media upload.

To facilitate the process, GazeLink implements a multipurpose, easy-to-learn UI with several parts:

- *Data Interface* displays a log that tracks the user's gaze gestures chronologically and data that summarizes the general performance of the user.
- *Text-entry Interface* allows users to type sentences with only eye gestures. It implements the multi-language eye-gesture keyboard with LLM enhancements and text-to-speech functionalities.
- *Quick Chat Interface* allows users to rapidly type 25 common sentences for the communication of ALS patients with only eye gestures.
- *Settings Interface* displays adjustable configurations like recognition sensitivity and text-entry method. The interface also presents the user's current gaze gesture and inputs for testing.
- *Calibration Interface* facilitates the calibration of each gaze gesture. The system will use the templates for template-matching to accurately detect gaze gestures and store them locally for the next use.

DOI: 10.54254/2755-2721/88/20241639

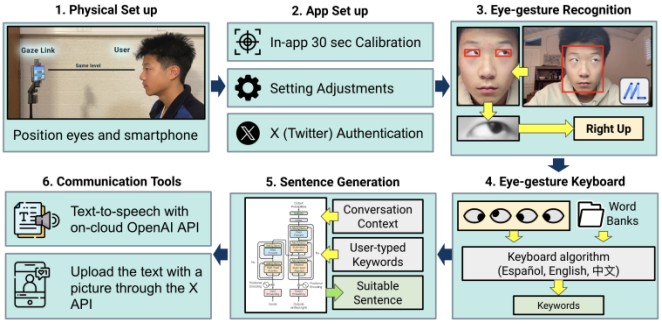

Figure 1. An overview of the GazeLink system. First, the eye gestures are recognized. Then, they are converted to keywords through the eye-gesture keyboard. Next, the model uses keywords to generate a complete sentence. Finally, GazeLink includes text-to-speech and social media connection services for verbal and digital communication.

GazeLink only requires the front-end webcam, touch screen, and speaker of a smartphone to function. An Internet connection is also required for LLM services on the cloud. We recommend the user to pair the smartphone with a phone stand with adjustable height and orientation.

The main components of the system are: 1) an eye gesture interpreter, 2) a multi-language eye-gesture keyboard engine with DVB 3) multi-language LLM implementations through cloud 4) a visual-aligned modular UI.

4. Eye gesture recognition

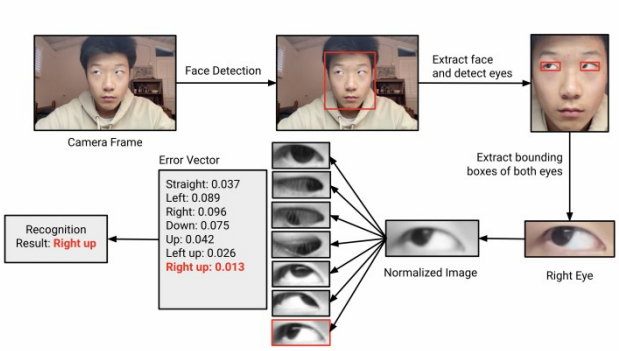

Figure 2. Eye gesture recognition flowchart. First, we detect the face and extract eye landmarks with Google ML Kit. Then, the image is resized and normalized. Finally, we compare the processed image with labeled templates to determine the most likely eye gesture. All steps are computed on the device.

GazeLink recognizes up to seven eye gestures (left, right, left up, right up, up, center, closed) with promising accuracy for both eyes (see Figure 2). The recognition system uses Google ML Kit for face and eye detection and OpenCV for image processing [13].

There are many Machine Learning (ML) algorithms available for eye gesture detection, including template-based and data-driven methods. Previous works conclude that, for eye gesture recognition systems with fewer templates, template-based recognition methods are more accurate and adaptable compared to data-driven methods. Template-based algorithms first collect labeled templates of distinct eye gestures and compare them in real-time to determine the most probable eye gesture [14].

4.1. Calibration

For first-time users, the system must collect templates for every gaze gesture. The process begins with an assistant pressing the “Continue” button on the device. The application will provide audio guidance such as “look left,” and users are expected to hold the gaze gesture until the assistant stores the camera

DOI: 10.54254/2755-2721/88/20241639

frame with the "collect" button. Participants are asked to perform slightly exaggerated eye gestures and eschew vibrating the phone or moving their heads.

The process takes around 30 seconds on average. Calibration is required for the first usage and recommended with different users or dramatic lighting changes. After capturing the frame, the system processes and stores the template with the following steps:

1. *Eye landmark extraction*: With the Google ML Kit library, we detect the face from the frame and extract the contour of the left and right eye.
2. *Bounding Box Calculation*: We take the 4 bounding points (maximum x and y, minimum x and y) in the contour to extract the bounding box for the eyes.
3. *Image Processing*: The bounding box is grayscaled, normalized, and resized with OpenCV
4. *Data Storage*: We save the processed calibration templates on the Android device. When the user reopens the application, the system automatically loads the templates from the device.

4.2. Recognition Algorithm

After calibration, the user can switch to the settings interface to test the accuracy and adjust settings accordingly. We use the risk function Mean Squared Error (MSE), which measures the average of squared errors between two sets of values, to compare the processed image with the labeled templates. The template with the lowest loss is the most probable eye gesture.

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2 \tag{1}$$

One difficulty in eye-typing is distinguishing accidental glances from intentional inputs. To ensure that the system does not recognize users' unintentional gestures as legitimate inputs, we set a dwell time of 0.25 seconds (3 frames). Users must hold a gaze gesture until they hear a "beep" sound, which indicates that their input is recognized.

