http://dx.doi.org/10.7236/JIIBC.2013.13.6.137
JIIBC 2013-6-18

모바일 단말기를 위한 추천 소프트 키보드

Preliminary Study on Soft Keyboard with Recommendation for Mobile Device

황기태*, 이재문**

Kitae Hwang, Jae-Moon Lee

요 약 최근 대부분의 모바일 단말기는 LCD 터치 화면에 소프트 키보드를 장착한다. 그러나 소프트 키보드의 터치 화면의 크기가 작기 때문에, 인접키가 실수로 눌러지고, 한 키를 여러 키 입력에 중복 사용함에 따라 사용자의 키 입력 오류가 많이 발생한다. 본 논문에서는 사용자가 텍스트를 입력하는 동안 적절한 어휘를 추천하여 키 입력 오류를 쉽게 수정하도록 돕는 알고리즘을 제안하고 MissLess 소프트 키보드를 구현한 내용에 대해 기술한다. 3개의 간단한 테스트 셋을 작성하여 MissLess 키보드의 추천 성능을 평가한 결과 다소 차이가 있지만 90%에 달하는 추천 성능을 얻었다. 그러나 추천 워드가 4개인 것을 고려하면 추천 성능을 해석할 필요가 있다.

Abstract Recently most mobile devices have soft keyboards on their LCD touch screens. Because of the tiny size of the touch screen of the soft keyboard, adjacent keys are mistakenly typed. Also utilizing a key for multiple key inputs causes key type errors. In this paper, we proposed an algorithm to recommend proper words to the user while the user continues to type keys, which helps to easily correct key type errors. In addition, we presented a soft keyboard called MissLess which implemented the recommendation algorithm. We evaluated recommendation performance of MissLess keyboard through experiments by using 3 test sets. The test results showed that the success ratio of recommendation reached up to about 90% although there were some differences between results. However it is needed to be considered that we recommended 4 words for an input word in this experiments

Key Words : Android, Soft keyboard, Recommendation, Mobile device, Touch screen

Ⅰ. 서 론

현재 터치 스크린을 가진 모바일 혹은 태블릿 장치들이 폭발적으로 확산되고 있다[1,2].이들은 하드웨어 키보드 대신 터치 스크린 상에 그래픽으로 그린 소프트 키보드를 사용한다[3,4]. 소프트 키보드는 크기, 키 배치, 언어에 제약 받지 않고, 자유롭게 터치 스크린 상에서 구현되는 장점을 가진다. 그러나 모바일이나 태블릿 장치의 터치 스크린의 크기 한계로 인해, 소프트 키보드의 키 크기가 작고 인접한 키의 거리가 좁고, 접촉 피드백의 전달이 부족하므로, 키 입력에 따른 오류 발생이 많은 편이다[5]. 또한 하드웨어 키보드의 경우 100개 내외의 키를 제공하고 있는 반면, 소프트 키보드는 좁은 공간 때문에 충분한 개수의 키를 배치할 수 없어 하나의

*정회원, 한성대학교 컴퓨터공학과 (교신저자)
**정회원, 한성대학교 멀티미디어공학과
접수일자 2013년 10월 11일, 수정완료 2013년 11월 26일
게재확정일자 2013년 12월 13일

Received: 11 October, 2013 / Revised: 26 November, 2013
Accepted: 13 December, 2013
*Corresponding Author: calafk@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

키 버튼으로 여러 키를 공유한다. 이로 인해 키 입력의 불편함과 키 입력 오류가 잦다[5,6].

소프트 키보드의 출현과 함께 소프트 키보드의 키 입력 오류를 줄이고 키 입력 속도를 높이는 방법에 대한 많은 연구가 진행되어 왔다. 입력되는 키 스트림에서 다음 키를 예측하여 예측된 키의 너비를 동적으로 늘이거나[6], 키 하이라이팅을 이용하여 사용자의 시각적인 관심을 유도하여 자연스럽게 키 입력을 성공시키거나[7], 사용자의 텍스트 입력 패턴을 분석하고 그에 따라 키 배치를 조절하여 입력되는 키 거리를 줄이는 방법 등이 연구되었다[8].

키 배치를 조절하는 방법은 새로운 키보드를 익혀야 하는 불편함을 동반하며, 사용자가 키를 입력하는 동안 다음 키를 예측하여 키의 크기를 조절하는 방법은, 사용자의 예측과 일치하지 않았을 때, 오히려 오타의 가능성이 있고, 자판이 불규칙하게 움직이게 되면 오히려 불편함을 줄 수 있다.

지금까지 연구의 대부분은 키 배치나 사용자 입력을 예측하여 추천하는 방법을 통해 사용자 오타 입력을 줄이는데 초점이 맞추어져 왔다. 그러나 본 연구에서는 오타의 경우를 줄이기보다는, 텍스트를 입력하는 동안 사용자가 의도하는 입력을 추론하여 이를 추천함으로써 오타가 발생하는 경우 사용자가 빠르게 교정할 수 있게 하여 사용자의 키 입력 편리를 도모하는 추천 키 입력 시스템을 제안하고 MissLess로 불리는 소프트 키보드를 구현하였다.

