The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 14, No. 4, pp.187-195, Aug. 31, 2014. pISSN 2289-0238, eISSN 2289-0246

http://dx.doi.org/10.7236/JIIBC.2014.14.4.187
JIIBC 2014-4-27

추천 소프트 키보드 MissLess의 성능 평가

Performance Evaluation of MissLess Soft Keyboard with Recommendation

황기태*, 김태완**, 조혜경***

Kitae Hwang*, Tae-Wan Kim, Hye-Kyung Cho*****

요 약 본 논문은 선행 연구에서 개발한 MissLess 키보드의 성능을 평가한다. MissLess 키보드는 모바일 단말기 내의 사전에 저장되는 모든 어휘에 해시 코드 값을 할당한다. 그리고, 사용자가 입력한 어휘와 일정 범위의 해시 코드를 가진 어휘를 골라내는 해시필터링, 골라낸 어휘와 입력된 어휘의 철자 유사성을 실시간으로 비교하여 유사 단어를 추려내는 철자 유사성 기반 정렬, 그리고 마지막으로 사용자의 사용 빈도가 높은 어휘를 우선적으로 골라내는 사용자 패턴 기반 추천의 3 단계를 거쳐 어휘를 추천한다. 본 논문은 추천 알고리즘의 각 단계별로 성능 요소를 분석하고, 이들 요소가 미치는 추천 성능의 영향을 평가하고 평가된 결과를 보인다.

Abstract In this paper, we evaluated the performance of recommendation and run time of the MissLess soft keyboard developed in the previous research. The MissLess keyboard assigns a hash code per each word for all words within its mobile dictionary. It decides recommendation words through three consecutive processes such as hash filtering, sorting based on spelling similarity, and finally recommendation based on frequency of use. Each process has some factors to have an impact on the recommendation success. We conducted experiments in an Android mobile device running the MissLess keyboard and measured performance of recommendation and run time overhead according to the impact factors. In this paper, we showed the experiment results.

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

Ⅰ. 서 론

소프트 키보드는 터치 스크린 상에 구현된 키보드로서, 하드웨어 키보드를 장착할 수 없는 모바일 단말기나 태블릿 장치에 주로 사용된다[1,2]. 그러나 이들 장치의 크기 한계로 인해 소프트 키보드가 작아서 인접한 키의 거리가 좁고, 키 타이핑 시 사용자에게 전달되는 접촉 피드백도 부족하므로, 키 입력 오류가 많이 발생한다[3].

이러한 소프트 키보드의 단점을 극복하기 위해 많은 연구가 진행되어 왔다. 계속되는 키 입력으로부터 다음 키를 예측하여 예측된 키의 너비를 동적으로 늘이거나[4], 터치 스크린 상의 키 부분에 그래픽 하이라이팅을 통해 사용자의 시각적인 관심을 유도하거나[5], 사용자의 일반적인 텍스트 입력 패턴을 분석하고 그에 따라 입력되는 키 거리가 최소가 되도록 키를 배치하는 방법 등이 연구되었다[6].

*정회원, 한성대학교 컴퓨터공학과 (교신저자)
**준회원, 한성대학교 컴퓨터공학과
***정회원, 한성대학교 정보통신공학과
접수일자 : 2014년 7월 9일, 수정완료 : 2014년 7월 29일
게재확정일자 : 2014년 8월 8일

Received: 9 July, 2014 / Revised: 29 July, 2014
Accepted: 8 August, 2014
*Corresponding Author: calafk@hansung.ac.kr
Dept. of Computer Engineering, Hansung University, Korea

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[7].

본 연구 팀은 선행 연구[7]를 통해, 사용자가 텍스트를 입력하는 동안 의도하는 한글 단어나 어절을 추론하여 추천하여, 한글 입력에 오타가 발생하는 경우 사용자가 빠르게 교정할 수 있게 하여 사용자의 키 입력을 도와주는 추천 키 입력 시스템을 제안하고 MissLess로 불리는 소프트 키보드를 구현하였다.

