Error Correction Using LLMs for Sentence Estimation from Ambiguous Inputs via Wearable Keyboards

Matsuri Iwasaki*, Masanobu Abe* and Sunao Hara*
* Okayama University, Okayama, Japan
E-mail: px9h3vtm@s.okayama-u.ac.jp, {m-abe,hara}@okayama-u.ac.jp

***Abstract*—This study develops a communication aid system that enables eyes-free input using finger movements. In particular, we focus on improving sentence estimation accuracy using an error correction method based on large language models (LLMs). The proposed system allows users to convey their thoughts through synthesized speech by employing a wearable keyboard that captures finger movement. The proposed system estimates intended sentences from finger movement sequences. However, because there is no one-to-one correspondence between fingers and characters, the mapping is ambiguous. To address this problem, the proposed system first applies neural machine translation (NMT) to estimate a sentence, followed by LLMs to correct estimation errors. Experiments were conducted to evaluate the accuracy of the NMT sentence estimation model and the performance of the LLMs-based error correction methods. The NMT model achieved symbol-level and sentence-level accuracies of 95.3% and 50.3%, respectively. In addition, the LLMs successfully corrected 52.5% of the sentences under the condition that the correct word was included in the n-best hypotheses generated by the NMT. These results suggest that error correction using LLMs has the potential to enhance the performance of sentence estimation from ambiguous inputs.**

I. Introduction

In recent years, the intelligibility and naturalness of text-to-speech (TTS) systems have significantly improved due to advances in deep neural networks. A promising application of TTS technology is to support communication for individuals with speech impairments, such as those with dysarthria or articulation disorders, by conveying their thoughts through synthesized speech [1], [2]. However, most TTS systems rely on a keyboard as the input device. Because users must look at the device while inputting text, the range of use is limited, particularly while walking. To overcome this limitation, we proposed an input method using a wearable keyboard that captures finger movements [3]. The proposed input method is based on the concept of touch typing. As shown in Figure 1, a mapping was created between the fingers and the letters for touch typing. Based on this finger-to-letter mapping, the system estimates the input sentence from finger movements.

A significant challenge with the proposed approach was the mapping ambiguity. For example, the number 1, representing the little finger of the left hand, covers "q," "a," "z," and "1." Therefore, numbers 1, 4, and 3 may represent the words "are" or "ate." Our previous work [3] attempted to address this ambiguity using neural machine translation (NMT) to translate finger movements into sentences. However, the sentence-level accuracy remained limited because NMT did not fully capture contextual information in some cases.

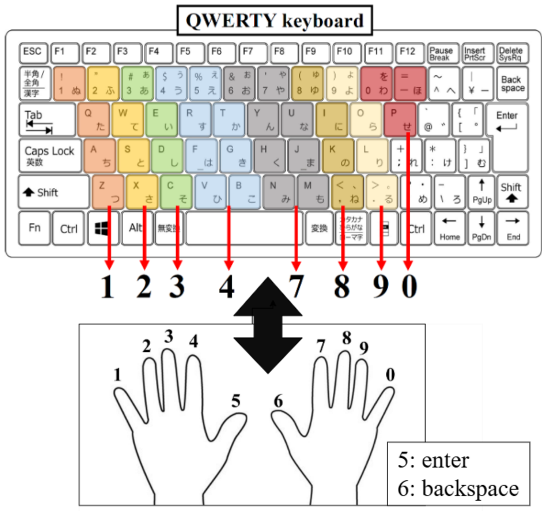

Fig. 1. Mapping between QWERTY keybord and fingers

To overcome this limitation and improve sentence-level accuracy, we propose an error correction system using large language models (LLMs). The proposed method first identifies errors in estimated sentences and then corrects them by selecting the most appropriate word from a candidate list extracted from the n-best hypotheses. Previous research has demonstrated that LLMs achieve high performance in error correction for automatic speech recognition based on contextual understanding [4], [5]. LLMs are expected to correct errors during finger movement-to-text conversion. This study aimed to demonstrate how LLMs can enhance sentence estimation through error correction. In addition, three prompt designs were evaluated to determine effective LLM-based error correction strategies.

