Journal of Korea Multimedia Society Vol. 21, No. 11, November 2018(pp. 1263-1270)
https://doi.org/10.9717/kmms.2018.21.11.1263

인접-오타를 이용한 소프트 키보드의 동적 적응 연구

고 석 훈†

A Study on Dynamic Adaptation of Soft Keyboard Using Adjacent-Typo

Seokhoon Ko†

ABSTRACT

Dynamic adaptation method is an effective technique to enhance the usability by personalizing the soft keyboard layout using the user's key input information. In this paper, we propose a dynamic adaptation method of a keyboard by automatically extracting typos from key input information and using adjacent-typo information classified through the relationship between typos. This technique does not limit a range of adaptation to the inside of the key but extends the range to the neighbor key so that the adaptation effect can be achieved in a wide range at a high speed, thereby the proposed method improves the usability of the keyboard with a small number of inputs. The proposed method showed 25% increase in usability compared to the existing method through experiment and it was confirmed that usability improves up to 33% when used with the existing method.

Key words: Soft Keyboard, Dynamic Adaptation, Adjacent-Typo, Usability

1. 서 론

스크린에 키보드 이미지를 표시하고 이를 터치하여 문자를 입력하는 터치스크린 키보드는 하드웨어 키보드를 사용하는 것과 유사한 사용자 경험을 제공하는 장점으로 인해 터치스크린이 장착된 기기에서 디팩토 스탠다드(de facto standard)로 사용되고 있다. 터치스크린 키보드는 소프트웨어로 구현되기 때문에 '소프트 키보드'라고도 부르는데, 소프트웨어를 이용하여 다양한 형태로 구성하거나 기능을 부여할 수 있지만, 키의 형태를 촉각으로 느낄 수 없기 때문에 문자 입력 속도와 오타 발생 확률 측면의 사용성(usability)이 떨어지는 특징이 있다. 특히, 스마트폰과 같은 작은 기기의 경우에는 소프트 키보드의 크기가 작기 때문에 사용성은 더욱 나빠질 수밖에 없다[1].

소프트 키보드의 사용성을 높이기 위한 다양한 시도로서, Fitt's Law에 근거하여 새로운 키보드 레이아웃을 도입하거나[2,3], 사전을 참조하여 완성 단어를 추천하거나[4], 터치한 다음 드래그 하여 여러 개의 글자를 동시에 입력하는 방법[5], 경계부근을 터치하면 언어 모델을 근거로 확률이 높은 글자를 선택하는 기법[6] 등이 연구 되었다. 그 중 적응형 키보드(adaptive keyboard)는 키보드 터치 정보를 이용하여 사용이 편리한 형태로 개인화된 키보드를 제공하는 기법으로 스마트폰과 같이 작은 스크린에서 사용성을 높이는데 효과적인 방법이다[7,8]. 기존의 적응형 키보드는 사용자가 올바르게 입력한 정타를 대상으로 적응을 수행하고, 잘못 입력한 오타에 대해서는 관심을 두지 않았다. 이런 방식은 적응 기준 데이터

※ Corresponding Author : Seokhoon Ko, Address: (17035) 81 Oedae-ro, Mohyeon-myeon, Cheoin-gu, Yongin-si, Gyeonggi-do, Korea, TEL : +82-31-330-4574, FAX : +82-50-4189-4098, E-mail : shko99@gmail.com
Receipt date : Sep. 16, 2018, Revision date : Oct. 22, 2018
Approval date : Oct. 23, 2018
† Institute of Information and Industrial Engineering, Hankuk University of Foreign Studies
※ This work was supported by Hankuk University of Foreign Studies Research Fund.

의 범위가 정타로 눌린 키 내부로 한정되기 때문에 효과가 점진적으로 나타나는 특징이 있는데, 만일 인접한 오타에 대해서도 적응을 수행한다면 훨씬 넓은 범위에서 빠른 시간 내에 적응의 효과를 얻을 수 있을 것이다. 본 논문에서는 오타 정보를 이용하여 넓은 범위에서 적응을 수행하는 알고리즘에 대해 설명한다.