본 논문은 선행 연구를 통해 설계 구현된 MissLess 키보드의 추천 알고리즘을 보완하고, 추천 시스템이 가지는 성능 요소를 보다 세밀히 분석하고 다양한 요소들이 성능에 어떤 영향을 미치는지 평가 하고자 한다.

본 논문은 다음과 같이 구성된다. 2 장에서 연구 배경으로서 선행 연구 MissLess 키보드의 추천 시스템을 요약 기술하고 3 장에서는 개성된 추천 알고리즘과 성능 요소를 본석한다. 4 장에서는 성능 평가한 내용을 기술하며 5 장에서 결론을 맺는다.

II. MissLess 추천 시스템

1. 추천 시스템 구성

MissLess 추천 키보드에 구현된 MissLess 추천 알고리리즘은 사용자가 한글 텍스트를 입력하는 도중 계속적으로 사용자가 입력하리라고 예상하는 전체 워드를 추천한다. 현재 입력된 한글 음소나 글자의 다음 글자나 단어를 추천하는 것이 아니다. 예를 들어 사용자가 “사진”을 입력하려다 “서ㅈㄴ”을 입력한 경우 “사진”을 추천하여 사용자가 선택할 수 있게 한다.

그림 1은 어휘 사전을 만드는 과정을, 그림 2는 본 논문에서 제안한 추천 시스템의 구조를 보여준다.

인터넷에서 많이 사용하는 어휘를 추려내어 어휘 사전을 구성한다. 어휘 사전은 초기에 설정된 워드 사전에 사용 빈도수가 낮은 워드를 삭제하고 사용자가 새로 사용한 워드를 추가하여 관리한다. 그리고 각 워드별로 표 1의 한글 음소에 대한 코드 값을 참조하여 해시 정수 값이 계산되어 저장된다.

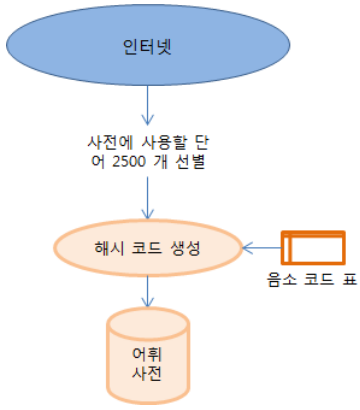


그림 1. 어휘 사전을 만드는 과정
Fig. 1. Process to construct the vocabulary dictionary

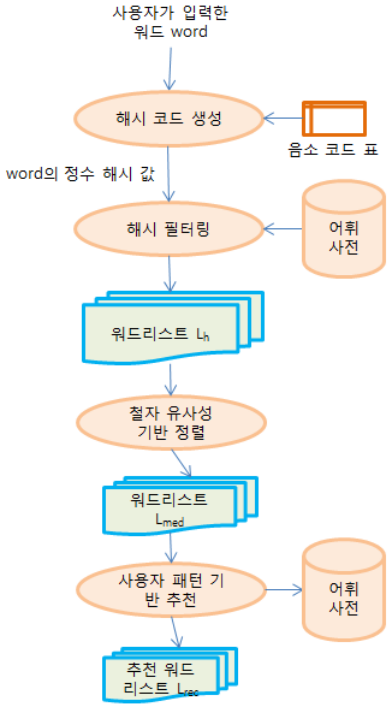


그림 2. 사용자 입력 키에 대한 워드 추천 프로세스
Fig. 2. Recommendation Process

2. 추천 알고리즘

MissLess 추천 알고리즘은 그림 2와 같이 4 개의 세부 알고리즘이 연속적으로 실행되도록 구성된다.