II. Related works

Text entry is an essential function not only for desktops and personal computers but also for mobile devices. For text entry methods in ubiquitous computing, an eyes-free feature that allows users to input text without looking at the keyboard is required for several use cases. An eyes-free method enables users to maintain visual attention on other tasks during text input in mobile scenarios. Speech input is a promising candidate; however , it is not appropriate for every situation because of noisy environments and privacy concerns. Therefore, touch-based and eyes-free text entry methods have been widely used [6], [7]. The minimum effort required to learn to type is an essential requirement; thus, the QWERTY layout has been used in many cases. The greatest drawback of this method is that the eyes-free touch system is inaccurate on a touch panel [8]. Thus, errors in text entry occur in most cases.

In terms of input devices, we use a wearable QWERTY keyboard based on the "FingeRing" concept [9]. Examples of this method can be found in previous studies [10], [11]. A previous study [12] eliminated and replaced the keyboard with gloves that only sense the pressure of a depressed finger and not the location. The study attempted to estimate words using lexical-level matching and frequency-ranking estimation. Although this method outperformed other mobile keyboards in terms of productivity, the absence of semantic-level analysis limited its estimation accuracy.

To improve the sentence estimation accuracy, we previously employed a NMT-based approach to estimate the intended sentences from ambiguous input sequences [3]. Although this method demonstrated the potential of the system as a speech communication aid, it achieved only 50% sentence-level accuracy, indicating the need for further improvements. Therefore, in this study, we extend our previous approach by applying LLMs to perform error correction.

III. Proposed Method

We propose a system comprising NMT-based sentence estimation and LLM-based error correction. Figure 2 shows an overview of the proposed system.

The proposed system employs wearable devices that detect finger movements. Users wear these devices on both hands and move their fingers as if typing. Each finger is assigned digits 0–9 (Figure 1), converting finger movements into a numeric sequence. The proposed system receives the numeric sequence as input and estimates the corresponding sequence of phonemes and accent symbols based on a predefined mapping between fingers and keyboard letters. For example, the numeric sequence 21412871318418723833275 corresponds to the letter sequence "watasihadaigakuseidesu (meaning "I am a college student" in English)." The proposed system is expected to estimate the prosodic representation as follows: ^ w a [t a sh i w a # d a [i g a] k u s e e d e s u $. Here, ^, [,], #, and $ indicate the start of a sentence, F0-rise, F0-fall, accent phrase boundary, and the end of a sentence, respectively.

A. Sentence Estimation Using Neural Machine Translation

The proposed system uses NMT to estimate phonemes and accent symbols from numeric sequences. As shown in Figure 1, the mapping between numbers and characters is ambiguous. For example, the number 1 can represent "q," "a," "z," or "1." Therefore, the sequence 1, 4, and 3 may correspond to either the word "are" or "ate." In addition, homophones must be considered. For example, the input "hashi" may refer to either "chopsticks" or "bridge," depending on the context. The NMT model is expected to resolve such ambiguities based on contextual information.

The NMT model is trained sentence by sentence. We used a sequence-to-sequence model with a stacking bidirectional long short-term memory (Bi-LSTM) for the encoder and an LSTM for the decoder. In the encoder, each character is first represented by a one-hot (1-of-K) vector and then compressed via an embedding layer. The obtained hidden vector is inputted into the Bi-LSTM. The hidden vector is inputted into the Bi-LSTM of the next layer and is saved as input to the attention mechanism [13]. Thus, learning the context and alignment of characters using the attention mechanism is essential because the proposed sentence estimation method has ambiguity in the transformation.

The reason for estimating phonemes and accent symbols is that these elements contain sufficient information to produce natural-sounding speech. One of their key features is the ability to capture accent differences that help distinguish homophones. For example, in Japanese, the word "hashi" can mean "chopsticks" (h a] sh i) or "bridge" (h a [sh i), which are distinguished by pitch accent. Because phoneme and accent symbols explicitly represent pitch accent, they enable the synthesized speech to reflect those differences.

In contrast, some homophones share the same accent pattern, such as "k a w a," which can mean either "river" or "leather." Because our final goal is speech synthesis, it is unnecessary to distinguish those words during the estimation phase. Therefore, phonemes and accent symbols help minimize ambiguity by focusing on important distinctions for speech synthesis.

B. Error Correction Using LLMs

The proposed system uses LLMs to correct estimation errors. The NMT model generates n-best hypotheses and selects the one with the highest score to be the final estimation. However, the sentence with the highest score is not always an accurate estimation. LLMs are employed to correct errors by reselecting phonemes and accent symbols from the list of hypotheses. LLMs are expected to correct errors based on the context of the sentence.

