Tutoring System for Smartphone Text Input for Older Adults using Statistical Stumble Detection

Toshiyuki Hagiya

ABSTRACT

Many older adults are interested in smartphones which offer opportunities to improve their lives. However, most of them encounter difficulty in text input, which is essential for various applications. This difficulty makes some older adults give up using a smartphone and go back to their old feature phone. This creates a digital divide even between those of the same generation. Therefore, providing appropriate support for text input is very important.

The goal of this study is to propose and develop a tutoring system for text input that can perform the role of a human tutor who suggests what the next operation should be, exactly when necessary. The tutoring system automatically detects input stumbles and provides instructions that help users to resolve the stumbles by themselves.

This study addresses three scenarios of the tutoring system. The first is a typing practice application with which users input presented texts. In the second scenario, the system is extended to allow free text input for daily use. Finally, the modality of input is extended to include voice input.

A common approach is adopted in these three scenarios. First, a user study is conducted to collect input data of older adults, which are used to identify and classify input stumbles. Then, statistical machine learning is conducted to build an automatic stumble detector, and instructions for detected stumbles are designed to develop a tutoring system. Finally, an evaluation experiment is conducted with older adults.

Chapter 1 introduces the background, the problems and the approaches presented in the thesis.

Chapter 2 reviews text input methods and support technologies that enhance accessibility to older adults.

Chapter 3 describes an assistive typing practice application as the initial step in this study. First, the problems which novice older adults encounter when inputting

presented sentences using a software keyboard are investigated. Next, the acceptability of the instructions provided for the stumbles is confirmed by a Wizard-of-Oz (WoZ) manner. Then, the assistive typing application is implemented. Ten classes of stumbles are identified by a decision tree with precision of 98% (F-measure 94%). An evaluation experiment with novice older adults showed that by using the system, the typing speed measured by the characters per minute (CPM) is increased by 17.2%, and the stumble ratio with respect to chances for stumbles is reduced by 59.1% from the users' initial rate.

Chapter 4 presents Typing Tutor, an individualized tutoring system that allows free text entry. First, a user study is conducted to clarify the problems when inputting free text, and 23 stumble classes are defined. In addition, the relationship between the input stumbles and the user's skill level is analyzed. Second, several statistical machine learning methods are compared. As a result, SVM achieves the best performance on the input stumble detection (precision of 90%; F-measurer of 70%) and the skill level classification (accuracy of 88%), which are used to build the Typing Tutor. The effect of the Typing Tutor was evaluated in two-week field trials by older adults. The result demonstrated that the system significantly reduced the number of stumbles and improved the text input speed, especially in the initial stage of usage. Furthermore, the applicability of the statistical input detection to other keyboards and languages is discussed with the QWERTY layout for Japanese and English.

Chapter 5 presents a tutoring system for voice input. With the same approach as the software keyboard input, classification of input stumbles and construction of a tutoring system are conducted. In this scenario, eleven classes are identified by a decision tree with precision of 83% (F-measure of 73%). In an evaluation experiment, the number of input stumbles and the sentence completion time were significantly reduced when subjects used the tutoring system compared to those who did not use the system.

Chapter 6 concludes the thesis and provides a brief outlook of future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

In recent years, smartphones have gradually come into widespread not only with young people but also with older adults [71]. Smartphones offer new opportunities to improve the lives of older adults by providing various creative activities and the ability to communicate with a wider circle [59]. Although these individuals have an interest in smartphones and would like to learn about them [6], those who have never used one may face difficulties. One reason for this is that the various forms of decline in functionality that people experience as they age may hinder them from interacting with smartphones which are not adapted to their needs [103]. Some researchers have pointed out that older adults find computer-based technologies especially difficult to learn because they became adults before computers were widely available [17]. For these reasons, some of them give up using a smartphone and go back to using their old feature phone because they cannot get used to the operation. This creates a digital divide even between those of the same generations. Therefore, providing appropriate support for text input is very important.

This study investigates how older adults learn to interact with a smartphone and provides an assistive system. In particular, text input is important to make use of many functions and applications such as web search, text messaging, and social networking service (SNS). The operation to input text is done by pressing keys or talking into the microphone. As trivial as it might seem, several studies have shown how such input techniques can pose problems for users who are not familiar with the new technology, as well as for those who have some physical or cognitive impairment [95]. An ideal approach to achieving mastery is to receive assistance from a human tutor whenever required, but this is not always possible. This thesis focuses on supporting text input for a smartphone.

1.2 Conventional Approaches and Problems

1.2.1 Input Methods for Smartphones

First, conventional approaches and problems related to text input are reviewed. Smartphones have two several kinds of text input methods: software keyboard and voice input. In both input methods, two approaches are mainly taken to improve usability. The first one is to improve input function directly such as reducing misrecognition and displaying appropriate candidates. The second one is cognitive support such as providing an explanation of the operation method.

Regarding text entry using a software keyboard, many researchers have adopted the former approach. They have tackled the issues associated with making text entry easier using a number of strategies, e.g., changing the layout, adjusting the key target areas, and presenting appropriate suggestions [8, 29, 34, 91, 101]. These aids are effective for users who are already accustomed to smartphone operation. However, smartphone novices, especially older adults, tend to have cognitive problems, such as forgetting and losing how/what to type next [70]. They, therefore, need patient support during the initial stages of use. According to a study [59], older adults tend to prefer an instruction manual to trial-and-error. One study [55] hypothesized that a tutoring system might reduce time and effort when learning a new type of text entry method and proposed a guideline policy. However, practical tutoring systems for text entry have not been developed.

On the other hand, automatic speech recognition (ASR) technology has been studied over the past decade [46, 60, 65, 68, 77]. Over the last few years, the accuracy of speech recognition has dramatically improved through the use of Deep Neural Networks [41, 78]. Hence, voice input has become an important text input option for those who are unfamiliar with touch input. However, to input what they intended, users need to get accustomed to ASR interface. ASR also needs to be used along with the software keyboards when precisely input sentences are required, e.g., when dealing with documents. Therefore, it is essential to learn to input or correct by keyboard. However, a tutoring system for voice input also has not been developed as well as the text entry.

1.2.2 Support Method for Novice Older Adults

Attempts to make new technology accessible to older adults can be roughly divided into three approaches: design for older adults, adaptation to older adults, and assistance for older adults [102].

The first approach, design for older adults, has been mentioned as universal usability [86]. Universal usability refers to the design of information and communications products and services that are usable by anyone. Some researchers proposed guidelines for older adults based on universal usability [84]. This can be helpful, but older adults' needs are varied and change over time, making it unlikely to find a universal usability that fits all of their needs [32].

The second approach, adaptation to older adults, has explored automatic user interface generation [27], adaptation to different user skills [44], and adaptation to changing user contexts [48] by capturing, measuring, and modeling the abilities of diverse users. However, there is a trade-off between the ease-of-use and learnability because users need to learn new user interfaces that change according to skills as they improve their skill level [74]. The multi-layered user interface can reduce learnability, especially for beginners or older adults [58].

The third approach, assistance for older adults to enable them to perform operations by themselves, is further classified into tutorials and contextual assistance. In tutorials by video or animation, users have to stop operating their devices to watch the content. On the other hand, contextual assistance allows users to receive assistance in the actual interface they are interacting with. Rogers *et al.* [83] found that interactive tutorials were the most effective approach in the learning process for older adults. Researchers have explored a range of tutorial systems that are presented in the context of the application [33, 49, 94]. These systems track the operational flow or capture the screenshot, and detect the context by pattern matching approaches or vision-based analysis, comparing what is occurring with an already registered pattern. These methods to detect the context are difficult to apply to text input support because text input has a high degree of freedom.

In text input, it is most important not to disturb the operation because making sentence has two processes, thinking to create a sentence and inputting the text. Therefore, the contextual assistance of the third approach is used to provide instruction.

1.3 Challenges and Approaches

From the above viewpoints, this study focuses on cognitive support to mastery of both software keyboard and voice input for older adults. The goal of this study is to propose and develop a tutoring system for text input that can perform the role of a human tutor who indicates the next action. Specifically, the tutoring system detects input stumbles and provides instructions that help users resolve such stumbles by themselves [35, 36]. The word “input stumble” is defined as an event when a user makes a mistake or does not know how/what to type next. The tutoring system uses statistical approaches to detect input stumbles in text input, and displays the corresponding instructions contextually. The target user group is older adults who have never used a smartphone but have owned a feature phone because most older adults in developed countries own a mobile phone.

There are three challenges. The first is to clarify the common difficulties that novice older adults encounter in using software keyboard and voice input. The second is to develop a method of detecting input stumbles for both text input methods. The third is to develop a tutoring system and evaluate the acceptability and the effect of the system.

This study has three scenarios as shown in Figure 1.1. The first scenario is the support of a typing practice application with a software keyboard. The aim is to confirm the acceptability of the tutoring system for text input because the prediction of input stumbles is not so difficult because the characters to be input are limited by focusing on the typing application. In the second scenario, the target is expanded to free text input for daily use. Assuming daily use, individualization is an important factor. An individualized tutoring system for free text input named Typing Tutor is proposed. In the final scenario, the modality of input is expanded to include voice input. In this study, a common approach is adopted in these three scenarios as shown in Figure 1.2. First, a user study is conducted to collect input data of older adults, which are used to identify and classify input stumbles. Then, statistical machine learning is conducted to build an automatic stumble detector, and instructions for detected stumbles are designed to develop a tutoring system. Finally, an evaluation experiment is conducted with older adults.

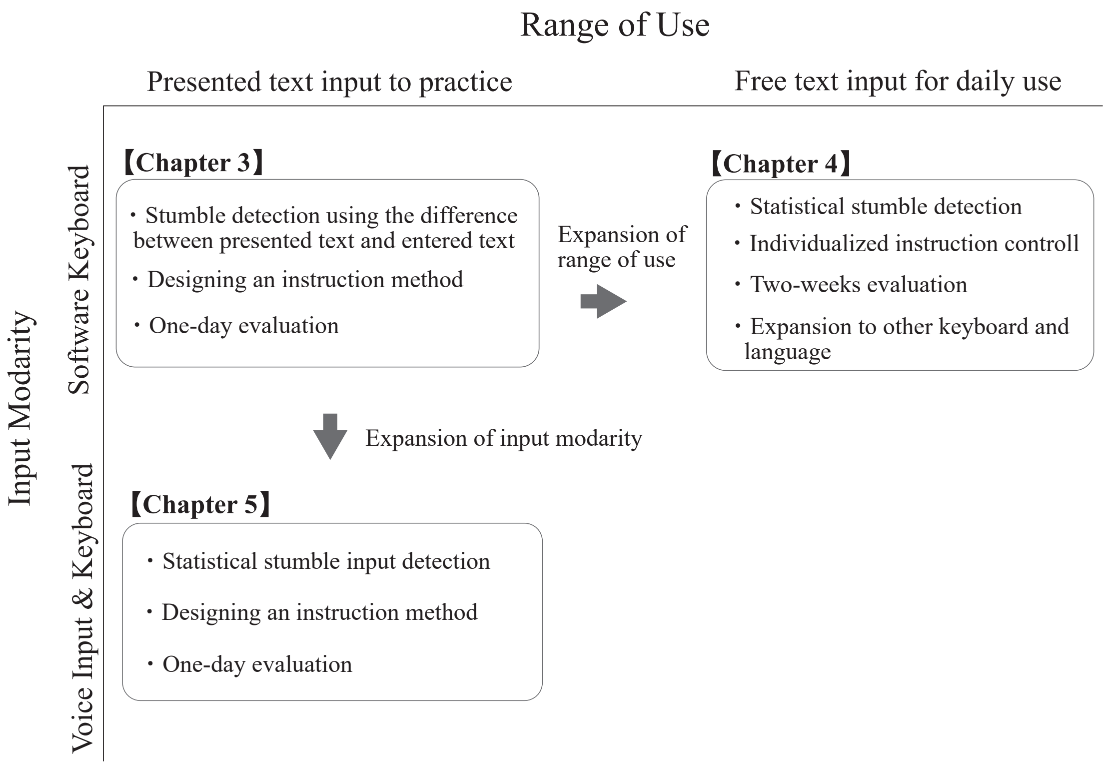


Figure 1.1: Overview of studies investigated in this thesis

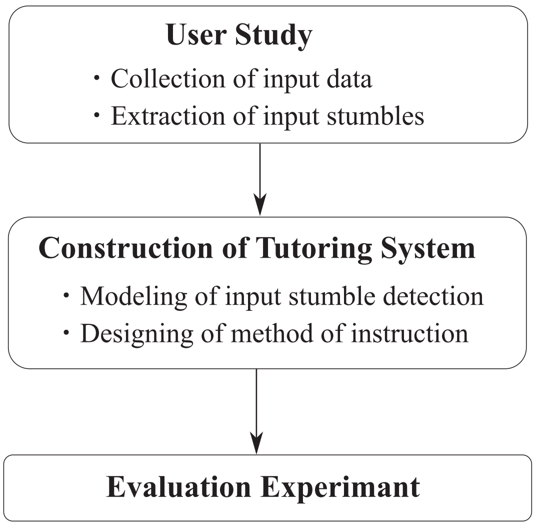


Figure 1.2: Common approach of this studies

1.4 Organization of this Thesis

The rest of this thesis is organized as follows. The next chapter reviews text input methods and related work that makes new technologies accessible to older adults.

Chapter 3 presents an assistive typing practice application. First, problems in which novice older adults encounter when inputting presented sentences using a software keyboard are investigated. Next, the acceptability of presenting instructions for text entry is confirmed by Wizard-of-Oz (WoZ) [50]. Finally, an assistive typing practice application is constructed, and its effect and acceptability are evaluated.

In Chapter 4, an individualized tutoring system for free text input named Typing Tutor is proposed. First, a user study to clarify the problems older adults encountered in text entry is described. In addition, the relationship between the input stumbles and the user's skill is analyzed Second, the statistical input detection method is constructed and the structure of Typing Tutor is designed. The performance of Typing Tutor was evaluated in a two-week experiment.

In Chapter 5, a tutoring system for voice input is proposed. First, a user study that clarified the problems that older adults encounter with voice input is described. Second, the statistical input detection method and the structure of the tutoring system are designed. Finally, the performance of the resulting tutoring system is evaluated.

The study is concluded in chapter 6 and a brief overview of future work is provided.

Chapter 2

Related Work

This chapter introduces text input methods and studies that enhance accessibility to older adults. First, studies on text input related to software keyboard and voice input are described. Subsequently, assistive technologies for novice older adults are introduced.

2.1 Text Input for Mobile Devices

First, comparative studies into methods of inputting text on a smartphone are explained. Next, studies on text input using a software keyboard are presented. Lastly, studies on voice input are explained.

2.1.1 Comparative Research into Text Input Method

Some studies compared usability among the various text input methods. Karat *et al*. [66] investigated voice input as opposed to using a keyboard and mouse. Although voice input was a faster text input method, it took nearly three times as long for users to correct errors with voice input compared to input with a software keyboard. Cox *et al*. [15] compared voice input with a 12-layout keyboard. They measured speed in terms of words per minute (WPM), word error rate, and workload. Voice input was shown to be the faster, with a higher level of satisfaction and lower workload compared to the 12-layout keyboard. For novice older adults, Rau and Hsu [81] compared several input methods for completing tasks on the web. They hypothesized that direct manipulation interfaces (handwriting, soft keyboard and voice input) would outperform keyboard and mouse. Hence, voice input has become an important text input option for those who are unfamiliar with touch input.

2.1.2 Text Input by Software Keyboard

Text Input Framework

Touch interfaces lack both the mechanical stability and tactile feedback of a keyboard, making it harder to accurately select targets [42].

Some studies proposed methods for transforming detection areas based on language models [40, 52] or users' touch distribution [26, 30, 82, 89] or a combination of both models [8, 29, 34, 91, 101]. In one study, the keys with high probabilities for a character sequence (N-grams) were displayed with bigger key sizes [52]. However, the performance of the language-model-based approach is heavily dependent on the dictionary. Unregistered words, such as new words, abbreviations, and colloquial expressions, are ignored.

In contrast, in touch-model-based approaches, some studies proposed transforming the keyboard layout according to the actual touch distributions [26, 30, 82, 89]. In these studies, the center of each key was moved to follow the centroid of the actual touch distributions, and the boundary was set in the middle of the keys. However, this method failed to take into account, i.e. the previous key. The studies also reported that some users were confused by the dynamic changes in the key layout.

Furthermore, some studies proposed methods for transforming the detection areas based on both the language-model-based and touch-model-based approaches [8, 29, 34, 91, 101]. The probability of a particular character being input next was modeled by the product of two probabilities, viz. one from the language model and the other from the touch model [30, 34]. Joshua *et al.* [30] used character N-grams for the language model and the Gaussian mixture model (GMM) of stylus input positions in the touch model. Asrla *et al.* [34] proposed methods for transforming the detection areas based on the typing history in the language model, although the central region of the key remained fixed to suppress excessive transformation. The occurrence probability of a character is given by the product of the probability of the touch model and that of the language model using Bayes' theorem as in

$$P(c_j|(x_t,y_t)) = \frac{p((x_t,y_t)|c_j)\cdot p(c_j)}{p(x_t,y_t)} \tag{2.1}$$

$$\propto p((x_t,y_t)|c_j)\cdot p(c_j) \tag{2.2}$$

$$= p_T \cdot p_L \tag{2.3}$$

where $P(c_j|(x_t,y_t))$ represents the probability of a character c_j for a input of the coordinate (x_t,y_t), p_T is the probability of the touch model, and p_L is that of the language model. The probability of touch model p_T can be represented by a single Gaussian distribution in the simplest case. In practice, the performance was improved by a Gaussian mixture model (GMM) because the touch distribution is different depending on hand postures, such as the typing hand. The generative model outputs the probability p_T using the following equation

$$p_T = \sum_{k=1}^{K} \omega_k \cdot N((x_t,y_t)|\mu_{jk},\Sigma_{jk}) \tag{2.4}$$

$$\Sigma_{jk} = diag(\sigma_x^2,\sigma_y^2) \tag{2.5}$$

where N denotes a Gaussian distribution with a mean vector μ_{jk} and a diagonal covariance matrix Σ_{jk} and K is the number of mixtures. On the other hand, the language model, which formulates the probabilities of the subsequent characters, is implemented by a character N-gram because the probability of c_j heavily depends on the history of character sequences. The probability p_L is shown as follows

$$p_L = p(c_j|h) \tag{2.6}$$

where h is the history of the N-gram character sequences. The probability of a sequence is given by the product of the probabilities in the sequence. The touch model is adapted to an individual user based on the maximum likelihood linear regression (MLLR) algorithm that has been reported to be effective for key input [37]. In MLLR, the means of the normal distributions are transformed and re-estimated as

$$\mu_k' = \mathbf{A}\mu_k + b \tag{2.7}$$

where μ_k is the mean vector of Equation (2.4), and μ_k' is the adapted mean vector. ($\mathbf{A}$, b) are obtained by maximum-likelihood estimation.

Practical issues in the real environment have also been tackled. Some studies analyzed the touch distributions for different hand postures, and proposed a keyboard that adapted to the users and the hand postures [29, 101]. Goel *et al.* [29] proposed switching various keyboard models by detecting the hand posture based on tap sizes and elapsed time between taps. Yin *et al.* [101] defined the hierarchical submodels depending on the user and the hand posture in order to adapt each model properly according to the amount of training data. Some studies analyzed the differences in touch distributions between sitting and walking, and proposed adaptation techniques for these contexts [28].

As a different approach, Bi *et al.* [8] optimized an algorithm for presenting appropriate suggestions to make correction and completion easier.

Characteristics of Older Adults in Text Entry

Weilenmann [95] revealed some of the challenges older adults face when learning text entry on mobile devices based on interaction analysis of a video showing older adults learning to use at mobile phone. Older adults have problems understanding how to press of keys sequentially, which is needed to perform a number of functions on a mobile phone, which includes texting. Kurniawan [56] reported that older adults usually dislike text-prediction features. Rodrigues *et al.* [82] analyzed how the typing behavior of older adults could be influenced by varying the keyboard, including the color and the width of keys.

On the other hand, various typing apps have been made available in the smartphone market. In most typing apps, users are prompted with a text showing what they are required to type. When users mistype a character, they are notified through auditory and visual feedback, such as beeps and underlines. This is effective for highly motivated users with a certain degree of knowledge and skill. However, Leung *et al.* [59] indicates that older adults tend to prefer an instruction manual to trial-and-error. Nicolau *et al.* [70] reported detailed analyses on how older adults learn text entry, and found their most common errors were due to cognitive problems in the initial learning stage. This thesis focuses on addressing cognitive problems rather than physical aspects such as mistyping.

2.1.3 Voice Input

Automatic Speech Recognition (ASR) Framework

Automatic speech recognition (ASR) technology has been stduied over the past decades [46, 60, 65, 68, 77]. ASR is formulated by identifying the most likely sequence of words that can explain an observed sequence of input speech. Such an approach is based on two statistical models estimated from data: the acoustic model, which provides the likelihood of an acoustic feature sequence given a word sequence; and the language model which provides a prior probability for a word sequence. A feature extraction module assumes the task of converting the speech signal into feature parameter vectors, and the lexicon acts as a map between the words in the language model and the subword units that comprise the acoustic models. Given some input speech, the most likely sequence of words is found using a search, or decoding process by combining the acoustic and language models. The acoustics are typically statistically modeled such as by Gaussian mixture models (GMMs) sequenced using hidden Markov models (HMMs). All the results presented here have been conducted in this framework, with front end parametrization being perceptual linear prediction coefficients (PLP) which are one of the best-known features for ASR.

Speaker-adaption reduces variability among different speakers including older adults. Normalization and adaptation can be divided into two categories. The first one is responsible for feature vector normalization and transformation. Cepstral mean normalization, cepstral variance normalization, and linear discriminant analysis are examples of feature space normalization and transformation techniques. The second one is related to model space transformation techniques such as maximum-a posteriori (MAP) and maximum likelihood linear regression (MLLR).