사용자가 입력한 한글 워드는 음소 코드 표를 참고하여 해시 함수에 의해 정수 값을 생성하고, 이 정수 값은 해시 필터링 과정을 통해 어휘 사전의 각 단어의 해시 코

드와 비교하여 일정 범위의 차 이내에 있는 워드들을 골라 워드 리스트 L_h를 구성한다. 그리고 L_h를 대상으로 철자의 유사성을 비교하여 가장 유사한 순서로 정렬하여 L_{med}를 구성한다. 이제 L_{med}에서 사용자의 사용 빈도수가 높은 워드를 몇 개 추천하여 L_{rec}를 구성한다. 이 리스트의 워드들이 사용자에게 보여지며, 사용자는 이 중에서 한 워드를 선택한다. 이 과정은 그림 3에서 보여진다.

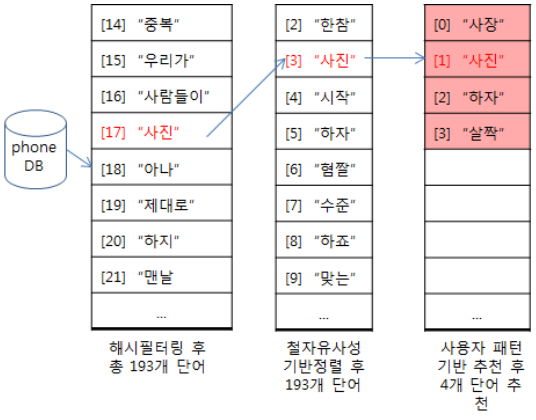


그림 3. 알고리즘의 실행 예
Fig. 3. A running sample of recommendation algorithm

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
				ㄷ	21
ㅌ	80000	ㅜ	4000	ㄹ	12
ㄸ	90000	ㅠ	4050	ㄹㄱ	83
		ㅐ	2100	ㄹㅁ	84
ㅂ	100000	ㅒ	2150	ㄹㅂ	85
ㅍ	110000	ㅔ	2600	ㄹㅅ	86
ㅃ	120000	ㅖ	2650	ㄹㅌ	87
ㅅ	130000	ㅚ	8500	ㄹㅍ	88
		ㅘ	8550	ㄹㅎ	89
ㅎ	140000	ㅙ	8650	ㅁ	62
ㅆ	150000	ㅝ	6050	ㅂ	31
ㅈ	160000	ㅟ	6000	ㅂㅅ	90
ㅊ	170000	ㅞ	6150	ㅅ	41
				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

Ⅲ. 추천 알고리즘과 성능 파라미터 분석

이 장에서는 추천 알고리즘의 각 단계를 분석하고 추천 성능과 관계된 변수들을 도출한다.

1. 해시 코드 생성과 코드 부여 알고리즘

가. 해시 코드 생성 방법

한글 워드 W는 식 (1)과 같이 n개의 글자(w_1, w_2, ...)로 구성되며 다시 각 글자 w_n은 e_{ni} 의 음소(초성, 중성, 종성)들로 구성되므로 다음과 같이 표현된다.

$$W = (w_1, w_2, \ldots, w_n) \quad (1)$$

$$w_n = (e_{n1}, e_{n2}, \ldots, e_{nm}) \quad (2)$$

해시 코드의 목적은 워드들의 유사성을 쉽게 분류하는데 있다. 본 논문에서는 해시 코드를 생성하기 위해 표 1과 같이 각 자음, 모음에 기본 코드를 부여하였다.

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

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

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

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

여기서 α(alpha)는 한 워드에서 동일한 글자들이 있을 때, 예를 들어 '이용'과 '용이'가 서로 다르게 평가되도록 설정하는 계수(Letter Weight)이다.

나. 한글 음소가 성능에 미치는 영향

이 단계에서, 추천 성능을 높이기 위해서는 사용자가 입력한 워드 W와 유사한 워드들의 해시 값의 차이가 가

능하면 작게 나도록 하고, 엉뚱한 워드의 해시 값이 사해지지 않도록 분리하는데 있다.

이를 위해 MissLess 키보드에서 한글 음소에 대한 기본 코드 값의 배치 전략은 다음과 같다.