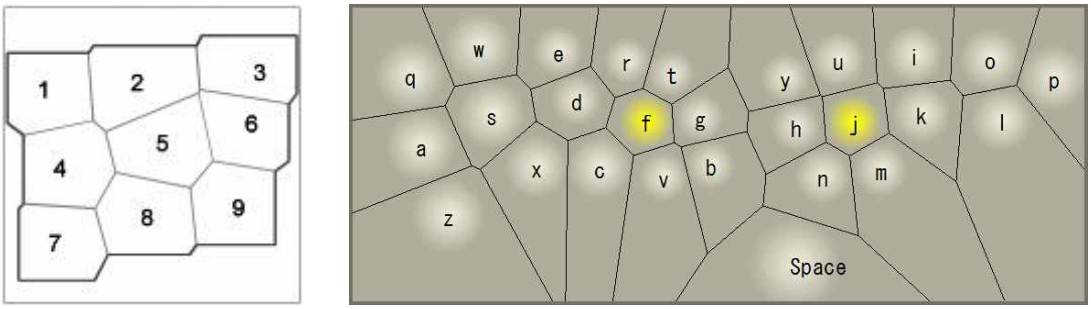
본 논문의 구성은 다음과 같다. 2장에서는 관련 연구를 소개한다. 3장에서는 유용한 오타 정보를 추출하는 방법과 이를 바탕으로 소프트 키보드의 터치 영역을 조정하는 적응 기법을 설명한다. 4장에서는 실험을 통해 본 논문에서 제시한 방법의 효과를 평가한다. 마지막으로 5장에서는 연구 결과를 요약하고 향후 연구 방향을 제시한다.

2. 관련 연구

Nokia 연구소의 Himberg는 9개의 터치스크린 숫자키에 대해 키가 눌려진 위치로 키의 중심을 이동하는 적응형 키보드를 제안하고, 실험을 통해 각 키가 손가락으로 누르기 가장 편한 위치로 이동하였음을 보여주었다(Fig. 1(a))[7]. Go는 Himberg의 적응형 키보드를 Qwerty 레이아웃에 적용한 CATKey (Customizable and Adaptable Touchscreen Keyboard)를 소개하였는데, 키의 위치와 크기 조정이 쉽도록 보로노이 다이어그램(Voronoi diagram)으로 키보드를 모델링하여 키의 형태가 불규칙한 거품 모양으로 만들어지는 특징을 갖는다(Fig. 1(b))[8]. Microsoft 연구소의 Gunawardana는 적응형 키보드가 과도하게 조정되면 오히려 사용자가 원하는 글자를 입력하기 어려워지는 상태를 되는 것을 지적하고, 이를 방지하기 위해 키의 가운데 영역을 앵커(anchor)로 고정하여 적응의 범위를 제한하는 방법을 고안하였고[9], 한국과학기술원의 Hong은 Gunawardana의 적응형 키보드를 안드로이드 플랫폼의 한국어 키보드에 적용하는 연구를 하였다[10]. Findlater는 적응형 키보드의 시각화와 사용성의 관계를 연구하여, 레이아웃이 고정된 비시각-적응형(Non Visual-adaptive) 키보드에 비해 적응으로 인해 변화되는 레이아웃을 사용자에게 보여주는 시각-적응형(Visual-adaptive) 키보드가 추가적인 시각적 인지 비용을 발생시켜 사용성이 떨어지는 것을 확인하였다[11].

사용자가 키보드로 입력하는 데이터의 양이 많지 않기 때문에[12], 이 데이터만으로 적응의 효과를 얻기까지는 많은 시간이 필요하다. Wassdahl은 단 시간 내에 적응의 효과를 얻기 위해 과거에 사용자가 작성한 Facebook 메시지를 가져와서 각 알파벳 별로 그 다음에 나오는 글자의 빈도를 기준으로 적응을 수행한 레이아웃을 만들고, 글자를 입력할 때마다 해당 글자의 레이아웃을 적용하는 방식으로 입력 확률이 높은 키의 터치 영역(key-target area)이 커지도록 하였다[13]. 이 방법은 키보드를 사용하기 전에 적응을 완료할 수 있다는 점에서 효과적이지만, 수집된 데이터의 내용이 현재 입력하는 문장과 상이한 경우 오히려 사용성이 나빠질 수도 있다.

적응형 키보드 연구는 올바른 키가 눌려진 정타에 대해서만 적응을 수행하며, 잘못된 키가 눌리는 오타에 대해서는 관심을 갖지 않았다. 오타와 관련된 연구로 Korea 대학의 Kwon이 제안한 오타를 고려한



(a) (b)

Fig. 1. (a) Himberg's adaptive numeric keypad (from [7]; copyright ACM) and (b) Go's bubble-shaped CATKey after adaptation process (from [8]; copyright Springer).