Over the last few years, the accuracy of speech recognition has dramatically improved by a deep neural network (DNN) [41, 78]. There are several ways in which neural networks can be applied to speech recognition. The most widely used method is called "DNN-HMM hybrid architecture", which combines a DNN with HMMs. In this framework, the HMMs capture the temporal dynamics of the speech signal and the DNN estimates the observation probabilities given the acoustic observations. Each output neuron of the DNN corresponds to the tied tri-phone state. This improvement achieved through DNN approaches has made voice input on smartphones more practical.

Characteristics of Older Adults in Voice Input

With aging, several changes occur in the respiratory system, larynx and the oral cavity which form the human speech production mechanism [64]. Significant changes affecting speech include loss of elasticity in the respiratory system leading to decreased lung pressure, calcification of the laryngeal tissues causing instability of vocal fold vibrations, loss of tongue strength, tooth loss, and changes in the dimensions of the oral cavity [80]. Aging affects many acoustic parameters of the speech such as fundamental frequency, jitter, shimmer and harmonic-noise ratios [99]. Aging voices are also characterized by increased breathiness and slower speaking rates. All these changes in the acoustics of older voices have an impact on ASR: One ASR system used with older adult voices (aged 60+) demonstrated a word error rate increase of 50% in comparison with younger adults (aged 18-59) [96]. Another study showed that word errors increased more rapidly for those older than 65 years [92]. Therefore, ASR systems adapted to older adults have been developed.

In addition to physical factors, aging influences the cognitive functions such as reasoning [43] and working memory [85]. Therefore, the interface is one of the most critical factors in the success of ASR. Some studies into voice input interfaces have proposed the inclusion of the assumption that misrecognition occurs. Goto *et al.* [31] proposed a system to alter a sentence when users hesitate by lengthening a vowel during a phrase. Ogata *et al.* [72] proposed a system that displays the result of ASR with alternative word candidates. A user who finds a recognition error can just select the correct word from among the candidates. An API of Speech Recognizer for Android[1] gives feedback on the results of errors, such as audio recording errors and network-related errors. This kind of support technology for voice input is essential. However, a cognitive support system for voice input has not been studied as it has for text entry.

2.2 Assistive Technology for Novice Older Adults

Studies to make new technologies accessible to older adults can be roughly divided into three approaches, design for older adults, adaptation to older adults, and assistance for older adults [102]. Figure 2.1 summarizes the relationship between the user and a system for the three different approaches [23]. The approach by design is for a user

[1] https://developer.android.com/index.html

whose abilities match those presumed by the system. The adaptation technology is designed to accommodate the user's abilities. It may adapt or be adapted to them. The assistance approach is largely that of fitting non-standard users to standard technology through an assistive component.

2.2.1 Design for Older Adults

A serious problem that novice users face with existing interfaces is that they are complicated and include too many options [17]. This complexity results in users becoming confused. Universal usability provides guidelines for designing interfaces that are usable by the widest range of people [86, 90]. Universal usability is equally concerned with disparities of access and use owing to age, culture, and so on. Fisk *et al.* [84] provided a list of guidelines for older adults. These guidelines emphasize interface designs that respond to age-related changes in perception and cognition: larger displays, fonts, buttons, accessibility improvements in specific contexts (navigation [84], web [67], email [18], etc.). While these guidelines are beneficial, adopting them requires dedicated effort in the design and implementation phases, thus excluding many currently existing applications. Moreover, it is difficult to come up with a universal solution that can meet the needs of all older adults, as needs can differ among individuals and change over time [32].

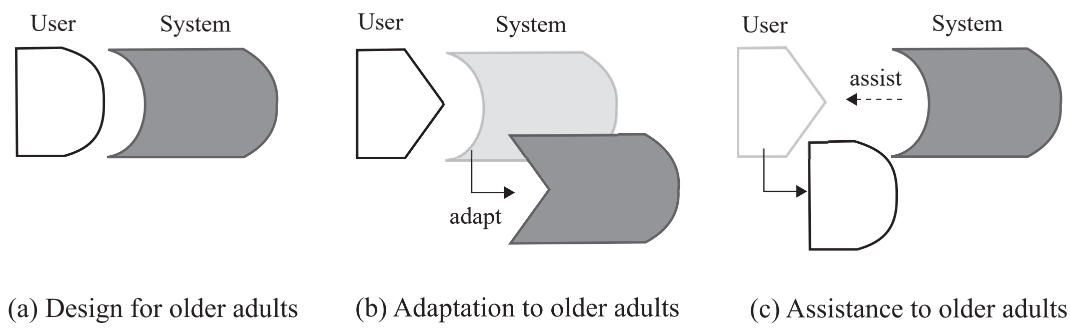


Figure 2.1: Summary of three research approaches: design, adaptation, and assist to older adults

2.2.2 Adaptation to Older Adults

To respond to the various needs of older adults, adaptive solutions have been investigated. Several models for adaptation can be found in some studies [20, 57]. Davis [16] proposed technology acceptance model (TAM). This is a model that shows how users come to accept and use technology and is still used as a baseline model today [51, 67]. TAM contains three main indicators: perceived usefulness, perceived ease-of-use, and attitude toward using/behavioral intention [19, 51]. TAM has also been used in studies of older adults and their adoption of communication technologies [98]. Wobbrock *et al.* [98] defined this shift as ability-based design. Adaptive interfaces can be configured according to individual user's profiles such as behaviors [87], knowledge [98], and skills [22]. Since determining a user's profile can be difficult, systems may adapt based on the automatic collection of their information [1]. However, there is a trade-off between ease-of-use and learnability. A multi-layered interface for adaptation can reduce learnability, especially for novice older adults [58]. One prototype system used two layers, one with reduced functionality and the other with full functionality. The simplified interface improved the novice older adults' task completion times compared to the full-featured interface. Although people using the simplified user interface needed more time to learn the full-featured user interface after mastering the simplified user interface, the two-layer interface did lower the overall learning barrier. This trade-off has been the subject of active investigation in recent studies.

2.2.3 Assistance for Older Adults

The third approach is to provide assistance. The focus of this approach is mainly entails fitting non-standard users to standard technology through an assistive component, often an add-on inserted between the user and the system [98]. Early human-computer interaction (HCI) research recognized the challenge of providing efficient help systems for software applications [10], and established the benefits of minimalist and task-centric help resources, and user-centered designs for help systems [53, 75]. Since the idea of online help was conceived, there have been explorations into many types of learning aids, such as tutorials [38, 73, 75] and contextual assistance [2].

There have been several approaches to designing effective and engaging tutorials. Software video tutorials were proposed in several early studies [38, 73]. More recently, it has been identified when animated assistance can be beneficial. MixT [11], which is a multimedia tutorial system, showed that short video demonstrations are useful for demonstrating individual tutorial steps involving dynamic actions. Pause-and-Play [75] improves user's ability to follow video demonstrations by automatically pausing and resuming a video to stay in sync with the user's progress.

Contextual assistance has been shown to be effective for learning graphical user interfaces [7, 33, 49]. Unlike traditional tutorials, contextual assistance allows users to receive assistance in the actual interface they are interacting with. There is a range of tutoring systems that are presented in the context of the application [24, 33, 49]. Kelleher *et al.* [49] developed an overlay step-by-step instructions on the interface and limit interaction to UI elements related to the current step. Sketch-Sketch Revolution [24] is a drawing tutorial that assists the user in completing drawing tasks, enabling the user to experience success while learning. ToolClips provides users with short video clips and other rich help content next to the toolbar buttons contextually [33]. However, these contextual assistances are limited to specific applications or are suitable only for a single, specific task. Therefore, Yeh *et al.* developed a tool [100] that allows designers to create contextual assistance across applications by taking screenshots. EverTutor [94] uses a computer vision method to locate the component on the user's screen that matches the screenshot. These systems track the operational flow or capture the screenshot, and detect the context by pattern matching approaches or vision-based analysis by comparing the pattern registered beforehand.

Chapter 3

Assistive Typing Application to Practice Text Entry

3.1 Introduction

In this chapter, to support self-instruction for learning text entry techniques on smartphones, an assistive typing practice application, that is, a tutoring system for entering presented sentences is proposed. The main purpose is to confirm the acceptability of the instruction from a tutoring system as the initial step in the study.

There are four steps to construct an assistive typing practice application. First, the ways that novice older adults have problems with text entry on smartphones in typing application are investigated through a user study. Second, the acceptability of being provided with instructions for text entry is confirmed by Wizard-of-Oz (WoZ) [50]. Next, an assistive practice typing application, which automatically detects input stumbles and provides instructions on the typing of presented sentences is constructed. This is built on the basis of the collected data, including operational stumbles and effective instructions found in the previous two studies. Finally, the assistive typing practice application is evaluated by comparing its performance with human tutors.

3.2 Related Work

3.2.1 Interface Design for Older Adults

Many studies have attributed older adults' difficulties in learning to use technology to a number of user characteristics, such as a decline in perceptual performance and lack of experience in the relevant technology. Rama *et al.* [79] noted that older adults have difficulty in learning current user interfaces because they generally have less experience with current devices than younger adults, and need to learn different types of user interfaces from former technologies. In addition, user interfaces often have buttons that perform different context-dependent tasks. To resolve such problems, Fisk *et al.* [84] provided a guideline for older adults. The guideline emphasizes interface changes that respond to age-related changes in perception and cognition: larger displays, fonts, and buttons, and accessibility improvements in specific contexts. With respect to the touch interface, Jin *et al.* [47] have investigated optimal target size, spacing, and position to derive recommendations and general guidelines for older adults. While these guidelines are very useful, needs can differ among individuals [32]. To respond to the various needs of older adults, researchers have considered adaptive solutions based on user profiles such as behaviors [87], knowledge [98], and skills [22].

3.2.2 Text Entry for Older Adults

Touch interfaces lack both the mechanical stability and tactile feedback of a keyboard, making it harder to accurately select targets [42]. Some studies proposed methods for transforming detection areas based on language models [40, 52] or users' touch distribution [26, 30, 82, 89] or the combination of both models [8, 29, 34, 91, 101]. For example, Rodrigues *et al.* [82] analyzed the way to influence the typing behavior of older adults by varying the keyboard, including the color and the width of keys. Gunawardana *et al.* [34] proposed methods for varying the detection areas based on typing history in the language model. As a different approach, Bi *et al.* [8] optimized an algorithm for presenting suitable suggestions to make correction and completion easier. However, Kurniawan [56] reported that older adults usually dislike text-prediction features. Therefore, Komninos *et al.* [54] proposed a keyboard that makes users aware of any errors by highlighting text in the body of the message and using a color bar at the top of the keyboard.

On the other hand, various typing apps have been proposed in the smartphone market. In most typing apps, users are prompted with text showing what they are required to type. When users mistype a character, they are notified through auditory and visual feedback, such as beeps and underlines. This is effective for highly motivated users with a certain degree of knowledge and skill. However, Leung *et al.* [59] indicates that older adults tend to prefer an instruction manual to trial-and-error. Nicolau *et al.* [70] reported detailed analyses on how older adults learn text entry, and found their most common errors were due to cognitive problems in the initial stage.

3.2.3 Tutoring System for Older Adults

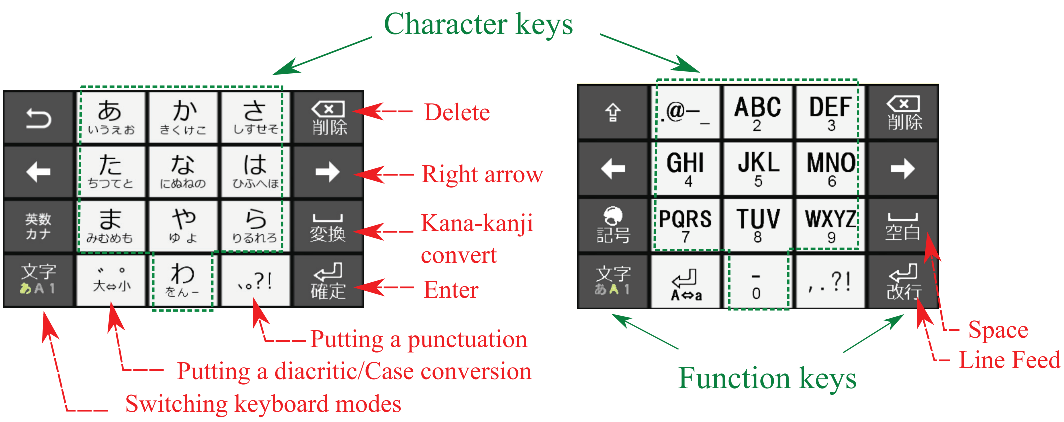
A wealth of research has largely focused on designing better instructional resources to assist older adults when learning how to use desktop computers. For example, Hickman *et al.* [39] have studied the type of guidance most suitable for older adults. Morrell *et al.* [69] have studied what the optimal amount of guidance is. Rogers *et al.* [83] investigated the kind of resources most useful in the learning process, and found that step-by-step interactive tutorials were the most effective approach in the learning process for older adults. With respect to using smartphones, Leung *et al.* [59] surveyed and investigated how older adults learned. According to their report, older adults tend to prefer an instruction manual to trial-and-error. Kelleher *et al.* [49] proposed stencil-based tutorials that overlay step-by-step instructions on the screen. Kristensson *et al.* [55] hypothesized that tutoring system might reduce time and effort when learning a new type of text entry method and proposed five design dimensions: automaticity, error correction, coverage, feedback, and engagement. However, practical tutoring systems for text entry have not been developed, and the effectiveness of the tutoring system has not been evaluated.

3.3 User Study to Clarify How Novice Older Adults Make Input Stumbles

First, a user study is conducted to investigate the problems and the characteristic actions of novice older adults when making text entries on smartphones.

3.3.1 Participants

Thirty participants, fifteen males and fifteen females between the ages of 60 and 83 (mean 72.1, sd = 8.2), took part in the user study. They were recruited from a local social institution. None of them had any previous experience using a smartphone, but all had owned a feature phone for more than one year, i.e. one with a physical keyboard with a 12-key layout. They were familiar with this kind of keyboard because the same



(a) Kana keyboard mode

(b) Alphabetic keyboard mode

Figure 3.1: Key layout

layout has been used in feature phones for a long time. Twenty-six of them had entered text with a feature phone before, while the others had never done so, using the phone only for calling. Ten had used their own PC in the usual ways. None of them had a tremor disorder, eye problems or other health problems.

3.3.2 Apparatus

The software keyboard was designed for multi-tap input with a 12-key layout for Japanese as shown in Figure 3.1. The smartphone recorded all touch events using the standard Android API and all linguistic information, e.g., typed keys, displayed characters, and suggestions. During use, an overhead video camera recorded participants' operations as well. All participants used a Samsung Galaxy S3 running Android 4.1.2 with a 4.8-inch screen having a resolution of 1280 × 720 pixels (306 ppi).

Characteristics of 12-key multi-tap for Japanese

Next, the characteristics of the keyboard with a 12-key layout are described. The 12-key layout is common in Japanese smartphones. The layout has several points in common with the QWERTY layout, such as function keys (arrow, switching keyboard, and case conversion) and a suggestion list. A character string being edited is not finalized until

Table 3.1: Japanese syllabic alphabet represented by Roman alphabet letters

	A	I	U	E	O
	a (あ)	i (い)	u (う)	e (え)	o (お)
K	ka (か)	ki (き)	ku (く)	ke (け)	ko (こ)
S	sa (さ)	si (し)	su (す)	se (せ)	so (そ)
T	ta (た)	ti (ち)	tu (つ)	te (て)	to (と)
N	na (な)	ni (に)	nu (ぬ)	ne (ね)	no (の)
H	ha (は)	hi (ひ)	hu (ふ)	he (へ)	ho (ほ)
M	ma (ま)	mi (み)	mu (む)	me (め)	mo (も)
Y	ya (や)	-	yu (ゆ)	-	yo (よ)
R	ra (ら)	ri (り)	ru (る)	re (れ)	ro (ろ)
W	wa (わ)	-	-	-	wo (を)
	nn (ん)	-	-	-	-

either a suggestion is selected or an enter key is pressed. A point of difference is that the characters are input by the combination of a key and a number being pressed. For example, the letter "c" is input by typing the "a" key three times. Therefore, when typing two consecutive characters allocated to the same key, the first character must be fixed by pressing the right arrow key before typing the second character.

Next, the characteristics of Japanese input are described. Kana characters, which comprise Japanese phonetics, are combinations of a consonant and a vowel as shown in Table 3.1. In this keyboard, a key corresponds to a consonant, and the number of presses corresponds to a vowel. For some characters, the diacritic key needs to be used after inputting kana. For example, the kana of "gu (ぐ)" is input by pressing a diacritic key after the kana for "ku (く)". Although the suggestion list usually shows the auto-complete list predicted by a typed string, the kanji list is shown after typing the kana-kanji convert key.

3.3.3 Procedure

First, the participants were given three minutes of explanations and examples of how to operate a smartphone, including touch and swipe operations, by a human tutor. Then, they tried to operate the smartphone for one minute. After that, they were briefly

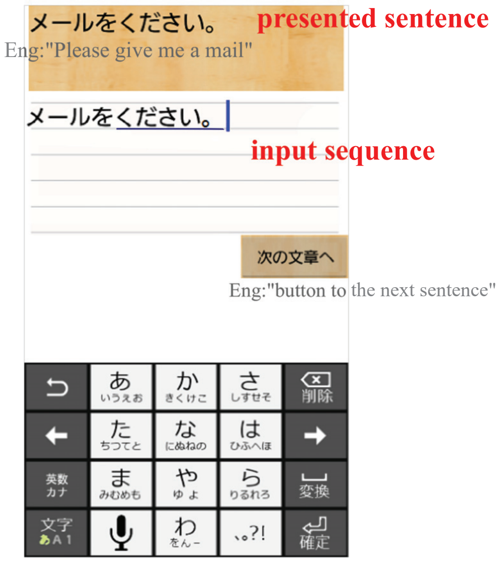


Figure 3.2: Screenshot of a typing application

instructed on how to use the software keyboard, including the correspondence between each key and a character and how to correct text or select suggestions by watching a video tutorial on the smartphone. Afterwards, they typed twelve sentences in Japanese by using a typing application as shown in Figure 3.2. The sentences were selected from an email corpus gathered originally from personal conversations via email, such as greetings and appointments to meet. The corpus contained roughly 30,000 sentences (average character length = 23.6). The experimental period was limited to 75 minutes. Participants had a one-minute rest after each sentence. After the experiment, they took part in an interview.

They operated the smartphone while holding it in their hand or resting it on a desk while sitting on a chair. A human tutor sat next to them. They were instructed to type by themselves if at all possible, with no help from the human tutor. However, they were permitted to ask the human tutor when they lacked the confidence on what to do next. They received operating instructions on request and when the human tutor judged that they needed assistance.

3.3.4 Annotation

Three annotators independently extracted the pattern of input stumbles from the logs and the videos of the study to maintain inter-rater reliability, and classified the input stumbles into ten categories based on discussion. Table 3.2 shows the input stumbles as categorized. Next, the three annotators labeled input stumbles and possible stumble opportunities according to the logs of each input. The possible stumble opportunity was labeled when an annotator judged that an input stumble was likely to occur based on the time series information before an input, such as the unfixed string and the elapsed time from previous input. The labels given by at least two annotators were adopted. The concordance rate using Fleiss' kappa [25] was 0.82.

3.3.5 Results of the User Study

Twenty-eight participants completed the task. The average completion time was 43.4 minutes (sd = 12.0). The other two participants did not complete the task within 75 minutes. These were the two who had no experience entering text with their own feature phone. A total of 355 sentences were typed, with an average character length of 26.1.

Every category was classified in terms of the two ratios in Table 3.2. One is the input stumble ratio (ISR) with respect to chances, which is the percentage of stumbles compared with the possible stumbles in each category. The other is the ISR with respect to participants, which is the percentage of participants who stumbled in the category at least once.

These ISRs were higher than average in both metrics: "(1) How to type characters allocated to the same key consecutively", "(2) How to fix a sentence", "(3) How to select a suggestion", "(4) How to switch keyboard modes", and "(5) How to insert a linefeed". These operations have strong cognitive aspects. The input stumbles on operations (1) and (2) were typical stumbles of omission due to cognitive lapses. The common point is that no character or string changed when these keys were typed. Many participants seemed not to understand that they needed to type these keys until the human tutor provided instructions several times. In terms of operation (3), many participants found it difficult to judge the timing for selecting the desired suggestion. They focused so much on typing that they forgot to select the target suggestion until they had typed a long sentence. When typing a long sentence, they tended to mistype or they had to

Table 3.2: Input stumbles with two frequency metrics

The operation in which an input stumble has occurred	ISR with respect to chances (%)	ISR with respect to participants (%)
(1) How to type characters allocated to the same key consecutively	27.1	86.7
(2) How to fix a sentence	17.3	63.3
(3) How to select a suggestion	15.2	90.0
(4) How to switch keyboard modes	15.0	50.0
(5) How to insert line feeds	13.0	53.3
(6) How to delete previous character	12.7	26.7
(7) How to convert to a small letter	9.4	20.0
(8) How to enter a symbol	5.0	40.0
(9) How to enter a diacritic	4.6	43.3
(10) How to enter punctuation	4.3	20.0
Average	12.4	49.3

decide where the sentence was divided. Therefore, it was useful to recommend selecting the suggestion with each short sentence. Input stumbles of operation (4) occurred frequently because the participants had no previous experience of such an operation on their own feature phone. Operation (5) is confusing because operations (2) and (5) are assigned to the same key and its response changes with the context. Although the concept is the same as that on the keyboard of a feature phone, some participants did not understand the changeable response.

In contrast, there were few stumbles associated with operations (6)–(10) because participants only had to check the final character and the next key position. Once they had received instructions, most of them were able to perform these operations.

This experiment confirmed that older adults needed more repetitive support for complex operations, such as (1)–(5). In addition, instructions that included the purpose of an operation were more effective than instructions that simply said what to do next. However, only a few repetitions of the instructions were required for the simple operations ((6)–(10)), such as the operation to point to the next position.