본 논문에서 제안하는 알고리즘은 대략 다음과 같이 작동한다. 사용자가 소프트 키보드로 텍스트를 입력하면 키가 입력될 때마다 입력된 워드 T의 해시 코드 C를 계산하고 C 값에서 일정 범위의 해시 값을 가진 워드를 어휘 사전에서 추출한다. 그리고 추출된 워드들과 T의 편집 거리(Edit Distance)를 계산하여 T와 편집 거리가 가장 가까운 순서로 정렬하고, 높은 순위 워드들을 대상으로 사용자의 사용 빈도수를 반영하여 추천 워드를 결정한다.

본 논문은 다음과 같이 구성된다. 2 장에서 연구 배경을 기술하고 3 장에서는 추천 알고리즘에 대해 기술한다. 4 장에서는 안드로이드 스마트폰에 MissLess 키보드를 구현한 내용을 기술하며 5 장에서 성능을 평가하고 6 장에서 결론을 맺는다.

II. 연구 배경

1. 관련 연구

이미 1990년대 초반부터 모바일 장치의 텍스트 입력에 관한 연구가 진행되어 왔다. 모바일 단말기의 터치 스크린에 그래픽 이미지로 구현한 소프트 키보드의 경우, 사용자 키 타입에 대한 피드백이 없기 때문에, 키 입력 오류가 많아지게 되었고, 키 입력 오류를 줄이기 위한 많은 연구들이 진행되어 왔다. 데이터베이스를 기반으로 키 배치를 자동으로 수행하는 방법[8], 사용자의 다음 입력을 예상하여 키의 크기를 조절하는 방법[5,6,7], 사용자의 손 위치에 따라 키보드가 터치 스크린을 움직이는 방법[9], 체스쳐(gesture)를 이용하여 키를 입력하는 방법[10] 등이 제안되고 현재도 개발되고 있다.

최근에 와서는 구글 키, 삼성 키, 스위프트 키 등 안드로이드 단말기나 애플의 아이폰에서 작동되는 추천식 소프트 키보드들이 상용으로 배포되고 있다.

그러나 현재 추천 소프트 키보드의 알고리즘이나 추천 성능에 대한 공개된 평가 설명된 문헌이 거의 없다.

2. 편집 거리(Edit Distance)

본 논문에서 제안하는 알고리즘은 사용자가 입력한 텍스트에 대해 추천 워드를 선택하기 위해, 두 워드 사이의 유사성을 판단하는 과정을 가지고 있다.

편집 거리(Edit Distance)[11]는 Levenshtein에 의해 제안된 알고리즘으로서, 두 개의 문자열이나 패턴이 얼마나 유사한지를 평가한다. 만일 두 문자열 A, B가 있다고 할 때, A가 B가 되기 위해 필요한 삽입(insert), 삭제(delete), 대치(replacement) 연산의 최소 횟수를 Edit Distance라고 부른다.

두 문자열 A와 B가 유사할수록 Edit Distance는 작게 계산된다. 예를 들어 tax가 taxi가 되려면 i가 삽입되어야 하므로 두 단어의 Edit Distance는 1이다. 또한 delegate가 delete가 되려면 g, a가 삭제되어야 하므로 두 단어의 Edit Distance는 2가 되며, park이 spark가 되기 위해서는 s를 삽입하고, r을 k로 대치하고, k를 e로 대치해야 하므로 Edit Distance는 3이 된다. 본 논문에서는 Edit Distance 알고리즘을 수정한 Modified Edit Distance를 제안하고 사용하였다.

Ⅲ. MissLess 추천 시스템

1. 추천 알고리즘

사용자가 터치 스크린 상에서 텍스트를 입력할 때마다, 본 추천 시스템은 사용자가 입력하리라고 예상하는 워드를 추천한다. 예를 들어 사용자가 “안녕”을 입력하려다 “안뇽”을 입력한 경우 “안녕”을 추천하여 사용자가 선택할 수 있게 한다. 추천 워드의 개수는 사용자가 결정할 수 있으나 현재 MissLess 테스트 키보드에서는 4개로 하였으며, 이들은 키보드 상단에 출력된다. 4 개의 추천 워드 중에 사용자가 선택할 수 있으며, 이때 추천 성공이 된다.