첫째, 초성의 코드 값을 종성의 코드 값보다 크게 하여 종성의 유사성이 초성의 유사성에 영향을 미치지 못하게 한다. 예를 들어 '갖"과 '작'은 매우 다른 글자로 인식되게 한다.

둘째, 현재 천지인 타입의 키보드에 적용하고 있기 때문에 동일한 키보드에 중복된 키들은 다른 키에 설정된 글자보다 유사성을 높이기 위해 기본 코드 값의 차이를 작게 설정하였다. 표 1에서 보면, 예를 들어 ,'ㄱ', 'ㅋ', 'ㄲ' 에 연속된 기본 키 값을 배정하고 있다.

다. α (alpha) 값이 성능에 미치는 영향

또한 식 (4)에서 정의한 α(alpha)는 각 글자의 해시 값이 차이나도록 하는 요소이다. 예를 들어 '이용'의 해시 값을 계산할 때, '이'의 해시 값은 19350이고 '용'의 해시 값은 196611이다. 이 두 정수 값은 단순히 더하여 '이용'의 해시 값으로 하게 되면, '용이'의 해시 값과 같아지게 되어 '이용'와 '용이'는 같은 단어인 것으로 평가될 수 있다. 그러나 α(alpha) 값을 100으로 주어 식 (4)를 계산하면 '이용'은 19350x100 + 196611x50 = 29180550이고, '용이'는 196611x100 + 19350x50 = 29336100이 되어 서로 다른 워드로 구분된다.

본 논문은 성능 평가를 통해 α(alpha) 값에 따라 추천 성능이 어떻게 변하는지 평가하였다.

2. 해시 필터링

가. 해시 필터링 알고리즘

해시 필터링 알고리즘은 사용자가 키를 입력할 때마다 현재 입력된 텍스트 T에 대해 해시 코드 C_T 생성하고 식 (5)를 이용하여 이 값을 어휘 사전에 있는 모든 워드의 해시 값과 비교하여 차이가 일정 범위 내인 것들을 골라 워드 리스트 L_h를 생성한다.

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

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 기준으로 해시 코드의 범위를 지정하는 계수이다.

나. β 값이 성능에 미치는 영향

β가 클수록 사용자가 입력한 어휘 T와 모양이 다른 어휘를 L_h에 포함하게 될 가능성과 L_h의 크기가 커져 해시 필터링 처리 시간 및 뒤이어지는 알고리즘 실행 시간이 길어질 가능성이 높다. β 값은 추천 성공률과 추천에 걸리는 시간에 영향을 주는 요소이다.

3. 철자 유사성 기반 정렬

이 알고리즘은 사용자가 입력한 텍스트 T와 철자가 유사한 워드들을 사전에서 골라내는 과정으로서, 선행 논문[7]에서는 Modified Edit Distance(MED)로 명명하였다. 기존의 Levinstein 알고리즘[8]을 따라 사용자가 입력한 텍스트 T와 사전에 있는 워드 W와의 편집 거리를 계산하는 식을 ed(T, W) 라고 하고, 단어 W의 철자 개수를 Size(W)라고 할 때, T와 W의 편집 거리 med는 다음 식 (6)과 같이 계산한다.

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

T와 W가 같으면 ed(T,W)는 0이 되고, med(T,W)는 100이 된다. 철자 유사성 기반 정렬 결과 L_h 리스트는 med 값이 낮은 순서로 정렬되어 L_{med} 리스트로 변경된다.

기본적으로 ed(T,W)를 계산하는 과정은 많은 루프를 포함하여 많은 시간이 소요되므로, L_h 리스트의 크기를 줄여야 추천에 걸리는 시간을 단축시킬 수 있다. 해시 필터링의 과정을 통해 2500 개의 사전 어휘로부터 해시 값이 비슷한 어휘를 선별하여 L_h 리스트를 만드는 과정은 사전의 모든 어휘를 대상으로 med를 계산하는 오버헤드를 줄이기 위한 과정이다.