The error correction process comprises two steps. The first step is error detection using an LLM. In this step, the LLM receives a sequence of phonemes and accent symbols as input and determines whether the sentence contains any errors. If an error is detected, the index of the accent phrase in which the error occurs is also identified. An example is shown in Figure 2: the estimation sentence "^ f u [w a f u w a n a # a [s a b u r o $ (fluffy-looking morning baths)" is provided, and the LLM outputs that the second accent phrase "a [s a b u r o (morning baths)" contains an error.

The second step is error correction using LLMs. Two elements are provided to an LLM: (1) a sequence of phonemes and accent symbols in which the accent phrase containing an error is replaced with a `<space>` symbol, and (2) a

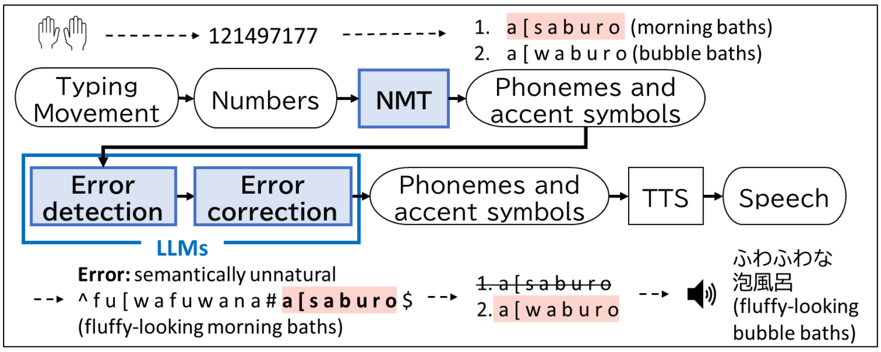

Fig. 2. Overview of proposed system

list of candidate hypotheses for the `<space>` position. The LLM then selects the most appropriate candidate to fill the `<space>`. For example, as shown in Figure 2, the input sequence "f u [w a f u w a n a # `<space>` $" and the hypotheses "1. a [s a b u r o (morning baths) and 2. a [w a b u r o (bubble baths)" are provided to an LLM, and it selects "a [w a b u r o (bubble baths)" as the most appropriate candidate.

IV. Experiment Data

To confirm the feasibility of the proposed method, experiments were conducted on Yahoo! Chiebukuro data (3rd edition): FY2023 distribution version [14] data were obtained. This corpus consists of 32 million Japanese sentences posted on the Yahoo! Chiebukuro bulletin board system. The corpus includes vocabulary on various topics, and most of the sentences in the corpus are casual registers in the form of question-and-answer style texts.

A parallel corpus was generated for NMT model training. The corpus comprises pairs of a sequence of phonemes and accent symbols and corresponding sequences of numbers. First, morphological analysis is performed to assign Roman character notations to correspond to Kanji characters. Second, a number is generated by referring to a table that maps QWERTY keyboards to respective fingers corresponding to the keys at the time of touch typing. Phonemes and accent symbols are generated using the grapheme-to-phoneme function in ESPnet2 [15].

V. Evaluation of Sentence Estimation Using NMT

A. Experimental conditions

The NMT estimation performance was evaluated based on the accuracy of the estimated sentences. The evaluation indices were the accuracies of symbols, accent phrases, and sentences. AccSymbol measures the percentage of estimated symbols that match the reference. AccPhrase evaluates the percentage of accurately estimated accent phrases, where each phrase is defined as a segment between "#" symbols. AccSentence refers to the percentage of estimated sentences that completely match the reference. The references are the phoneme sequences and accent symbols generated by the grapheme-to-phoneme function. Table I shows the configuration and training conditions of the NMT model.