3.4 User Study with Wizard-of-Oz

To confirm the acceptability of being provided with instructions, a user study is conducted using Wizard-of-Oz (WoZ) [50]. In this study, the tutoring system of WoZ provides instructions when a human tutor judged that participants needed instructions.

3.4.1 Participants

Five participants took part in the user study, three males and two females ranging in age from 65 to 77 years (mean age 73.2, sd = 5.5) who were recruited from a local social institution and did not participate in the preliminary experiments in this study. They had no previous experience with a smartphone or tablet, but had owned a feature phone with a 12-key physical keyboard for more than one year and had sent emails more than twice a week. None of them had a tremor disorder, eye problems or other relevant health problems.

3.4.2 Apparatus

A Samsung Galaxy S3 running Android 4.1.2 and with a 12-key multi-tap layout was used in this experiment, as in the first study. The keyboard recorded all touch events using the standard Android API and all linguistic information, e.g., typed keys, displayed characters and suggestions.

The WoZ system included a video camera, a monitor, and a PC that was connected to the smartphone using Bluetooth. All participant operations were recorded by the video camera and observed by a human tutor via the monitor. The human tutor was able to send an instruction by selecting from a list of instructions in a PC application. The list contained ten instructions appropriate for the input stumbles observed in the first study. The instruction was provided to the participant by a combination of text instruction, key highlighting overlaid on the key, and voice instruction that matched the wording of the text instruction. The highlighted key had a yellow rectangle overlaid on the key. The voice instruction was produced by Text-to-Speech. Figure 3.3 shows one of the instructions responding to input stumble "(2) How to fix a sentence". The text instruction and voice instruction say *"Please fix the sentence by typing bottom-right key"*, with the "fixing" key highlighted.

Figure 3.3: Screenshot of a WoZ instruction

3.4.3 Procedure

First, the participants were given explanations and examples of how to operate a smartphone, including touch and swipe operations, by a human tutor. Next, they were briefly instructed on how to use the software keyboard, including the correspondence between each key and a character, and how to correct input text or select suggestions. Then, they typed ten sentences in Japanese by using the same typing application as in the previous study. They operated the smartphone while holding it in their hand or resting it on a desk while sitting on a chair. When the human tutor observing through the monitor judged that the participant needed help with an operation in the instruction list, the human tutor sent the instruction to the smartphone used by the participant. Participants could rest for one minute after writing each sentence. The experimental period was limited to 60 minutes. After the experiment, they filled out a 5-point Likert-scale questionnaire [63] in which five indicated strong agreement, and took part in an interview.

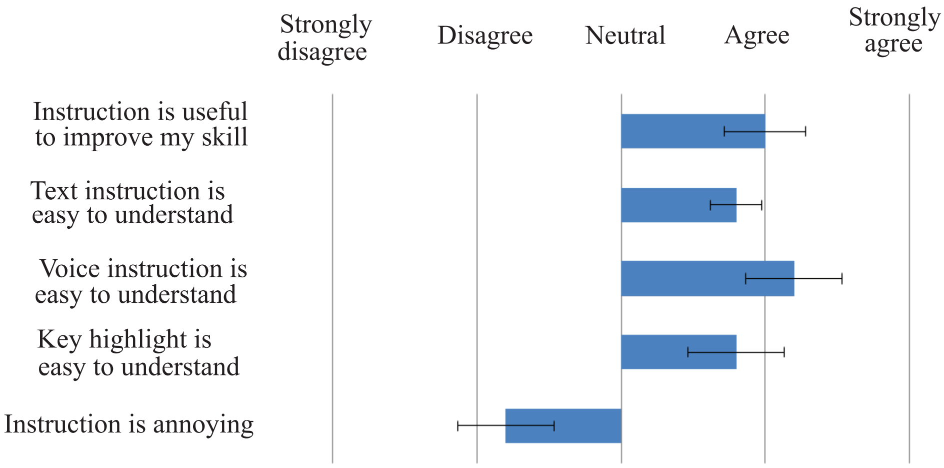


Figure 3.4: Questionnaire responses regarding WoZ study

3.4.4 Results of WoZ Study

All participants completed the task. The average completion time was 40.2 minutes (sd = 12.0). Figure 3.4 shows the results of the subjective questionnaires for the tutoring system using WoZ. There was a positive response to the item "Instruction is useful to improve my skill". The instruction methods were also positively rated, particularly voice instruction. These results showed that the tutoring system was generally supported by the participants.

On the other hand, the following negative opinions were expressed;

> *"Some instructions were provided even when I didn't make a mistake. Incorrect instructions induce a feeling of anxiety."*

This opinion indicated that precision is more important than recall in designing an automated tutoring system for text entry.

In addition, two participants made the following comment in regard to the manner of instruction;

> *"First, I didn't realize that I could press the overlaid yellow key before the highlight disappeared. It's confusing to me."*

This study showed that the tutoring system was generally supported by the participants. However, there is room for improvement of the way of instruction.

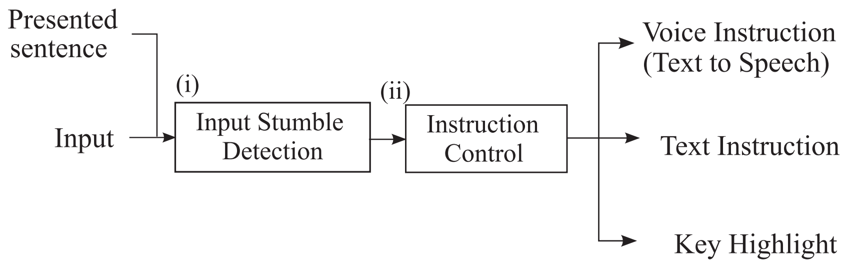


Figure 3.5: Architecture of the assistive typing practice application

3.5 System Design of Assistive Typing Practice Application

To support self-instruction for learning text entry techniques on smartphones, an assistive typing practice application, that is, a tutoring system for entering presented sentences is proposed. The application has two functions: detecting input stumbles and providing instructions. On the basis of the recorded data and instructions from the previous two studies, an input stumble detection and instruction controls are constructed. Figure 3.5 shows the architecture of the application. First, the application observes the input data, such as the typed key and suggestions. Then, the application detects input stumbles based on the input stumbles detection that was constructed using data from the earlier study. Lastly, instructions for the next action are provided simultaneously by voice instruction, text instruction, and key highlighting based on the instruction model.

3.5.1 Input Stumble Detection

The input stumble detection included two steps: stumble detection and stumble classification.

Input stumbles were detected when the user stopped typing for longer than a certain threshold time ϵ, defined as $\mu_t + 2\sigma_t$. The variables μ_t and σ_t are the mean and the standard deviation of the touch intervals of the thirty participants in the preliminary experiment. 97.8% of the touch intervals labeled input stumbles exceeded the threshold time. The stumble classifier used with the machine learning technique classifies input data into 11 categories: the ten input stumble categories in Table 3.2, and a category

Table 3.3: The classification performance of ten input stumble categories of each model

Classification model	F-measure (Precision)
SVM	0.941 (0.982)
C4.5	0.942 (0.982)
NB	0.938 (0.980)

of no input stumble. The performances of three models, a support vector machine (SVM) [13], a C4.5 [76] and a Naive Bayes (NB) [14], were compared in order to select a proper learning machine. Each model was trained by using data labeled 11 categories from the first experiment in Section 3.3. A total of 44 features was obtained on the basis of input sequences including unfixed string and fixed string, presented sentence, and suggestions. The features were selected from three points of view. One was characteristics of the input string itself, such as the number of phrases in an unfixed sentence and the category ID of the morpheme of the last phrase. Another was the keyboard mode. The third considered whether the partial input string or the suggestion was consistent with the presented sentence. Each model adopted some of the feature amounts so that the accuracy rate by F measure was the highest in the evaluation of each participant in a 30-fold cross-validation.

Table 3.3 shows the performance of each model. The F measures were 0.941 with SVM, 0.942 with C4.5, and 0.938 with NB. Anova (significance level $\alpha = 0.05$) showed no significant differences between the three ($F(2, 58) = 0.74$), so C4.5 with the highest accuracy rate was adopted. The features adopted by C4.5 were the ten shown in Table 3.4. Two features directly related to the input string referred to the type of the last character in an unfixed string and the length of the unfixed string. Two features were relevant to the keyboard mode: the current type of keyboard and the keyboard to be used next. There were six either-or features: whether the fixed string was consistent with the presented sentence, whether the unfixed string was consistent with the beginning of the rest of the presented sentence, whether the word to be input next was in the suggestion list, whether the last character in the unfixed string should include a diacritic, such as a voiced sound, whether the last character in the unfixed string should be converted to a small letter, and whether the character to be input next was allocated to the same key, such as “kaki (かき)”.

Table 3.4: Effective features to classify input stumbles with C4.5

ID	Features
1	The type of the last character in U
2	The length of U
3	The type of the current keyboard
4	The type of the keyboard to be used next
5	The binary whether F is consistent with P
6	The binary whether U is consistent with the beginning of the rest of P
7	The binary whether the word to be input next is in the suggestion list
8	The binary whether the last character in U should include a diacritic
9	The binary whether the last character in U should be converted to a small letter
10	The binary whether the character to be input next is allocated to the same key

F: fixed string, U: unfixed string, P: presented sentence

3.5.2 Instruction Control

Instruction control, which provides an instruction corresponding to each type of input stumble, was provided manually based on the effective advice observed from human tutors. Concretely, the instruction model provides the procedure and the purpose for operations (1)–(5), while providing only the procedure for (6)–(10). For instance, the instruction in operation (1) is *"Please tap the right arrow key to move the cursor for typing characters allocated to the same key consecutively"* and that in operation (6) is *"The delete key is upper right"*. The instruction model produces instructions in the form of voice, text, and a highlighted key with an overlaid finger animation. The procedure for indicating the key position was improved based on the results of the other preliminary experiment because participants in section 3.4 pointed out that simply indicating the key with a yellow overlay was confusing.

Finally, the assistive typing practice application was implemented on the smartphones. Figure 3.6 shows that the assistive typing practice application detects input

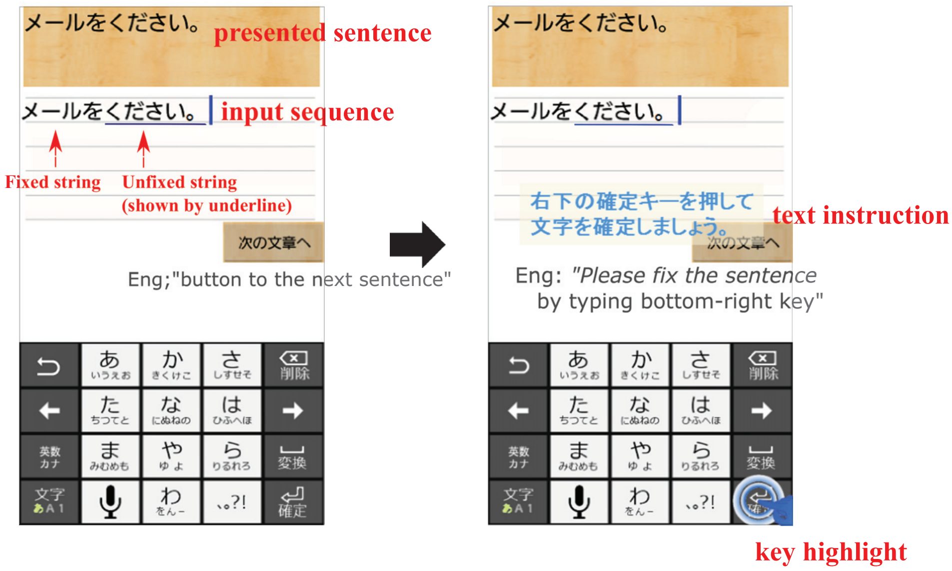


Figure 3.6: Screenshot showing the provision of instruction

stumbles of punctuation and provides instructions, such as *"Please fix the sentence by typing bottom-right key"* which is the instruction given in response to input stumble "(2) How to fix a sentence", with the text instruction and the voice instruction given at the same time as the key is indicated by the overlaid finger animation.

3.6 Evaluation of Assistive Typing Practice Application

In this section, the performance of the assistive typing practice application is evaluated by comparing it with that of a human tutor.

3.6.1 Participants

Twenty-four participants, twelve males and twelve females between the ages of 60 and 82 (mean 70.2, sd = 6.5) who did not participate in preliminary experiments in this study, took part in this experiment. They were recruited from a local social institution. None of them had previous experience with a smartphone, but all had owned a feature phone for more than one year. Nineteen of them had entered text using a feature phone before, while the others had never done so, using their feature phones only for calling.

Nine of them had used their own PC in the usual ways. None of them had a tremor disorder, eye problems or other health problems.

3.6.2 Apparatus and Procedure

The tutorial in the experiment on how to use the smartphone and the software keyboard was the same as in the previous experiment. After the tutorial, the participants typed twenty sentences in Japanese (average character length of 16.5) using the typing application within a 75-minute time limit.

When the input sequence matched the presented sentence correctly, the next sentence was presented. The participants were divided into three groups. In group A, twelve participants used the proposed assistive typing practice application. In group B, six participants typed the sentences with support from a human tutor, who gave instructions based on the same procedure as the assistive typing practice application, but only when the participants did not type for a set period. In group C, the other six participants typed the sentences with support from a human tutor who gave instructions without any control. Any comparison with a group typing without instructional support was not made because it was clearly too difficult for them to complete the task without any instruction. After finishing the task, the participants in group A answered a questionnaire about their impressions and the effectiveness of the assistive typing practice application.

3.6.3 Experimental Results

All participants completed the task. As in the study in section 3.3, three annotators labeled input stumbles and possible stumble opportunities according to the logs of each input. The labels that were applied by more than two annotators were adopted. The concordance rate using Fleiss' kappa was 0.83. The input stumble ratio (ISR) with respect to chances, i.e., the percentage of possible stumbles that actually occurred, was calculated.

The performance was evaluated by three metrics: the typing speed, the input stumble ratio (ISR) with respect to chances, and the subjective questionnaire.

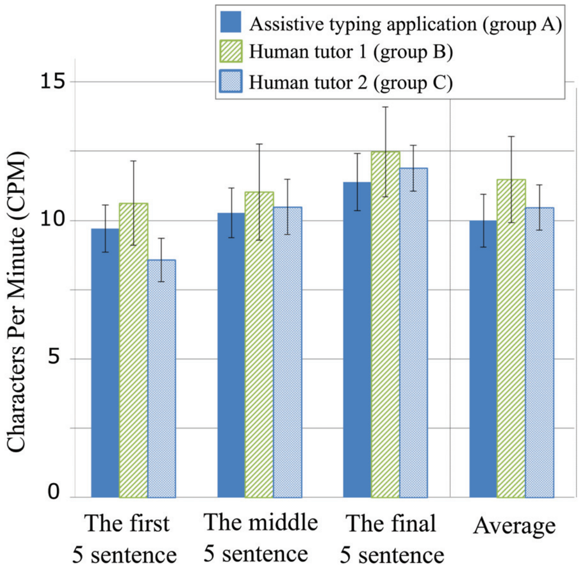


Figure 3.7: User progress in CPM for each group

Typing Speed

Typing speeds were evaluated by the metrics of characters per minute (CPM) [4] not including the time used by instructions. Figure 3.7 shows the measured CPM of all groups for the first five sentences, the middle five (from the 8th to the 12th), and the last five, and the overall average. All participants completed the tasks. The CPM improved over the course of the sentences in all groups. The Anova (significance level $\alpha = 0.05$) did not show any significant differences between three groups in any stage; F(2,21) was 0.44 in the first five, 0.12 in the middle five, 0.96 in the last five and 1.14 in the overall average. However, focusing on the improvement rate as measured by CPM, the improvement rate of group C was greater than the others: the rates were 17.2% in group A, 17.5% in group B and 35.5% in group C.

Input Stumble Ratio

The average number of input stumbles per participant was 1.17 per a sentence. The input stumble ratios (ISR) with respect to chances were compared. Figure 3.8 shows the ISRs of all groups at each step. The ISR for all groups decreased with practice.

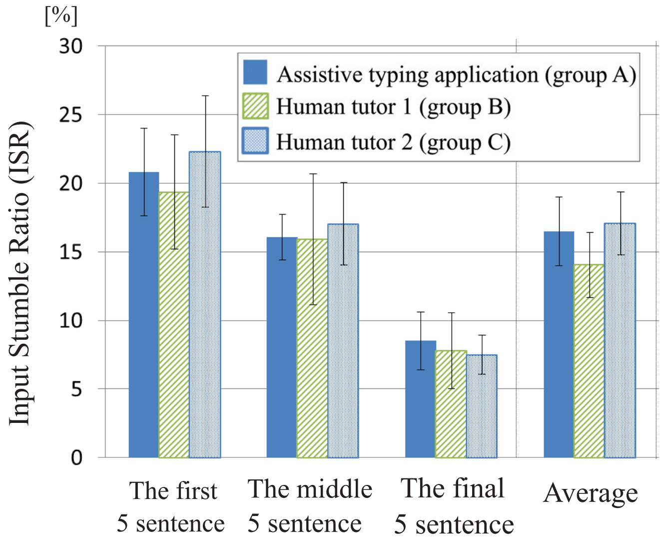


Figure 3.8: User progress in the input stumble ratio (ISR) with respect to the chances for each group

Anova (significance level $\alpha = 0.05$) showed no significant differences between the three methods in any stage. The improvement rate of this measure did not show a noticeable difference from that of the ISR; the rates were 59.1% in group A, 59.8% in group B and 66.4% in group C.

Questionnaire and Interview

Figure 3.9 shows the results of the subjective questionnaires on the assistive typing practice application. It shows that the assistive typing practice application was generally well accepted. The instruction methods were also positively rated, with voice instruction the same as in the WoZ study in section 3.4.

Participants expressed the following positive opinions:

> *"Multimodal instructions were easy to understand."*
>
> *"Instructions were helpful to learn smartphone."*

On the other hand, there were some negative opinions on the instruction,

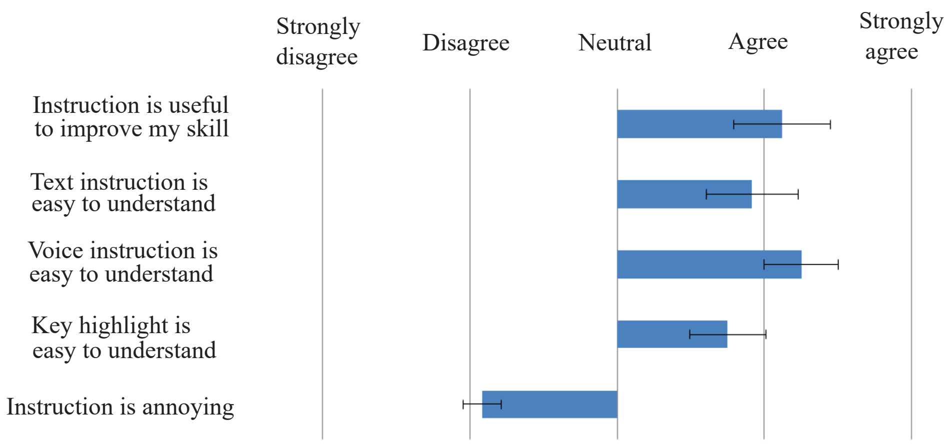


Figure 3.9: Questionnaire responses regarding the assistive typing practice application

> *"An instruction was provided once when I didn't need it. The instruction made me confused."*

The precision for detecting input stumbles was 97.8%. Four participants were provided incorrect instructions at least once. On the other hand, one interviewed participant responded:

> *"I didn't mind the misdirection because the instruction didn't interrupt my operation."*

The comments suggest that it is essential to provide instructions in a way that does not interrupt the user's operations.

3.7 Discussion

The discussion first considers the results of the evaluation experiment. Anova for CPM and ISR showed no significant differences among the three groups. Additionally, the improvement rates of participants using the assistive typing practice application were almost the same as those with human tutors. This shows that the assistive typing practice application works effectively and prompts the user's operations well. In an interview, there were similar comments:

> *"First, I didn't know operations at all. But I could begin to operate gradually as I understood the difference between my feature phone and the smartphone. These kinds of instructions were useful for me."*

This comment indicates that the assistive typing practice application is effective for users who use their feature phones to email frequently but have trouble with self-instruction.

From a different perspective, although Anova did not show any significant difference, the CPM and ISR of participants using the assistive typing practice application were lower than those of participants given instructions without any control. Participants made not only single stumbles but also complicated stumbles that included multiple components. This was probably because they took more time to resolve these complicated stumbles with the simple operations. The provision of step-by-step instructions for a complicated task is currently difficult to achieve. This difficulty is due to the requirement for a detailed understanding of what stumbles are currently occurring and what the user wants to do next. These challenges form the basis of one of the future research tasks.

On the other hand, a participant made a negative comment in the interview:

> *"Instructions were provided sometimes. But I didn't need the instructions because I preferred trial-and-error."*

He was one of the most skillful participants from the beginning. His CPMs were 13.2 in the first 5 sentences and 19.5 in the final 5 sentences. This comment indicates that the assistive typing practice application is not needed for users who prefer trial-and-error and have high proficiency. Therefore, providing instructions that consider skill level is essential. Skill estimation is one of the future tasks.

Finally, the next goal, which is providing support for free text input, is noted. Some participants commented:

> *"I want to learn to type in free text input practically rather than in the typing application."*

It is difficult to estimate what may be typed next with free text input, unlike with the presented sentences in this study. Therefore, further development of the input stumble detection algorithm is needed.

3.8 Conclusion of Chapter 3

This chapter proposed an assistive typing practice application for text presented entry that detected input stumbles and provided instructions to help users to resolve stumbles on their own, instead of receiving instructions from a human tutor. First, data including operational input stumbles and effective instructions were collected through a user study that included thirty participants. Next, the acceptability being provided with instructions for text entry is confirmed using Wizard-of-Oz (WoZ). Then, a prototype of the assistive typing practice application was created based on the previous two studies. An evaluation experiment with novice older adults (60+) showed that the assistive typing practice application increased characters per minute (CPM) by 17.2% and reduced the stumble ratio with respect to chances for stumbles by 59.1% compared with users' initial rate. Improvement rates were almost the same as those achieved with human tutors. The subjective assessment and the interview showed that the assistive typing practice application was generally-supported.