그림 1은 본 논문에서 제안한 추천 시스템의 구조를 보여준다. 어휘 사전은 일반적으로 많이 사용하는 워드와 사용자가 주로 사용하는 워드가 합쳐져 구성되며, 각 워드별로 하나의 해시 정수 값이 할당되어 함께 저장된다. 사용자가 키를 입력할 때마다 현재 입력된 텍스트 Word를 해시 정수로 변환하고, 어휘 사전에 있는 각 워드의 해시 값과 비교하여 일정 범위 내의 차이를 가진 모든 워드를 해시 값 순서로 정렬하여 워드 리스트를 만든다(해시필터). 그리고 다시 Word와 워드 리스트의 각 워드에 대해 철자 유사성을 비교하여 철자 유사성이 높은 순위로 재정렬한다(철자유사성 기반 정렬). 그리고 다시 워드리스트에 있는 상위 워드 중에서 사용자의 사용 빈도수가 높은 워드를 우선적으로 N개(현재 4개) 선정하여 최종 추천한다(사용자패턴기반 추천).

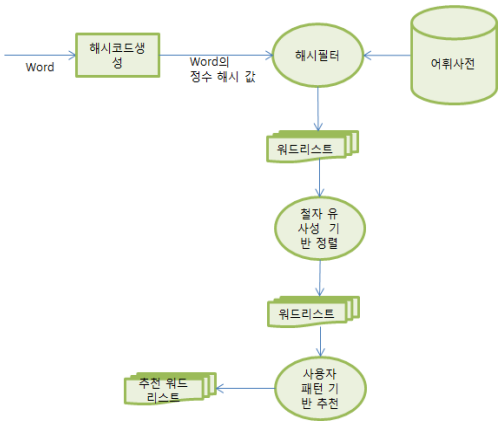

그림 1. 사용자 입력 키에 대한 워드 추천 프로세스
Fig. 1. Recommendation Process for Input Word

2. 해시 코드 생성

본 논문에서는 사용자가 스페이스(Space)로 구분하여 입력하는 한글의 어절을 ‘워드’로 지칭한다. 워드 W는 식 (1)과 같이 n개의 글자로 구성된다.

$$W = w_1+w_2+ \cdots +w_n \quad (1)$$

각 글자 w_n은 한글 문법에 맞는 ‘가’, ‘방’, ‘에’ 등의 글자뿐 아니라, 음소 ‘ㄱ’, ‘ㅓ’ 등도 가능하다. 그러므로 예를 들면 본 논문의 사전에 기록되는 워드에는 다음과 같은 것들이 존재한다.

‘안녕’, ‘컴퓨터’, “ㅠㅠ“

다시 w_n은 초성, 중성, 종성의 음소들로 구성되므로 다음과 같이 표현된다.

$$w_n = e_{n1}+e_{n2}+ \cdots +e_{nm} \quad (2)$$

본 논문에서는 어휘 사전에 저장된 각 워드마다 해시 코드 정수 값을 할당한다. 해시 코드의 목적은 워드들의 유사성을 쉽게 분류하기 위해서이다. 해시 코드를 생성하기 위해 표 1과 같이 한글의 모든 자음, 모음에 정수 값의 기본 코드를 부여한다. 각 워드의 해시 값은 유일하지 않으며, 동일한 해시 값을 가진 워드들이 존재할 수 있다.

해시 코드를 생성하는 원칙은 다음과 같다.

첫 번째, 비슷한 두 워드는 가능하면 가까운 해시 값이 되도록 한다. 비슷한 두 개의 워드가 구성하는 음소들이 거의 같다면, 이들의 해시 값 차이는 매우 근소하게 된다.

두 번째, 사용자들의 입력 습관상 오타는 발견 즉시 수정하기 때문에 여러 글자로 이루어지는 워드에서 뒤에 입력된 글자에 오타가 생길 가능성이 매우 높고, 첫 번째 글자가 서로 다른 두 워드는 유사성이 매우 낮다. 따라서 두 워드에서 동일한 글자가 첫 글자인 경우와 마지막 글자인 경우 해시 값에 다른 영향을 주도록 계수(Letter Weight) α(alpha)를 도입하여 해시 함수를 구성한다.

세 번째, 키보드의 특징을 반영하여 음소에 코드 값을 부여한다. 동일한 키에 중복된 한글 음소들은 가까운

값들로 기본 코드 값을 부여하였다.

표 1. 한글 음소에 대한 코드 값
Table 1. Codes for Hangul Phonemes

초성		중성		종성	
음소	코드	음소	코드	음소	코드
ㄱ	10000	ㅏ	2000	ㄱ	1
ㅋ	20000	ㅑ	2050	ㅋ	2
				ㄲ	3
ㄲ	30000	ㅓ	2500	ㄱㅅ	80
ㄴ	40000	ㅕ	2550	ㄴ	11
ㄹ	50000	ㅗ	6500	ㄴㅈ	81
ㄷ	70000	ㅛ	6550	ㄴㅎ	82
		ㅜ	4000	ㄷ	21
ㅌ	80000	ㅠ	4050	ㄹ	12
ㄸ	90000	ㅐ	2100	ㄹㄱ	83
				ㄹㅁ	84
ㅂ	100000	ㅒ	2150	ㄹㅂ	85
ㅍ	110000	ㅔ	2600	ㄹㅅ	86
ㅃ	120000	ㅖ	2650	ㄹㅌ	87
		ㅚ	8500	ㄹㅍ	88
ㅅ	130000	ㅘ	8550	ㄹㅎ	89
ㅎ	140000	ㅙ	8650	ㅁ	62
ㅆ	150000	ㅝ	6050	ㅂ	31
ㅈ	160000	ㅟ	6000	ㅂㅅ	90
				ㅅ	41
ㅊ	170000	ㅞ	6150	ㅆ	43
ㅉ	180000	ㅡ	4500	ㅇ	61
ㅇ	190000	ㅣ	3500	ㅈ	51
ㅁ	200000	ㅢ	5000	ㅊ	52
				ㅌ	22
				ㅍ	32
		.	5500	ㅎ	42