5. Keyboard design

5.1. Multi-language Eye-gesture Keyboard

Compared to GazeSpeak's keyboard which only allows the user to form sentences using words in the preset corpus, we propose a new multi-language eye-gesture keyboard that allows users to type an enormous range of words rapidly without any assistance. Considering it is inefficient and fatiguing for users to specify each character of a word using limited types of eye gestures, GazeLink incorporates all 26 letters of the alphabet with four "blurry inputs" (A–F, G–M, N–T, and U–Z), each one representing a range of letters. For example, letters A, B, C, D, E, and F will all correspond to the blurry input A-F. After the user completes typing the blurry input, GazeLink lists out all the possible words, sorted by frequency, by searching through a word bank of 5000 commonly used English words. A trie data structure optimizes the search to $O(n)$ time complexity, with n representing the number of blurry inputs [9].

To adapt to two other languages, Spanish and Chinese, the eye-gesture keyboard uses different vocabulary banks for each. Since Spanish has many conjugations for its verbs, the vocabulary bank includes all the infinitives instead of all separate conjugations. In case of adjective-noun agreement, we only included adjectives and nouns in the masculine and singular form. All the verbs, adjectives, nouns, and other words are sorted by frequency to maximize efficiency.

For Chinese, we first computed all the possible combinations of pronunciations and organized them in a table. Then, we extracted the top 5000 words in a collection of the most frequent Chinese characters by Jun Da and determined the most common pronunciations. We sorted the pronunciation based on the frequency and used it as the vocabulary bank for the Chinese eye-gesture keyboard.

GazeLink's text-entry interface incorporates two types of UI layout elements: non-interactive text boxes to display current inputs and eye gesture buttons activated with either eye gesture or touchscreen. To leverage the small number of distinct eye gestures without over-complicating the system, the interface employs two switchable modes, a letter mode that types the blurry input and a word mode that

DOI: 10.54254/2755-2721/88/20241639

selects the intended word. All buttons have one functionality for each mode. The closed-eye gesture button is always used to switch between modes.

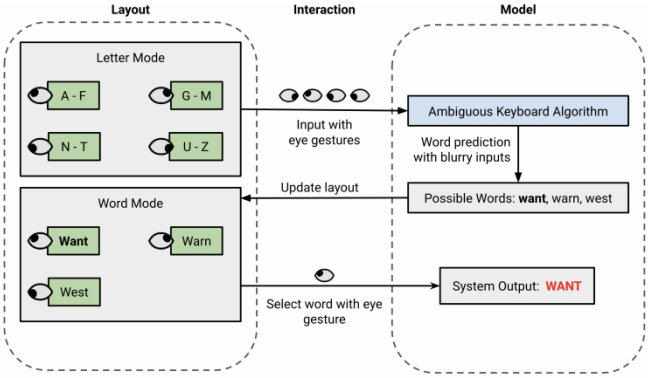

Figure 3. A diagram of the text-entry process involving the UI layout, system model, and interaction between device and user. First, the user types the blurry input of the word with eye gestures. The system processes the blurry input and updates the layout with possible matching words. Finally, the user selects the intended word with an eye gesture.

For the user's convenience, the eye gesture buttons are split between alphabetic buttons and modifier/control buttons. Alphabetic buttons (left up, right up, left, and right gesture) manage the actual text entry, including blurry input and word selection. In contrast, modifier/control buttons (up and closed) perform special actions, including delete or speak. The interface distinguishes between the two types with color and size.

5.2. Text-entry Interface

In addition to eye gesture buttons, the interface includes three non-interactive text boxes that display the following: 1) the context of the conversation inputted through voice recording or hand-typing, 2) current text that the user typed, 3) the sentence generated by LLM model.

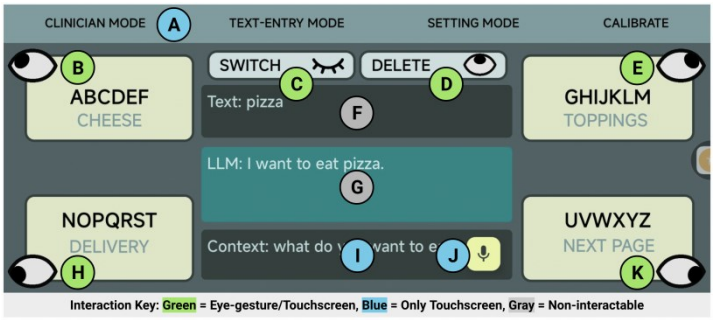

Figure 4. Text Entry interface: (A) Top bar to change modes; (B) Left up eye gesture button to type A-F or word 1; (C) Close eye gesture button to switch between letter and word mode; (D) Up eye gesture button to delete or speak input, depending on the mode. (E) Right up eye gesture button to type G-M or word 2. (F) Text that displays user input. (G) Text that displays the generated sentence. (H) Left down eye gesture button to type N-T or word 3. (I) Text that displays the context of the conversation. (J) Record button to enter context with Android's voice-to-speech. (K) Left down eye gesture button to type U-Z or switch pages for word mode.

DOI: 10.54254/2755-2721/88/20241639

6. LIM implementations

6.1. Multi-language Keyword-based Sentence Generation

To ensure accurate and rapid text entry for the user, GazeLink uses Large Language Models (LLMs) such as GPT-3.5 and GPT-4 for word generation, next-word prediction, and sentence retrieval. Both GPT-3.5 and GPT-4 are models based on the transformer architecture introduced in 2017 by Google [11].

GazeLink improves keystroke savings by implementing a keyword-based sentence generation model with a fine-tuned GPT-3.5 model. Instead of requiring users to type complete sentences, the system only uses keywords and the conversation context to automatically generate the intended response, drastically reducing the number of keystrokes. Additionally, generated sentences ensure grammatical consistency, which is a limitation of the GazeSpeak text-entry system.

To train the sentence generation model, we created a training and validation dataset named the GazeLink dataset. Each entry in the dataset includes 1) a target sentence, the expected output for the model, 2) conversation context, the previous turn in the dialogue, and 3) keywords, a subset of keywords in the target sentence that preserves most of the meaning. To obtain the target sentence, we leveraged a crowdsourced corpus of augmentative and alternative communication (AAC)-like communications [15]. The corpus contains approximately 6000 sentences based on telephone conversations, social media, and newswire text that encompass common communications of AAC users. However, the sentences provided in the corpus do not contain the conversation context or keywords. Therefore, we generate the conversation context and keywords with surrogate models.

