A Demonstration of a Tinkerable Augmentative and Alternative Communication Keyboard

Boyin Yang
University of Cambridge
Cambridge, United Kingdom
by266@cam.ac.uk

Per Ola Kristensson
University of Cambridge
Cambridge, United Kingdom
pok21@cam.ac.uk

ABSTRACT

Augmentative and alternative communication (AAC) provides communication assistance for nonspeaking individuals with motor disabilities. Literate AAC users typically rely on a keyboard interface coupled with a speech synthesizer to communicate with speaking partners. However, due to both the nature of a keyboard interface and motor disabilities, communication rates are frequently low, resulting in a communication gap between the AAC user and the speaking partner. As a result, AI methods that can predict words, phrases, and sentences can potentially reduce this gap. However, despite tremendous progress in such AI methods in recent years, most of these developments have not yet resulted in effective and efficient AAC systems. We here demonstrate a tinkerable AAC (TAAC) system that allows AAC users and professionals to explore a wide variety of AI methods and provide functions for manipulating parameters and user interface elements, as well as built-in methods for recording system and user behavior and analyzing such data at runtime.

CCS CONCEPTS

• **Human-centered computing** → **Accessibility design and evaluation methods**; **Accessibility systems and tools**.

KEYWORDS

accessibility, text entry, augmentative and alternative communication

ACM Reference Format:
Boyin Yang and Per Ola Kristensson. 2023. A Demonstration of a Tinkerable Augmentative and Alternative Communication Keyboard. In *28th International Conference on Intelligent User Interfaces (IUI '23 Companion), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3581754.3584153

1 INTRODUCTION

Nonspeaking individuals with motor disabilities, for example, people with motor neuron diseases (MND), such as amyotrophic lateral sclerosis (ALS) and stroke, frequently face severe difficulties in movement and communication. They rely on augmentative and alternative communication (AAC) devices to communicate with other people. For literate AAC users, such systems typically provide a keyboard interface that enables users to type and have the text read out using speech synthesis. Due to motor disabilities, the entry rates tend to be very low. Thus to bridge the communication gap between AAC users and their speaking partners, AAC systems need to rely on AI techniques to predict or generate AAC users' intended text.

Unlike utterance-based AAC systems where users can only select the preset sentences, keyboard-based AAC text entry systems enable users to type anything and have it spoken by the system.

In recent years, there have been many studies about enhancing the text entry rate for keyboard-based AAC systems, such as adopting context information for sentence retrieval [2], applying large language models (LLM) for language prediction with keywords [6], and utilizing LLMs for context-aware abbreviation expansion to generate sentences [1].

There are still questions on how to effectively exploit advanced AI methods for predicting and generating text in a way that is applicable to practical AAC. Despite the rapid development of natural language processing (NLP) technologies, interactive accessible controllers, and human-computer interaction theories, AAC technologies are still lagging behind the latest technology developments.

A primary reason for this state is that AAC is a highly interdisciplinary domain, requiring close cooperation between designers, machine learning scientists, linguists, AAC researchers, AAC users, and AAC professionals. This opens up a gap between *what is possible* on the technology side given the tremendous progress in AI in recent years versus *what we can realistically deploy and study* on the application side, given that we cannot expect AAC users, professionals, and designers to be experts in machine learning.

We hypothesize that a way to bridge this gap is to enable the AAC community to *tinker* with a prototype AAC system that has been bestowed with a range of word, phrase, and sentence prediction and generation methods [8]. This allows experimentation and reflection on how different AI methods may give rise to positive, or negative, qualities in AAC interaction.

To realize this, we have built a tinkerable AAC (TAAC) system that allows people to manipulate a wide range of parameters in an AAC system. Specifically, Table 1 illustrates the associations between the stakeholders' attributes and requirements, and the functions supported by the system. Aside from the motivation of enabling the AAC community to tinker with a flexible AAC system, a second motivation is that we feel a demonstration of this system is a way to engage the IUI community by demonstrating concrete means IUI research can impact AAC.