4. 사용자 패턴 기반 추천

가. 사용자 패턴 기반 추천 알고리즘

L_{med} 리스트의 워드에서 사용자가 자주 사용하는 워드를 우선적으로 추천하기 위해 동일한 med 값을 갖는 워드들을 그룹으로 묶고, 그룹 단위로 med 값이 낮아지는 순서로 정렬한다.

그리고 사용자가에게 추천할 워드의 개수가 N이라고 할 때, 정렬된 L_{med}에서 첫 번째 그룹을 다시 사용자의 참

조 카운트 순으로 정렬하고 이 중에서 참조 카운트가 높은 N개를 추천한다. 만일 첫 번째 그룹의 워드 개수가 N보다 작으면 이들을 모두 추천하고, 다음 그룹으로 이동하면서 동일한 방법으로 N개를 추천할 때까지 계속된다.

나. 추천 워드 개수 N 값이 성능에 미치는 영향

N 값은 추천에 소요되는 시간에 전혀 영향을 주지 않는다. 다만, N 개 안에 사용자가 원하는 어휘가 있을 때 추천 성공이라고 판단한다면, N 값이 클수록 추천 성공률이 높아지는 것은 당연하다. 모바일 단말기의 특성상 추천 창의 크기가 작기 때문에 4개보다 많은 어휘를 추천하는 것은 다소 무리가 있기 때문에, 본 논문에서는 N의 최댓값을 4로 하고, N(1~4) 값에 따라 추천 성공률이 어떻게 변하는지 관찰하였다.

V. 성능 평가

1. 성능 파라미터 도출

추천 과정에 적용되는 각 알고리즘에서 성능과 관련된 요소를 분석한 결과, 추천 성공률과 추천에 걸리는 시간에 영향을 주는 다음 요소들을 도출하였다.

- 한글 음소의 기본 코드 값 배정
- 해시 코드 생성에 사용되는 α(alpha) 값
- 해시 윈도우의 크기를 결정하는 β(beta) 값
- 추천 창의 크기 N

2. 성능 평가 지수

본 연구의 성능 평가 지수는 추천 성공률과 추천에 걸리는 시간으로 한다. 사용자가 입력하고자 하는 어휘가 추천 창에 존재하면 추천 성공으로 보며, 추천 성공률은 다음과 같이 정의하며, 추천 성공률 S는 다음 함수이다.

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

$$S = f(\text{한글음소기본코드}, \alpha, \beta, N) \quad (8)$$

추천에 걸리는 시간 T_{rec}는 사용자가 어휘를 입력한 직후 N개의 추천 리스트를 완성하는데 걸리는 알고리즘 시간으로 한다.

3. 실험 방법 및 추천 성공 판단

실험 방법은 선행 연구[7]의 방법을 그대로 따랐다. 사용자의 행위를 미리 테스트 셋으로 만들어 두고, 알고리즘이 구현된 키보드 소프트웨어에 배치(batch) 방식으로 전달하여, 성능을 테스트하였다. 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 선행 연구[7]에서 구성한 3개의 테스트 셋, TS1, TS2, TS3 을 이용한다.

가. 테스트 셋 1(TS1)

'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.

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	그랩
그래	그랩	오늘	오른	진짜	친짜
교수	겨수	대박	대갑	잘자	잘ㅈ거

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.

4. 실험 및 결과

가. 테스트 셋에 따른 추천 성공률

이 절에서는 전체 실험 결과를 이해하도록 하기 위해 간단히 선행 논문의 실험 결과[7]를 요약한다. 그림 4와

같이 테스트 셋에 따라 추천 성공률이 다르게 나타났으며, TS2의 추천 성공률이 97%에 이르렀다. 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.