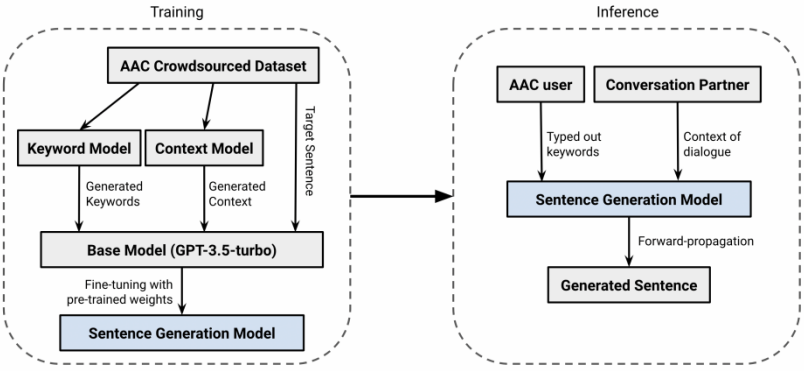

Figure 5. The overall process of the sentence generation model. First, keywords and context are generated from the AAC Crowdsourced Dataset with surrogate models. Then, the base model is fine-tuned with entries of keywords, context, and target to create the sentence generation model. During inference, the model takes keywords and context as input to generate a suitable sentence.

First, to generate conversation context from the AAC crowdsourced corpus, we trained a surrogate context model with few-shot learning of GPT-4 and the ConvAI2 challenge dataset [16]. The ConvAI2 challenge dataset contains 11000 dialogues of natural conversations between crowd workers.

We also employ the GPT-4 model and few-shot training for bag-of-keyword extraction from the target sentences. Rather than using previous datasets, we obtain targets in the training data by recruiting human participants to extract keywords from 20 sentences.

After completing the GazeLink dataset with the surrogate context and keyword models, we convert the entries into JavaScript Object Notation Line (JSONL) format and upload the file to the OpenAI developer platform for fine-tuning with a gpt-3.5-turbo-1106 model. The fine-tuned model is stored in the cloud and has a training loss of 0.0496 after 3 epochs and 29670 tokens for English (See Figure 5.) Our system can access the model via the internet and the OpenAI API when the user inputs a keyword (see Figure 5).

DOI: 10.54254/2755-2721/88/20241639

We also implemented a Google Gemini model with a similar methodology inside Gaze Link. Future work will include accuracy and latency comparisons between different fine-tuned LLMs.

6.2. Dynamic Vocabulary Bank

GazeLink applies a GPT-3.5 model to generate words for the DVB. Previous works' eye-gesture keyboard restricts the user's vocabulary to a word bank of 5000 words. However, users may use specific or technical nouns, like France or pizza, outside of the bank during their conversations. Therefore, we propose a DVB that extends the traditional word bank by leveraging the dialogue's context and previous inputs, providing users with a wider range of semantic expressions (see Figure 6)

With the application's UI, the conversation partner enters the dialogue context into GazeLink through an editable textbox or a voice recording button. The system recognizes the change in context and sends a prompt like "Generate 50 words related to the context: What do you want to eat?" via the internet to access the GPT-3-turbo model with the OpenAI API. We leverage prompt engineering concepts like specificity and role-play to improve the model's accuracy. The model then generates several words related to the context, such as burger, sushi, or fries, and transmits the words to the device via the internet. We insert the 50 dynamic words into the DVB and display them in front of static words. Dynamic words from previous turns in the dialogue are deleted to ensure efficiency.

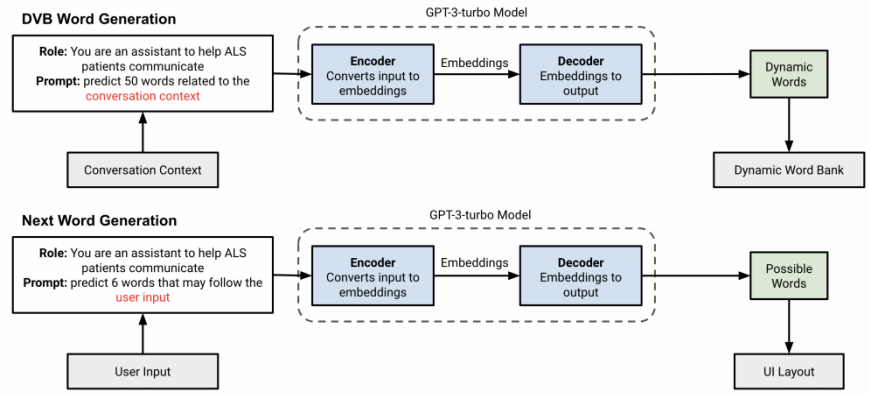

Figure 6. A flowchart of DVB and next word prediction with prompt engineering.

6.3. Next Word Prediction

GazeLink also implements next-word prediction with the GPT-3.5 model to improve the text-entry rate. After a user enters a word, the system ranks possible succeeding words with the dialogue context and the last typed word. For example, for the word "sleep", the system displays predicted words like "now", "peacefully", or "later" in the user interface.

7. Experiments

7.1. Multi-language Sentence Generation Model Evaluation

We first evaluate the effectiveness of the model analytically by examining the trade-off between semantic similarity (%) and keystroke savings. Semantic similarity measures the closeness in meaning between the target sentence and the generated sentence. Since all users have different text-entry rates, we quantify the model's speed with keystrokes, the minimum number of eye gesture inputs required to type a sentence. We used a Windows 11, 16G RAM, CPU Intel i9, GPU Nvidia RTX 4060, 1T storage computer to experiment.

There are multiple model-independent evaluation metrics to measure sentence similarity, including overlap-based and embedding-based metrics. Overlap-based metrics such as the Bilingual Evaluation Understudy Score (BLEU) analyzes the amount of word overlap between two sentences. However,

DOI: 10.54254/2755-2721/88/20241639

according to recent studies, BLEU only considers the lexical similarity and neglects the overall meaning of the sentence, which is more significant than word overlaps in terms of accurate communication.