TABLE I
Training Conditions

Model	RNN-based encoder-decoder model
(Encoder)	-
RNN	Bi-LSTM
#layers	5 layers
#units	512 units in each layer
(Decoder)	-
RNN	LSTM
#layers	5 layers
#units of 1st layer	1024 $\rightarrow$ 512
#units of 2nd–5th layers	512 $\rightarrow$ 512
(Linear)	-
#units	512 $\rightarrow$ number of symbol types
activation function	Softmax
mini-batch	128 sentences
optimizer	Adam
loss function	Log-Softmax cross entropy
training data	32,143,621 sentences
validation data	2,000 sentences
test data	2,000 sentences

B. Evaluation Results

Table II presents the experimental results. In the table,"Phoneme & Accent" indicates the accuracy of all estimated phonemes and accent symbols; "Phoneme & #" indicates the accuracy of phoneme symbols and accent clause boundaries, "#." For this calculation, any accent symbols, except the "#," are removed from the sentences. "Phoneme" indicates the accuracy of phoneme symbols, calculated after removing all accent symbols from the estimated sentences.

For phoneme and accent symbols, the accuracies of the symbol, accent phrase, and sentence were 95.3%, 82.6%, and 50.3%, respectively. Although the symbol- and accent phrase-level accuracies are high, only approximately half of the estimated sentences are accurate. The sentence-level accuracy is insufficient because even a single symbol error can change the intended meaning.

In addition, accuracies of Phoneme are higher than those of Phoneme & Accent. This indicates that estimating accent symbols is more challenging than estimating phonemes. One possible reason for this is that the input numeric sequences do not contain explicit information about accent symbols, and the estimation relies on contextual information.

TABLE II
ACCURACY OF NMT MODELS

	AccSymbol	AccPhrase	AccSentence
Phoneme & Accent	95.3%	82.6%	50.3%
Phoneme & #	96.1%	85.7%	52.9%
Phoneme	96.7%	-	55.4%

TABLE III
ESTIMATED SENTENCES FOR TEST DATA.
WORDS WRITTEN IN RED ITALICS INDICATE ERRORS

answer 1	^ r a] b e r u # sh i] i r u n i # k i] r e e n i #···
estimated 1	^ *t a [b e] r u* # sh i] i r u n i # k i] r e e n i #···
answer 2	··· h a [j i] m e t e # u [N t e N m e] N ky o o # t o] cl t e # n o [cl t a # n o [g a _ g e [N ts u k i b a] i k u d e s u k a r a _ r o [j o o o # ···
estimated 2	··· h a [j i] m e t e # u [N t e N m e] N ky o o # t o] cl t e # *h o] o o # h a [sh i] g u d e s u* $

C. Analysis of Estimated Errors

Table III presents the estimated sentences for the test data. Words written in red italics indicate errors. On the test dataset, two types of errors were found.

The first type of error is due to word conflicts. As mentioned in Section III, a number sequence can correspond to more than one word. For example, in the estimated 1, "tabete (eating)" is estimated from 414347 although "rabere (label)" is correct. Although 414347 corresponds to "taberu," the meaning "eating" does not fit the context ("I would like to apply a (eating / label) sticker perfectly"). Although the NMT model correctly learns the mapping between numbers and characters, it has limitations in capturing the context and naturalness of sentences.

The second type of error is missing input information. For example, in estimated sentence 2, the symbols after "t o] cl t e" do not appear in the estimated sentence, resulting in missing information. One possible cause is that the attention mechanism in the NMT model skipped over the missing part and focused on the following characters. This type of error occurs in sentences that are composed of multiple phrases.

VI. EVALUATION OF ERROR DETECTION USING LLMS

A. Experimental conditions

The performance of LLMs in detecting errors was evaluated. The performance was evaluated by comparing the model predictions (error or no-error) with the ground truth labels. The evaluation indices were accuracy, precision, and recall. In the evaluation, the label "error" was treated as positive when calculating precision and recall.

Figure 3 shows the prompt used in the evaluation. The original prompt was written in Japanese, which we translated into English for this paper. An LLM received phoneme and accent symbol sequences estimated by the NMT model and determined whether each sentence contained an error. If the sentences had an error, the index of the accent phrase with the error was also identified.

We used GPT-4.1 [16] for this evaluation. The test dataset consisted of 2,000 sentences estimated by NMT during the evaluation in Section V. We generated input sequences for the LLM from these by removing all accent symbols except for the "#." Of the 2,000 sentences, 942 and 1,058 contained and did not contain errors, respectively.

