Journal of the Korean Society of Marine Engineering, Vol. 39, No. 6 pp. 656~661, 2015
J. Korean Soc. of Marine Engineering (JKOSME)
http://dx.doi.org/10.5916/jkosme.2015.39.6.656

ISSN 2234-7925 (Print)
ISSN 2234-8352 (Online)
Original Paper

한국어 단어 자동완성 시스템의 성능 분석 및 새로운 평가 방법

이성욱†

(Received April 9, 2015 ; Revised June 17, 2015 ; Accepted June 30, 2015)

Performance Analysis of a Korean Word Autocomplete System and New Evaluation Metrics

Songwook Lee†

요약: 본 연구의 목적은 스마트폰이나 태블릿 PC와 같이 문자 입력이 수월하지 않은 모바일 기기에서 사용자로 하여금 최소한의 키입력을 통해 최대한 빠르고 정확히 원하는 단어를 얻을 수 있도록 도와주는 단어 자동완성 시스템의 성능을 평가하는 것이다. 우리는 트위터에서 대량의 데이터를 수집하였으며, 수집된 데이터의 사용빈도에 따라 유니그램(unigram) 사전과 바이그램(bigram) 사전을 각각 구축하였다. 구축된 사전을 사용한 단어 자동완성 시스템의 성능을 평가하였으며 기존의 평가방법보다 단어 자동완성 기능의 특성을 잘 반영한 키입력 수익률과 복원율을 새로운 평가 방법으로 제안하였다.

주제어: 단어자동완성, n그램, 키입력 수익률, 복원율

Abstract: The goal of this paper is to analyze the performance of a word autocomplete system for mobile devices such as smartphones, tablets, and PCs. The proposed system automatically completes a partially typed string into a full word, reducing the time and effort required by a user to enter text on these devices. We collect a large amount of data from Twitter and develop both unigram and bigram dictionaries based on the frequency of words. Using these dictionaries, we analyze the performance of the word autocomplete system and devise a keystroke profit rate and recovery rate as new evaluation metrics that better describe the characteristics of the word autocomplete problem compared to previous measures such as the mean reciprocal rank or recall.

Keywords: Autocomplete, Ngram, Keystroke profit rate, Recovery rate

1. 서 론

단어 자동완성 기능은 사용자가 어떤 단어를 입력하고 있을 때 그 단어의 나머지 부분이나 문장이 자동으로 입력되는 기능이다. 단어 자동완성 기능은 웹 브라우져의 URL 입력, 이메일 클라이언트의 주소 입력, 검색엔진의 질의어 추천, 소스코드 편집기의 코드추천, 스마트폰의 텍스트 입력 등에 광범위하게 사용되고 있는 기능이다. 이러한 단어 자동완성 기능은 해상에서 위급한 상황이 닥쳤을 때나 신속한 정보의 전달이 필요할 때 유용하게 사용될 것이다.

본 연구의 목적은 모바일 환경에서 주로 사용하는 단어 자동완성 시스템의 성능을 평가하는 것이다. 성능 평가에 사용된 시스템은 A사의 상용 시스템으로서 모바일기기에 실제 사용되고 있는 시스템이다. 시스템에서 사용하는 어휘사전의 크기와 종류에 따라 성능에 차이가 있을 수 있다. 우리는 모바일기기로부터 입력된 문장을 수집하기 위하여 대표적인 모바일 서비스 중 하나인 트위터로부터 대용량의 말뭉치를 수집하여 어휘사전을 구축한다. 언어모델의 사용유무에 따른 성능을 파악하기 위하여 유니그램(unigram) 어휘사전과 바이그램(bigram) 어휘사전을 각각 구축한다.

평가집합은 단어 자동완성 기능의 성능을 평가하며 향후 다른 시스템과의 성능비교 및 성능향상을 위한 지표로써 사용될 수 있다. 기본적으로 평가집합은 사용자가 입력한 키입력(keystroke)에 대한 완성단어의 쌍으로 구성된다.

따라서 본 연구의 중점 사항은 어휘사전과 평가집합을 구축하여 단어 자동완성 시스템의 성능을 평가하는 것이며, 단어 자동완성 기능의 성능평가를 위한 적절한 평가지표를 제시한다.

2. 관련 연구

단어 자동완성 기능은 트라이 자료구조**[1]**와 같은 이진

† Songwook Lee (ORCID: http://orcid.org/0000-0002-6224-4241): Department of Computer Science and Information Engineering, Korea National University of Transportation, 157, Cheoldoparkmulkwanro, Uiwang-si, Gyeonggido, 437-763, Korea, E-mail: leesw@ut.ac.kr, Tel: 070-8855-1686

탐색트리를 이용하거나 역색인 알고리즘을 이용하여 구현할 수 있는데 트라이 자료구조는 자연어처리의 형태소 사전 등에 광범위하게 사용된다[2]. 트라이 자료구조에 기반한 시스템은 먼저 어휘사전에 있는 모든 어휘로 구성된 탐색트리를 만든다. 그런 후에 사용자의 키입력으로부터 생성 가능한 모든 어휘를 빠르게 탐색하고 어휘 가중치에 따라 추천 단어열을 만들어 낸다. 트라이 자료구조는 탐색 알고리즘의 복잡도가 낮고, 트리를 만드는 데 걸리는 시간이 짧은 특성이 있다. 역색인 알고리즘에 기반한 시스템은 어휘사전의 모든 어휘의 접두어(prefix)가 각각의 완성 어휘집합을 색인하도록 만든 것이다.