k번째의 글자 w_k의 해시 값 H_k는 w_k를 구성하는 각 음소의 코드 값의 합이므로 음소의 개수가 m인 경우 다음과 같이 표현된다.

$$H_k = \sum(ei_k\text{의 코드값}),\ \text{for } i=1..m \qquad (3)$$

이제 워드 W가 n 개의 글자로 구성될 때, 해시 값 C를 계산하는 함수는 식 (4)와 같다.

$$C = \sum H_k \times (\alpha/k),\ \text{for } k=1..n \qquad (4)$$

여기서 α(alpha)는 한 워드에서 첫 번째 글자인 경우와 그 다음 글자의 경우, 서로 구분을 명확히 하기 위해 설정하는 계수(Letter Weight)이다. 즉, 첫 번째 글자에는 해시 값에 α를 곱하고, 두 번째 글자에는 α/2를 곱하고, 세 번째 글자에는 α/3을 곱한다. 식 (4)에서 α를 100으로 하여 "이용"과 "용이" 두 단어의 해시 코드를 계산해보자.

"이용"의 해시 값 = (초성ㅇ코드 + ㅣ코드)*(100/1) + (초성ㅇ코드 + ㅛ코드 + 종성ㅇ코드)*(100/2)
= 19350000 + 9830550 = 29180550

"용이"의 해시 값 = (초성ㅇ코드 + ㅛ코드 + 종성ㅇ코드)*100/1 + (초성ㅇ코드 + ㅣ코드)*(100/2)
= 19661100 + 9675000 = 29336100

두 번째 글자의 해시 값 가중치를 (100/2)로 하였기 때문에 "이용"과 "용이"는 서로 다른 해시 값으로 계산된다.

3. 어휘 사전

본 논문에서 다루는 사전은 인터넷에서 주로 사용되거나 사용자가 자주 사용하는 워드로 구성된다. 각 워드 W는 참조 카운트 R, 해시 코드 C와 함께 다음과 같이 사전에 저장된다.

$$Dic = \{(W_1, C_1, R_1), (W_2, C_2, R_2), .. , (W_n, C_n, R_n)\}$$

사전에 포함되는 어휘 개수는 모바일 단말기의 메모리 량을 고려하여 적당히 결정한다. 본 연구에서 안드로이드 스마트폰에 2500개 어휘를 사용하였다.

4. 해시 필터링

사용자가 키를 입력할 때마다 현재 입력된 텍스트 T에 대해 해시 코드 C_T 생성한다. 그리고 어휘 사전에 있는 모든 워드와 해시 값을 비교하여 차이가 일정 범위 내인 것들을 골라 워드 리스트 L_h를 생성한다.

$$L_h = \{\ W_i\ \},\ |C_{wi} - C_T| < \text{HashWindow} \qquad (5)$$

여기서 HashWindow는 정수로서 $C_T \times \beta$에 의해 결정되며, β(0<β<1의 실수)는 사용자가 현재 입력한 텍스트 T와 유사한 모양의 워드를 골라내기 위해 C_T를 기

준으로 해시 코드의 범위를 지정하는 계수이다. β가 클수록 T와 모양이 다른 사전 어휘를 L_h에 포함하게 될 확률과 L_h의 크기가 커지고, 해시 필터링 처리 시간 및 뒤이어지는 알고리즘 실행 시간이 길어질 가능성이 높다.

해시 필터링은 사용자의 입력 텍스트와 유사한 워드를 추려 내기 위한 쉽고도 간단한 과정이다. 뒤이어지는 철자 유사성 기반 정렬에 사용하는 Levenshtein 알고리즘은 많은 시간을 소모하기 때문에 일차적으로 유사한 단어를 골라내기 위한 선행 작업이다.