On the other hand, embedding-based metrics represent the sentence in a latent space instead of comparing words on the surface like BLEU [17]. Consequently, embedding-based metrics demonstrate a higher correlation with human judgment and examine the semantic similarity more robustly [18]. However, embedding-based metrics is slower and requires training. We resolve this limitation by employing an open-sourced embedding-based sentence transformer model named all-MiniLM-L6-v2 on HuggingFace as the evaluation metric [19].

The experiment leverages a portion of the GazeLink dataset created with the AAC crowdsourced corpus and surrogate models. Distinct from the training data, the testing data contains 40 entries of target sentences with conversation context and keywords of different sizes (min: 1, max: 5).

For each entry, we study how the size of keywords affects the semantic accuracy of the generated sentence and the keystroke saving, the percentage of keystroke reduction from the baseline. The baseline does not implement any LLMs, so every word in the sentence must be typed. However, the sentence generation model only requires keywords as input.

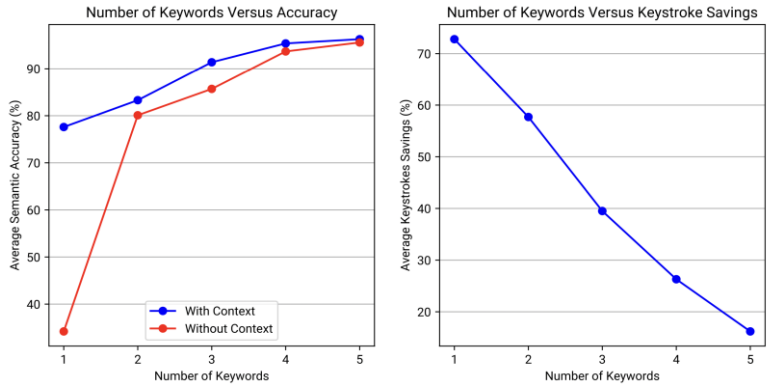

Figure 7. Two graphs to present how the number of keywords affects average semantic accuracy (%) and average keystrokes savings (%) over 38 samples. The blue line is the model with context, and the red line is the model without context.

The left graph of figure 7 shows that as the number of keywords increases, the average semantic accuracy (%) increases, which signifies that the generated sentence more closely corresponds to the target sentence. This reveals that if more information is provided, the model can more accurately generate a sentence. However, the model's pattern of change differs depending on the presence of the conversation context. At a low level of information (keywords = 1), context is exceptionally important in preserving semantic accuracy because it provides essential background knowledge for the model. For example, if the target sentence is "I want to eat pizza," a context like "What do you want to eat?" would only require the keyword "pizza" for an accurate prediction. When keywords increase from 1 to 2, a rapid climb in accuracy is observed in the model without context because the second keyboard is crucial in forming a sentence. For example, the word "water" by itself has many possibilities as a sentence, but two words "drink, water" narrows it down significantly. As information further increases, the model reduces its dependency on the context, and both line plateaus at around 96% semantic accuracy.

The right graph of figure 7 demonstrates that as the number of keywords increases, keystrokes savings decreases. Since typing more keywords requires more keystrokes, it reduces the user's text-entry rate. Therefore, although more information benefits semantic accuracy as shown in the left graph, it also slows the speed of text input.

To further evaluate the trade-off between speed and accuracy, this research determines the average keystrokes savings for each semantic similarity (%) threshold (min: 50, max: 100, step: 5) The experiment is also repeated without including the context.

DOI: 10.54254/2755-2721/88/20241639

Similar to the results of Figure 7, Figure 8 shows the inverse correlation between semantic accuracy and keystroke savings. Since the curve is concave down, significantly more information is required to achieve a high semantic accuracy threshold like 90%. The presence of conversation context also has a notable impact on the percentage of keystrokes saved. The keystrokes savings tend to be approximately 10% lower when context is not present. The graph is beneficial in quantifying the relationship between speed and accuracy. We conclude that, when context is present, a semantic accuracy threshold of 85% can save 55.5% of keystrokes, which potentially reduces more than half of the text entry time. To put in perspective, the sentences "I want to sleep later" and "I want to sleep in a while" have a semantic similarity of 84.7%.

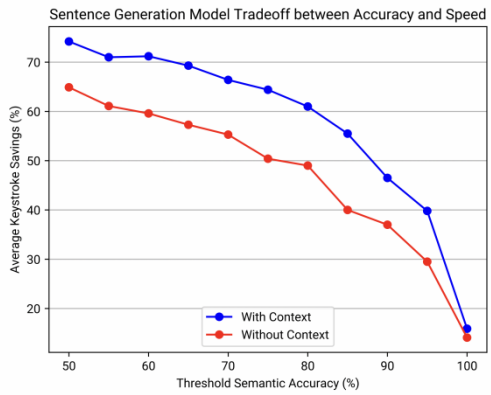

Figure 8. Graph that demonstrates the trade-off between semantic accuracy (%) and keystrokes saving (%). The keystrokes savings without conversation context (red) are also included.

This research also evaluated the effectiveness of sentence generation in Spanish and Chinese. The procedure is similar except for the dataset creation. New datasets are needed for the different languages since Spanish and Chinese are fundamentally different from English. In the Spanish dataset, I included vocabulary with accents. In the Chinese dataset, I used the pronunciations (Pin Yin) for the Chinese characters as the keywords and Chinese characters for the context to align with the eye-gesture keyboard.