##Step
1.**Hiragana Conversion**:Convert the input phoneme sequence into a sentence written in hiragana, following the rules for Step 1. The resulting sentence is referred to as hiragana.
2. **Kanji Conversion** : Convert the hiragana sentence into a kanji-mixed sentence. The resulting sentence is referred to as kanakanji.
3. **Naturalness Judgment**: Determine whether a kanakanji sentence contains any semantically unnatural parts. If the sentence is entirely natural, output "1." If there are unnatural parts, output "0" and provide the index of the phrase that contains the unnatural part.

Fig. 3. Prompt for error detection

TABLE IV
RESULT OF ERROR DETECTION (SENTENCES)

		GPT	
		error	no-error
Ground Truth	error	727	215
	no-error	310	748

Accuracy	73.8%
Precision	70.1%
Recall	77.2%

B. Evaluation Results

Table IV presents the experimental results. Of the 942 sentences containing errors, 727 were accurately detected. Assuming that all detected errors were appropriately corrected, the sentence accuracy would increase by 36.4 points compared with the original NMT estimation. The precision and recall are 70.1% and 77.2%, respectively. Improving recall is necessary to increase sentence-level accuracy.

C. Analysis of Estimated Errors

Table V shows the estimated sentences for the test data. Words written in red italics indicate errors. These errors are categorized into two types.

The first type of error occurs when a sentence is semantically natural despite containing an error. For example, in error 1, the error sentence means "Why do you sell it?" In contrast, the ground truth is "Why do you oppose it?" Because the LLM is instructed to detect errors based on naturalness, this type of error is not detected. Incorporating broader contextual information is considered effective in addressing such cases.

The second type of error occurs when the LLM fails to recognize the unnaturalness of the erroneous word. For example, although a phrase such as "ky o] r i n o # b o [o g e N (abusive language of distance)" is not meaningful, the LLM mistakenly identifies it as natural.

VII. EVALUATION OF ERROR CORRECTION USING LLMS

A. Experimental conditions

We evaluated three different prompts for error correction. In all prompts, the inputs consisted of (1) a phoneme and accent symbol sequence in which the accent phrase containing the error was replaced with a `<space>` symbol, and (2) a list of candidate hypotheses for that `<space>`. The output was a corrected sentence. As the evaluation metric, we used symbol, accent phrase, and sentence accuracy.

TABLE V
EXAMPLES OF FAILED ERROR DETECTION.
WORDS WRITTEN IN RED ITALICS INDICATE ERRORS

answer 1	^ n a] z e # h a [N t a i # sh i [t e # i [r u # ···
error 1	^ n a] z e # *h a [N b a i* # sh i [t e # i [r u # ···
answer 2	^ ky o] g i n o # t o] o b e N o # sh i [t a # ···
error 2	^ *ky o] r i n o # b o [o g e N* o # sh i [t a # ···

TABLE VI
STEPS INCLUDED IN EACH PROMPT FOR ERROR CORRECTION

Steps	Prompt 1	Prompt 2	Prompt 3
Blank Filling			○
Japanese Conversion			
–sentence after blank filling			○
–Estimated sentence and hypotheses	○	○	
Context Understanding		○	○
Hypothesis Selection	○	○	○
Confirmation		○	

Blank Filling: Fill each hypothesis into the <space> in the input sentence .
Japanese conversion: Convert phonemes into Japanese.
Context Understanding: Understand the context surrounding the <space >.
Hypothesis Selection: Compare all hypotheses and select the hypothesis that is most natural in context.
Confirmation: Confirm whether the selected hypothesis is natural.

Table VI illustrates the prompts used in the experiments. Circles indicate that the step is included in a prompt.

- Prompt 1 (conversion_insertion): The estimated sentence and each candidate hypothesis are independently translated into Japanese. The model then selects the most appropriate hypothesis based on naturalness.
- Prompt 2 (conversion_context): The model is explicitly instructed to understand the context of the estimated sentence before performing hypothesis selection, following the Japanese conversion steps.
- Prompt 3 (insertion_conversion): Each hypothesis is inserted into the `<space>` position in the estimated sentence, and each resulting sentence is translated into Japanese. The model compares all resulting sentences and selects the most appropriate one.

We used two experimental datasets for the evaluation: the correctable and LLM-detected error datasets.

The correctable error dataset consisted of 341 sentences with estimation errors, where the correct word was included in the candidate hypothesis list. We used this dataset to evaluate the performance of the LLM-based error correction model under ideal conditions under which corrections are theoretically possible. To construct this dataset, we extracted sentences with errors from the NMT estimation results (Section V). We then identified the specific phrase containing the error using dynamic programming matching between the estimated and reference sentences. Finally, we replaced the identified erroneous phrase with a `<space>` symbol to mark the location for correction.