5. 철자 유사성 기반 정렬

본 연구에서는 입력된 텍스트 T와 철자가 유사한 워드들을 사전에서 골라내기 위해 Edit Distance 알고리즘을 수정하여 Modified Edit Distance(MED)를 고안하였다. 사용자가 입력한 워드 T와 사전에 있는 단어 W와의 편집 거리를 ed(T, W) 라고 하고, 단어 W의 철자 개수를 Size(W)라고 할 때, T와 W의 med는 다음과 같이 계산한다.

$$med(T,W)=(1-ed(T,W)/Size(W))*100 \quad (6)$$

기존의 편집 거리(ED)는 1, 2, 3 과 같은 정수로 표기되나, MED의 경우 단어의 길이를 고려하며 백분위 값으로 표기하기 때문에 더 세밀하게 어휘 유사성을 표현할 수 있다. 만일 T와 W의 두 단어가 같다면 ed(T,W)는 0이 되고, med(T,W)는 100이 된다. 철자 유사성 기반 정렬 결과 L_h 리스트가 MED 값이 낮은 순서로 정렬되어 L_{med} 리스트로 변경된다.

6. 사용자 패턴 기반 추천

L_{med} 리스트의 워드 중에서 사용자가 자주 사용하는 워드를 우선적으로 추천한다. MED 값이 같은 워드가 여러 개 있을 수 있기 때문에 동일한 MED를 갖는 워드들을 그룹으로 묶는다. 추천 워드의 개수가 N이라고 할 때, L_{med}에서 첫 번째 그룹을 대상으로 참조 카운트 순으로 정렬하고 이 중에서 참조 카운트가 높은 N개를 추천한다. 만일 첫 번째 그룹의 워드 개수가 N보다 작으면 이들을 모두 추천하고, 다음 그룹에서 동일한 방법으로 정렬하고 앞서 추천하고 남은 개수만큼 높은 참조 카운트 값을 가진 워드를 추천한다. N 개를 모두 추천할 때까지 다음 그룹으로 이동하면서 반복한다. N을 4로 하여 실험한 결과 처음 두 그룹 내에서 거의 추천이 이루어졌다.

7. 참조 카운트 갱신

워드들은 사용빈도수와 함께 사전에 저장된다. 인터넷 상에서 크롤링하여 선택하였기 때문에 각 워드의 사용빈도수는 인터넷을 사용하는 사용자들의 평균적인 빈도이다. 본 연구에서는 키가 입력될 때마다 사용자가 입력한 워드의 사용 빈도를 지속적으로 변경한다.

사용자가 띄어쓰기를 하거나 전송/Enter 버튼을 누르거나 추천 창에 있는 워드를 선택하였을 때, 하나의 워드가 완성된 것으로 간주한다. 그리고 이때 어휘 사전에 저장한다. 띄어쓰기나 전송/Enter 버튼이 눌러졌을 경우, 추천 과정이 진행되는 도중에 사용자가 입력한 워드와 어휘 사전의 워드들을 비교하여 같은 단어가 있으면, 현재 해쉬 필터 과정에서 검색된 워드 리스트 L_h 중에서 참조 카운트(reference count) 값이 제일 큰 워드를 선택하고 이 워드의 참조 카운트 값에 10을 더한 값을 구하여 이 값을 사용자가 입력한 워드의 참조 카운트 값으로 설정한다. 만일 사용자가 입력한 단어가 어휘 사전에 없는 경우, 사용자가 오타를 입력하였거나 어휘 사전에 없는 단어를 입력하였을 수 있기 때문에, 임시 저장소에 저장해 둔다. 임시 저장소에 만일 이미 이 워드가 있으면, 이 워드를 오타인지 정타인지 판단할 필요 없이 사용자가 즐겨 사용하는 단어로 판단하여 어휘 사전에 저장하고, 현재 참조 카운트 값이 가장 작은 워드를 어휘 사전에서 제거한다.

사용자가 추천 창에서 워드를 선택하는 경우, 해시 필터 과정에서 검색된 워드 리스트 L_h 중에서 가장 높은 참조 카운트 값에 10을 더해서 추천된 워드의 참조 카운트 값을 갱신한다.

8. 알고리즘 실행 예

본 논문에서 제안하는 추천 알고리즘의 실행 예는 그림 2와 같다. 사용자는 '사진'을 입력하고 하였는데 잘못하여 '사잔'을 입력하였다. '사잔'과 phone DB의 2500 개 워드와 해시 코드를 비교하는 해시필터링을 거쳐 193 개의 워드를 추려내고, 철자 유사성 기반으로 정렬한 후, 사용 빈도수를 고려하고 4 개의 어휘를 추천한다. 이곳에 '사진'이 포함되어 있다. 그림 4는 구현된 결과를 보여준다.

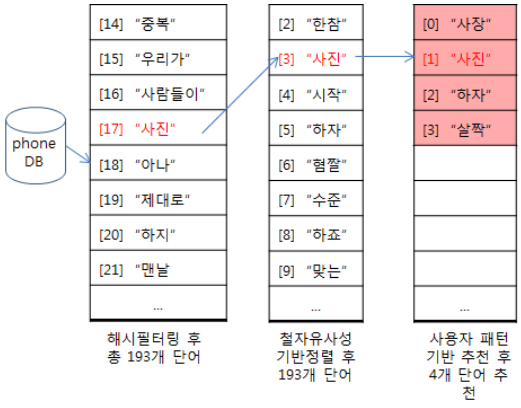

그림 2. 알고리즘의 실행 예
Fig. 2. A running sample of recommendation algorithm

Ⅳ. 시스템 구현

본 논문에서 제안한 알고리즘을 테스트하기 위해 안드로이드 플랫폼에 그림 3과 같은 구조로 소프트 키보드를 구현하였다. PC에서 웹 상에서 사용되는 빈도수가 높은 워드 중 2500개를 추려내고, Hash Code Generator를 이용하여 각 워드의 해시 코드를 생성하여 Phone DB를 구성한다.