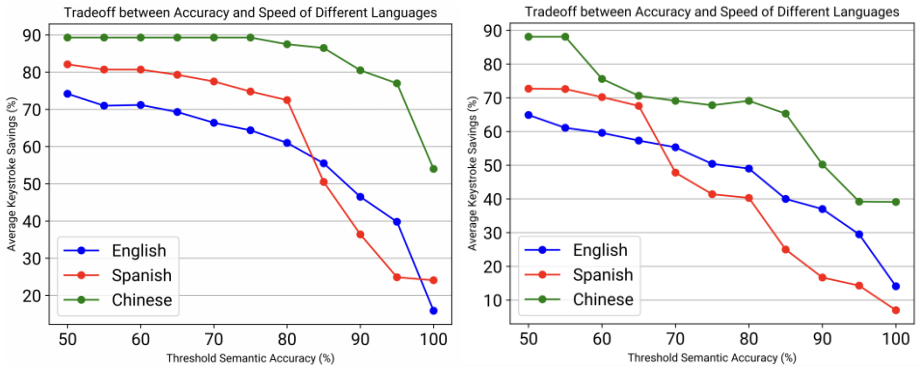

Figure 9. Graphs of the performance of LLMs for each language with context (left) and without context (right).

All three LLM sentence generation models can reduce at least 50% of keystrokes while maintaining a semantic accuracy of 85%, which increases by text entry rate by around two times. The Chinese LLM performs the best, reducing 81% of keystrokes while maintaining 90% of semantic accuracy.

7.2. Usability Testing

We recruited 30 able-bodied participants (5 for pilot testing) within the community to evaluate the effectiveness of the GazeLink system. Based on convenience sampling, participants within a high school

DOI: 10.54254/2755-2721/88/20241639

community are contacted through email and asked to sign a human consent form before the experiments. For participants under 18, the form also requires a signature from a parent or guardian. Since all participants do not have previous experiences with a similar text-entry system, their testing provides valuable data on GazeLink's learnability for first-time users. The participants have an average age of 22.17 (min: 12, max: 57). 24 were male and 6 were female. 12 wore glasses while others had perfect vision or contact lenses. The eye and skin color of participants also varied.

The testing session has 3 parts: layout learnability, eye gesture recognition, and system testing (35 minutes). Most testing was conducted in the Thornton Lab of The Webb Schools in Claremont, LA, between 2024/1/14 and 2024/2/2.

For all experiments, we use the Android smartphone Xiaomi MIUI 14.0.10 with 8.0 + 3.0 GB RAM, Octa-core MAX 3.0GHz CPU, and a 50MP camera. We stabilize the phone on a tripod placed on a table and adjust the height to the participant's eye level. Only the front-facing camera of the smartphone is used so the participant can see the screen at all times. Since glasses significantly affect the ML eye detection functionality, we ask participants to take off their glasses during the experiments and adjust the phone's distance so the text is visible.

After setup, we play a 3-minute tutorial that introduces the product and text-entry method so all participants possess a controlled understanding of the system. Then, we test the participants on two key components of GazeLink: the eye-gesture keyboard and eye gesture recognition. Finally, the users combine their previous experience to type complete sentences with only eye gestures. We also ask participants to complete a questionnaire for qualitative data concerning their experience of using the system.

7.3. Layout Learnability

Before using eye gestures to input text, all participants must first learn the eye-gesture keyboard. Since this experiment only assesses the learnability of the text-entry interface, we ask the participants to speak their intended input instead of using eye gestures. For example, the user will verbally say "left up," and then we will physically enter the input via the touchscreen. This interaction prevents eye fatigue as many trials are conducted to examine the learning curve. For 10 trials, the participants will utter the eye gesture to type out a sentence with Gaze Link while we time and observe their input.

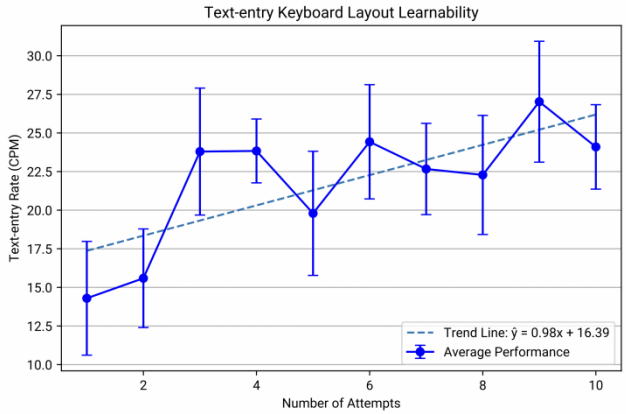

Figure 10. A plot that indicates the change in average text-entry rate (Characters per Minute) and variability as the number of attempts increases. The error bars are the standard deviation of the average text-entry rate of each attempt. The trend line is the line of best fit.

Figure 10 shows the learning trend of participants testing out GazeLink's text-entry keyboard. The line of best fit signifies that for every new attempt, the text-entry rate of participants increases by 0.98 characters per minute. We observe that participants achieve a maximum performance of 27.0 characters per minute, a 12.8 increase from the first attempt. A one-way ANOVA test shows that the two trials are significantly different (f-score=39.3). The error bars show the standard deviation of each trial (average

DOI: 10.54254/2755-2721/88/20241639

= 3.4). Overall, the graph verifies that with only 10 attempts, the participants already demonstrated a significant improvement.

7.4. Eye Gesture Recognition Performance

After completing the text-entry learnability study, we examine GazeLink's ability to recognize the eye gestures of the participant. After calibration and setting adjustments, we record the recognition of each gaze gesture for 5 trials. For each trial, we instruct the participant to perform an eye gesture (up, left, right, right up, left up, closed) in random order. Participants must maintain the gesture until they hear a "beep" sound from the device, signaling that the eye gesture is recognized, or 3 seconds have passed. If the eye gesture is recorded within the period and matches the instruction, the trial is considered successful. Else, it is unsuccessful. The participant may request a break anytime during the experiment.