Chapter 4

Typing Tutor: Individualized Tutoring for Free Text Input

4.1 Introduction

In the previous chapter, an assistive typing practice application for software keyboard was constructed. In this chapter, a tutoring system named Typing Tutor that help to resolve user's input stumble for free text input is proposed. First, the common difficulties that novice older adults encounter are clarified in free text entry, and the relationship between the input stumbles and the user's skill is analyzed. In addition, an individualized tutoring system which detected input stumbles using a statistical approach with a 12-key layout keyboard for Japanese is developed. Moreover, the applicability of the input stumble detection method to other keyboards and languages with a QWERTY layout for Japanese and English are demonstrated.

There are five steps to construct Typing Tutor. First, a user study to clarify the problems in text entry is conducted. Second, the structure of Typing Tutor on the basis of the previous user study is designed. Then, Typing Tutor is improved through two trials. Subsequently, the performance of Typing Tutor is evaluated in a two-week experiment. Finally, the applicability of the statistical input detection to other keyboards and languages is evaluated with a QWERTY layout for Japanese and English.

The contributions of this chapter are three-fold. The first is to clarify the common difficulties that novice older adults encounter in text entry, and to analyze the relationship between the input stumbles and the user's skill. The second is to develop statistical input stumble detection. The thrid is to develop an individualized tutoring system based on user studies.

4.2 User Study to Clarify How Novice Older Adults Make Input Stumbles

This user study has two purposes. One is to clarify how novice older adults make input stumbles in a smartphone text entry, and analyze the relationship between the input stumbles and the user's skill. Second is to discover and evaluate the necessity of support functions which should be implemented in Typing Tutor.

4.2.1 Participants

Thirty-five participants took part in this experiment. Five were colleagues who had advanced IT skills, four males and one female, ranging in age from 29 to 48 years with a mean age of 36.6 (sd = 6.6). The other thirty were older adults who were recruited from a local social institution. There were fifteen males and fifteen females, ranging in age from 60 to 83 years with a mean age of 72.1 (sd = 8.2). None of them had any previous experience of using a smartphone, but had owned a feature phone for more than one year, i.e., one with a physical keyboard with a 12-key layout. They were familiar with this kind of keyboard because the same layout has been used in feature phones for a long time. Twenty-six of them had often communicated via e-mail, while the others had seldom done so, using the phone only for making phone calls. Twenty-eight had previously entered text with PCs. Ten had used their own PC routinely. None of them had a tremor disorder, eye problems or other relevant health problems.

4.2.2 Apparatus

The software keyboard of the smartphone had a 12-key multi-tap layout as shown in Figure 3.1 which is the same layout of the experiment of chapter 3. The smartphone recorded all touch events in an input log using the standard Android API, plus all linguistic information, e.g., typed keys, displayed characters, and suggestions. All participant operations were recorded by an overhead video camera and were displayed on a monitor that a human tutor observed.

All colleagues and the fifteen older adult participants used a Samsung Galaxy S3 running Android 4.1.2 with a 4.8-inch screen having a resolution of 1280 × 720 pixels (306 ppi). The other fifteen used an Asus ZenFone5 whose size and weight are similar to that of a Galaxy S3, running Android 4.4 with a 5.0-inch screen having a resolution of 1280 $\times$ 720 pixels (294 ppi). The difference in the two conditions was due to a delay in the study. However, it is assumed that there would be little influence on cognitive factors because the shapes of these mobile phones are similar and layouts of the keyboard and the applications are same.

4.2.3 Procedure

First, the participants were given explanations and examples of how to operate a smartphone, including touch and swipe operations, by a human tutor. Next, they were instructed on how to use the software keyboard, including the correspondence between each key and a character, and how to select suggestions.

Then, they typed ten sentences in Japanese using an e-mail application. Participants created the sentences as a response to given sentences, such as "You expect to be ten minutes late for a meeting with a friend. Please create a sentence to let him know." and "Please create a sentence to ask when your son or daughter will arrive home." They operated the smartphone while holding it in their hand and sitting on a chair. Participants were instructed to type by themselves if at all possible, with no help from the human tutor who sat next to them. However, they were permitted to ask the human tutor only when they lacked the confidence on what to do next. The experimental period was limited to 60 minutes. After the experiment, they took part in an interview.

4.2.4 Annotation

First, three annotators independently extracted the pattern of input stumbles from the logs and the videos of the study, and based on the discussion classified the input stumbles into 30 classes. The 30 classes of input stumbles were grouped into 10 categories based on similarities. Table 4.1 shows the list of classified input stumbles along with the categories. Next, the three annotators labeled input stumbles and possible stumble opportunities according to the logs of each input. The possible stumble opportunity was labeled when an annotator judged that an input stumble was likely to occur based on the time series information before an input, such as the unfixed string and the elapsed time from the previous input. The labels given by at least two annotators were adopted. The concordance rate using Fleiss' kappa [25] was 0.44. The reason for the low kappa value was that occasionally there were conflicts in determining when an input stumble started even if annotators labeled the same input stumble as belonging to the same

Table 4.1: Input stumble categories; the asterisk mark (*) shows the particular input stumble in Japanese.

Input stumble category	Input stumble class
(1) Typing characters allocated to the same key consecutively	(1-1) Typing characters allocated to the same key consecutively
(2) Selecting a suggestion	(2-1) Not selecting despite displayed the desired word (2-2) Not displayed the desired word due to a long input (2-3) Not displayed the desired word because there are too many suggestions (2-4) Not knowing how to manipulate the suggestion list to find the desired word (2-5) Not knowing how to select a word in the suggestion list in kana-kanji convert mode* (2-6) Not displayed the desired word due to a long input in kana-kanji convert mode* (2-7) Not displayed the desired word because there are too many suggestions in kana-kanji convert mode* (2-8) Not knowing how to switch the modes*
(3) Enter Key	(3-1) Fixing the sentence (3-2) Linefeed
(4) Deleting character	(4-1) Deleting the last character (4-2) Deleting the character before several characters
(5) Switching keyboard modes	(5-1) Switching the keyboard from kana mode to alphabetic or numeric mode (5-2) Switching the keyboard from alphabetic or numeric mode to kana mode
(6) Moving cursor	(6-1) Moving cursor to the previous position (6-2) Moving cursor to the end of sentence (6-3) Moving cursor in order to change target of kana-kanji conversion*
(7) Entering a diacritic	(7-1) Not knowing how to select the diacritic (7-2) Selecting the symbol key by mistake (7-3) Selecting the case conversion by mistake
(8) Case conversion	(8-1) Case conversion of alphabetic character (8-2) Case conversion of kana character*
(9) Entering a symbol	(9-1) Not knowing how to select a period or exclamation mark (9-2) Not knowing how to select a question mark (9-3) Selecting the diacritic key by mistake
(10) Other	(10-1) Touching multiple keys simultaneously (10-2) Failure of key response due to short-duration touch (10-3) Over toggling (10-4) Not knowing how to type emoji (pictorial symbols)

Table 4.2: The distribution of labeled skill levels for the initial and final sentences of the participants

Skill level	**1**	**2**	**3**	**4**	**5**
Initial sentence	8	11	9	3	4
Final sentence	7	8	11	6	6
All sentence	56	96	93	43	47

operation. The kappa value of seven classes (4-1), (4-2), (5-1), (6-1), (6-2), (6-3), and (10-4) was under 0.20. The kappa value without the seven classes was 0.68.

At the same time, the logs for each sentence were also labeled according to five skill levels. To determine the criteria, the annotators watched a video in which two sentences were being typed and defined the skill level as being from one to five. A sentence of skill level one was typed at an average rate of 4.1 characters per minute (CPM) [4]. On the other hand, that of skill level five was typed at an average rate of around 20.3 CPM, which is comparable to the rate for an average smartphone user with this kind of keyboard. A sentence of skill level two-four was defined between one and five subjectively. The average rate of typing speed was 7.9 CPM for skill level two, 11.8 CPM for skill level three, and 16.4 CPM for skill level four. After watching the video, the annotators placed the sentences into one of five skill levels subjectively from the standpoints of the typing speed, the typing errors and the number of input stumbles. The labels given by at least two annotators were adopted. The concordance rate by Fleiss' kappa was 0.72.

4.2.5 Results of the User Study

All colleagues and the twenty-eight older adults completed the task. The average completion time for colleagues and the older adults was 12.4 minutes (sd = 6.0) and 32.4 minutes (sd = 10.9), respectively. The other two participants did not complete the task within 60 minutes. These were the two who had had no previous experience in entering text with their own feature phones.

Table 4.2 shows the distribution of labeled skill levels for the initial and the final sentences of the participants. Most of the sentences are below level three, whereas the final sentences of all colleagues are at level five. Comparing the skill level for the initial

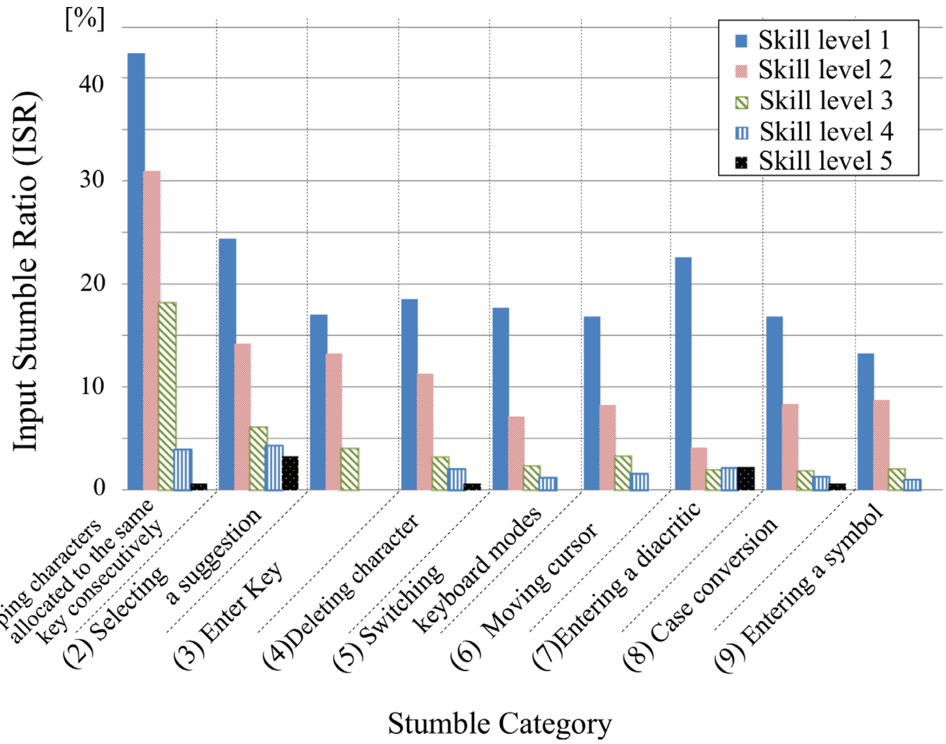


Figure 4.1: The relationship between the input stumble ratio (ISR) and the input stumble category for each skill level

sentences with that for the final sentences, the skill level generally improved. Figure 4.1 shows the relationship between the input stumble ratio (ISR) and the input stumble category for each skill level. The ISR was defined as the percentage of the number of input stumbles in the total of possible stumble opportunities for each category. The category "(10) Other" was excluded because it was difficult to define the possible stumble opportunity uniquely. The overall average of ISR, which is the percentage of the number of input stumbles in the total of possible stumble opportunities of all categories, was 8.3%. A Mann-Whitney U-test ($\alpha = 0.05$) did not show a significant difference between the two different smartphones.

Participants with higher-skills had a tendency of lower ISRs. The category with the highest ISR was "(1) Typing characters allocated to the same key consecutively". This means that when typing two characters allocated to the same key consecutively, the first character must be fixed by pressing the right arrow key before typing the second character. This operation is confusing because no character or string changes when this key is pressed. It was not until the human tutor had repeated instructions several times that many participants realized that they needed to press this key. The category with the second highest ISR was "(2) Selecting a suggestion". This category includes eight classes. These operations need deep knowledge, especially in Japanese because the suggestion list contains the results of auto-complete and kana-kanji conversion. Therefore, this category had the highest ratio among sentences whose skill level was four or five. Many participants found it difficult to judge the timing for selecting the desired suggestion. The ISR of "(3) Enter key" was higher than that of the overall average. The enter key has two functions allocated to it, (3-1) fixing the sentence and (3-2) linefeed. Some participants did not understand the difference so they became confused. On the other hand, the ISR of other operations were lower than that of the overall average participants had only to check the last character entered and the method of the next operation. Once they had received instructions, most of them were able to carry out these operations.

These results were also supported by comments of the interview after the tasks were completed.

> *"The tutor's instructions were helpful. But I sometimes couldn't understand why I should do something. If I'm not given the reasons, I will encounter the same problems again."*

> *"We [older adults] can use a smartphone if we get used to it. But it will be tough to keep using it without any support if we only receive instructions from the tutor on one occasion."*

The experiment extracted two things that need to be incorporated into the instructions provided through the tutoring system: One is that older adults need instructions to be repeated for more complex operations. The other is that they not only need instructions on how/what to type next but also the reasons and tips on how to perform these operations.

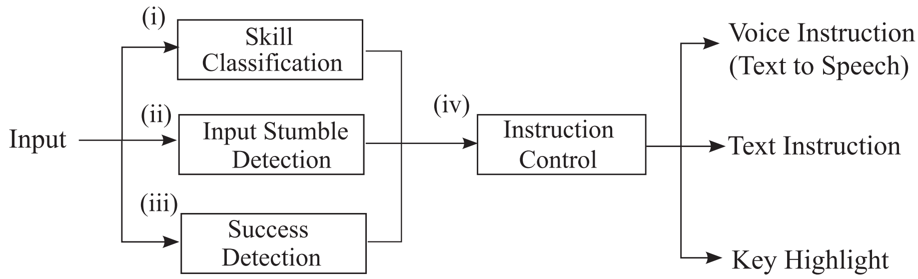


Figure 4.2: Block diagram of Typing Tutor

4.3 System Design of Typing Tutor

In this section, the system design of Typing Tutor based on the previous user study is described.

4.3.1 System Design

In the previous user study, the tendencies of an input stumble were quite different according to the skill level. In addition, older adults needed instructions to be repeated for more complex operations, and needed not only the instructions on how/what to type next but also the reasons and tips on the operations. Therefore, Typing Tutor is designed based on the following principles: The instructions should be provided based on the skill level and whether the user has previously succeeded or not. Typing Tutor should provide not only instructions on how/what to type next but also provide the reasons and tips on how to perform difficult operations.

Following these principles, Typing Tutor having four functions is designed as shown in Figure 4.2: (i) skill classification, (ii) input stumble detection, (iii) success detection and (iv) instruction control. First, Typing Tutor monitors the input data, such as the keys typed and suggestions. Then, the function of skill classification identifies the skill level. At the same time, the functions of input stumble detection and success detection work in concert. These functions detect whether the operation was a stumble or successful. Lastly, the function of instruction control determines whether the instruction should be provided or not, based on information collected through the other three functions. Specifically, Typing Tutor provides instructions even for easy operations when the skill level is low. In addition, no instructions are provided for easy operations that

Table 4.3: Accuracy of skill classification

Number of touches	Accuracy rate	Accuracy within ± 1
10	81.7	96.3
20	82.1	97.5
50	84.4	98.4
100	88.4	98.4

the user has already succeeded in doing without any instruction. This makes it possible to provide only those instructions that are needed by the individual user. The details of each function are described in this section.

4.3.2 Skill Classification

The skill classification model was constructed based on the data identifying the five skill levels. A linear support vector machine (SVM) classifier was trained based on the skill level data. The models were constructed for the number of touches of 10, 20, 50 and 100 from the beginning of typing. The features for the skill classification were nine dimensions as follows: the average and standard deviation of pressing time, those of the time required to type the same/different key, the number of input characters, the number of suggestions used, and the number times the delete key was used. The cross-validation (CV) for evaluation was conducted 35-fold for each user. As shown by the accuracy rate in Table 4.3, the greater the number of touches, the higher the accuracy rate. The accuracy rate of the 10-touch model was 81.7%. Accuracy within ± 1 skill level is greater than 95%, even for the 10-touch model.

4.3.3 Input Stumble Detection

Statistical models for input stumble detection were implemented in Figure 4.2 (ii). To select an appropriate learning machine, the performances of four methods were compared. The four methods were two kinds of SVM [9] (linear, RBF), a logistic regression, and a C4.5 [76]. The number of covered stumble classes was 23 after excluding 7 classes for which the kappa value was under 0.2 in the annotation as described in the previous section. Each model was trained by using the data labeled into 24 classes: the 23 stumble classes and no input stumble. Eighty-two features were obtained on the basis of the input sequence which includes unfixed string, fixed string, typing histories, and

Table 4.4: Classification performance of input stumbles detection for each model

Classification method	F-measure (Precision)
Linear-SVM	0.67 (0.89)
RBF-SVM	0.70 (0.90)
Logistic Regression	0.68 (0.88)
C4.5	0.66 (0.89)

suggestions. Features for adopting two SVMs and a logistic regression were selected by L1 regularization [9]. Those for a C4.5 were selected by stepwise backward selection of each participant 35-fold cross-validation (CV). Each method was evaluated by the F-measure from the 35-fold CV. Table 4.4 shows the performance of each model. The F-measure was 0.67 with linear-SVM, 0.70 with RBF-SVM, 0.68 with logistic regression and 0.66 with C4.5. One of the causes of the misdetection is that the concordance rate of the annotation is not high. Especially, annotations of "selecting a suggestion" tended to conflict in determining when an input stumble started. This reduces the accuracy of input detection. However, from the viewpoint of the timing of the instruction, this always do not make the usability reduce. this kind of misdetection always does not make the usability reduce.

Anova ($\alpha = 0.05$) showed no significant differences among these models, so RBF-SVM is adopted hereinafter, with the highest accuracy rate. The reason why the F-measure of RBF-SVM was the highest is that the linear classification is difficult to detect input stumbles because complicated features are needed to classify. The effective features adopted by RBF-SVM are shown in Table 4.5.

Figure 4.3 shows the relationship between the training data per participant and the F-measure. The performance of the model trained from the data of N persons was evaluated by the following procedure. First, the model was trained using the data of N persons, then the F-measure was calculated using the data of the remaining 35-N persons. Next, N variable selection was randomized and calculations were repeated a fewer number of times: all possible combinations of N or 5000 times. The F-measure of the model trained from the data of N persons was defined as the average value. In Figure 4.3, the F-measure increases as N increases. The F-measure exceeds 50% with the training data of 10 persons and becomes saturated with the training data of 20 persons.

Table 4.5: Ten effective features to classify input stumbles with RBF-SVM

ID	Features
1	The time interval between touches
2	The difference from the average in the time interval
3	The touch holding time
4	The type of current keyboard
5	The number of simultaneous touches
6	The type of typed key
7	The number of consecutive touches of the same key
8	Whether the final character can take diacritic
9	The type of morpheme of the last word
10	The number of words in an unfixed string

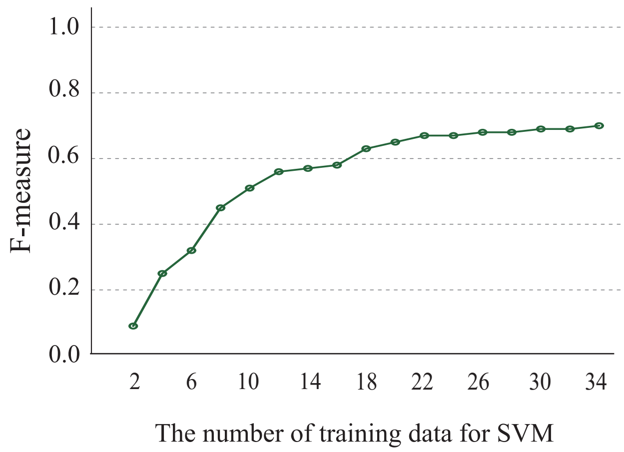


Figure 4.3: Relationship between F-measure of RBF-SVM and the number of training data

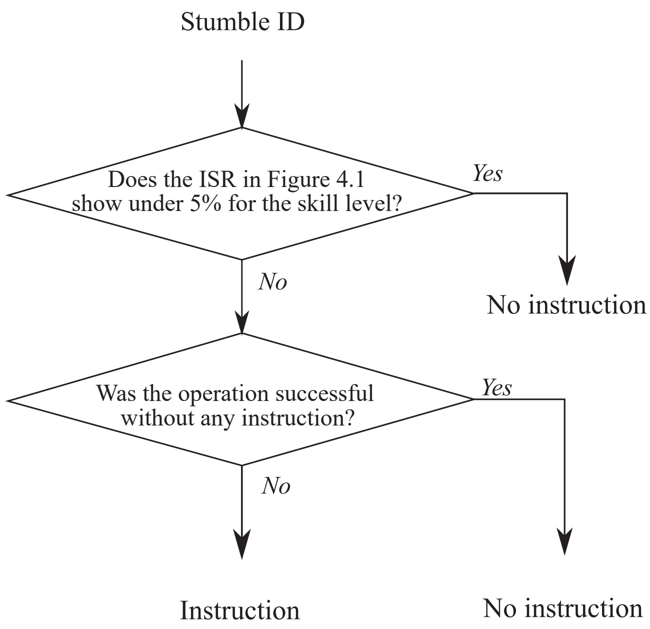


Figure 4.4: Flowchart for instruction control

4.3.4 Success Detection

This function detected successful operations after their completion, unlike input stumble detection. This after-the-fact characteristic made detection easier. However, a premature evaluation can lead to errors, so successful operations were registered only if the operation was not canceled before the input of the next two characters.