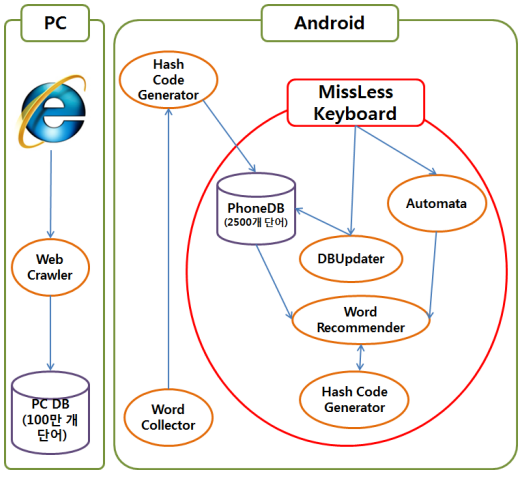

그림 3. 시스템 모듈 설계
Fig. 3. System Module Design

사용자가 키를 입력할 때마다 Automata(오토마타 모듈)로 전달되고, Automata는 키를 조합하여 한글 문자로 변환하고 화면에 출력한다. 동시에 현재까지 입력된 텍스트를 Word Recommender(추천 모듈)로 전달한다. Word Recommender는 Hash Code Generator를 호출하여 입력된 텍스트를 해시 코드로 변환하고, 본 논문에서 제안된 추천 알고리즘을 이용하여 사전을 검색하여 워드를 추천한다. 추천된 워드는 추천 창에 출력한다.

그림 4는 구현된 MissLess 추천 키보드의 사용 실례를 보여준다. MissLess 키보드는 사용자가 원하는 이미지를 키보드에 배치할 수 있다. 그림 5에서 사용자는 '사진'을 입력하려다가 '사잔'을 입력한 경우이며, 추천 창에는 '사진'이 추천된 것을 볼 수 있다.

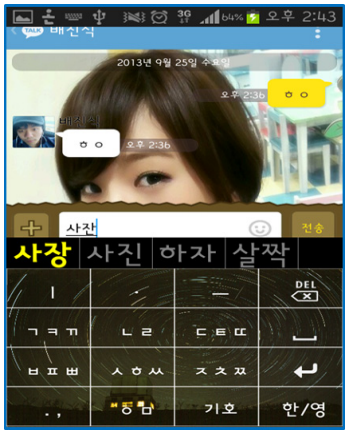

그림 4. 본 연구에서 구현된 추천 키보드
Fig. 4. Soft keyboard implemented in this research

Ⅴ. 성능 평가

1. 성능 평가 지수

본 연구의 성능 평가 지수는 추천 성공률이다. 사용자가 입력하고자 하는 어휘가 추천 리스트에 존재하면 추천 성공으로 본다. 추천 성공률 S는 다음과 같다.

$$S = \text{추천 성공 회수} / \text{입력된 어휘 개수} \quad (7)$$

추천 리스트 N의 크기가 크면 당연히 추천 성공률이 올라가므로 N의 크기를 현실적으로 설정하는 것이 중요하다. 본 연구에서는 LCD 크기를 고려하여 N을 4로 설정한다. 알고리즘의 파라미터로 α(alpha)는 100으로, HashWindow를 결정하는 β는 실험에서 변화시키면서

성능을 관찰한다.

2. 실험 방법 및 추천 성공 판단

본 연구에서 설계된 알고리즘의 성능을 평가하기 위해서는 알고리즘이 구현된 키보드 상에서 사용자가 직접 워드를 입력하게 하고, 오타에 대해 시스템이 추천하는 워드에 사용자가 즉각적으로 추천 성공을 평가하는 것이 옳은 방법이지만, 평가가 쉽지 않다.

그러므로 본 연구에서는 실제 상황과 유사한 실험을 위해, 사용자의 행위를 사전에 테스트 셋으로 만들어 두고 알고리즘이 구현된 키보드 소프트웨어에 배치(batch) 방식으로 전달하여 성능을 테스트한다. 테스트에 사용되는 단위 데이터는 '정상 워드'와 '오타 워드'로 구성되며, 오타 워드를 입력으로 주었을 때, 정상 워드가 추천 리스트에 포함되면 추천 성공으로 판별한다.

본 연구에서는 실험을 위해 TS1, TS2, TS3 등 3개의 테스트 셋을 구성하였다.