The LLM-detected error dataset consisted of 1,037 sentences that were detected as estimation errors by the LLM in Section VI. We used this dataset to evaluate the correction performance under realistic conditions. This included cases in which the original NMT output was correct but was flagged as an error by the LLM, leading to unnecessary correction. The phrases identified as errors by the LLM were replaced with a `<space>` symbol.

We used beam search to extract candidate hypotheses from the top 10 sentences generated by the NMT model for both datasets. For the correction stage, we used GPT-4.1 [16] with a temperature of 1.0 and a frequency penalty of 0.

B. Evaluation Results

Tables VII and VIII show the accuracies after error correction for the correctable and LLM-detected error datasets over the three different prompts.

1) Defferece between prompts: In addition, the results demonstrate that the prompt design affects error correction performance. As shown in Table VII, Prompt 1 (conversion_insertion) had a significantly lower sentence-level accuracy of 27.3% than Prompts 2 and 3. This indicates that instructing LLMs to understand the context of sentences is highly effective in improving accuracy. The accuracy of the symbol and accent phrase of Prompt 1 is lower than that of the original NMT output. One possible cause of this degradation is that the LLM occasionally produces outputs that do not adhere to the phoneme formatting rules. For example, rather than outputting "sh i" as a single phoneme unit, the model may incorrectly produce "s h i" by separating each character, resulting in symbol mismatches.

2) Effectiveness of error correction: Table VII shows that Prompts 2 and 3 correct 51.3% and 52.5% of the sentences, respectively. This result indicates that the LLM-based error correction is highly effective under ideal conditions, where the correct word is guaranteed to be within the candidate list and the error phrase is correctly identified.

The performance on the LLM-detected error dataset (Table VIII) reveals a significant decrease in accuracy under realistic conditions. The sentence accuracies for Prompts 2 and 3 are 25.8% and 25.0%, respectively, which are lower than the NMT estimation baseline (29.9%). Our analysis shows that overcorrection is a major reason for this degradation. Specifically, 310 sentences in the dataset were incorrectly identified as erroneous even though their original NMT output was correct. During the error correction phase, 114 of these 310 sentences were unnecessarily changed. The LLM judged these sentences as unnatural during the error detection phase. Consequently, the model is unlikely to select the originally correct but unnatural candidate in the subsequent error correction phase. This highlights that improving the accuracy of the error detection component is a critical challenge, because it affects the overall performance of the error correction system.

Furthermore, the accuracy improvement is also limited by the quality of the candidate list. In the LLM-detected error dataset, only 246 of 727 sentences contained the correct candidate among the top 10 hypotheses. This suggests that the current candidate list lacks sufficient variety. Prompt 3 successfully corrected 66 of the 246 correctable sentences, demonstrating its potential when the correct answer is avail-

TABLE VII
ACCURACY AFTER ERROR CORRECTION (CORRECTABLE ERROR DATASET)

prompt	AccSymbol	AccPhrase	AccSentence
NMT estimation	94.5%	82.2%	0.0%
1: conversion_insertion	90.3%	80.7%	27.3%
2: conversion_context	96.9%	89.5%	51.3%
3: insertion_conversion	96.2%	88.4%	52.5%

TABLE VIII
ACCURACY AFTER ERROR CORRECTION
(LLM-DETECTED ERROR DATASET)

prompt	AccSymbol	AccPhrase	AccSentence
NMT estimation	93.7%	80.2%	29.9%
1: conversion_insertion	93.2%	79.9%	24.4%
2: conversion_context	93.3%	80.3%	25.8%
3: insertion_conversion	93.4%	80.3%	25.0%

able. Therefore, to bridge the gap between ideal and realistic performance, it is crucial to improve both the accuracy of the error detection module to avoid overcorrection and the quality and diversity of the candidate hypotheses to ensure that the correct candidate is available.

C. Analysis of Miscorrections

Correction failures are classified into two types.

The first type of error occurs when the hypothesis list contains multiple contextually natural hypotheses. For example, answer 1 in Table V is not corrected because both of "h a [N t a i (oppose)" and "h a [N b a i (sell)" are natural in the given context. To address this type of error, the use of broader contextual information is considered effective.