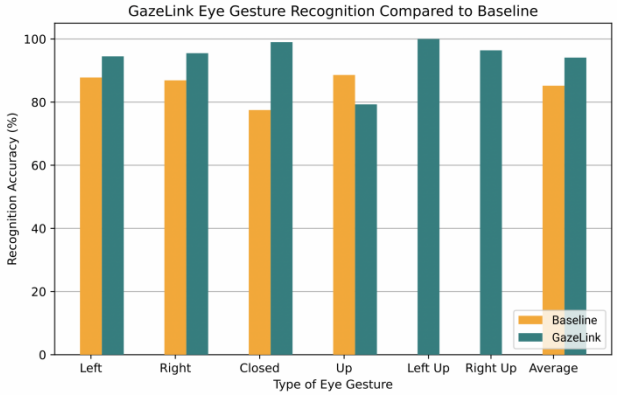

Figure 11. A bar graph that compares baseline eye gesture recognition (orange) with GazeLink eye gesture recognition (green). GazeLink recognized two extra eye gestures, left up and right up, that the baseline did not recognize.

Figure 11 shows that, compared to the baseline (GazeSpeak system), GazeLink's eye gesture recognition is overall superior. The system also recognizes two extra eye gestures, left up and right up, with no baseline data. We observe GazeLink's robust performance in recognizing the closed and left-up gesture with 99.0% and 100% accuracy respectively. This is because the calibration templates of these two gestures are usually very distinct from other templates, which makes them easier to distinguish. Similarly, the substandard performance of the up gesture is likely due to its resemblance with the left-up and right-up gestures. The baseline does not recognize them and thus performs better. Overall, the system has an average recognition accuracy of 94.1%, an 8.9% increase from the baseline.

7.5. Overall System Testing

After the participants are familiar with both gaze recognition and text-entry UI, we conduct a comprehensive system testing to examine their ability to combine both skills and type suitable sentences with only eye gestures. Since all participants are first-time users of the text-entry system, we begin with a lower difficulty and gradually increase the keyword size within the participant's ability.

For the first trial, an entry from the GazeLink testing dataset with only 1 required keyword is selected randomly. We enter the context into the system through voice and inform the participant of the keyword. After the participant enters the keyword through eye gestures, the system will generate a suitable sentence based on the keywords and context. We record the time difference between the first and last input as the text-entry time. We also ask the user to subjectively rate the semantic similarity between the generated and target sentence on a scale of 1-10. Then, we randomly select an entry with one extra keyword and repeat the process for 2-3 trials, depending on the ability and eye fatigue of the participant.

DOI: 10.54254/2755-2721/88/20241639

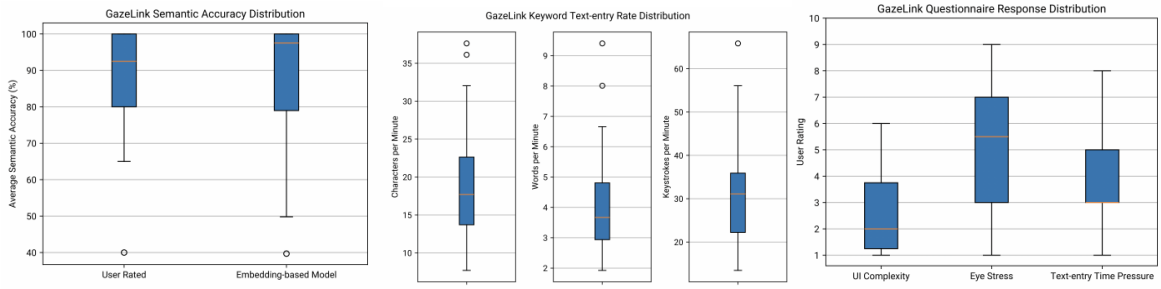

Figure 12. Three box plots to show Gaze Link's semantic accuracy, text-entry rate, and questionnaire response distribution.

The box plots in Figure 12 demonstrate the high semantic accuracy of GazeLink during system testing verified with two different metrics, user rating and embedding-based model. The median of user rating and embedding-based model is 92.5% and 97.5% respectively. One possible reason for this is that the user rating is based on a 10-point scale with a step of 0.5. Therefore, some sentences with extremely close meanings are at most rated 9.5 (95%). However, the embedding-based model can output more specific values like 97.6%. The 25th percentile (1st quartile) of both metrics is 80, meaning that 75% of the generated sentences will have a semantic accuracy between 80% to 100%.

The box plots in Figure 12 analyze the text-entry rate for entering keywords during system testing with different units. The system can type 17.7 characters per minute (IQR=9.0, min=7.7, max=37.6), 3.7 words per minute (IQR=1.9, min=1.9, max=9.4), or 31.1 keystrokes per minute (IQR=14, min=13.4, max=65.8).

This research evaluates the user experience of GazeLink through recording observations during the experiments and collecting an optional questionnaire inspired by the NASA Task Load Index (NASA TLX) scale, which measures the mental workload of a participant while performing a task [20]. The questionnaire asks the participant to quantify the physical (eye stress level), temporal (text-entry time pressure), and mental (UI complexity) demands of entering text through GazeLink from a scale of 1 to 10. Participants can also leave suggestions and remarks concerning their experience.

The results from Figure 12 demonstrate that most users do not consider GazeLink to be mentally or temporally demanding. The median for the UI Complexity and Text-entry Time Pressure rating are respectively 2 (IQR=3, min=1, max=6), and 3 (IQR=2, min=1, max=8). However, the eye stress rating is relatively higher than the other two demands with a median of 5.5 (IQR=4, min=1, max=9). This is because first-time participants are not familiar with using an eye-gesture communication tool, so their eyes may be stressed and uncomfortable after many trials.

Comments of first-time participants include: 1) *The text entry is surprisingly functional after calibrations and a few attempts*, 2) *The UI is "simple" and "easy to grasp."*, 3) *Eye gesture recognitions are accurate except for one particular gesture.*, 4) *Eyes are stressed and tired after the experiments*. Recalibration and interval breaks were able to reduce the limitations that the users mentioned.