IUI '23 Companion, March 27–31, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0107-8/23/03.
https://doi.org/10.1145/3581754.3584153

	AAC User	AAC Researcher	Natural Language Processing (NLP) Researcher	Human-Computer Interaction (HCI) Researcher
Stakeholders' attributes and requirements	1. Unique motor capabilities 2. Developing physical conditions 3. Unique communication needs 4. Rely on text prediction functions to type	1. Have access to AAC users 2. Language threapists, psychologists, or other specialists who have deep understanding in the AAC domain and user needs 3. Use existing tools to conduct experiments	1. Investigate state-of-the-art NLP models for various downstream tasks 2. Develop dedicated language models for AAC purposes	Bridge stakeholders to enable the design and development of AAC systems for users, professionals and researchers
Relevant functions in the TAAC system	1. Text entry system with word and sentence prediction functions 2. Tinkerable keyboard layout 3. Tinkerable conversation modes with different interaction approaches	1. Multiple text input methods 2. Different text prediction algorithms for different scenarios	1. Tinkerable parameters for each prediction method. 2. The system is modualized for quick integration of up-to-date NLP models	Quanlitative analysis functions for automatically tracking and evaluating human performance

Table 1: The stakeholders' attributes and requirements and the relevant TAAC system's functions.

2 SYSTEM DESCRIPTION

The TAAC system consists of three main modules: (1) the soft keyboard and the text display that supports the core predictive text entry activities; (2) the tinker panel that allows users to adjust the user interface and manipulate parameters of the algorithms; and (3) the trace analysis panel that allows researchers to evaluate performance.

The TAAC system adopts a simple software architecture, the model-view-controller (MVC), to modularize the visual interface, the interaction, and the underpinning AI models and algorithms. This design allows developers to easily extend the system with, for example, additional AI algorithms, without changing a significant amount of code.

2.1 User Interface

As shown in Figure 1, the user interface of the TAAC system includes the keyboard, the word and sentence predictions, the input text display, and the menu bar.

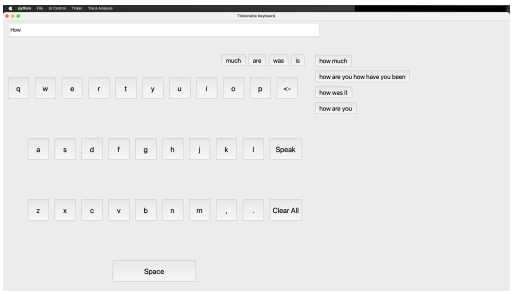

Figure 1: The user interface of the tinkerable AAC system.

The keyboard allows users to type or delete letters and punctuation. During typing, word and sentence predictions are updated based on the user's input and the user's chosen language models. Selecting a suggested word or sentence will result in the corresponding text being displayed in the text box. Thereafter, pressing the *Speak* button reads out the text using speech synthesis.

The menu bar is positioned at the top of the screen and is divided into four sections: *File*, *UI Control*, *Tinker*, and *Trace Analysis*. *File* allows the user to save the current settings and reload previous settings as readable text files that can thus be edited outside the system. *UI Control* allows the user to move keys on the keyboard around by dragging them, thus changing the keyboard's layout and geometry. Similarly, the position of word and sentence prediction slots can also be changed by dragging them. Further system tinkering features associated with the back-end are integrated into the *Tinker Panel* (see Figure 2a and Figure 2b), which can be accessed by choosing *Tinker* in the menu. Moreover, *Trace Analysis* supports quantitative analysis of performance by tracking system and user operations (see Figure 2c).

2.2 Tinker Panel

The *Tinker Panel* provides two main tabs for sentence prediction settings (see Figure 2a) and word prediction settings (see Figure 2b) in terms of the prediction display and the prediction method, respectively. For word predictions, the system enables adjusting the maximum prediction number between 1 and 4 and choosing the word display location either on a fixed position, or above the last pressed key on the keyboard. In addition, three predictive methods are supported, including BM25 [7], RoBERTa [3], and GPT-2 [4]. When typing letters, word predictions are automatically displayed on the keyboard area.

Similarly, for sentence prediction the system also enables users to alter the maximum prediction number between 1 and 4. The predictions are ordered by how likely they support the user's input. More than four prediction slots are usually not helpful. The system provides two types of sentence entry approaches: (1) conventional left-to-right text entry; and (2) keyword-based text entry. The system provides two generative sentence prediction methods. The first, GPT-2 [4], is used for left-to-right sentence prediction in the system, while KWickChat [6] is a keyword-based language model designed specifically for AAC use. Figure 2a shows a list of tuneable parameters for the KWickChat model. The semantic similarity-based (e.g. Sentence-BERT [5]) and text similarity-based sentence retrieval