The second type of error occurs when the estimation of accent phrase boundaries is incorrect. Sentences are written in Japanese without spaces between words; therefore, accent boundaries play a critical role in segmenting sentences into individual words. Errors in estimating these boundaries can lead to sentence misinterpretations. Because the current prompt does not include instructions for handling such errors, these misinterpretations can result in inappropriate words being selected during correction.

VIII. CONCLUSION

In this study, we proposed using an LLM-based error correction method for a communication aid system based on finger movement input. The following conclusions are drawn from the experimental results: (1) The NMT model estimates 95.3% of the symbols and 50.3% of the sentences. (2) The proposed error correction method demonstrated a strong potential, successfully correcting 52.5% of the sentences under ideal conditions. However, a significant gap was observed between this ideal scenario and realistic conditions, primarily due to the challenges in error detection and the limited variety of candidate hypotheses.

Considering real conversational situations, using previous utterances for error correction is a promising approach for future work. Because the vocabulary in the utterance may be related to that of previous utterances, NMT will naturally introduce such an assumption. In addition, we will conduct evaluations with wearable devices in real-world scenarios.

REFERENCES

[1] J. Yamagishi, C. Veaux, S. King, *et al.*, "Speech synthesis technologies for individuals with vocal disabilities: Voice banking and reconstruction," *Acoustical Science and Technology*, vol. 33, no. 1, pp. 1–5, 2012.

[2] K. Matsubara, T. Okamoto, R. Takashima, *et al.*, "High-intelligibility speech synthesis for dysarthric speakers with lpcnet-based tts and cyclevae-based vc," in *ICASSP 2021*, IEEE, 2021, pp. 7058–7062.

[3] M. Iwasaki, S. Hara, and M. Abe, "Speech synthesis using ambiguous inputs from wearable keyboards," in *APSIPA ASC 2023*, 2023, pp. 1172–1178.

[4] C.-H. H. Yang, Y. Gu, Y.-C. Liu, *et al.*, "Generative speech recognition error correction with large language models and task-activating prompting," in *ASRU 2023*, 2023, pp. 1–8.

[5] R. Ma, M. J. Gales, K. M. Knill, *et al.*, "N-best t5: Robust asr error correction using multiple input hypotheses and constrained decoding space," in *INTERSPEECH*, 2023.

[6] L. Yiqin, Y. Chun, Y. Xin, *et al.*, "Blindtype: Eyes-free text entry on handheld touchpad by leveraging thumb's muscle memory," *IMWUT 2017*, 2017.

[7] Z. Suwen, L. Tianyao, B. Xiaojun, *et al.*, "Typing on an invisible keyboard," in *CHI 2018*, 2018, pp. 1–13.

[8] Y. Wang, C. Yu, J. Liu, *et al.*, "Understanding performance of eyes-free, absolute position control on touchable mobile phones," in *MobileHCI 2013*, 2013, pp. 79–88.

[9] M. Fukumoto and Y. Tonomura, "Body Coupled FingeRing: Wireless wearable keyboard," in *CHI 1997*, ACM Press, 1997, pp. 147–154.

[10] R. Rosenberg and M. Slater, "The chording glove: A glove-based text input device," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 29, no. 2, pp. 186–191, 1999.

[11] Tap, *Introducing tap*. [Online]. Available: https://www.tapwithus.com/introducing-tap/ (visited on 06/29/2025).

[12] M. Goldstein, R. Book, G. Alsiö, *et al.*, "Non-keyboard qwerty touch typing: A portable input interface for the mobile user," in *CHI 1999*, 1999.

[13] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *EMNLP 2015*, 2015, pp. 1412–1421.

[14] Yahoo Japan Corporation, *Yahoo! chiebukuro data (3rd edition)*, Informatics Research Data Repository, National Institute of Informatics. (dataset), 2023. [Online]. Available: http://id.nii.ac.jp/1457/00004881/.

[15] T. Hayashi *et al.*, "Espnet-tts: Unified, reproducible, and integratable open source end-to-end text-to-speech toolkit," in *ICASSP 2020*, 2020, pp. 7654–7658.

[16] OpenAI, *Gpt-4.1 (2025-04-14 version)*, https://platform.openai.com/docs/models/gpt-4.1, 2024.