7.6. System Comparisons

We evaluate the contribution of GazeLink by comparing its text-entry performance to two other systems: A low-tech gaze-transfer board named E-tran and another mobile gaze-tracking communication system named GazeSpeak [9]. We extract the data of these two systems from a previous work that also recorded the time taken to type out a sentence with only eye gestures. Although the previous work uses another phrase set created by Mackenzie and Sourkeroff while this research uses an AAC crowdsourced dataset, both datasets are taken from online and are similar in representing daily communications [21]. We calculate the text-entry rate of baseline data by dividing the total time taken by the number of words typed.

DOI: 10.54254/2755-2721/88/20241639

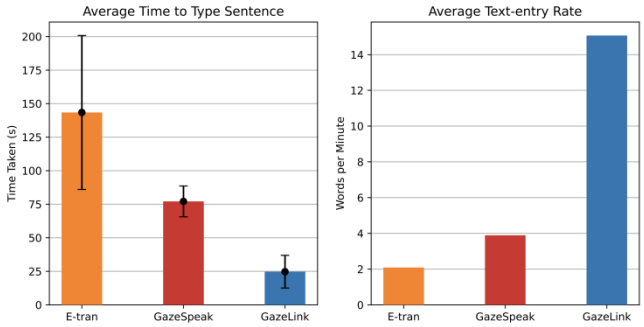

Figure 13. Bar graph comparing three eye-gesture communication systems: E-trans, GazeSpeak, and GazeLink in terms of time taken to type a sentence (s) and the average text-entry rate (wpm). The error bars on the left graph represent standard deviation.

Figure 13 shows that GazeLink is significantly faster than baseline systems in text-entry rate. While E-tran and GazeSpeak took 143.4 and 77.1 seconds on average to complete a sentence, GazeLink only took 24.7 seconds on average, which is 3.12x faster than GazeSpeak. However, both systems had approximately the same standard deviation (GazeSpeak=11.5, GazeLink=12.2). When converted to text-entry rate, GazeLink (15.07wpm) is around 3.87x faster than GazeLink (3.89wpm) and 7.2x faster than E-tran (2.09wpm).

8. Discussion

This research presents GazeLink, a low-cost, accessible, and portable communication system for PALS, to resolve the limitations of low-tech and high-end devices on the market. GazeLink is scalable to any mobile device and only requires a standard smartphone to type grammatical sentences independently. The only running cost of GazeLink is cloud-based LLM services for text generation. Assuming that the user can type 1000 words (1334 tokens) a day, the cost per month would be around $0.06. This price is significantly lower than physical products with specialized hardware such as the E-trans board ($100) or the Tobii Dynavox TD I-13 ($8800 $19000) [5]. As a mobile application, GazeLink is distributed online and avoids the complications of shipment and repair, facilitating access for PALS in remote areas. The system is highly portable since smartphones are smaller and lighter than existing AAC devices like E-trans boards or Tobii Dynavox's TD Pilot [4, 22].

In addition, GazeLink demonstrates robust efficiency, learnability, and usability, which aligns with the hypothesis. The system incorporates on-cloud LLMs to enhance the current keyboard in many aspects, including an AI-driven DVB that adds new vocabulary to a static word bank according to the conversation context, a next-word prediction model that provides possible words from the current text input, and a sentence retrieval model that generates grammatical sentences based on keywords and the context. Computer simulations and system testing validate LLM's effectiveness in improving text-entry rate while preserving semantic accuracy. This research analyzes previous eye-tracking systems and designs a visual-aligned, modular UI for different roles. The text-entry UI allows PALS to type and speak the text with only eye gestures. Usability studies demonstrate that most participants swiftly comprehend the text-entry UI and text-entry method. A questionnaire collected after the experiments shows that although some participants experienced eye fatigue, the cognitive task load of text entry was low.

GazeLink also includes many personalizable features to accommodate different users and conditions. When the user or lighting condition changes, a quick re-calibration can drastically improve accuracy and consistency. Users can also adjust the sensitivity of eye gesture recognition. Finally, GazeLink provides three selectable text-entry methods (letter-by-letter, eye-gesture keyboard only, eye-gesture keyboard with LLMs) depending on the user's need and internet access.

To download the GazeLink mobile application or watch a video demonstration, see section 9 for hyperlinks.

DOI: 10.54254/2755-2721/88/20241639

8.1. Errors and Limitations

One possible error is that the usability studies are conducted in varying lighting environments, which may cause the data for the eye gesture recognition accuracy to vary. Although calibration allows GazeLink to adapt to different lighting, it would be more insightful to analyze how variables like brightness or contrast level impact recognition accuracy.

One limitation identified during the usability study and user feedback is that, for a few participants, one particular eye gesture may be hard to recognize, preventing the participant from performing certain operations. A solution is to verify the validity of eye gestures during calibration to ensure that no eye gesture is unrecognizable or too similar to another eye gesture. Enlarging the eye frames and adjusting sensitivity may also resolve the issue.

8.2. Future Work

Although this research only tested with able-bodied participants, it is necessary to evaluate GazeLink's baseline capabilities before recruiting PALS for more experiments. Since usability studies demonstrate GazeLink's robustness, the next stage of GazeLink's development is to collaborate with ALS research teams or hospitals and conduct system testing with PALS after obtaining authorization from the government. We are already in contact with numerous ALS associations. These experiments will reinforce GazeLink's validity to help underserved PALS globally communicate.

The AAC crowdsourced dataset used to train text generation LLMs may not fully encompass the verbal requirements of all users since every individual's speaking habits vary. For personalization, further studies can collect participants' previous conversations and train a personalized LLM that mimics their tone and style. We can also extract the user voice through ML models and incorporate it into the text-to-speech service for more realism in conversations. We can test the features in longitudinal studies, which are difficult to conduct but extremely valuable in examining the long-term learning rate and effect of GazeLink on PALS.