단어 예측 기법이 있다. Kwon은 사용자가 항상 오자를 누를 수 있다고 가정하여, 키가 입력될 때마다 입력된 키와 좌우로 인접한 위치의 키를 모두 조합하여 교정어 후보군을 만들고, 이를 바탕으로 사전을 검색하여 예측 단어를 제시하도록 했다[14]. Kwon의 방법에서 현재 입력된 키가 좌우에 인접한 키의 오타일 수 있다는 가정은 본 연구의 인접-오타 구분 방법을 고안하는데 기본 아이디어로 사용하였다. 동강대학교 Jeoung은 정전 용량 방식 터치 패널을 사용하는 소프트 키보드에서 손가락이 패널에 살짝 닿는 순간 해당 지점의 키를 확대하여 보여주는 방법을 제안했는데, 이때 오타 비율에 따라 확대 배율을 결정하도록 하였다[15]. 이 방법은 키를 터치하기 직전에 올바른 키가 선택되었는지 쉽게 확인하는 것을 목적으로 하였기 때문에 터치 영역 자체는 변화하지 않는다는 특징을 갖는다.

3. 제안한 방법

3.1 배경

인지 과학의 정보처리 모델에 따르면 사용자가 키를 누르는 과정은 자극 인식, 인지 처리 그리고 반응 실행의 세 단계로 구분된다[16].

(1) 자극 인식: 현재 입력중인 문장의 상태를 시각 기능으로 인식

(2) 인지 처리: 인식된 정보를 바탕으로 어떤 문자 키를 누를 것인지 선택

(3) 반응 실행: 선택된 키의 영역을 손가락으로 누르는 행동 실행

각 단계에서는 서로 다른 이유로 오타가 발생할 수 있다. 자극 인식 단계에서는 주의를 게을리 하여 현재 입력중인 위치를 놓치거나 엉뚱한 위치를 보아 현재까지 입력된 텍스트의 정보를 잘못 인식할 수 있고, 인지 처리 단계에서는 단어의 스펠링을 잘못 기억하여 엉뚱한 문자를 입력하도록 선택할 수 있다. 마지막으로 반응 실행 단계에서는 손가락의 해부학적인 요인으로 누르고자 하는 키가 아닌 인접한 다른 키를 터치하는 실수를 할 수 있다. 자극 인식 단계와 인지 처리 단계의 실수로 발생하는 오타는 적응형 키보드 기법으로 개선할 수 있는 범위의 문제가 아니다. 하지만, 반응 실행 단계의 실수로 발생하는 오타는 키의 터치 영역의 크기를 조정하여 오타 발생 확률을 낮출 수 있다.

본 논문에서 제안하는 소프트 키보드의 동적 적응 기법은 사용자의 키 입력을 모니터링 하여 자동으로 오타를 추출하고, 반응 실행 단계에서 발생한 오타를 구분하여 그 정보를 이용하여 키보드의 위치를 조정하는 적응을 수행하는 단계로 구성된다(Fig. 2).

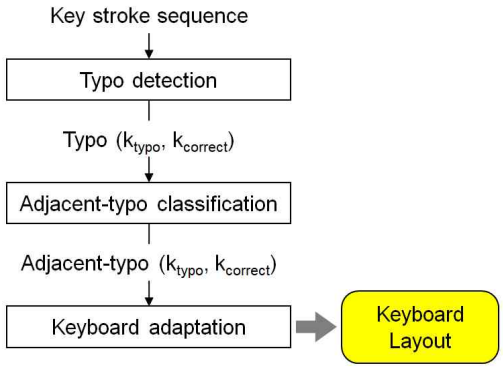


Fig. 2. Process of dynamic adaptation of soft keyboard using adjacent-typo.

3.2 오타 추출

키보드로 글자를 입력하는 통상적인 사용자 경험을 살펴보면, 글자 입력 중에 오타가 발생하면 BS (backspace)키를 눌러 오타로 입력된 문자를 삭제하고 교정할 문자를 입력한다. 따라서 키 입력 시퀀스를 모니터링 하여 BS키가 눌린 위치를 분석하면 오타로 눌린 키와 원래 입력하려 했던 문자에 해당하는 키를 알아낼 수 있다.