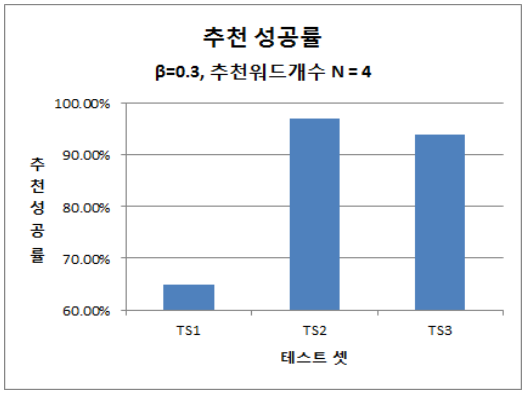


그림 4. β =0.3일 때, 3 개의 테스트 셋의 추천 성공률
Fig. 4. Success ratios of recommendation for three test sets for β =0.3

나. 한글 음소의 코드 부여 알고리즘에 따른 추천 성공률

해시 코드를 만들기 위한 각 한글 음소의 코드 값을 정하는 방법은 추천에 지대한 영향을 미친다. 유사 단어가 비슷한 해시 값이 나오도록 유도하고 다른 단어는 해시 값에 많은 차이가 나도록 하는 것이 MissLess 키보드의 코드 할당 전략이다.

MissLess 키보드의 코드 부여 기법의 성능의 우수성을 보이기 위해 다른 코드 부여 방법과 비교 평가하였다. 이를 위해 총 MissLess 키보드를 포함하여 총 6가지의 코드 부여 방법을 비교 평가하였다. 중성은 표 1의 코드 체계를 그대로 사용하였다.

방법 1. 종성 코드를 초성의 평균 1.5배로 부여할 때
방법 2. 종성 코드를 초성의 평균 2배로 부여할 때
방법 3. 초성 값을 랜덤으로 부여하고, 중성과 종성은 표 1의 코드를 사용할 때
방법 4. 초성과 종성 모두 랜덤으로 주었을 때
방법 5. 초성과 종성 같은 값으로 주었을 때
방법 6. MissLess 키보드. 종성 코드를 초성의 5000~10000배 값으로 부여

α=100, β=0.3, N=4로 설정하고 실험한 결과, 6가지 방법론의 추천 성공률은 그림 5와 같이 평가되었다. 초성과 종성의 코드 차이를 크게 하고, 한 키에 중복된 음소들은 매우 유사한 코드를 부여하는 MissLess 키보드의 성능이 월등이 좋은 것으로 평가되었다.

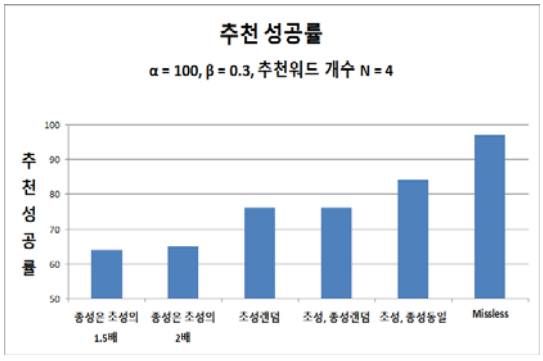


그림 5. 한글 음소의 코드 부여 방식에 따른 추천 성공률
Fig. 5. Success ratios of recommendation according to code values of hangul phonemes

다. α 값에 따른 추천 성공률

그림 6은 α 값을 바꾸면서 3 개의 테스트 셋에 대해 추천 성공률을 실험한 결과를 보여준다. 이 실험의 결과, α 값에 따라 추천 성공률이 거의 달라지지 않는 것으로 결론 내린다. 그 이유를 분석한 결과 철자 유사성 정렬의 성능이 매우 좋기 때문에 α값이 잘못 설정된 경우에도 유사성이 떨어지는 단어를 배제하는 효과가 크게 발생하기 때문이었다.