9. Conclusion

Currently, advanced systems for ALS patients are overly expensive and difficult to access, while low-tech systems are inefficient and require human assistance. In addition, both solutions include specialized hardware that is difficult to deliver, carry, and fix. To mitigate the limitations, this research proposes GazeLink, a low-cost multi-language mobile application that helps ALS patients to communicate efficiently and independently. Compared to existing products, GazeLink is distributed online, costs approximately less than $0.1 per month, and only requires a standard smartphone to function. With the device's front-end webcam, the system can robustly detect six eye gestures in real-time with a 94% accuracy on average and convert them to inputs on an eye-gesture keyboard. Computational evaluations demonstrate that LLMs used in GazeLink can increase baseline text-entry rate by 81% while maintaining more than 90% semantic accuracy. Usability studies and questionnaires affirm the system's learnability and low mental load, rated 2.7 (very low) out of 10 (very high) on average. With only 20 minutes of training, most first-time users can independently type sentences at 15.1 words per minute on average, which is 3.87x faster than the baseline and 7.2x faster than the low-tech solution E-tran. Overall, experiment results reinforce GazeLink's potential to notably assist the communication of PALS in their daily life. The next step is to directly evaluate the text-entry system on PALS and introduce GazeLink to more target users with internet services.

Acknowledgments

I express my sincere gratitude to Dr. Joseph Martin, Ms. Nacionales, and Professor Zachary Dodds for their constant support and inspiration throughout my project. I also greatly appreciate Mr. Boyin Yang of Cambridge University for counseling me via email and Zoom calls. Finally, a huge thank you to my family and friends for believing in my ability to complete this strenuous project.

DOI: 10.54254/2755-2721/88/20241639

References

[1] Kiernan, M. C., Vucic, S., Cheah, B. C., Turner, M. R., Eisen, A., Hardiman, O., ... & Zoing, M. C. (2011). Amyotrophic lateral sclerosis. The lancet, 377(9769), 942-955.

[2] Inform, N.(2023). Motor neurone disease(mnd). https://www.nhsinform.scotillnesses-and-conditions/brain-nerves-and-spinal-cord/motor-neurone-disease-mnd. Accessed: 2024-02-15] .

[3] Kang, B.-H., Kim, J.-I, Lim, Y-M., and Kim, K-K.(2018). Abnormal oculomotor functions in amyotrophic lateral sclerosis.Journal of Clinical Neurology, 14(4):464-471.

[4] Solutions, L.T.(2016). E-tran(alphabet). https://store.lowtechsolutions.org/e-tran-alphabet/. Accessed:2024-02-15] .

[5] Dynavox, T.(2024a). Td i-series. https://us.tobiidynavox.com/pages/td-i-series.Accessed: 2024-02-15] .

[6] Sarcar, S., Panwar, P, and Chakraborty, T.(2013). Eyek:anefficient dwell-free eye gaze-based text entry system. In Proceedings of the 1lth asia pacifc conference on computer human interaction, pages 215-220.

[7] Fan, M., Li, Z., and Li, F.M.(2020). Eyelid gestures on mobile devices for people with motor impairments. In Proceedings of the 22nd International ACM SIGACCESS Conference on Computers and Accessibility, pages 1-8.

[8] Cecotti, H., Meena, Y.K., and Prasad, G.(2018).A multimodal virtual keyboard using eye-tracking and hand gesture detection. In 2018 40th Annual international conference ofthe IEEE engineering in medicine and biology society (EMBC), pages 3330-3333.IEEE.

[9] Zhang, X., Kulkarni, H., and Morris, M.R.(2017). Smartphone-based gaze gesture communication for people with motor disabilities. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pages 2878-2889.

[10] Shen, J., Yang, B., Dudley, J.J., and Kristensson, P.0.(2022). Kwickchat: A multi-turn dialogue system for aac using context-aware sentence generation by bag-of-keywords. In 27th International Conference on Intelligent User Interfaces, pages 853-867.

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

[12] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I, Aleman, F.L., Almeida, D., Altensch midt, J., Altman, S., Anadkat, S., et al.(2023). Gpt-4 technical report. arXiv preprint arXiv:23 03.08774. [Amy and pALS, 2018] Amy and pALS (2018). Communication basics for people with als (pals). https://amyandpals.com/communication-solutions-gallery/.Accessed:2024-02 -15].

[13] Bradski, G.(2000).The opencv library. Dr.Dobb'sJournal:Software Tools for the Professional Programmer, 25(11):120-123.

[14] Li, J., Ray, S., Rajanna, V., and Hammond, T.(2021). Evaluating the performance of machine learning algorithms in gaze gesture recognition systems. IEEEaccess, 10:1020-1035.

[15] Vertanen, K.and Kristensson, P.0.(2011). The imagination of crowds: conversational aac language modeling using crowdsourcing and large data sources.In Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pages 700-711.

[16] Dinan, E., Logacheva, V., Malykh, V., Miller, A., Shuster, K., Urbanek, J., Kiela, D., Szlam, A., Serban, I, Lowe, R., et al.(2020). The second conversational intelligence challenge(convai2). In The NeurIPS'18 Competition:From Machine Learning to Intelligent Conversations, pages 187-208.Springer.

[17] Papineni, K., Roukos, S., Ward, T, and Zhu, W.-J.(2002). Bleu:a method for automatic evaluation of machine translation. In Proceedings of the 40th annualmeeting of the Association for Computational Linguistics, pages 311-318.

[18] Reimers, N. and Gurevych, I. (2019).Sentence-bert: Sentence embeddings using siamese bert-networks. arXivpreprint arXiv:1908.10084.

DOI: 10.54254/2755-2721/88/20241639

[19] HuggingFace (2024). Sentence similarity. https://huggingface.co/tasks/sentence-similarity. Accessed:2024-02-15].
[20] Hart, S.G. and Staveland, L.E.(1988). Development of nasa-tlx(taskload index): Results of empirical and theoretical research.In Advances in psychology, volume 52, pages 139-183.Elsevier.
[21] MacKenzie, I.S. and Soukoreff, R.W.(2003). Phrase sets for evaluating text entry techniques.In CHI03 extended abstracts on Human factors in computing systems, pages 754-755.
[22] Dynavox, T. (2024b). Td pilot.https://us.tobiidynavox.com/products/td-pilot.Accessed: 2024-02-15].