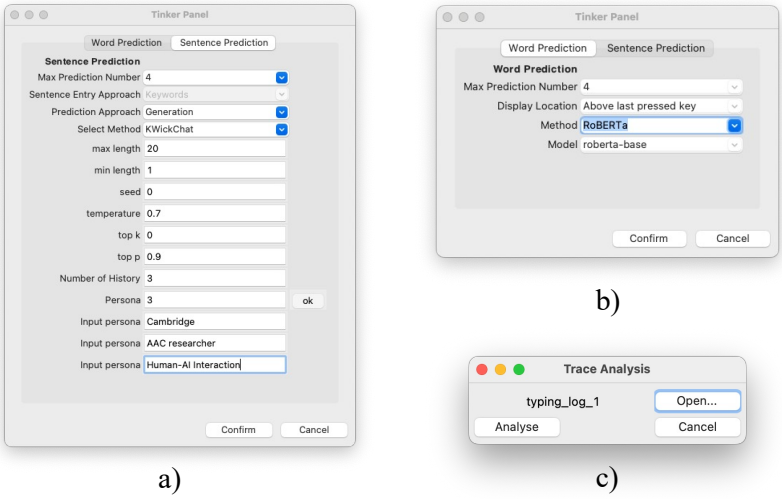

Figure 2: The tinker panel for sentence prediction (a) and word prediction (b), and the trace analysis panel (c).

methods (e.g. BM25 [7]) support both left-to-right text entry and a keyword-based text entry approaches.

2.3 Trace Analysis Panel

The *Trace Analysis* function (see Figure 2c) tracks system and user operations. This information can assist researchers in reasoning about the user experience under different settings and enables the generating of models to analyze interaction, text entry strategies, etc. As a complementary approach to conventional qualitative user experience evaluation methods, these analyses provide a quantitative angle for examining the user experience, thereby providing evidence for system design optimization. The tinkerability of this system facilitates AAC stakeholders, especially AAC researchers, to easily conduct experiments for studying and comparing different configurations.

3 CONCLUSION AND FUTURE WORK

This demonstration presents a tinkerable AAC text entry system. We hope the functions provided by the system help bridge the gap between machine learning and software specialists and AAC users and professionals by democratizing access to new AI methods and their various parameterizations. In addition, the tinkerable system introduces emergent and promising LLM technologies into the AAC domain, which opens up opportunities for the application of machine-generated text. Our future work is to continue to improve the tinkerable AAC system by integrating up-to-date LLM technologies, such as ChatGPT, into the system. Moreover, we plan to evaluate this system by conducting user studies with AAC users and professionals.

4 OPEN SCIENCE

The source code for the system is here: https://doi.org/10.17863/CAM.91650.

REFERENCES

[1] Shanqing Cai, Subhashini Venugopalan, Katrin Tomanek, Ajit Narayanan, Meredith Morris, and Michael Brenner. 2022. Context-Aware Abbreviation Expansion Using Large Language Models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Seattle, United States, 1261–1275. https://doi.org/10.18653/v1/2022.naacl-main.91

[2] Per Ola Kristensson, James Lilley, Rolf Black, and Annalu Waller. 2020. A Design Engineering Approach for Quantitatively Exploring Context-Aware Sentence Retrieval for Nonspeaking Individuals with Motor Disabilities. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–11. https://doi.org/10.1145/3313831.3376525

[3] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (2019). arXiv:1907.11692 http://arxiv.org/abs/1907.11692

[4] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.

[5] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. (2019). arXiv:1908.10084 http://arxiv.org/abs/1908.10084

[6] Junxiao Shen, Boyin Yang, John J Dudley, and Per Ola Kristensson. 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* (Helsinki, Finland) *(IUI '22)*. Association for Computing Machinery, New York, NY, USA, 853–867. https://doi.org/10.1145/3490099.3511145

[7] Andrew Trotman, Antti Puurula, and Blake Burgess. 2014. Improvements to BM25 and Language Models Examined. In *Proceedings of the 2014 Australasian Document Computing Symposium* (Melbourne, VIC, Australia) *(ADCS '14)*. Association for Computing Machinery, New York, NY, USA, 58–65. https://doi.org/10.1145/2682862.2682863

[8] Boyin Yang and Per Ola Kristensson. in press. Tinkerable Augmentative and Alternative Communication for Users and Researchers. In *Cambridge Workshop on Universal Access and Assistive Technology*. Springer.