오타 추출 방법: 키 입력 시퀀스 $k_0, \cdots, k_{(i-2)}, k_{(i-1)}, k_i, k_{(i+1)}, k_{(i+2)}, \cdots, k_m$이 있고, k_i부터 $k_{(i+n-1)}$이 n개의 연속된 BS키 입력이라고 할 때, 오타키(typo key) $k_{typo} = k_{(i-n)}$이며, 오타를 수정하기 위해 입력한 교정키(correct key) $k_{correct} = k_{(i+n)}$이다.

Fig. 3은 BS키를 이용하여 오타를 추출하는 예를 보여준다. Fig. 3(a)의 경우, k_3에서 BS키가 한번 눌렸으므로 i=3, n=1이 되어 $k_{typo} = k_{(3-1)} = k_2 = 'r'$이고, $k_{correct} = k_{(3+1)} = k_4 = 't'$가 된다. 즉, 't' 키 대신 'r' 키를 오타로 잘못 눌렀음을 알 수 있다. Fig. 3(b)의 경우, k_4에서 BS키가 3번 눌렸으므로 i=4, n=3이 되

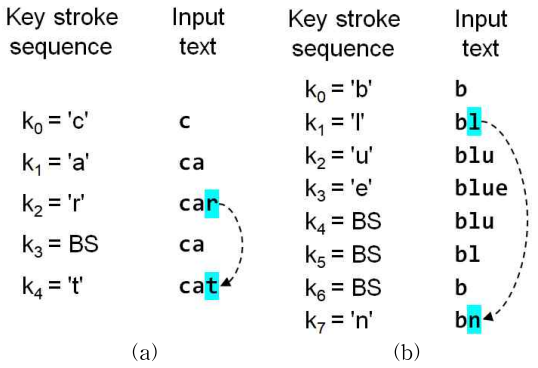


Fig. 3. Two examples of typo extracting from key stroke sequence. (a) Single backspace and (b) Multiple backspaces.

어 $k_{typo} = k_{(4-3)} = k_1$ = 'l'이고, $k_{correct} = k_{(4+3)} = k_7$ = 'n'이 된다. 즉, 'n' 키 대신 'l' 키를 오타로 잘못 눌렀음을 확인할 수 있다.

3.3 인접-오타 구분

반응 실행 단계를 원인으로 발생하는 오타는 사용자가 손가락으로 누르려고 의도한 위치가 아닌 다른 곳이 눌려서 발생한다. 그러므로 원래 눌려야 하는 키의 위치와 오타로 눌린 키의 위치는 손가락 크기 이내로 가까울 확률이 높다. 따라서 오타 추출 단계에서 찾아낸 오타키 k_{typo}와 교정키 $k_{correct}$의 키보드 상의 위치를 비교하여 서로 인접한 경우에 한하여 반응 실행 단계의 실수로 간주할 수 있으며, 우리는 이것을 인접-오타(adjacent-typo)라 부른다.

Fig. 4는 오타의 원인을 분석하는 예를 보여준다. Fig. 4(a)과 같이 k_{typo} = 'l'과 $k_{correct}$ = 'n'의 키보드 상의 위치가 멀리 떨어져 있다면 자극 인식 또는 정보 해석 단계의 오류로 발생한 오타일 가능성이 높다. 반면에 Fig. 4(b)와 같이 k_{typo} = 'r'과 $k_{correct}$ = 't'가 키보드에서 좌우로 인접해 있는 인접-오타라면 반응 실행 단계의 오류로 발생한 오타로 간주할 수 있다.

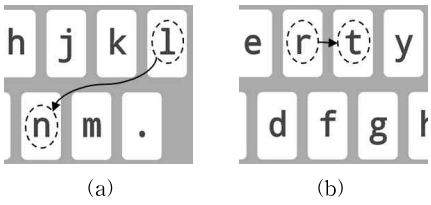


Fig. 4. Two types of typo. (a) k_{typo} = 'l', $k_{correct}$ = 'n' and (b) k_{typo} = 'r', $k_{correct}$ = 't'.

인접-오타는 상하로 인접한 경우에도 발생할 수 있지만, 소프트 키보드의 키 배치가 행 별로 엇갈려 있어 상하로 인접한 키의 거리는 좌우로 인접한 키의 거리보다 멀고, 키의 크기를 세로 방향으로 길게 디자인하면 입력 오류가 크게 줄어들기 때문에[17] 상하로 인접한 경우의 인접-오타는 고려하지 않는다.