4.3.5 Instruction Control

This function determines whether the instruction on how to correct a detected input stumble is provided or not. The flowchart is shown in Figure 4.4. Even if an input stumble is detected, no instruction is provided when the relevant ISR in Figure 4.1 is under 5% for the user's skill level or when the success of the operation without any instruction was detected in the past. Otherwise, the instruction is provided. The instruction corresponding to each type of input stumble was prepared manually, based on the effective advice observed by human tutors in previous studies. Specifically, Typing Tutor provides only operating instructions for categories (4)-(10), but provides

Figure 4.5: Screenshot showing instruction provision

both instructions on the reason behind them and the tips for categories (1)-(3). Both text instruction and voice instruction are provided, plus overlaid key highlighting, the visibility of which was improved as shown in Figure 4.5.

4.4 Experiment to Improve Typing Tutor

In this section, the preliminary experiment of two trials designed to improve Typing Tutor is described.

4.4.1 Description of the Two Trials

Participants

In the first trial, six participants, three males and three females between the ages of 65 and 72 years with a mean age of 68.2 (sd = 2.9), took part. The second trial involved the same number of males and females between the ages of 66 and 74 years with a mean age of 69.4 (sd = 3.2). They were not participants in the other preliminary experiments, and were recruited from a local social institution. None of them had previous experience with a smartphone, but had owned a feature phone for more than one year. They had

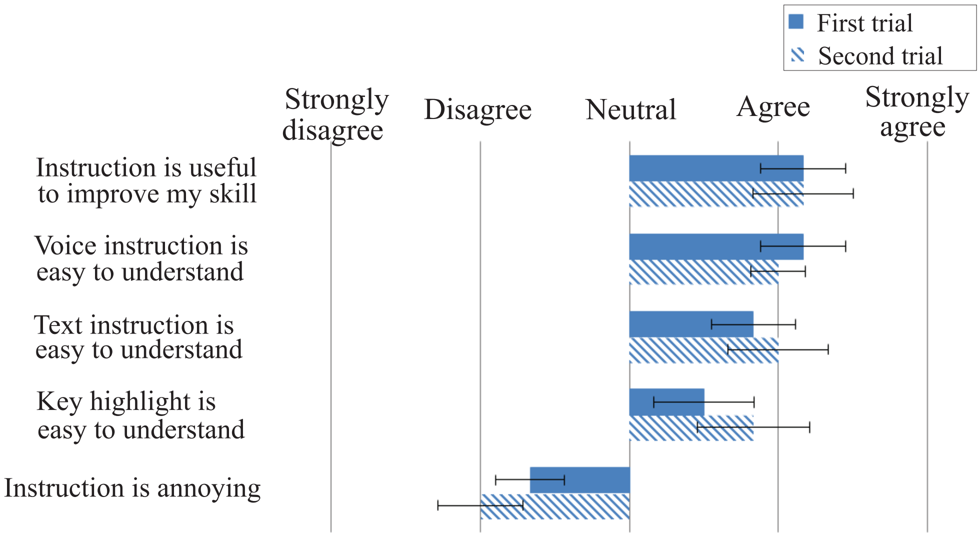


Figure 4.6: Questionnaire responses regarding the two trials

had previous experience in entering text using a feature phone. Four participants in each trial had used a PC to create documents when employed. Two in the first trial and three in the second trial had routinely used their own PCs. None had a tremor disorder, eye problems or other relevant health problems.

Procedure and Apparatus

The procedure and apparatus were the same in both trials. The tutorial in which the participants were shown how to use a smartphone and a keyboard was the same as that in the previous experiment. After the tutorial, they typed ten sentences as responses to given sentences, using an e-mail application. Participants in the first trial used Typing Tutor, which was described in the previous section. Participants in the second trial used Typing Tutor which was improved after the first trial. All of them made ten sentences as responses to given sentences related to personal conversations, just as in the previous user studies. The experiment period was limited to 60 minutes, with a one-minute rest after each sentence. After the experiment, they filled out a Likert-scale questionnaire and took part in an interview. An Asus ZenFone5 was used in both trials.

4.4.2 Results of the First Trial

All participants completed the task. The solid bars in Figure 4.6 show the results of the questionnaires administered in the first trial. There was a positive response to the item "Instruction is useful to improve my skill." The instruction method was also rated positively, with voice instruction the most favored. These results show that Typing Tutor was well accepted. The interviews yielded some important feedback:

> *"I am not confident that I can repeat an operation that I did today. I would like to receive instructions again."*
>
> *"I felt some instructions were annoying because I could understand where to type next, needing only the key highlight."*

These interviews indicated a need for two improvements: One is to repeat instructions sometimes even though participants have already succeeded in performing an operation. Another is to improve the handling of instructions for easy operations, which can be understood simply from the position and the label.

4.4.3 Improvement of Typing Tutor

The two points above, which were deemed to require improvement based on the first trial, were dealt with as follows.

Consideration of forgetting

Typing Tutor in the first trial did not provide further instructions after the participant had succeeded once. To allow the system to forget some successes, forgetting curves were used. This memory retention problem has been examined in many studies [3, 21, 97]. According to the review in [3], the major mathematical models can be classified into three categories: log-linear model, an exponential model, and hyperbolic model. The exponential model in [97] is used as follows:

$$f = \exp\left(-b_i\sqrt{t}\right) \tag{4.1}$$

Here f is the retention rate, t is the elapsed time since the last operation, i is the number of times the operation is executed and b_i is the coefficient difference based on i. In the system, the retention rates of each operation are computed, and when the retention rate falls below 50%, the instructions are provided again.

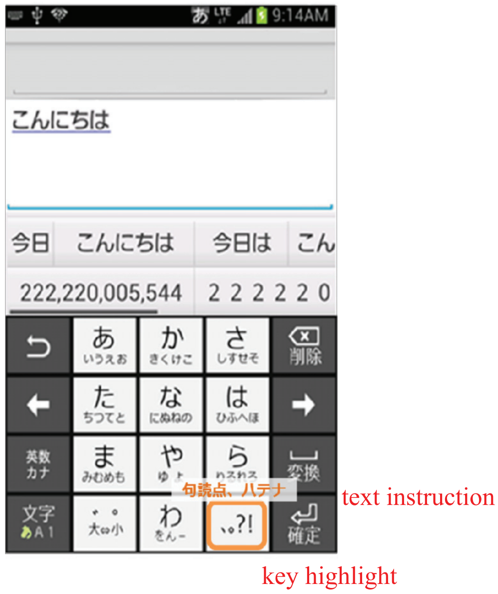


Figure 4.7: Screenshot of instruction for easy operations after improvement

Improvement of instructions for easy operations

Typing Tutor in the first trial provided instructions in a similar way regardless of the degree of difficulty of the operation. Therefore, the operations were divided into two difficulty levels by three annotators. The criterion was whether the operation could be understood simply from an indication of the key position and a word. The input stumbles made when executing the easy operations are the stumble classes (4-1,2), (5-1,2), (7-1,2), (8-1,2) and (9-1,2,3) as shown in Table 4.1. The instruction for the operations was improved by providing the key highlight and text above the key highlight, as shown in Figure 4.7.

Modification of instruction control

To take the two improvements into consideration, the function of the instruction control was modified. Figure 4.8 shows the flowchart of the instruction control. Even if an input stumble is detected, the instruction is provided only when the relevant ISR in Figure 4.1 is not under 5% for the user's skill level and when the retention rate of the operation

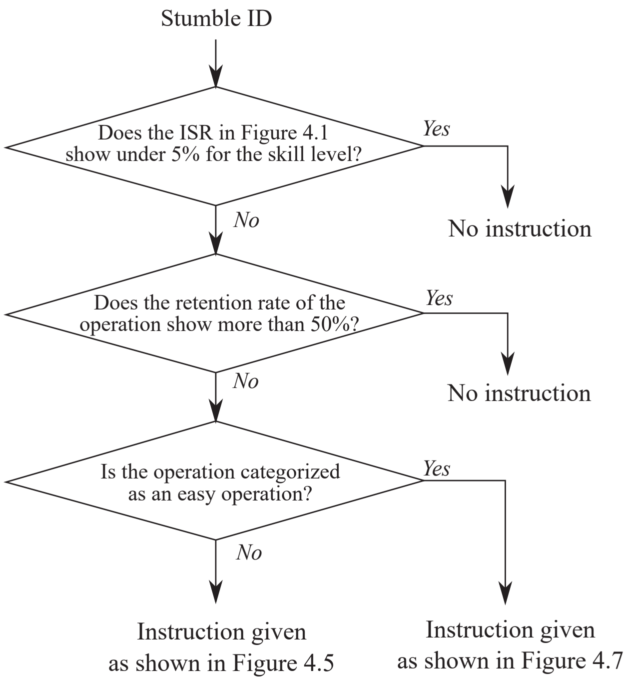


Figure 4.8: Flowchart for instruction control improved after the first trial

falls below 50%. Moreover, the way of instruction varies depending on the difficulty of the operation.

4.4.4 Results of the Second Trial

All participants completed the task. The hatched bars in Figure 4.6 show the results of the subjective questionnaires administered in the second trial. Compared to the results of the first trial, there appeared to be an improvement for the items "Key highlight is easy to understand" and "Instruction is annoying", although a Mann-Whitney U test ($\alpha = 0.05$) did not show a significant difference. The half of the participants responded negatively to the voice instruction:

> *"The voice instruction was easy to understand. But I wouldn't like to use it outside because it embarrasses me."*

Therefore, Typing Tutor was altered to allow the choice of using voice instruction or not.

4.5 Two-week Evaluation Experiment

Finally, the performance of Typing Tutor is evaluated by means of a two-week evaluation experiment. In this experiment, the participants would chat for one hour with an operator using the smartphone's chat application.

4.5.1 Descriptions of the Evaluation Experiment

Participants

Twenty participants, eleven males and nine females between the ages of 65 and 74 years with a mean age of 68.4 (sd = 4.6) who did not participate in the other experiments took part in this experiment. They were recruited from a local social institution. Ten of the participants were provided instructions by Typing Tutor, the input detection model of which was trained by RBF-SVM with the data of 35 participants (group A). The other ten participants were not provided instructions (group B). However, the participants of group B were permitted to ask for advice on how to operate the smartphones through a chat application, but only if necessary. None of them had previous experience with a smartphone, but all had owned a feature phone for more than one year. All of them had entered text using a feature phone before and had often communicated via e-mail. Seven in group A and seven in group B had routinely used their own PCs. None of them had a tremor disorder, eye problems or other relevant health problems.

Procedure and Apparatus

First, the participants were given explanations and examples of how to operate a smartphone and how to use the software keyboard, just as in previous experiments. After that, they were taught how to use the chat application, "Hangout", which is the default chat application for Android. They kept the smartphone for two weeks (twelve days from Monday to the second Friday) and had a chat with an operator for one hour every day except Saturday and Sunday. The chat topics were not limited and included such subjects as hobbies, their work, and daily events. After two weeks, interviews were held. An Asus ZenFone5 was used in the experiment.

Table 4.6: The distributions of estimated skill levels on the first day and the final day for each group

Skill level		1	2	3	4	5	Average skill level
Group A	First day	1	6	2	1	0	2.3
	Final day	0	1	3	4	2	3.7
Group B	First day	1	4	3	2	0	2.6
	Final day	0	1	4	2	3	3.7

4.5.2 Experimental Results

All participants sent more than six messages each day. The daily average number of messages and the average length of a message are 8.5 (sd = 1.7) and 43.2 characters (sd = 12.3) for group A, and 8.4 (sd = 1.5) and 42.2 characters (sd = 11.0) for group B. The performance was evaluated by four metrics: the skill level, the number of input stumbles, the typing speed and the typing accuracy. Each metric was tested by a Mann-Whitney U-test ($\alpha = 0.05$) each day.

Classified Skill Level

Table 4.6 shows the distributions of the classified skill level on the first day and the final day for each group. On the first day, the skill levels were almost the same. On the final day, the skill level of group B was higher than that of group A. However, a U-test did not show a significant difference between the two groups.

Number of Input Stumbles

Figure 4.9 shows the number of input stumbles in both groups. At the beginning of use, the number of input stumbles was almost the same in all groups. After the second day, there were fewer stumbles in group A than in group B. A U-test showed a significant difference from the third day to the tenth day. The number of input stumbles on the eighth day was higher than on the fifth day because the participants had not used their phones on the previous two days.

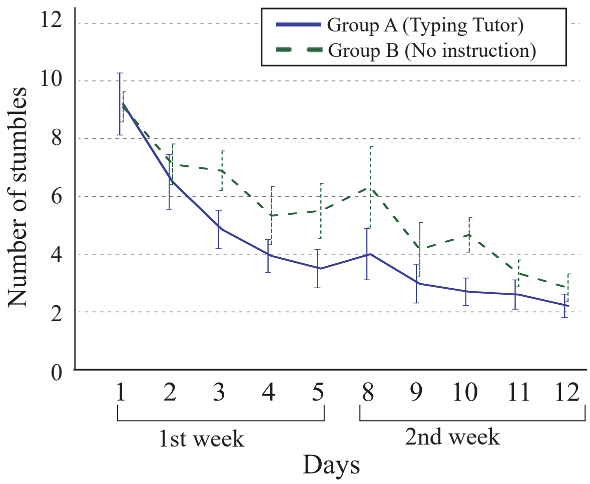


Figure 4.9: User progress in the number of input stumbles for each group; vertical bars show standard errors

Typing Speed

The typing speed was compared by the metric of characters per minute (CPM) [4] excluding the time used for instructions. Figure 4.10 shows the CPM of both groups over two weeks. The CPMs of both groups were almost the same on the first day. After that, the CPM of group A increased dramatically. In contrast, that of group B did not increase monotonically. This difference was maintained up until the final day. A U-test showed a significant difference except for the first day, the second day, and the fifth day.

Figure 4.11 shows the time taken by both groups to input character keys and non-character keys. The non-character keys refer to the function keys as shown in Figure 3.1 and the suggestion list, which requires a longer decision time to select than the character keys. The time taken by the two groups to input character keys decreases gradually from about 1500 ms to 1000 ms in a similar manner. However, the time taken by group A to input the non-character keys decreased markedly compared with group B, although times on the first day and the final day were almost identical in both groups. A U-test showed that a significant difference existed for each day from the third day to the eighth day.

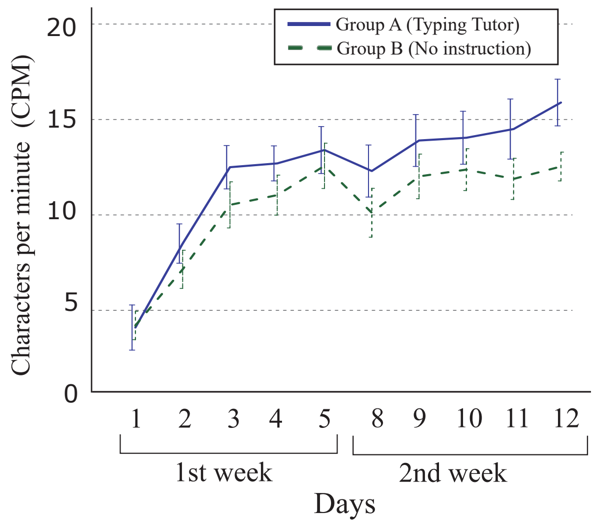


Figure 4.10: User progress in typing speed for each group; vertical bars show standard errors

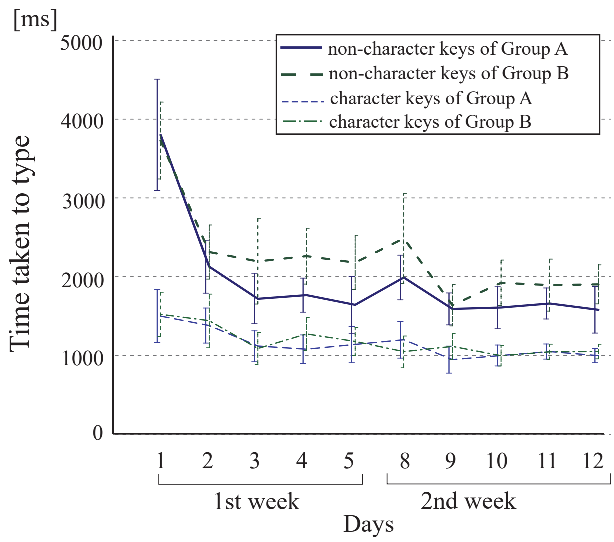


Figure 4.11: Time taken to type character keys and non character keys for each group

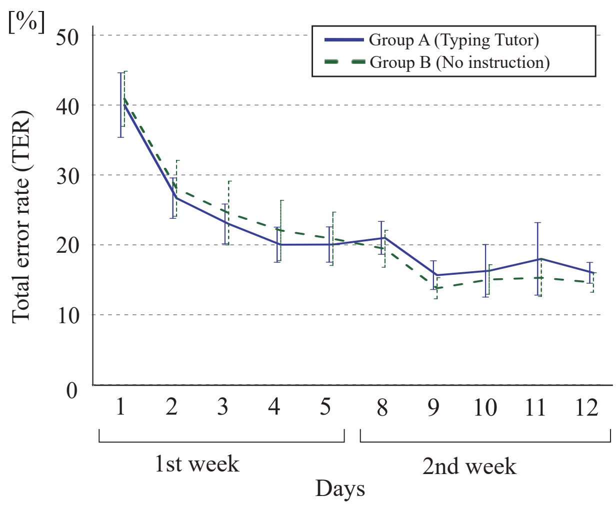


Figure 4.12: User progress in total error rate for each group; vertical bars show standard errors

Typing Accuracy

Typing accuracy was compared by the metric of total error rate (TER) [4]. Although TER is the metric for typing presented sentences including error correction, sent sentences were used as presented sentences. The difference between the two groups was negligible as shown in Figure 4.12. A U-test did not show a significant difference between the two groups. The metric is affected by a number of physical factors such as mistyping rather than by cognitive factors. Therefore, this metric is outside the scope of the evaluation of Typing Tutor.

4.6 Applicability to Other Keyboards and Languages

In the previous section, the effectiveness of Typing Tutor was confirmed. However, the applicability to other keyboards and languages was not taken into consideration. The skill classification is not heavily dependent on keyboards and languages because the features are not language dependent. On the other hand, input stumble detection is influenced by keyboards and languages because some features are language dependent. Therefore, the applicability of the input stumble detection is examined focusing on the QWERTY keyboard layout for Japanese and English.

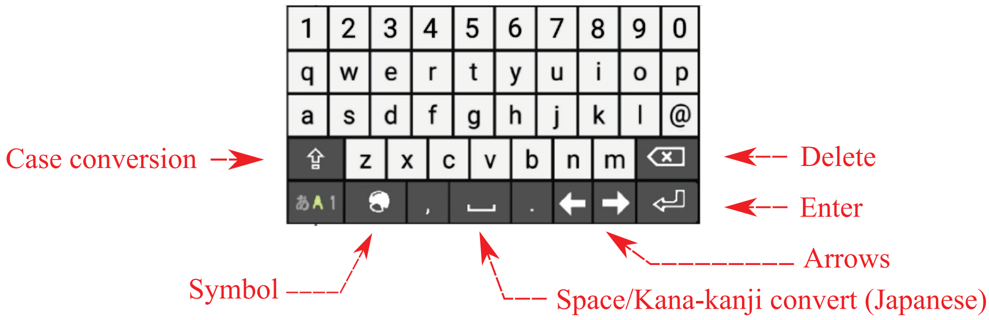


Figure 4.13: Key layout of QWERTY keyboard

4.6.1 Participants

Twenty-four participants took part in this experiment. Twelve of them, six males and six females between the ages of 61 and 77 (mean 68.5, sd = 4.9), typed Japanese with a QWERTY. The other twelve, six males and six females between the ages of 60 and 73 (mean 65.7, sd = 3.8), typed English with a QWERTY. They had used English for work such as composing documents when they were still employed. All of them were recruited from a local social institution. None of them had any previous experience of using a smartphone, but had owned a feature phone for more than one year. All of them had often communicated via e-mail with a feature phone and had previously entered text with PCs. Eight had used their own PC routinely. None of them had a tremor disorder, eye problems or other relevant health problems.

4.6.2 Procedure and Apparatus

The experiment was conducted in a similar way to the experiment in Section 4.2. In brief, first, the participants were given explanations and examples of how to operate a smartphone and how to use the software keyboard. Using a smartphone and while sitting on a chair, they created ten sentences in the form of a response to given sentences. They could rest for one minute after writing each sentence. They were instructed to type by themselves if at all possible. An Asus ZenFone5 was used in the experiment. The software keyboard of the smartphone had a QWERTY layout as shown in Figure 4.13.

Japanese characters are typed by combinations of a couple of Roman alphabet letters as shown in Table 3.1 in chapter 3. English letters are typed by tap input, which means one touch corresponds to one letter. The smartphone recorded all touch events using the standard Android API and all linguistic information. All participant operations were recorded by an overhead video camera.

4.6.3 Annotation

All of the older adult participants completed the task. The average completion time was 28.5 minutes (sd = 8.5) with a QWERTY Japanese keyboard and 40.5 minutes (sd = 15.0) with a QWERTY English keyboard.

Similar to the experiment in Section 4.2, three annotators independently extracted the pattern of input stumbles from the logs and the videos of the study. The input stumbles were classified into 22 classes under 7 categories for the QWERTY Japanese keyboard and 12 classes under 8 categories for the QWERTY English keyboard. Table 4.7 and Table 4.8 show the list of classified input stumbles along with the categories. Then, the three annotators labeled input stumbles according to the logs of each input. The labels given by at least two annotators were adopted. The concordance rate using Fleiss' kappa [25] was 0.54 in QWERTY Japanese and 0.42 in QWERTY English. The kappa value of input stumbles (3) and (4) for both keyboards was under 0.20. The kappa value without the input stumbles was 0.70 for QWERTY Japanese and 0.63 for QWERTY English.