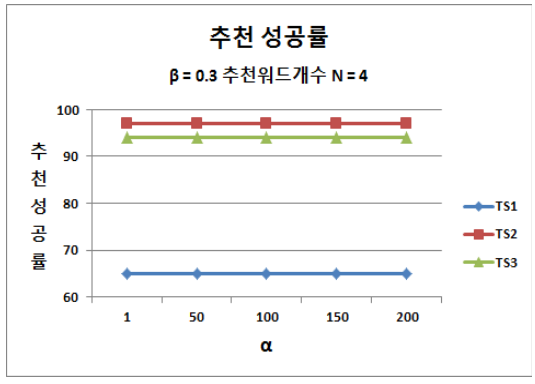
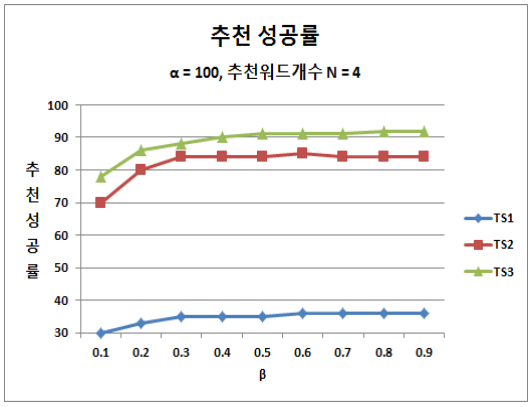


그림 6. α 값에 따른 추천 성공률
Fig. 6. Success ratios of recommendation according to α

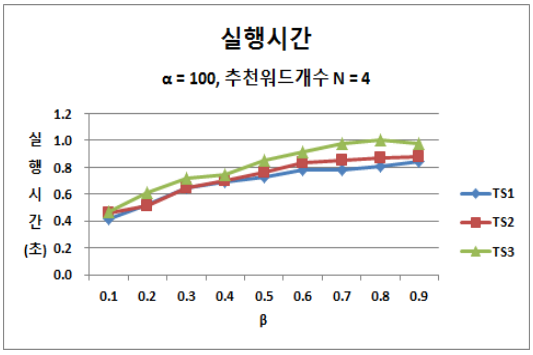
라. β 값에 따른 추천 성공률

그림 7은 β 값을 바꾸면서 3 개의 테스트 셋에 대해 추천 성공률을 실험한 결과를 보여준다. β 값이 0.3을 지나 거의 0.4를 넘어서면 추천 성공률이 거의 높아지지 않고 있는 것을 볼 수 있다.

β가 클수록 L_h의 크기가 커져 사용자가 입력하고자 한 목적 어휘가 L_h에 들어갈 확률이 높아지기 때문이다. 그러나 그림 7(b)에서 볼 수 있듯이 L_h가 커진다고 해서 추천 성공률은 더 이상 높아지지 않고, 실행 시간은 β에 비례하여 길어진다.



(a) β값에 따른 추천 성공률



(a) β값에 따른 추천 소요 시간

그림 7. β 값에 따른 추천 성공률
Fig. 7. Success ratios of recommendation according to β

마. N 값에 따른 추천 성공률

추천 단어가 N 개에 포함되면 추천 성공으로 판단하기 때문에, 당연히 N 값이 크면 추천 성공률이 올라갈 것으로 예측된다. 다만 이 실험의 목적은 N의 크기 차이에 따라 어느 정도 추천 성공률이 차이 나는지 보기 위함이다.

그림 8은 N 값에 따른 추천 성공률을 실험한 결과를 보여준다. TS1과 TS2의 경우 N이 1인 경우에도 거의 84%가 넘는 수준으로 평가되었다. N의 크기는 추천 실행 시간에 영향을 주는 요소는 아니다.