가. 테스트 셋 1(TS1)

본 연구를 도와주는 보조원들의 '카카오톡' 메신저 그룹 채팅에서 자주 사용된 단어를 랜덤하게 100개 추출하여 '정상 워드'로 구성하고, 각 정상 워드에 대해 직접 타이핑해보며 가장 많이 발생하는 오타를 '오타 워드'로 설정하였다. 채팅 방의 규모가 30명 인원이므로 평균적으로 사용되는 워드와는 약간의 거리가 있을 수 있다. 표 2는 몇 개의 샘플을 보여 준다.

나. 테스트 셋 2(TS2)

일반 커뮤니티 사이트를 통해 자주 사용된 워드 1500개 중에서 랜덤하게 100개를 추출하였다. 그리고 본 연구의 보조원들이 이 워드들에 대해서 직접 타이핑하면서 발생하는 오타로 테스트 셋을 구성하였다.

다. 테스트 셋 3(TS3)

일반 커뮤니티 사이트를 통해 추출한 워드 중에서 사용률이 가장 떨어지는 워드 1000개를 선정하고 이 중에서 100개를 랜덤하게 선택하였다. 그리고 본 연구의 보조원들이 이 워드들에 대해서 직접 타이핑하면서 발생하는 오타로 테스트 셋을 구성하였다.

표 2. 테스트 셋 TS1의 샘플
Table 2. Samples of TS1

정상 워드	오타 워드	정상 워드	오타 워드	정상 워드	오타 워드
사진	사잔	승리	슨리	그래	그램
그래	그램	오늘	오른	진짜	친짜
교수	겨수	대박	대갑	잘자	잘ㅈ거

3. 실험 및 결과

가. 테스트 셋에 따른 추천 성공률

추천 성공률을 실험한 결과는 표 3과 그림 5에서 보여준다. 실험 결과에 의하면 TS2의 추천 성공률이 97%에 이르렀지만 테스트 셋에 따라 추천 성공률이 다르게 나타났다. TS1의 추천 성공률이 특히 낮은데 이것은 TS1은 특정 사용자 집단의 '카카오 톡'에서 사용된 어휘를 대상으로 하였기 때문에, 인터넷에서 주로 사용되는 어휘를 대상으로 구성된 어휘 사전에는 없는 어휘가 많기 때문이다. 표 3의 결과와 같이 사전에 처음부터 없어서 추천되지 않는 경우 20%를 제거하면 성공률은 85%가 된다.

한편, 추천 창 N의 크기가 4이므로 정상 워드가 4개 중에 들기만 하면 추천 성공이다. 그러므로 하나만 추천하는 경우를 따진다면, 추천 성공률은 다소 떨어질 것이다.

β가 클수록 추천 성공률이 좋아지는 것은 그림 6에서 볼 수 있다. β가 크면 알고리즘에서 다루는 워드의 개수가 늘어나므로 추천 시간에 악영향을 미친다.

표 3. β =0.3일 때 추천 알고리즘 성능 결과
Table 3. Results of Performance Evaluation for β =0.3

테스트셋	추천 성공률	추천 워드가 L_h 에 포함되지 않아 실패한 경우	추천 워드 크기 N에 포함되지 않아 실패한 경우	추천 단어가 사전에 없어 실패한 경우
TS1	65%	4%	11%	20%
TS2	97%	1%	2%	0%
TS3	94%	3%	3%	0%

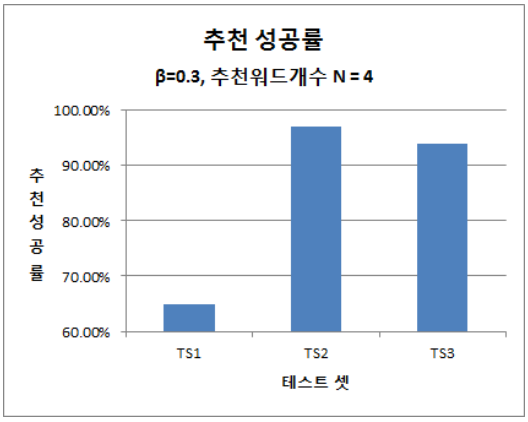

그림 5. β =0.3일 때, 3 개의 테스트 셋의 추천 성공률
Fig. 5. Success ratios of recommendation for three test sets for β =0.3

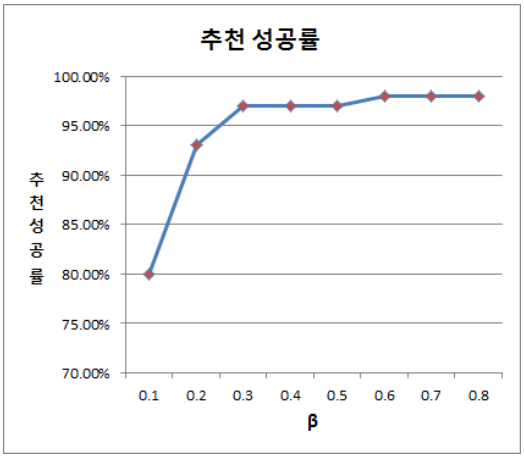

그림 6. TS2에 대해 β 값에 따른 추천 성공률
Fig. 6. Success ratios of recommendation for TS2 according to β

VI. 결론

본 논문은 터치 스크린을 이용하는 소프트 키보드에서 사용자의 오타를 줄일 수 있는 알고리즘을 설계하고 안드로이드 스마트폰에 구현하여 알고리즘을 현실화하는 기초 연구이다. 우리는 3 단계로 구성되는 워드 추천 알고리즘을 제안하고 성능을 평가하였으며, 좋은 성능 결과를 얻게 되었다.