3.4 소프트 키보드 동적 적응

인접-오타가 구분되면 오타키 k_{typo}와 교정키 $k_{correct}$가 마주하고 있는 인접 경계의 위치를 움직여 키의 터치 영역 크기를 조정하는 동적 적응 작업을 한다. 동적 적응을 수행하기 위해서는 오타가 발생한 위치의 좌표가 필요한데, 3.1에서 키 입력 시퀀스를 모니터링 할 때, 입력된 키 k_i와 함께 스크린을 터치한 위치 x_i를 함께 저장한다.

인접-오타를 이용한 동적 적응의 기본 아이디어는 $k_{correct}$의 터치 영역(key-target area)을 k_{typo}의 터치 영역 내부에 위치한 오타 발생 위치 x_{typo}까지 확장하는 것이다. Fig. 5는 k_{typo} = 'r'과 $k_{correct}$ = 't'인 경우에 실행되는 동적 적응의 예를 나타낸다. 그림에서 점선으로 된 사각형은 각 키의 터치 영역(key-target area)을 의미하고 각 키의 안쪽에 짙은색 사각형은

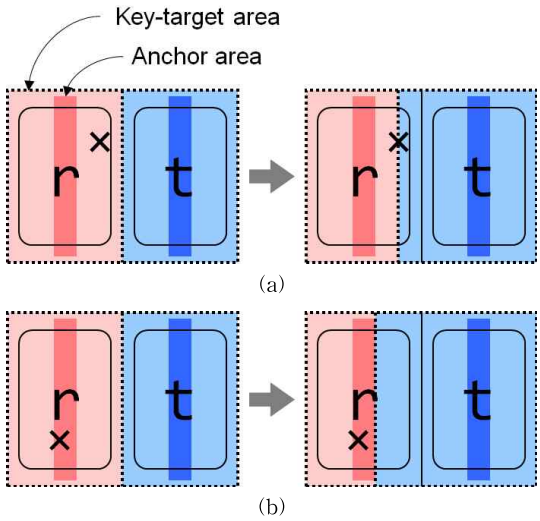


Fig. 5. Two examples of dynamic adaptation process. (a) Normal case and (b) Blocked by anchor case.

고정 영역(anchor area)를 의미한다. Fig. 5(a)의 왼쪽 그림은 인접-오타가 발생한 순간으로 X표 위치는 오타 발생 위치 x_{typo}를 의미한다. 이 상태에서 동적 적응을 수행하면, Fig. 5(a)의 오른쪽 그림과 같이 $k_{correct}$의 터치 영역의 왼쪽 경계가 x_{typo}까지 즉시 확장된다. 그것은 동시에 k_{typo}의 터치 영역이 그 만큼 줄어드는 것을 의미한다. 그리고 $k_{correct}$의 터치 영역의 확장 범위는 k_{typo}의 고정 영역을 넘어설 수 없다. Fig. 5(b)의 왼쪽 그림과 같이 x_{typo}가 k_{typo}의 고정 영역 또는 그보다 왼쪽에 위치하는 경우에 동적 적응을 수행하면, Fig. 5(b)의 오른쪽 그림과 같이 $k_{correct}$의 터치 영역의 확장 범위는 k_{typo}의 고정 영역의 오른쪽 경계로 한정된다. 고정 영역의 크기는 임의로 정할 수 있는데 Gunawardana의 실험 결과를 반영하여 터치 영역의 13%로 지정하였다[9].

4. 실험 및 고찰

4.1 실험 방법

본 논문에서 제안한 방법의 유용성을 확인하기 위해 안드로이드 플랫폼에서 동작하는 소프트 키보드 테스트 앱을 구현하여 실험을 진행하였다. 테스트 앱의 소프트 키보드는 Qwerty 배열 기준으로 불필요한 키를 제외하고 영문 알파벳 위주로 구성하였으며 알파벳 키의 크기는 가로 5.3 mm, 세로 8 mm, 키 사이의 공간은 가로 1 mm, 세로 1.5 mm로서 일반적인 스마트폰과 유사한 크기이다(Fig. 6).