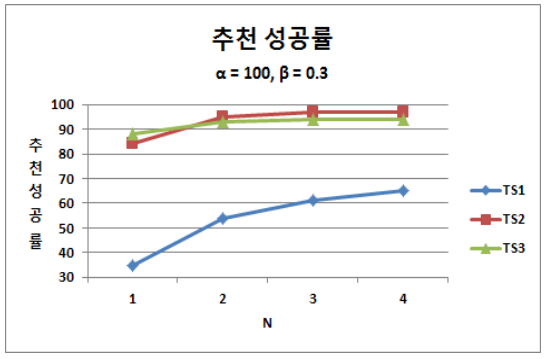


그림 8. N 값에 따른 추천 성공률
Fig. 8. Success ratios of recommendation according to N

바. 추천 알고리즘의 시간 요소 분석

추천 알고리즘의 각 단계별로 소요되는 시간을 측정하였다. 그림 9는 측정된 결과를 보여준다. '철자유사성 기반 정렬' 알고리즘이 가장 많은 소요 시 간을 차지하였다. 입력된 어휘의 해시코드를 생성하는데 소요되는 '해시코드생성' 시간은 매우 작아 그림 9의 그래프에 나타나지 않는다.

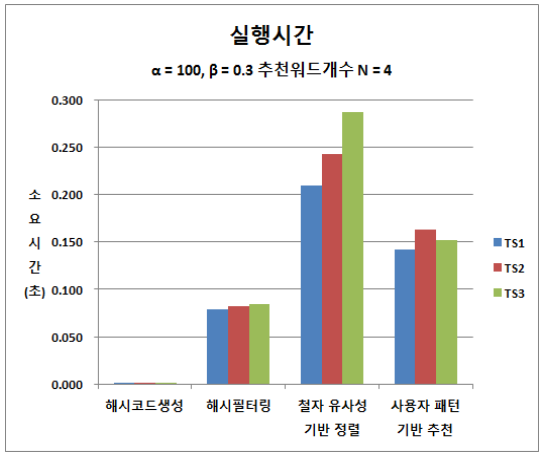


그림 9. 추천 알고리즘 실행 시간
Fig. 9. Run time of recommendation algorithm

VI. 결론

MissLess 키보드는 선행 연구[7]에서 완성한 추천 키보드로서, 해시필터링, 철자 유사성 기반 정렬, 사용자 패턴 기반 추천의 3 알고리즘 단계를 거쳐 단어를 추천한다.

본 논문에서는 성능 요소로 한글 음소 코드 값 부여 전략, α, β, N의 5가지로 분석하고, 이들이 추천 성능에 어떤 영향을 미치는지 실험하였다. 실험 결과 MissLess의 한글 음소 코드 부여 전략은 타 전략에 비해 추천 성능 향상에 현저한 영향을 미치는 것으로 나타났다.

그러나 α 값의 변화는 추천 성능에 크게 영향을 미치지 않았는데 그 원인은 철자 유사성 기반 정렬 과정에서 유사성이 떨어지는 어휘가 잘 걸러지기 때문이었다.

β 값은 0.3과 0.4 사이에서 추천 성능이 포화 상태에 다다랐으며, 더 이상 성능이 올라가지 않았다. 한편 β가 클수록 추천 시간이 오래 걸리는 단점이 있어 0.3으로 설정하는 것이 추천 성능과 추천 시간 면에서 가장 유리하다는 결론을 얻었다.

N 값이 크면 당연히 추천 성능이 올라가지만, N=4 일 때, N=1일 때 보다 4배의 성능이 올라가지는 않았으며, 조금 올라가는 수준으로 평가되었다. 필요에 따라 N을 무리하게 크게 잡지 않아도 필요한 추천 성능은 얻을 수 있다.

References

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

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

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] Kitae Hwang, Jae Moon Lee, “Preliminary Study on Soft Keyboard with Recommendation for Mobile Device”, The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), vol. 13, no. 6, pp.137-145, 2013.

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

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>

김 태 완(학생회원)

- 한성대학교 컴퓨터공학과 학사 과정

<주관심분야 : 모바일 시스템>

조 혜 경(정회원)

• 서울대학교 제어계측공학과 학사
서울대학교 제어계측공학과 석사
서울대학교 제어계측공학과 박사
• 경력
Carnegie Mellon Univ. 방문연구원
<주관심분야 : HRI/HCI for Learning>

※ 본 연구는 한성대학교 교내 학술 연구비를 지원받았음