Focusing on the differences in the keyboards by comparing the input stumbles for the 12-key layout in Table 4.1 with the QWERTY in Table 4.7, there were a lot of common points between the two keyboards. This was because the basic concept of operation was similar due to the same language. The number of stumble classes for the QWERTY layout was less than that of the 12-key layout. This was attributed to the fact that it is simpler to execute operations using the QWERTY layout than it is using the 12-key layout. Next, the differences in language are focus on by comparing input stumbles in Japanese (Table 4.7) and English (Table 4.8). Japanese was associated with a number of input stumbles in terms of selecting suggestions because Japanese needs kana-kanji conversion. English had the unique stumble of not inserting a space between words.

Table 4.7: Input stumble category by QWERTY Japanese; the asterisk mark (*) shows the particular input stumble in Japanese.

Input stumble category	Input stumble class
(1) Selecting a suggestion	(1-1) Not selecting despite displayed the desired word (1-2) Not displayed the desired word due to a long input (1-3) Not displayed the desired word because there are too many suggestions (1-4) Not knowing how to manipulate the suggestion list to find the desired word (1-5) Not knowing how to select a word in the suggestion list in kana-kanji convert mode* (1-6) Not displayed the desired word due to a long input in kana-kanji convert mode* (1-7) Not displayed the desired word because there are too many suggestions in kana-kanji convert mode* (1-8) Not knowing how to switch the modes*
(2) Enter Key	(2-1) Fixing the sentence (2-2) Linefeed
(3) Deleting character	(3-1) Deleting the last character (3-2) Deleting the character before several characters
(4) Moving cursor	(4-1) Moving cursor to the previous position (4-2) Moving cursor to the end of sentence (4-3) Moving cursor in order to change target of kana-kanji conversion*
(5) Case conversion	(5-1) Case conversion of alphabetic character (5-2) Case conversion of kana character*
(6) Entering a symbol	(6-1) Not knowing how to select a period or exclamation mark (6-2) Not knowing how to select a question mark (6-3) Selecting the diacritic key by mistake
(7) Other	(7-1) Touching multiple keys simultaneously (7-2) Failure of key response due to short-duration touch

Table 4.8: Input stumble category by QWERTY English.

Input stumble category	**Input stumble class**
(1) Selecting a suggestion	(1-1) Not selecting despite displayed the desired word (1-2) Not displayed the desired word due to a long input
(2) Enter Key	(2-1) Linefeed
(3) Deleting character	(3-1) Deleting the last character (3-2) Deleting the character before several characters
(4) Moving cursor	(4-1) Moving cursor to the previous position (4-2) Moving cursor to the end of sentence
(5) Case conversion	(5-1) Case conversion of alphabetic character
(6) Entering a symbol	(6-1) Not knowing how to select a question mark
(7) Inserting a space	(7-1) Not inserting space between words
(8) Other	(8-1) Touching multiple keys simultaneously (8-2) Failure of key response due to short-duration touch

Table 4.9: Classification performance of input stumble detection

Keyboard	F-measure (Precision)
QWERTY Japanese	0.63 (0.88)
QWERTY English	0.50 (0.82)
(12-key Japanese trained by 11 person in Fig 4.3)	0.56 (0.87)

4.6.4 Performance of Input Stumble Detection

A model of input stumble detection for each language was trained by RBF-SVM. The number of stumbles was 17 for QWERTY Japanese and 8 for QWERTY English, after excluding the stumbles for which the kappa value was under 0.2 in the annotation. The number of features obtained from input was 79 for Japanese and 70 for English. The features included unfixed string, fixed string, typing histories, and suggestions. Next, the features were reduced to 28 for Japanese and 26 for English by L1 regularization.

Table 4.9 shows the performance of each model evaluated by 12-fold cross-validation. The F-measures were 0.63 for QWERTY Japanese and 0.53 for QWERTY English. Comparing the F-measure of these two keyboards with that of 12-key Japanese trained by 11 persons in Figure 4.3, the F-measure of QWERTY Japanese was significantly larger than that of 12-key Japanese by an ANOVA with post hoc Tukey's test ($\alpha = 0.05$).

4.7 Discussion

4.7.1 Two-week Evaluation

The number of input stumbles in group A decreased more rapidly than in group B for the first three days as shown in Figure 4.9. In a similar way, the CPM of groups A increased more rapidly as shown in Figure 4.10. However, there were no significant differences between the two groups in the first two days. This can be attributed to the fact that the participants of both groups were in the process of learning by trial and error during this time. In contrast, significant differences were seen from the third day to the final day. It is considered that the trial-and-error period continued in group B.

Regarding the time taken to type characters, a difference existed between the time taken to type character keys and non-character keys, as shown in Figure 4.11. The time taken to type non-character keys in group A decreased more rapidly than that in group B as the number of input stumbles decreased. Typing Tutor focused on support for these operations which have a cognitive dimension. These results show that Typing Tutor is an effective way to improve the skills of participants from a cognitive viewpoint. However, in terms of the time taken to type character keys, the differences between the two groups were negligible. This metric is affected by the ability to construct sentences. These results indicate that Typing Tutor did not affect the performance.

In the interview, participants of group A made similar comments:

> *"Initially, I thought that the smartphone operations were markedly different from those of my feature phone. But as I came to realize the differences and common points between my feature phone and the smartphone through the instructions, I gradually became able to operate the smartphone. "*

These comments indicate that the Typing Tutor works effectively for novice older adults.

Finally, the participants for whom not much effect was observed are focused on. One participant had a classified skill level of two on the first day, and his level remained the same until the final day. His CPM was 3.33 on the first day and 8.15 on the final day, which was the worst result among all participants. During the interview, he made the following comment;

> *"The touch operation doesn't suit me because a key that I didn't intend respond. I prefer a feature phone with a physical keyboard."*

His TER exceeded 32%, which was also the worst rate among all participants. This was caused by physical rather than cognitive factors. Hence, it can be concluded that Typing Tutor was not effective in this case.

4.7.2 Applicability to Other Keyboards and Languages

First, the results regarding the different keyboards are discussed. Table 4.9 shows that the accuracy of input stumble detection of the QWERTY for Japanese is higher than that of the 12-key layout for Japanese. This would appear to be because the number of classes for input stumble using the QWERTY is fewer than that for the 12-key layout and the operation of the QWERTY layout is simpler than that of the 12-key layout. For example, in order to input the kana "gu (ぐ)", with a 12-key layout, it is necessary to press the diacritic key after the kana "ku (く)". In contrast, with QWERTY, only "gu" needs to be typed using alphabetical letters.

Next, the reason why the accuracy of input stumble detection of QWERTY for English is lower than that of the other two keyboards in Table 4.9 is discussed. One reason is the difference in the sentence structure between Japanese and English. Some input stumbles concentrate around the end of a sentence. The order of Japanese is subject-object-verb and Japanese does not have that many expressions that appear at the end of the sentence. This makes it easy to predict the end of a sentence. By contrast, the order of English is subject-verb-object, and the object can be post-modified. Therefore it can be difficult to predict the end of a sentence without information on the syntactic dependency or grammar, e.g., whether the morpheme at the end of a sentence is grammatically correct or not, which was not incorporated as an SVM feature in this study. The other reason is that the lower concordance rate of Fleiss' kappa was responsible for the lower accuracy of stumble detection. Although the annotation was controlled based on the discussion in advance, inconsistencies arose because the three annotators were non-native English speaker. On the other hand, as shown in Figure 4.3, the F-measure improved as the training data increased in the 12-key layout for Japanese. This means that the F-measure of other keyboards and languages may have the same tendency. These indicate that Typing Tutor for other keyboards and languages could be also effective.

4.8 Limitations and Future Work

In terms of the detection of input stumbles, the F-measure was around 0.7 in this study. It should be possible to improve the precision, but it is difficult to detect input stumbles with a perfect accuracy. Therefore, providing instructions in a way that does not interrupt the user's actions is essential. In addition, there is a possibility that the acceptability of the way instructions are given depends on the cultural background of the user. Thus, the acceptability of Typing Tutor for people in other cultures is needed to research.

On the other hand, Typing Tutor does not focus on problems by physical factors, e.g., the mistyping, even though this is also a serious problem for older adults [70]. In the field of text entry, many studies have focused on making it easier to type text by adjusting key target areas or presenting suitable suggestions [8, 34, 82]. It should be possible to support both cognitive and physical factors by combining Typing Tutor with these techniques. This is the issue to be addressed in the future.

4.9 Conclusion of Chapter 4

In this chapter, Typing Tutor, a individualized tutoring system for text entry that automatically detects input stumbles and provides individualized instructions for older adults, was presented. Through a user study, the common difficulties that older adults experience and how a user's skill level is related to input stumbles using a 12-key layout for Japanese were clarified. Based on the study, Typing Tutor was developed. A two-week evaluation experiment with novice older adults (65+) showed that Typing Tutor was an effective way to improve their proficiency in the area of text entry, especially in the initial stage of use compared with instructions through a chat application online. It was particularly beneficial for users who had previously used a feature phone but had trouble with self-instruction. In addition, the applicability of statistical input stumbles detection to other keyboards and languages was evaluated. From the result, the performance of the input stumble detection of QWERTY layout for Japanese and English was shown to be almost same as that of a 12-key layout for Japanese.

Chapter 5

Tutoring System for Voice Input

5.1 Introduction

Chapters 3 and 4 described tutoring systems for input using a software keyboard. On the other hand, as described in Chapter 2, voice input is also an important text input option for smartphone novice older adults. This chapter describes a tutoring system for voice input that detects input stumbles using a statistical approach, and then provides instructions that help users resolve input stumbles by themselves. There are three steps to develop the tutoring system. First, a user study that clarified the problems that older adults encounter with voice input is conducted. Second, the structure of the tutoring system on the basis of the user study is explained. Finally, the performance of the resulting tutoring system is evaluated.

5.2 Related Work

5.2.1 Interface Design for Voice Input

A book [12] described that the interface is one of the most critical factors in the success of any automated speech recognition (ASR) system, determining whether the user experience will be satisfying or frustrating, or even whether the user will remain a user. At present, ASR is still imperfect, always producing occasional misrecognitions, although many researchers have worked to improve its performance for not only general users [46, 60, 77, 65, 68, 61] but also for older adults [5, 93]. Therefore, some research

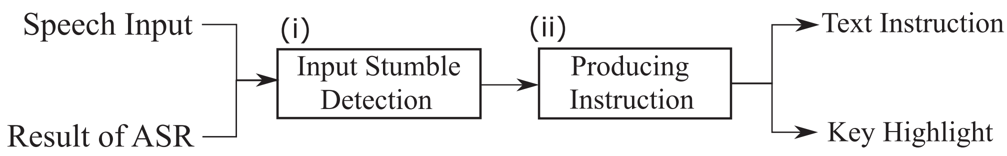


Figure 5.1: Block diagram of the voice input tutoring system

has been proposed for voice input interfaces to include the assumption that misrecognition occurs. Goto *et al.* [31] proposed a system to alter a sentence when users hesitate by lengthening a vowel during a phrase. Ogata *et al.* [72] proposed a system to display the result of ASR as a sentence that includes alternative word candidates. A user who finds a recognition error can just select the correct word from the candidates. Similarly, Liang *et al.* [62] proposed a simple gesture-based error correction interface where a user marks the error region once, and then the error region is replaced by the top candidate. Speech Recognizer for Android[1], an application programming interface (API), gives feedback on the results of errors, such as audio recording errors and network-related errors. This function mainly gives feedback on the cause of equipment-side rather than user-side problems.

5.3 Tutoring System for Voice Input

Although many studies deal with ASR, tutoring systems for voice input have not been studied. Therefore, a tutoring system for voice input is proposed. As shown in Figure 5.1, the tutoring system is composed of two functions, (i) input stumble detection and (ii) producing instruction. First, the tutoring system monitors the input data, such as the speech input and the result of ASR. Then, the function of input stumble detection detects whether the operation was a stumble based on a model trained by features such as acoustics and results of ASR. Finally, the function that produces instruction displays the instructions for the next operation, using text and key highlighting.

[1] https://developer.android.com/index.html

5.4 User Study to Clarify How Novice Older Adults Make Input Stumbles

This study has two purposes: One is to clarify how novice older adults make input stumbles in smartphone voice input. Second is to collect the input data to construct a tutoring system as described in the next section. In this study, participants first input a statement by voice, and then are permitted to modify what was recognized, using a combination of voice and touch.

5.4.1 Participants

Ten older adults were recruited from a local social institution to take part in this experiment. There were five males and five females, ranging in age from 65 to 72 years, with a mean age of 69.8 (sd = 3.8). None of them owned a smartphone but all had owned a featurephone and had used their own PC routinely. None of them had tremor disorders, eye problems, or other relevant health problems.

5.4.2 Apparatus

The Google Cloud Speech API[2] was used for ASR. As shown in Figure 5.2, a key for ASR was implemented on a software keyboard. After the key is pushed, a tone acknowledges the push, and the end of the tone announces the start of ASR. ASR terminates when the API detects the end of the speech or when five seconds have passed without input. Then Google Cloud Speech API outputs the N best candidates of recognition results and the highest confidence score between 0 and 1. The smartphone records the user's voice and the keys touched. All operations were also recorded by an overhead video camera. Participants used a Nexus 6 running Android 7.0.

5.4.3 Procedure

First, the participants were given explanations on how to operate a smartphone, including touch and swipe operations, by a human tutor. Next, they were instructed on how to use the voice input and the software keyboard. Then, they input sentences to match

[2]https://cloud.google.com/speech/

Figure 5.2: Screen of an input application and a software keyboard

a sentence presented in Japanese, using an application as shown in Figure 5.2. The sentences were selected from an email corpus collected originally from personal conversations via email. The corpus contained roughly 30,000 sentences (average character length = 23.6).

Participants were asked to input twenty sentences within 60 minutes, with a ten-minute break after completing ten sentences. They operated the smartphone while holding it in their hand and sitting on a chair. They were instructed to input by themselves if possible. However, they were permitted to ask the human tutor when they lacked the confidence to perform the next action. After the experiment, they took part in an interview.

5.4.4 Annotation

After all the participants had completed the experiment, three annotators independently extracted the patterns of input stumbles from the logs and the videos of the study and defined categories of the stumbles based on discussion. Next, the annota-

Table 5.1: Input stumble of voice input

	Input Stumble
(1)	Volume was too high for voice recognition
(2)	Volume was too low for voice recognition
(3)	Utterance started before ASR began working
(4)	Filler was inserted
(5)	Utterance paused due to long utterance
(6)	Intended homonyms were not displayed due to short utterance
(7)	Not knowing how to insert punctuation mark
(8)	Not knowing how to insert question mark
(9)	Not knowing how to select candidates
(10)	Not knowing how to delete a word
(11)	Forgetting to move cursor to end of sentence

tors labeled input stumbles of each input sentence. The labels given by at least two annotators were used in the next stage.

5.4.5 Results of the User Study

All participants completed the task, taking an average time of 37.3 minutes (sd = 7.4). Voice input stumbles fell into 11 categories, as shown in Table 5.1. Each input stumble is explained as following: "(1) Volume was too high for voice recognition" is a stumble that ASR does not cope with well due to sound distortion by loud voice input. On the other hand, "(2) Volume was too low for voice recognition" is a stumble where the volume of the voice is too low to be recognized voice through a microphone. "(3) Utterance started before ASR began working" is a stumble where the user starts an utterance before the end of the tone signals the start of ASR. "(4) Filler was inserted" is a stumble where fillers such as "Well.." or "Uhm.." are unconsciously being made during the utterance. "(5) Utterance paused due to long phrase" is a stumble where ASR terminates in the middle of an utterance since the phrase is too long. "(6) Intended homonyms were not displayed due to short utterance" occurs because there are a lot of candidates. "(7) Not knowing how to insert punctuation mark", "(8) Not knowing

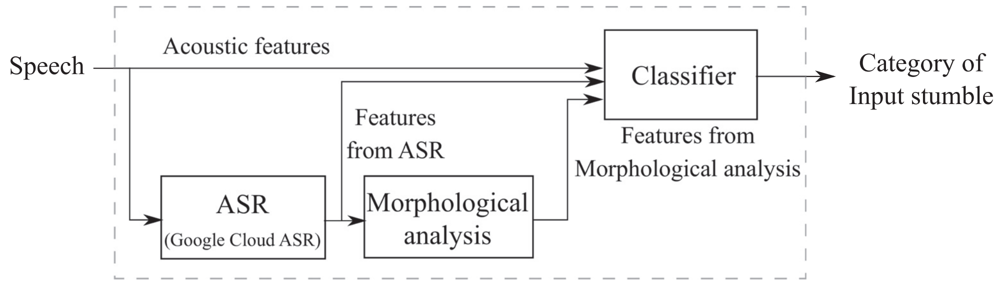


Figure 5.3: Block diagram of input stumble detection

how to insert question mark", "(9) Not knowing how to select candidates", "(10) Not knowing how to delete a word", and "(11) Forgetting to move cursor to end of sentence" are stumbles where the user forgets or does not know what to do. Input stumbles (1) to (6) are related to voice input. On the other hand, input stumbles (7)-(11) are related to operation after voice input, such as correction of input word and selection of candidates. The concordance rate of the annotation of the input stumble by Fleiss' kappa was 0.79.

In the interview, many comments similar to these were made:

> *"It was easier to input by voice than touch, but it was difficult to correct it when I made mistakes. So I wanted to know how I should correct and why I failed."*

This study showed that older adults have some difficulties with voice input and need instructions.

5.5 System Design of Tutoring System for Voice Input

In this section, the construction of the two functions of the tutoring system for voice input is described.

5.5.1 Input Stumble Detection

Input stumble detection was constructed by machine learning based on data from the previous user study. The block diagram of input stumble detection is shown in Figure 5.3. First, acoustic features were extracted from speech input. Simultaneously,

Table 5.2: Classification performance of input stumbles detection for each model

Classification method	F-measure (Precision)
Linear regression	0.70 (0.80)
C4.5	0.73 (0.83)
RBF-SVM	0.72 (0.82)
DNN	0.73 (0.83)

features from ASR including the confidence score and the candidates were output from Google Cloud ASR. Next, the features from the morphological analysis of the results of ASR were extracted using Lucene-gosen[3]. Finally, the classifier of input stumbles was trained using the features and the labels.

To select the best machine learning algorithm, the performance of four models was compared, a linear regression, a C4.5, a support vector machine (SVM) using RBF kernel, deep neural network (DNN) [88]. A DNN with five layers was trained by minimizing loss function with backpropagation and the stochastic gradient descent method with a dropout ratio of 50%.

A total of 28 features was used, 17 of which were acoustic features: the number of amplitude overflows, the duration of voice activity, five kinds of amplitude averages, namely the amplitude average in all voice activity, 0.5 sec before and after the start and the end of voice activity, and the differences in the five kinds of averages. The remaining 11 dimensions of the features were textual features related to the ASR and morphological analysis results: the number of candidates, the confidence score, the length of the sentences, the number of morphemes, the ID of morphemes of sentence terminations, the number of conjunctions, the number of end-forms, the number of fillers, the number of homonyms of the shortest words (except particles), the time to the next operation, and the position of the cursor.

Features used for SVM and logistic regression were selected by L1 regularization. Those for a C4.5 were selected by stepwise backward selection of each participant with 10-fold cross-validation (CV). Table 5.2 shows the performance of each model. The F-measure from the 10-fold CV was 0.70 with a linear regression, 0.73 with C4.5, 0.72 with SVM and 0.73 with DNN. Anova ($\alpha = 0.05$) showed no significant differences among

[3]https://github.com/lucene-gosen

Table 5.3: Effective features to classify voice input stumbles with C4.5

ID	Features
1	The amplitude average in all voice activity
2	The differences between the amplitude average of 0.5 sec before and after the start of voice activity
3	The number of amplitude overflows
4	The number of candidates
5	The number of morphemes
6	The ID of morphemes of sentence terminations
7	The number of end-forms
8	The number of fillers
9	The number of homonyms of the shortest words (except particles)
10	The time to the next operation

these models, so C4.5 is adopted hereinafter, as it had the highest rate of accuracy. The effective features adopted by C4.5 are shown in Table 5.3.

Figure 5.4 shows performance in terms of the detection of each input stumble with C4.5. The F-measures for stumbles after voice input of (6)-(11) were lower than that of stumbles of voice input of (1)-(5). It is considered that one of the reasons is that the concordance rate of the annotation is lower. The average concordance rate of input stumbles (6)-(11) was 0.58. In particular, the rate of the input stumbles (6) and (10), which were those having the lowest F-measure, were 0.40 and 0.33, respectively. This means that it is difficult to define the timing of these input stumbles even for humans.

5.5.2 Producing Instructions

The definition of instructions to be presented to users was done manually, based on the observation of effective instructions given by human tutors. The instruction was provided by text and with overlaid key-highlighting. The instruction provided for the input stumble of “(7) Not knowing how to insert a punctuation mark” is exemplified in Figure 5.5. The instruction was displayed when there was no operation for 5 seconds after detected input stumbles, and remained on the screen for 7 seconds or until the screen was touched. The time and the text of instructions were modified after discus-

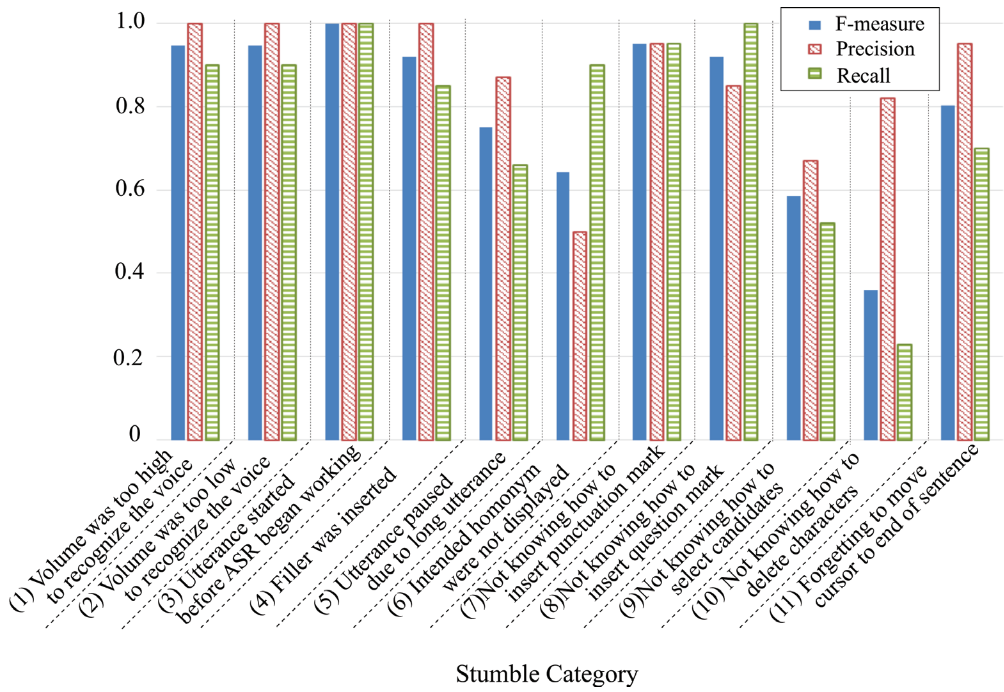


Figure 5.4: The performance of the detection of each input stumble with C4.5

sions with three older adults (60s) who did not participate in this study. The texts were accordingly modified to show not only the method of operation but also the reason for the correction, as well as tips for voice input.