그러나 본 연구는 앞으로 개선하고 연구할 몇 가지 부분이 있다. 첫째, 테스트 셋의 크기가 부족하고, 테스트 셋에 포함되는 '정상 워드'와 '오타 워드'가 실험자의 개인적인 것이어서, 실험의 현실성이 떨어진다. 둘째, 워드 리스트 L_h나 L_{med} 리스트의 크기에 비례하여 실행 시간이 길어지게 되므로 이에 대한 대책이 필요하다. 셋째, 사용자가 많이 사용하는 단어에 대한 추천 성공률을 높이는 알고리즘의 개선이 필요하다.

결론적으로 본 논문의 기여는 다음과 같다. 첫째, 제안된 알고리즘은 장치와 무관하게 키입력 오류를 줄인다. 둘째, 사용자의 입력 워드를 고려한 개인화(personalized) 키 입력 시스템이고, 맞춤법과 무관하게 작동하므로 사용자의 실제 습관에 부합된 키보드이다. 셋째, 온라인 키 추천을 지원한다. 키를 입력하는 매 순간 사용자의 키 입력 오류를 수정하거나 추천한다.

References

[1] Seokjin Im, Hee Joung Hwang, "Design and Implementation of Real-Time Health Information Sharing Framework for N-Screen Service", Journal of Korean Institute of Information Technology, vol. 11, no. 10, pp. 171-180, Oct. 2013

[2] Kitae Hwang, Hye-Kyung Cho, "A Design and Implementation of Prototype of Dual Screen Platform on Android", Journal of the Institute of Webcasting, Internet and Telecommunication, vol. 12, no. 3, pp. 163-169, June 2012

[3] I. Scott MacKenzie, "Introduction to This Special Issue on Text Entry for Mobile Computing", Human-Computer Interaction, vol. 17, pp.141 - 145, 2002.

[4] Per Ola Kristensson, "Five Challenges for Intelligent Text Entry Methods", Association for the Advancement of Artificial Intelligence, pp.85-94, 2009

[5] Asela Gunawardana, Tim Paek, Christopher Meek, "Usability Guided Key-Target Resizing for Soft Keyboards", In proceedings of International conference of Intelligent User Interface, pp.111-118, 2010.

[6] Seongmin Kimo, Nahyun Kim, Woongsub Byun, Junhwan Choi, Taeseok Kim, "Soft Keyboard

application for reducing the mis-typing ratio in the smartphones", In Proceedings of Fall Conference, Korea Institute of Information Scientists and Engineers, vol. 38, no.2, pp.89-92, 2011.

[7] L. Magnien, J. Bouraoui, and N. Vigouroux. "Mobile text input with soft keyboards: optimization by means of visual clues", In Proceedings of Mobile HCI, pp.337 - 341, Springer-Verlag, 2004.

[8] Kye-Suk Lee, Hwan-Seung Yong, "Research on the Automatic Software Keyboard Based on Database", Journal of Korea Multimedia Society, vol. 8, no. 1,pp.101-110, 2005.

[9] Lung-Pan Cheng, Hsiang-Sheng Liang, Che-YangWu, Mike Y. Chen, "iGrasp: Grasp-based Adaptive Keyboard for Mobile Devices", In Proceedings of CHI '13, pp.3037-3046, 2013.

[10] Shumin Zhai and Per Ola Kristensson, "The Word-Gesture Keyboard: Reimagining Keyboard Interaction", communications of the acm, vol. 55, no. 9, pp.91-101, 2012.

[11] A. Levenstein, "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," Soviet Physics-Doklady, vol. 10, no. 8, pp.707-710, 1966.

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음

저자 소개

황 기 태(정회원)

• 서울대학교 컴퓨터공학과 학사
서울대학교 컴퓨터공학과 석사
서울대학교 컴퓨터공학과 박사
• 경력
University of Callifornia, Irvine 방문 교수
University of Florida 방문 교수

<주관심분야 : 모바일 시스템>

이 재 문(정회원)

• 한양대학교 전자공학과(학사)
한국과학기술원 전기및전자공학과(석사)
한국과학기술원 전기및전자공학과(박사)
• 경력
한국통신 연구개발단

<주관심분야 : 기계학습, 게임프로그래밍, 감성컴퓨팅>