실험의 입력으로 사용하는 테스트 문장은 인접-오타 이외의 오타로 인한 영향을 줄이기 위해, 인식 처리 단계의 오타를 유발할 수 있는 스펠링이 어려운 단어를 배제하고 누구나 쉽게 외워서 입력할 수 있도록 "pineapple strawberry watermelon"으로 정하였다. 테스트 앱은 문장 입력 중에 오타가 발생하면 사용자가 즉시 알아챌 수 있도록 배경색을 바꿔 주며 백스페이스로 오타를 제거하기 전까지 새로운 문자가 입력되지 않도록 하였다.

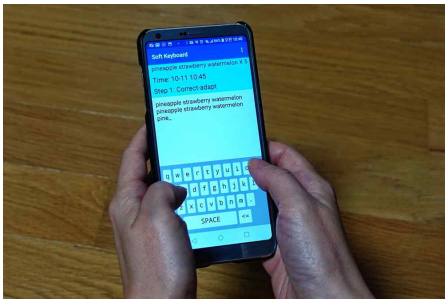


Fig. 6. Soft keyboard test app.

테스트 앱은 기존의 방법과 새로운 방법의 적응 결과를 비교할 수 있도록 설정을 통해 correct-adapt, typo-adapt 그리고 all-adapt 방법 중 한 가지를 선택할 수 있도록 하였다. 실험에는 총 7명의 실험자가 참여하였으며, 각 실험자는 세가지 적응 방법에 대해 각각 테스트 문장을 5회 반복 입력하는 것으로 실험을 진행하였다.

- Correct-adapt: 정타에 대해서만 적응을 수행하는 기존의 동적 적응 방법
- Typo-adapt: 본 연구에서 제안하는 인접-오타를 이용한 동적 적응 방법
- All-adapt: 위의 두 가지 방법을 모두 사용하는 동적 적응 방법

적응의 결과로 변화된 터치 영역으로 인해 오타 판정에 영향을 미치는 assist와 prevent 발생 회수를 측정하는 것은 적응형 키보드의 사용성을 평가하는 기준으로 매우 유용하다[13]. 테스트 앱은 문장 입력 중에 발생하는 assist와 prevent 케이스의 발생 횟수를 측정한다. Fig. 7은 't' 키를 눌러야 하는 시점에서 assist와 prevent 케이스가 발생되는 예를 보여준다.

- Assist: 적응에 의해 확장된 부분을 터치하여 오타 발생을 방지하는 경우
- Prevent: 적응 이전 기준으로는 터치 영역에 해당되지만, 인접키의 확장에 의해 터치 영역이 축소되어 오타로 판정된 경우

4.2 결과 및 고찰

Table 1은 실험 방법에서 설명한 correct-adapt,

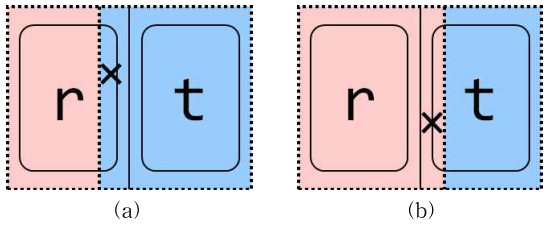


Fig. 7. Two result cases of adaptation affecting typos. (a) Assist case and (b) Prevent case.

Table 1. The average ratio of assist and prevent incidence for each adaptation method

Adapt method	Assist	Prevent	Assist/ Prevent
Correct-adapt	0.14	0.14	100%
Typo-adapt	0.71	0.57	125%
All-adapt	0.57	0.43	133%

typo-adapt 그리고 all-adapt 방법 별로 실험을 통해 측정된 평균 assist와 prevent 발생 회수 및 assist/prevent 비율을 나타낸다. Fig. 8은 실험 중 assist가 발생한 한 실험자의 실험 종료 후 캡처한 키보드 레이아웃으로서, 키의 터치영역을 구분하기 쉽도록 각 키의 터치영역에 서로 다른 색을 지정하여 칠하였다.

올바르게 입력된 키를 사용한 correct-adapt 방법의 경우 assist와 prevent 모두 비교적 낮은 값을 갖는다. 그 이유는 기존 동적 적응 방법의 적응 속도가 느리기 때문에 적응의 효과를 나타내기에 사용자의 키 입력이 충분하지 않았던 것으로 추정된다. Fig. 8(a)을 보면 correct-adapt 방법의 키 터치 영역의 변화가 typo-adapt 방법에 비해 상대적으로 크지 않음을 알 수 있다.