5.6 Evaluation Experiment with Tutoring System

In this section, the effect of the tutoring system is evaluated by comparing user performance, with and without use of the system.

5.6.1 Participants

Twenty older adults took part in this experiment. They were recruited from a local social institution and did not participate in the previous user study. There were ten males and ten females, ranging in age from 65 to 71 years, with a mean age of 68.6

Figure 5.5: Screenshot showing instructions provided after an input stumble (7)

(sd = 1.9). Ten participants did not use the tutoring system (group A), and the other ten used it (group B). None of these older adults owned a smartphone or had any previous experience of making a voice input, but all had owned a featurephone and all had routinely used their own PC. None of them had tremor disorders, eye problems, or other relevant health problems.

5.6.2 Procedure and Apparatus

The experiment was conducted in a similar way to the previous user study. In brief, the participants were first given explanations on how to operate a smartphone and how to use the voice input and the software keyboard. While seated in a chair, they input 40 sentences to match the presented sentences. Participants first used voice input, then were permitted to modify the result with a combination of voice and touch. They had a ten-minute break after completing twenty sentences. After the experiment, they filled out a 5-point Likert-scale questionnaire, and took part in an interview. After all the participants had completed the experiment, three annotators labeled input stumbles in the logs. The labels given by more than two annotators were adopted.

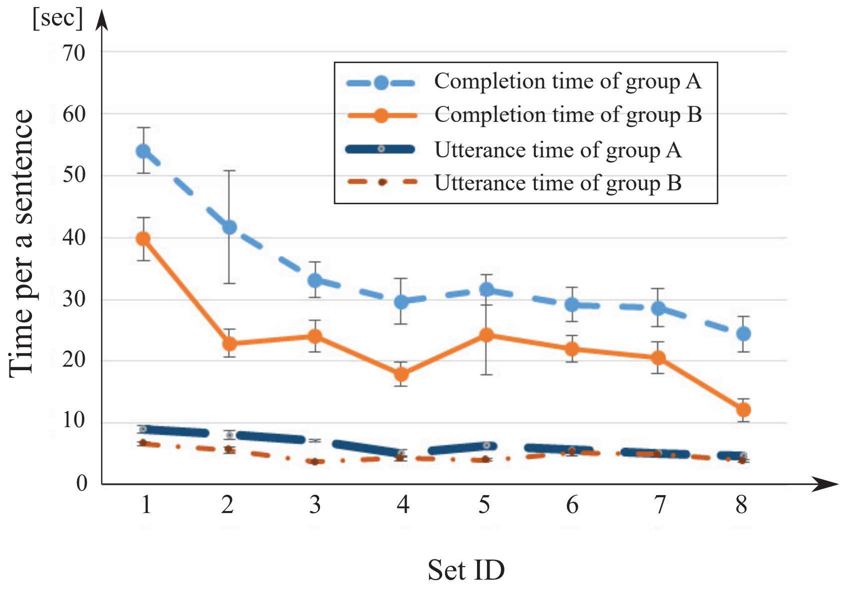


Figure 5.6: The average completion time and the utterance time for a sentence in each set; the Set ID is a block of 5 sentences divided into 8 set and arranged in chronological order

5.6.3 Experimental Results

All participants completed the task. The concordance rate of annotation of input stumble using Fleiss' kappa was 0.85. The detection performance of input stumble for group B using F-measure was 0.74. To evaluate the effect of the tutoring system, four metrics are adopted: the task completion time, number of input stumble, the performance of ASR, and the questionnaire.

Completion Time

To analyze the learning effect of the system, a chronological analysis of the performance of the subjects is conducted, by dividing all 40 sentences into 5 sets (8 sentences per set), and measuring the average completion time and utterance time for each set during the experiment. The results of the analysis are shown in Figure 5.6. As can be seen from the figure, the completion time of both groups decreased gradually. The completion

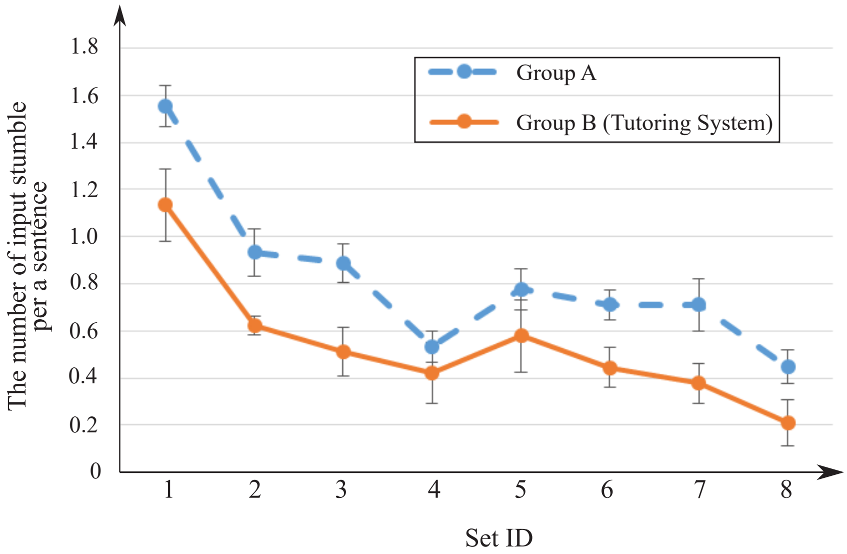


Figure 5.7: The number of input stumbles in a sentence in each set

times of group B were shorter than those of group A in all sets. Welch's t-test showed a significant difference between the completion time of both groups for the first set ($t(12) = 2.79, p < .05$) and the final set ($t(15) = 2.89, p < .05$). On the other hand, the utterance time of both groups decreased gradually, and there was no significant difference between the two groups.

Number of Input Stumbles

Figure 5.7 shows the number of input stumbles per sentence. As shown in the figure, the number of input stumbles of both groups decreased gradually, as did the completion time. The number of input stumbles in group B was less than in group A in all sets. Welch's t-test showed a significant difference between the number of input stumbles of both groups for the first set ($t(16) = 3.05, p < .01$) and final set ($t(9) = 8.65, p < .01$).

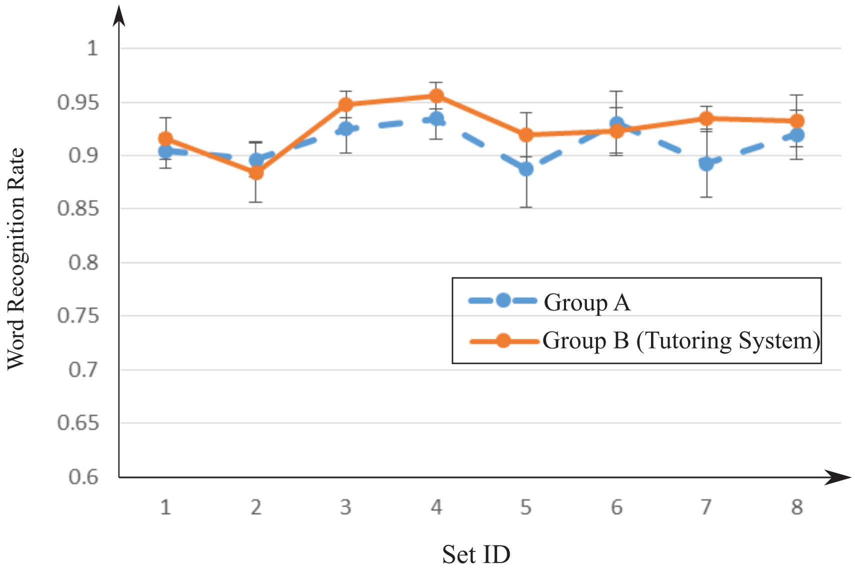


Figure 5.8: The word recognition rate in each set

Performance of ASR

Figure 5.8 shows the word recognition rate in each set. There was little change from the beginning to the end for both groups and the difference between the two groups was negligible. Welch's t-test ($\alpha = 0.05$) did not show any significant difference in rate between the initial set and final of both groups. This result shows that the tutoring system does not influence the performance of ASR.

Questionnaire Results

The results of questionnaires are shown in Table 5.4. The scale is 1-5, where 5 indicates the highest level of agreement. The score for the items *"The voice input was useful"* was 3.7 in group A and 3.6 in group B, showing no significant difference. On the other hand, there was a 0.5 point inter-group score difference for *"The correction was easy"* and *"I can learn the voice input method by myself"*. A Wilcoxon signed-rank sum test ($\alpha = 0.05$) showed a significant difference between the groups for the only latter item.

Table 5.4: Questionnaire responses of which the scales are 1-5, where 5 is higher agreement; the standard deviations are given in parentheses)

Question	group A	group B
The voice input was useful	3.7 (0.64)	3.6 (0.66)
The correction was easy	2.6 (0.66)	3.1 (0.70)
I can learn the way of the voice input by myself	3.7 (0.45)	4.2 (0.40)

5.7 Discussion

In summarizing the results of the evaluation, the completion time and the number of input stumbles of group B were significantly smaller than those of group A. These results indicate that the tutoring system works effectively for novice older adults. Focusing on the time, the time taken to find out how to modify and the time to needed to modify sentences decreased as a result of using the tutoring system because the ratio of the utterance time to the completion time was small. In addition, there is a possibility that the amount of modification decreased because instructions influenced the input efficiency of participants. Some participants of group B made a comment:

> *"I realized that ASR worked well when I want to input a long sentence or several sentences if I input each short phrase or sentence."*

These comments imply that the instruction provided by the tutoring system worked effectively.

On the other hand, some participants made multiple input stumbles simultaneously in one voice input. However, the tutoring system provides an instruction only for the most probable input stumbles. Therefore, in this case, some participants often did not know how to perform the next operation after resolving an input stumble. To detect multiple stumbles and provide step-by-step instruction is a topic for future research.

5.8 Conclusion of Chapter 5

This chapter proposed a tutoring system for voice input that detected input stumbles, and provided instructions that helped users resolve input stumbles independently.

First, a user study to clarify the problems older adults encountered in voice input was conducted. Second, statistical input detection method and the tutoring system were constructed. Finally, the performance of the tutoring system was evaluated. In the evaluation experiment, the number of input stumbles was significantly lower and the sentence completion time was significantly shorter when participants used the tutoring system compared to those who did not use the system. The results showed that the tutoring system resulted in improvement in the efficiency of voice input for novice older adults. Future research is to develop a system to detect multiple stumbles and provide step-by-step instructions to improve usability.

Chapter 6

Conclusion

This study presented a tutoring system for text input that detects input stumbles and provides instructions, assuming the role of a human tutor who indicates the next action. This chapter first reviews the contributions of this study, then discusses findings and suggests directions for future research.

6.1 Contributions and Summary

The main contributions of this thesis are three-fold. The first contribution (C1) was to clarify the common difficulties that novice older adults encounter in text input using a software keyboard and by voice input. The second contribution (C2) was to develop a method to detect input stumbles for both text input methods. The third contribution (C3) was to develop a tutoring system for text input and confirm the acceptability of the tutoring system. The following describes contributions and provides a summary of each chapter.

In Chapter 3, an assistive typing practice application for text entry, that is tutoring system for inputting presented sentences, was presented as the initial step of the research. First, the common difficulties encountered in typing presented sentences were clarified through a user study (C1). Then, a simple input stumble detector using the difference between the sentence to be input and the actual input sentence was developed from the data obtained in user studies (C2). Then, a prototype of the assistive typing practice application was developed and evaluated (C3). An evaluation experiment with novice older adults showed that the assistive typing practice application increased characters per minute (CPM) by 17.2% and reduced the stumble ratio with respect to chances for stumbles by 59.1% compared with the user's initial rate. Improvement rates were almost the same as those achieved with human tutors.

In Chapter 4, the target area was expanded to free text input assuming daily use. First, the common difficulties in free text input were clarified and the relationship between the input stumbles and the user's skill was analyzed through a user study (C1). Next, statistical input stumble detection was developed based on eighty-two features that were obtained from the input sequence which included unfixed string, fixed string, typing histories, and suggestions (C2). Then, a practical tutoring system that took skill level and forgetting into account was developed (C3). A two-week evaluation experiment with novice older adults showed that Typing Tutor was effective in improving text entry, especially in the initial stage of use. In addition, the applicability of statistical input stumbles detection to other keyboards and languages was evaluated. Based on the results, it was found that the performance of the input stumble detection for the QWERTY layout for Japanese and English was almost same as that for the 12-key layout for Japanese.

In Chapter 5, the input modality was expanded to include voice input. First, the common difficulties in voice input were also clarified through a user study (C1). Second, statistical input stumble detection was developed based on a total of 28 features including acoustic features and textual features related to the ASR and morphological analysis results (C2). Then, the tutoring system for voice input was developed (C3). In the evaluation experiment, the number of input stumbles was significantly lower and the sentence completion time was significantly shorter for those using the tutoring system compared to those who did not. The results showed that the tutoring system resulted in the efficiency of voice input being improved for novice older adults.

In all of these studies, the group that used the tutoring system improved input speed and decreased input stumbles compared with the group that did not use the system. These results indicate that the tutoring system for text input works effectively for novice older adults.

6.2 Discussion

This study focused on a tutoring system for older adults, although it needs to be pointed out that older adults are a highly diverse group. To respond to the variety of older adults, some studies have considered user models. User models are broadly categorized as ability [22], behaviors [87], preferences [45], and knowledge [98]. In terms of utilization of tutoring systems in user models, two points are relevant, "decision on whether to provide instruction" and "what kind of instruction should be provided." In chapter 4 of this study, users were classified based on ability, and the classification was used as a measure of whether instructions should be provided or not. In this section, user models are discussed from other perspectives.

First, in considering a user model in terms of behavior, correlations among input stumbles can be examined. Namely, this is the situation where users who make input stumble A tend to make input stumble B. As an example, in this study, it became evident that users who tended to input long sentences were not good at converting kana to kanji. By combining the correlation and the classification by input skills, it is thought instructions can be provided more efficiently.

Next, from the viewpoint of preferences, preference for a particular input modality can be examined. In chapter 6, the instruction method was designed based on the premise that corrections of misrecognition with voice input were made mainly by the software keyboard. However, for users who prefer voice input, providing instructions based on the premise of making corrections with voice input is also shown to be effective.

Finally, from the viewpoint of knowledge, background and knowledge of IT use can be examined. For example, focusing on background knowledge in IT, it is possible to recommend the use of the QWERTY keyboard to a user who is familiar with PCs. On the other hand, by taking IT knowledge into consideration, it is possible that brief instructions using technical terms can be provided to users who are familiar with new technology, and easy-to-understand instructions using simple expressions can be provided to users who are not so familiar with such technology. It is difficult to guess the user's knowledge accurately, but if such knowledge can be exploited, more effective instructions can be given.

As described above, users can be modeled from a plurality of viewpoints in terms of text input, and it is possible to develop a more intelligent tutoring system by effectively using it. This is the issue that needs to be addressed in the future.

6.3 Future Work

In addition to user modeling, two future studies are planned in order to make new technologies accessible to older adults: One is further improvement of smartphone text input support. The proposed tutoring system was developed for a keyboard having a basic layout and predictive conversion to assist older adults to master common smartphone text input. However, many studies have focused on making it easier to type text by adjusting key target areas or giving appropriate suggestions or auto-complements [8, 34, 82]. It should be possible to support both cognitive and physical factors by developing a tutoring system that incorporates both these techniques. In addition, the proposed tutoring system provides instructions only for the most probable input stumbles. Therefore, in this case, after resolving an input stumble, users often did not know how to perform the next operation. To detect multiple stumbles and provide step-by-step instructions is a topic intended to address in future work.

Another area of study is the expansion of the assist area. One application is how to utilize the other functions. Another application is the operation of other devices, such as smartwatches, smartglasses, and smartspeakers. The method used in this study needs to acquire data and annotate the data, so it takes time and effort to expand the other operations and devices. Therefore, endeavors need to be made to reduce the cost incurred for annotation by using other techniques such as semi-supervised and non-supervised classification and anomaly detection.

Acknowledgement

I would like to express my sincere appreciation for the unflagging support of my supervisor Professor Tatsuya Kawahara and for his supervision. His thoughtful advice for this thesis work encouraged me greatly. The time I spent in his lab during my doctorate course studies was a great opportunity to have different experiences, not only in my research work.

I am very grateful to Professor Sadao Kurohashi and Professor Toru Ishida of Kyoto University for their invaluable comments and helpful suggestions concerning this thesis.

I would also like to thank Professor Tetsuya Matsuda of Kyoto University, who supervised my master's degree program. Through his direction, I learned the correct attitude conducting research.

I also wish to express my deep appreciation to Dr. Yasuhiro Takishima, Executive Director of Human Communication Division of KDDI Research, Inc. for his continuous guidance from the time I entered the company. I would like to express my thanks to Dr. Keiichiro Hoashi, Group Leader of Intelligent Media Laboratory and Dr. Tomonori Yazaki, Group Leader of Future Design Division of KDDI Research, Inc., for giving me the opportunity to write this thesis.

I am also very grateful to Dr. Tsuneo Kato, Associate Professor of Doshisha University, and Dr. Toshiharu Horiuchi of Interaction Design Laboratory of KDDI Research, Inc. for their kind support throughout the process of the research on this thesis.

I also express my thanks to the members of Speech and Audio Processing Lab and the members of Intelligent Media Laboratory of KDDI Research, Inc. for their fruitful discussions.

Most importantly, I express special thanks to my family and especially my wife for her support and understanding.

Bibliography

[1] Julio Abascal and Colette Nicolle. Moving towards inclusive design guidelines for socially and ethically aware HCI. *Interacting with Computers*, Vol. 17, No. 5, pp. 484–505, 2005.

[2] Andrea L. Ames. Just what they need, just when they need it: An introduction to embedded assistance. In *Proceedings of the 19th Annual International Conference on Computer Documentation*, SIGDOC '01, pp. 111–115. ACM, 2001.

[3] Michel J. Anzanello and Flavio S. Fogliatto. Learning curve models and applications: Literature review and research directions. *International Journal of Industrial Ergonomics*, Vol. 41, No. 5, pp. 573–583, 2011.

[4] Ahmed S. Arif and Wolfgang Stuerzlinger. Analysis of Text Entry Performance Metrics. In *Proceedings of the International Conference on Science and Technology for Humanity*, TIC-STH '09, pp. 100–105. IEEE, 2009.

[5] Akira Baba, Shinichi Yoshizawa, Miichi Yamada, Akinobu Lee, and Kiyohiro Shikano. Acoustic models of the elderly for large-vocabulary continuous speech recognition. *Electronics and Communications in Japan (Part II: Electronics)*, Vol. 87, No. 7, pp. 49–57, 2004.

[6] Beverly Beisge and Marilyn Kraitchman. Senior centers: Opportunities for successful aging. *Springer Publishing Company*, 2003.

[7] Lawrence Bergman, Vittorio Castelli, Tessa Lau, and Daniel Oblinger. Docwizards: A system for authoring follow-me documentation wizards. In *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology*, UIST '05, pp. 191–200. ACM, 2005.

[8] Xiaojun Bi, Tom Ouyang, and Shumin Zhai. Both Complete and Correct? Multi-Objective Optimization of Touchscreen Keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 2297–2306. ACM, 2014.

[9] Christopher Bishop. *Pattern recognition and machine learning.* Springer, 2007.

[10] John Carroll and Mary Beth Rosson. Paradox of the active user in interfacing thought: cognitive aspects of human computer interaction. 11 1998.

[11] Pei-Yu Chi, Sally Ahn, Amanda Ren, Björn Hartmann, Mira Dontcheva, and Wilmot Li. Mixt: Automatic generation of step-by-step mixed media tutorials. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pp. 1499–1504. ACM, 2012.

[12] Michael H. Cohen, James P, Giangola, and Jennifer Balogh. *Voice User Interface Design.* Addison-Wesley Professional, 2004.

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning, 20*, 1995.

[14] Corinna Cortes and Vladimir Vapnik. On the optimality of the simple bayesian classifier under zero-one loss. In *Machine Learning, 29*, 1997.

[15] Anna L. Cox, Paul A. Cairns, Alison Walton, and Sasha Lee. Tlk or txt? using voice input for sms composition. *Personal Ubiquitous Comput.*, Vol. 12, No. 8, pp. 567–588, 2008.

[16] Fred Davis. A technology acceptance model for empirically testing new end-user information systems. 1985.

[17] Anna Dickinson, Peter Gregor, and Lucy Dickinson. Seeword: Rethinking interfaces, insights from word processing software for dyslexic readers. 2003.

[18] Anna Dickinson, Alan F. Newell, and Michael J. Smith. Introducing the internet to the over-60s: Developing an email system for older novice computer user. *Interacting with Computers*, Vol. 17, No. 6, pp. 621–642, 2005.

[19] Morgan Dixon and James Fogarty. Prefab: Implementing advanced behaviors using pixel-based reverse engineering of interface structure. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 1525–1534. ACM, 2010.

[20] Paul Dourish. The appropriation of interactive technologies: Some lessons from placeless documents. *Comput. Supported Coop. Work*, Vol. 12, No. 4, pp. 465–490, 2003.

[21] Hermann Ebbinghaus. Memory; a contribution to experimental psychology. *New York by Teachers College*, 1885.

[22] Rob Edlin-White, Sue Cobb, Mirabelle D'Cruz, Anne Floyde, Sarah Lewthwaite, and Johann Riedel. Accessibility for older users through adaptive interfaces: Opportunities, challenges and achievements. In *Human-Computer Interaction, Part III, HCII2011, LNCS6763*, pp. 483–489.

[23] Alistair D. N. Edwards. Extra-ordinary human-computer interaction. chapter Computers and People with Disabilities, pp. 19–43. Cambridge University Press, 1995.

[24] Jennifer Fernquist, Tovi Grossman, and George Fitzmaurice. Sketch-sketch revolution: An engaging tutorial system for guided sketching and application learning. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 373–382. ACM, 2011.

[25] Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, Vol. 76, No. 5, pp. 378–382, 1971.

[26] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Zhai Shumin. Effects of language modeling and its personalization on touchscreen typing performance. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '15.

[27] Krzysztof Z. Gajos, Daniel S. Weld, and Jacob O. Wobbrock. Automatically generating personalized user interfaces with supple. *Artificial Intelligence*, Vol. 174, No. 12, pp. 910–950, 2010.

[28] Mayank Goel, Leah Findlater, and Jacob Wobbrock. Walktype: Using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 2687–2696. ACM, 2012.

[29] Mayank Goel, Alex Jansen, Travis Mandel, Shwetak N. Patel, and Jacob O. Wobbrock. Contexttype: using hand posture information to improve mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2795–2798. ACM, 2013.

[30] Joshua Goodman, Gina Venolia, Keith, and Chauncey Parker. Language Modeling for Soft Keyboards. In *IUI '02 Proceedings of the 7th international conference on Intelligent user interfaces*, pp. 194–195. ACM, 2002.

[31] Masataka Goto, Katunobu Itou, and Satoru Hayamizu. Speech completion: On-demand completion assistance using filled pauses for speech input interfaces. In *Proceedings of the 7th International Conference on Spoken Language Processing*, ICSLP '02, pp. 1489–1492. World Academy of Science, Engineering and Technology (WASET), 2002.

[32] Peter Gregor, Alan F. Newell, and Mary Zajicek. Designing for dynamic diversity: Interfaces for older people. In *Proceedings of the Fifth International ACM Conference on Assistive Technologies*, Assets '02, pp. 151–156. ACM, 2002.

[33] Tovi Grossman and George Fitzmaurice. Toolclips: An investigation of contextual video assistance for functionality understanding. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 1515–1524. ACM, 2010.

[34] Asela Gunawardana, Tim Paek, and Christopher Meek. Usability Guided Key-Target Resizing for Soft Keyboards. In *Proceedings of the 15th International Conference on Intelligent User Interfaces*, IUI '10, pp. 111–118. ACM, 2010.

[35] Toshiyuki Hagiya, Toshiharu Horiuchi, and Tomonori Yazaki. Typing Tutor: Individualized Tutoring in Text Entry for Older Adults Based on Input Stumble Detection. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '16, pp. 733–744. ACM, 2016.

[36] Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tsuneo Kato, and Tatsuya Kawahara. Assistive typing application for older adults based on input stumble detection. *Journal of Information Processing*, Vol. 25, pp. 417–425, 2017.

[37] Toshiyuki Hagiya and Tsuneo Kato. Probabilistic Keyboard Adaptable to User and Operating Style Based on Syllable HMMs. In *Proceedings of the 21st International Conference on Pattern Recognition*, ICPR '12, pp. 65–68. IEEE, 2012.

[38] Susan M. Harrison. A comparison of still, animated, or nonillustrated on-line help with written or spoken instructions in a graphical user interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '95, pp. 82–89. ACM, 1995.

[39] Jamye M. Hickman, Wendy A. Rogers, and Arthur D. Fisk. Training older adults to use new technology. *Journals of Gerontology: Series B*, Vol. 62, pp. 77–84, 2007.

[40] Johan Himberg and Jonna Häkkilä and Petri Kangas and Jani Mäntyjärvi . Online Personalization of a Touch Screen Based Keyboard. In *IUI '03 Proceedings of the 8th international conference on Intelligent user interfaces*, pp. 77–84. ACM, 2003.

[41] Geoffrey Hinton, Li Deng, Dong Yu, George Dahl, Abdel rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Brian Kingsbury, and Tara Sainath. Deep neural networks for acoustic modeling in speech recognition. *IEEE Signal Processing Magazine*, Vol. 29, pp. 82–97, 2012.

[42] Eve Hoggan, Stephen A. Brewster, and Jody Johnston. Investigating the effectiveness of tactile feedback for mobile touchscreens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pp. 1573–1582. ACM, 2008.

[43] John L. Horn. *The Theory of Fluid and Crystallized Intelligence in Relation to Concepts of Cognitive Psychology and Aging in Adulthood*, pp. 237–278. Springer US, 1982.

[44] Amy Hurst, Scott E. Hudson, and Jennifer Mankoff. Dynamic detection of novice vs. skilled use without a task model. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '07, pp. 271–280. ACM, 2007.

[45] Gawesh Jawaheer, Peter Weller, and Patty Kostkova. Modeling user preferences in recommender systems: A classification framework for explicit and implicit user feedback. *ACM Trans. Interact. Intell. Syst.*, Vol. 4, No. 2, pp. 8:1–8:26, 2014.

[46] F. Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, Vol. 64, No. 4, pp. 532–556, 1976.

[47] Zhao Xia Jin, Tom Plocher, and Liana Kiff. Touch screen user interfaces for older adults: button size and spacing. In *Universal Access in HCI, Part I, HCII2007, LNCS4554*, pp. 933–941, 2007.

[48] Shaun K. Kane, Jacob O. Wobbrock, and Ian E. Smith. Getting off the treadmill: Evaluating walking user interfaces for mobile devices in public spaces. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '08, pp. 109–118. ACM, 2008.

[49] Caitlin Kelleher and Randy Pausch. Stencils-based tutorials: Design and evaluation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pp. 541–550. ACM, 2005.

[50] John F. Kelley. Cal - a natural language program developed with the oz paradigm: Implications for supercomputing systems. In *Proceedings of the First International Conference on Supercomputing Systems*, pp. 238–248. ACM, 1985.

[51] Catherine L. Kelly and Neil Charness. Issues in training older adults to use computers. *Behaviour & Information Technology*, Vol. 14, No. 2, pp. 107–120, 1995.

[52] F. Khaldoun, A, M. Mustapha, and V. Nadine. BigKey: A Virtual Keyboard for Mobile Devices. In *13th International Conference, HCI International 2009, Part III.*, pp. 3–10. Springer, 2009.

[53] Kevin Knabe. Apple guide: A case study in user-aided design of online help. In *Conference Companion on Human Factors in Computing Systems*, CHI '95, pp. 286–287. ACM, 1995.

[54] Andreas Komninos, Emma Nicol, and Mark D Dunlop. Designed with Older Adults to Support Better Error Correction in SmartPhone Text Entry: The MaxieKeyboard. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services*, MobileHCI '15, pp. 797–802. ACM, 2015.

[55] Per Ola Kristensson. Analysis of Text Entry Performance Metrics. In *Proceedings of the 15th International Conference on Artificial Intelligence in Education, Lecture Notes in Computer Science, vol 6738.*, AIED 2011, pp. 494–496. Springer, 2011.

[56] Sri Kurniawan. Older people and mobile phones: A multimethod investigation. *International Journal of Human Computer Studies*, Vol. 66,12, pp. 889–901, 2008.

[57] Cliff Lampe, Jessica Vitak, and Nicole Ellison. Users and nonusers: Interactions between levels of adoption and social capital. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work*, CSCW '13, pp. 809–820. ACM, 2013.

[58] Rock Leung, Leah Findlater, Joanna McGrenere, Peter Graf, and Justine Yang. Multi-layered interfaces to improve older adults’ initial learnability of mobile applications. *ACM Trans. Access. Comput.*, Vol. 3, No. 1, pp. 1:1–1:30, 2010.

[59] Rock Leung, Charlotte Tang, Shathel Haddad, Joanna Mcgrenere, Peter Graf, and Vilia Ingriany. How older adults learn to use mobile devices: Survey and field investigations. *ACM Transactions on Accessible Computing*, Vol. 4, No. 3, pp. 11:1–11:33, 2012.

[60] S. E. Levinson, L. R. Rabiner, and M. M. Sondhi. An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal*, Vol. 62, No. 4, pp. 1035–1074, 1983.

[61] Jinyu Li, Abdelrahman Mohamed, Geoffrey Zweig, and Yifan Gong. Lstm time and frequency recurrence for automatic speech recognition. In *IEEE Automatic Speech Recognition and Understanding Workshop*, 2015.

[62] Yuan Liang, Koji Iwano, and Koichi Shinoda. Simple gesture-based error correction interface for smartphone speech recognition. In *Proceedings of the 15th Annual Conference of the International Speech Communication Association International Conference on Science and Technology for Humanity*, INTERSPEECH '14, pp. 1194–1198. International Speech Communication Association (ISCA), 2014.

[63] Rensis Likert. A technique for the measurement of attitudes. *Archives of Psychology*, No. 140, 1932.

[64] Sue Ellen Linville. *Vocal Aging. Singular Thomson Learning*. Singular Thomson Learning, 2001.

[65] Lidia Mangu, Eric Brill, and Andreas Stolcke. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech & Language*, Vol. 14, No. 4, pp. 373–400, 2000.

[66] Clare marie Karat, Christine Halverson, Daniel Horn, and John Karat. Patterns of entry and correction in large vocabulary continuous speech recognition system. In *Conference Companion on Human Factors in Computing Systems*, CHI '99, pp. 568–575. ACM, 1999.

[67] S. Milne, Dickinson, A. Carmichael, D. Sloan, R. Eisma, and Peter Gregor. Are guidelines enough?: An introduction to designing web sites accessible to older people. *IBM Syst. J.*, Vol. 44, No. 3, pp. 557–571, 2005.

[68] Mehryar Mohri, Fernando Pereira, and Michael Riley. Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, Vol. 16, No. 1, pp. 69–88, 2002.

[69] Roger W. Morrell, Denise C. Park, Christopher B. Mayhorn, and Catherine L. Kelley. Effects of age and instructions on teaching older adults to use eldercomm, an electronic bulletin board system. *Educational Gerontology*, Vol. 26, No. 3, pp. 221–235, 2000.

[70] Hugo Nicolau and Joaquim Jorge. Elderly text-entry performance on touchscreens. In *Proceedings of the 14th International ACM SIGACCESS Conference on Computers and Accessibility*, ACCESS '12, pp. 127–134. ACM, 2012.

[71] Ministry of Internal Affairs and Communications. Result of the survey on communication trend in 2016 (Japanese).

[72] Jun Ogata and Masataka Goto. Speech repair: Quick error correction just by using selection operation for speech input interfaces. In *Proceedings of the 9th European Conference on Speech Communication and Technology*, Eurospeech '05, pp. 133–136. International Speech Communication Association (ISCA), 2005.

[73] Susan Palmiter and Jay Elkerton. Animated demonstrations for learning procedural computer-based tasks. *Human-Computer Interaction*, Vol. 8, No. 3, pp. 193–216, 1993.

[74] Tim F. Paymans, Jasper Lindenberg, and Mark Neerincx. Usability trade-offs for adaptive user interfaces: Ease of use and learnability. In *Proceedings of the 9th International Conference on Intelligent User Interfaces*, IUI '04, pp. 301–303. ACM, 2004.

[75] Suporn Pongnumkul, Mira Dontcheva, Wilmot Li, Jue Wang, Lubomir Bourdev, Shai Avidan, and Michael F. Cohen. Pause-and-play: Automatically linking screencast video tutorials with applications. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 135–144. ACM, 2011.

[76] J. Ross Quinlan. *Programs for Machine Learning*. Morgan Kaufmann Publisher, 1993.

[77] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. In *Proceedings of the IEEE*, Vol. 77, pp. 257–286. ACM, 1989.

[78] Abdel rahman Mohamed, George Dahl, and Geoffrey Hinton. Acoustic modeling using deep belief networks. *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 1, pp. 14–22, 2011.

[79] Mili D. Rama, Huib Ridder, and Herman Bouma. Technology generation and Age in using layered user interfaces. *Journals of Gerontechnology*, pp. 1(1):25–40, 2001.

[80] Lorraine Olson Ramig, Steven Gray, Kristin Baker, Eugene Buder Kim Corbin-Lewis and, Erich Luschei, Hilary Coon, and Marshall Smith. The aging voice: A review, treatment data and familial and genetic perspectives. Vol. 53, pp. 252–65, 2001.

[81] Pei-Luen Patrick Rau and Jia-Wen Hsu. Interaction devices and web design for novice older users. *Educational Gerontology*, Vol. 31, No. 1, pp. 19–40, 2005.

[82] Élvio Rodrigues, Micael Carreira, and Daniel Goncalves. Enhancing typing performance of older adults on tablets. *Universal Access in the Information Society*, pp. 1–26, 2014.

[83] Wendy A. Rogers, Elizabeth F. Cabrera, Neff Walker, D. Kristen Gilbert, and Arthur D. Fisk. A survey of automatic teller machine usage across the adult lifespan. *Human Factors*, pp. 38,1; 156–166, 1996.

[84] Wendy A. Rogers and Arthur D. Fisk. Toward a Psychological Science of Advanced Technology Design for Older Adults. *J Gerontol B Psychol Sci Soc Sci*, Vol. 65, .

[85] Timothy A. Salthouse, Renȩe L. Babcock, and Raymond J. Shaw. Effects of adult age on structural and operational capacities in working memory. Vol. 6, pp. 118–127, 1991.

[86] Ben Shneiderman. Issues in training older adults to use computers. *Behaviour & Information Technology*, Vol. 14, No. 2, pp. 107–120, 1995.

[87] David Sloan, Matthew Tylee Atkinson, Colin Machin, and Yunqiu Li. The potential of adaptive interfaces as an accessibility aid for older web users. In *Proceedings of the 2010 International Cross Disciplinary Conference on Web Accessibility 35*, pp. 35:1–35:10. ACM, 2010.

[88] Yuan Tang. Tf.learn: Tensorflow's high-level module for distributed machine learning. In *arXiv*, 2016.

[89] Ahmet Cunyed Tantug. A Probabilistic Mobile Text Entry System for Agglutinative Languages. In *IEEE Transactions on Consumer Electronics, Vol. 56, No. 2, May 2010*, pp. 1018–1024. ACM, 2010.

[90] Gregg Vanderheiden. Fundamental principles and priority setting for universal usability. In *Proceedings on the 2000 Conference on Universal Usability*, CUU '00, pp. 32–37. ACM, 2000.

[91] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. VelociTap: Investigating Fast Mobile Text Entry using Sentence-based Decoding of Touchscreen Keyboard Input. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '15, pp. 659–668. ACM, 2015.

[92] Ravichander Vipperla, Steve Renals, and Joe Frankel. Longitudinal study of asr performance on ageing voices. In *In Proceedings of the 9th Annual Conference of the International Speech Communication Association*, INTERSPEECH '08, pp. 2550–2553, 2008.

[93] Ravichander Vipperla, Steve Renals, and Joe Frankel. Ageing voices: The effect of changes in voice parameters on asr performance. *EURASIP J. Audio Speech Music Process.*, pp. 5:1–5:10, 2010.

[94] Cheng-Yao Wang, Wei-Chen Chu, Hou-Ren Chen, Chun-Yen Hsu, and Mike Y. Chen. EverTutor: Automatically Creating Interactive Guided Tutorials on Smartphones by User Demonstration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pp. 4027–4036. ACM, 2014.

[95] Alexandra Weilenmann. Learning to text: An interaction analytic study of how seniors learn to enter text on mobile phones. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pp. 1135–1144. ACM, 2010.

[96] J. G. Wilpon and C. N. Jacobsen. A study of speech recognition for children and the elderly. In *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, Vol. 1, pp. 349–352 vol. 1, 1996.

[97] John T. Wixted. Analyzing the empirical course of forgetting. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, Vol. 16, No. 5, pp. 927–935, 1990.

[98] Jacob O. Wobbrockand, Shaun K. Kane, Krzysztof Z. Gajos, Susumu Harada, and Jon Froehlich. Ability-based design: Concept, principles and examples. *ACM Transactions on Accessible Computing*, Vol. 3,3, No. 9, 2011.

[99] Steve An Xue and Grace Jianping Hao. Changes in the human vocal tract due to aging and the acoustic correlates of speech production. Vol. 46, pp. 689–701, 2003.

[100] Tom Yeh, Tsung-Hsiang Chang, Bo Xie, Greg Walsh, Ivan Watkins, Krist Wongsuphasawat, Man Huang, Larry S. Davis, and Benjamin B. Bederson. Creating contextual help for guis using screenshots. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pp. 145–154. ACM, 2011.

[101] Ying Yin, Tom Y. Ouyang, Kurt Partridge, and Shumin Zhai. Making Touchscreen Keyboards Adaptive to Keys, Hand Postures, and Individuals - A Hierarchical Spatial Backoff Model Approach. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 2775–2784. ACM, 2013.

[102] Jason Chen Zhao, Richard C. Davis, Pin Sym Foong, and Shengdong Zhao. Cofaçade: A customizable assistive approach for elders and their helpers. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pp. 1583–1592. ACM, 2015.

[103] Martina Ziefle and Susanne Bay. *Mental Models of a Cellular Phone Menu. Comparing Older and Younger Novice Users*, pp. 25–37. Springer Berlin Heidelberg, 2004.

Appendix A

List of Publications

Journal Articles

1. Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tatsuya Kawahara, "Typing Tutor: Individualized Tutoring in Text Entry for Older Adults Based on Statistical Input Stumble Detection", Journal of Information Processing (in press)

2. Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tsuneo Kato, Tatsuya Kawahara, "Assistive Typing Application for Older Adults Based on Input Stumble Detection", Journal of Information Processing Vol. 25 (2017) pp. 417–425

3. Toshiyuki Hagiya, Tsuneo Kato, "Probabilistic Touchscreen Text Entry Method Incorporating Gaze Point Information", Journal of Information Processing Vol. 23 (2015) No. 5 pp. 708–715

4. Toshiyuki Hagiya, Tsuneo Kato, "HMM-Based Probabilistic Flick Keyboard Adaptable to Individual User", Journal of Information Processing Vol. 22 (2014) No. 2 pp. 410–416

International Conferences

1. Toshiyuki Hagiya, Keiichiro Hoashi, Tatsuya Kawahara, "Voice Input Tutoring System for Older Adults using Input Stumble Detection", In Proceedings of In-

ternational Conference on Intelligent User Interfaces Companion 2018 (IUI 2018) (in press)

2. Jianming Wu, Toshiyuki Hagiya, Yujin Tang, Keiichiro Hoashi, “Effects of Objective Feedback of Facial Expression Recognition during Video Support Chat”, In Proceedings of 16th International Conference on Mobile and Ubiquitous Multimedia (MUM 2017)

3. Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tsuneo Kato, “Typing Tutor: Individualized Tutoring in Text Entry for Older Adults Based on Input Stumble Detection”, In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2016), Pages 733–744

4. Toshiyuki Hagiya, Toshiharu Horiuchi, Tomonori Yazaki, Tsuneo Kato, “Typing Tutor: Automatic Error Detection and Instruction in Text Entry for Elderly People”, In Proceedings of the 17th International Conference on Human-Computer Interaction with mobile devices and services Adjunct (MobileHCI 2015), Pages 696–703

5. Toshiyuki Hagiya, Tsuneo Kato “Probabilistic Touchscreen Keyboard Incorporating Gaze Point Information”, In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI 2014), Pages 329–333

6. Toshiyuki Hagiya, Tsuneo Kato “Adaptable probabilistic flick keyboard based on HMMs”, In Proceedings of International Conference on Intelligent User Interfaces Companion 2013 (IUI2013), Pages 71–72

7. Baba, S., Kaji, K., Kawaguchi, N., Hagiya, T., Kato, T., “An Intuitive Multi-Touch Interface Using Finger Posture for Operating 3D Objects”, In Proceedings of 2013 International Workshop on Advanced Image Technology (IWAIT 2013), Pages 244–249

8. Toshiyuki Hagiya, Tsuneo Kato, “Probabilistic keyboard adaptable to user and operating style based on syllable HMMs”, In Proceedings of 21st International Conference on Pattern Recognition (ICPR 2012), Pages 65–68

Domestic Conferences

1. 萩谷 俊幸, 帆足 啓一郎, 河原 達也, “スマートフォンにおけるつまずき検出に基づく音声入力支援システム”, 第 175 回 HCI 研究会 (2017)
2. 呉 剣明, 住友 亮翼, 萩谷 俊幸, 徐 シン, 矢崎 智基, “パーソナライズ情報を提供するクロスデバイス型対話エージェント”, IEICE 総合大会 2016
3. 萩谷 俊幸，堀内 俊治, 矢崎 智基, 加藤 恒夫, “つまずき検出に基づくスマートフォン文字入力練習支援システム”, FIT2015
4. 萩谷 俊幸，加藤 恒夫, “注視点情報を用いた確率的フリック入力方式の提案”, FIT 2014
5. 萩谷 俊幸，加藤 恒夫, “注視点情報を用いた確率的ソフトウェアキーボード入力方式”, IPSJ 全国大会 2014
6. 萩谷 俊幸，加藤 恒夫, “HMM に基づくフリックキーボード入力方式の評価”, IPSJ 全国大会 2013
7. 萩谷 俊幸，加藤 恒夫, “HMM に基づくフリックキーボード入力方式”, FIT2012
8. 萩谷 俊幸, 上向 俊晃，加藤 恒夫, “確率モデルに基づくキーボード入力方式”, IPSJ 全国大会 2012