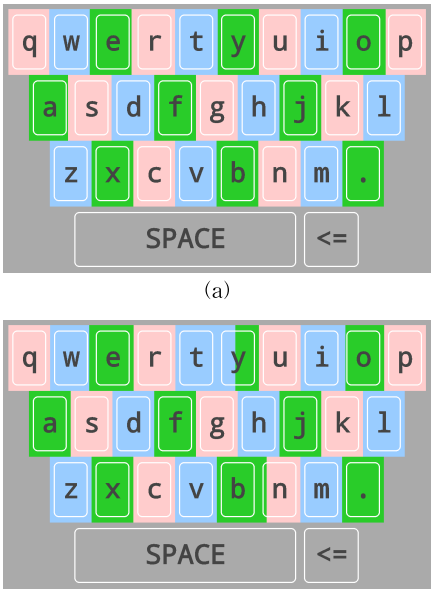


Fig. 8. Layout of the keyboard with adaptation through experiments(key-target area divided by color) (a) Adapted key layout of correct-adapt method and (b) Adapted key layout of typo-adapt method.

인접-오타를 이용하는 typo-adapt 방법의 경우 assist의 발생 회수가 correct-typo의 약 5배 증가하였는데 인접-오타를 이용한 적응 방법이 적은 수의 키 입력만으로도 사용성 향상의 효과를 나타내는 것을 알 수 있다. Fig. 8(b)를 보면 assist가 발생한 't' 키와 'b' 키의 터치 영역의 크기 변화가 큰 것을 시각적으로 확인할 수 있다. 아울러 prevent 발생 회수도 correct-typo의 약 4배 증가하는데 적응 영역의 변이가 크기 때문에 발생하는 부작용이라 생각된다. Fig. 8(b)에서 'y' 키의 터치 영역을 보면 prevent가 발생하기 쉬운 상태임을 추정할 수 있다. 그럼에도 불구하고 사용성의 기준이 되는 assist/prevent 비율은 125%로서 correct-typo에 비해 25% 증가하여 기존 방법 대비 사용성이 향상되었음을 알 수 있다.

두 가지 방법을 동시에 사용하는 all-adapt 방법의 경우 assist와 prevent 모두 correct-adapt와 typo-adapt의 중간 범위의 값을 갖는다. 하지만 assist/prevent 비율이 133%로 가장 높은 결과를 나타냈는데, 이러한 결과는 typo-adapt 방법을 통해 이른 시간에 적응이 시작되고 correct-adapt 방법에 의해 과하게 증가한 적응 영역에 대한 보정이 이루어지기 때문인 것으로 분석된다. 따라서 typo-adapt 방법을 단독으로 사용하는 것보다 기존 방법과 함께 사용하는 것이 사용성 향상에 보다 큰 도움이 된다는 것을 알 수 있다.

5. 결 론

본 논문에서는 소프트 키보드의 동적 적응 방법에 있어, 키 입력 정보로부터 자동으로 오타를 추출하고 오타 간의 관계를 통해 구분한 인접-오타 정보를 이용하여 키보드의 동적 적응을 수행하는 기법을 제시하였다. 이 기법은 적응의 범위를 키 내부로 한정하지 않고 인접키까지 확장하여 넓은 범위에서 빠른 속도로 적응의 효과를 낼 수 있어 적은 수의 입력으로도 키보드 사용성을 향상시킬 수 있다. 실험을 통해 기존 방법과 비교하여 25%의 사용성 증가를 확인했고, 기존 방법과 함께 사용하는 경우 33%까지 사용성이 향상되는 것을 확인하였다.

본 연구는 키보드 동적 적응 방법에 오타 정보를 활용하는 방안을 제시했다는 점에서 큰 의의가 있다.

제안하는 기법은 스마트폰과 같이 입력하는 문장의 길이가 짧은 환경에서[17] 효용 가치가 더욱 클 것으로 예상한다. 현재는 오타 정보를 개별적으로 분석하여 적응에 이용하지만, 향후 과제로 오타 발생 빈도에 따른 가중치를 적용하는 적응 방법과 오타와 정타를 종합적으로 이용하는 적응 방법에 대해 연구할 계획이다.

REFERENCE

[1] B. Kim, T. Kim, Y. Lim, and E.S. Jung, "Usability Evaluation of the Size of Small Touch Keys for the Smart Phone," *Journal of the Korean Institute of Industrial Engineers*, Vol. 38, No. 2, pp. 80-88, 2012.

[2] I.S. MacKenzie and S.X. Zhang, "The Design and Evaluation of a High-Performance Soft Keyboard," *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 25-31, 1999

[3] S. Zhai, M. Hunter, and B.A. Smith, "The Metropolis Keyboard: An Exploration of Quantitative Techniques for Virtual Keyboard Design," *Proceeding of the 5th International Conference on Intelligent User Interfaces*, pp. 119-128, 2000.

[4] T9 is a Predictive Text Technology for Mobile Phones (2009), https://en.wikipedia.org/wiki/T9_(predictive_text) (accessed Aug., 23, 2018)

[5] S. Kwon and H. Kim, "Keypad Method Using Touch-based Drag," *Journal of Korea Multimedia Society*, Vol. 17, No. 10, pp. 1213-1219, 2014

[6] J. Goodman, G. Venolia, K. Steury, and C. Parker, "Language Modeling for Soft Keyboards," *Proceeding of the 7th International Conference on Intelligent User Interfaces*, pp. 194-195, 2002.

[7] J. Himberg, J. Hakkila, P. Kangas, and J. Mantyjarvi, "On-line Personalization of a Touch Screen Based Keyboard," *Proceeding of the 8th International Conference on Intelligent User Interfaces*, pp. 77-84, 2003.

[8] K. Go and Y. Endo, "CATKey: Customizable and Adaptable Touchscreen Keyboard with Bubble Cursor-like Visual Feedback," *Human-Computer Interaction - INTERACT 2007*, LNCS 4662, Part I, pp. 493-496, 2007.

[9] A. Gunawardana, T. Paek, and C. Meek, "Usability Guided Key-Target Resizing for Soft Keyboards," *Proceeding of the 8th International Conference on Intelligent User Interfaces*, pp. 111-118, 2010.

[10] T. Hong, B. Goh, and K.E. Kim, "Adaptive Soft Keyboard via Key-Sequence Prediction : A Development Case Study on Android Platform," *Proceeding of the Korea Information Science Society*, pp. 1767-1769, 2014

[11] L. Findlater and J.O. Wobbrock, "Personalized Input: Improving Ten-Finger Touchscreen Typing through Automatic Adaptation," *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 815-824, 2012.

[12] A. Battestini, V. Setlur, and T. Sohn, "A Large Scale Study of Text Messaging Use," *Proceeding of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, pp. 229-238, 2010.

[13] H. Wassdahl and K. Cho, "Personalized Stroke Order Dependent Keyboard with Adaptive Key-Target Areas Using User Generated Data," *Proceeding of the HCI Society of Korea*, pp. 108-113, 2016.

[14] O.S. Kwon, *Implementation of Input Suggestion System using Neighbor Miss Touch Correction Method on Touch Screen Smartphones*, Master's Thesis of Korea University, 2012.

[15] Y.S. Jeoung and D.M. Choi, "User Adaptive Variable Keyboard for Smart Devices," *Journal of the Korea Institute of Electronic Communication Sciences*, Vol. 12, No. 6, pp. 1167-1172, 2017

[16] C.H. Park, H.W. Kwak, S.I. Kim, Y,J. Kim,

J.W. Kim, K.H. Lee, et al., *Cognitive Engineering Psychology*, Sigma Press, Seoul, Korea, 2009.

[17] J. Choi, "A Comparison of Usability between the Height Adjustable Keyboard and the Adaptive Touch Keyboard on Smartphones," *Journal of the Ergonomics Society of Korea*, Vol. 36, No. 2, pp. 145-156, 2017.

고 석 훈

1993년 2월 동국대학교 전자계산학과 졸업
1995년 2월 KAIST 전산학 석사
1995년 1월~1997년 12월 LG종합기술원 주임연구원
1997년 12월~1999년 11월 SK텔레텍 선임연구원
2000년 4월~2007년 3월 ㈜신지소프트 대표이사
2010년 2월 동국대학교 컴퓨터공학 박사
2012년 2월~2013년 1월 서강대학교 컴퓨터공학과 강의전담교수
2017년 3월~현재 한국외국어대학교 정보산업공학연구소 교수
관심분야 : HCI, 임베디드 소프트웨어, 컴퓨터 교육