이러한 트라이 자료구조나 역색인 알고리즘의 공간 효율과 탐색 속도를 향상시키기 위한 연구가 많이 이뤄져 왔다. [3]은 적응형(adaptive) 알고리즘을 사용하여 캐쉬 효율을 높인 트라이 탐색 방법을 제안하였다. 트라이의 루트 노드에 가까운 노드는 먼 노드에 비해 자식노드의 개수가 더 많은 특성을 가지고 있으므로 자료 구조를 다르게 설계하고 컴퓨터 메모리에 잘 맞도록 자료구조를 변형하였다. [4]는 대용량의 키집합을 위한 트라이 구조가 차지하는 공간 비용의 효율을 개선하기 위해 키의 전방과 후방을 각각 압축하여 두 개의 트라이를 구성하는 방법을 제안하였다. [5]은 한 개의 단어가 아니라 구문 단위의 자동완성 방법을 제안하였는데 어절 단위의 접미사 트리를 구성함으로써 n그램 언어모델을 구현한 방법으로 볼 수 있다. 질의어 자동완성 도메인에서는 사용 이력을 반영한 방법이 제안되었다. [6]는 시소러스에 존재하는 개념과 사용자의 문맥의 연관성을 규칙으로 정의하여 질의어 자동완성을 하였다. [7]은 웹검색에 사용자의 검색기록(log)과 입력질의어 사이의 유사도를 다양한 방법으로 측정하여 가장 근접한 질의어를 추천하였다. [8][9]는 역색인을 위한 자료구조를 개선하여 더 적은 메모리를 차지하면서 더 빠른 탐색이 가능한 새로운 자료구조를 제안함으로써 단어 자동완성의 효율성을 높였다. [10]은 질의어 자동완성을 위해 문맥을 이용하는데 먼저 문맥 단어가 존재하는 문서 집합으로 탐색 범위를 좁힌 후, 사용자가 입력한 접두어가 존재하는 단어들의 가중치와 문서의 가중치를 이용하여 질의어를 추천하였다. [11]은 트라이가 이전 단어의 문자도 저장하고, 현재 단어를 완성하면서 다음 단어도 예측하도록 트라이를 확장하였다. 또한 최근에 사용한 단어들의 정보를 트라이에 저장하여 사용자의 사용이력을 반영하였다.

3. 트라이를 이용한 단어 자동완성 알고리즘

평가에 사용한 시스템의 단어 자동완성 알고리즘은 트라이 사전에 기반한 알고리즘이다. **Figure 1**은 평가 시스템의 알고리즘을 나타낸 것이다. 먼저 사용자의 입력자소에 대해 트라이의 루트노드부터 순서대로 탐색을 하게 된다. 만약 트라이에 노드가 존재한다면, 사용자의 입력자소를 접두어로 갖는 단어가 사전에 존재하는 것이므로 현재 노드의 모든 자손노드에 존재하는 단어들이 후보단어가 된다. 만약 사용자의 키입력에 대한 노드가 트라이에 존재하지 않는다면, 사용자의 키입력과 가장 유사한 노드를 편집거리[12]를 이용하여 탐색한다. 이 유사한 노드들의 자손노드들에 존재하는 단어들이 후보단어가 된다. 후보단어가 결정되면 각 후보단어의 우선순위는 각 후보단어의 가중치로 결정하는데, 단어 발생확률의 로가리즘(logarithm)을 가중치로 사용한다.

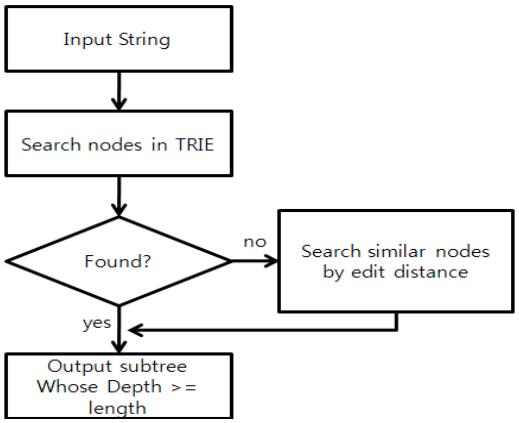


Figure 1: Autocomplete algorithm with a Trie dictionary

Figure 2와 같은 트라이 사전에 대해 단어 자동완성 알고리즘을 적용해 보자. 트라이 사전에서 사전 탐색은 루트노드에서 시작하여 입력 자소에 해당하는 자식노드들을 따라가면서 이뤄진다.

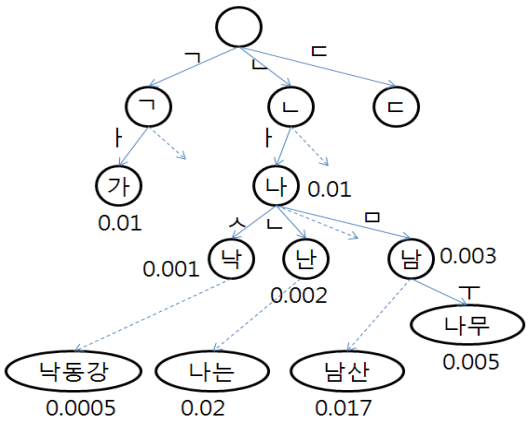


Figure 2: An example of a Trie dictionary

만약 사용자가 “ㄴ ㅏ ㅁ”을 입력한다면 트라이에서 해당 자소들을 루트 노드로부터 순차적 탐색하여 ‘남’에 해당하는 노드를 탐색한다. 이 때, 후보단어들은 ‘남’의 자손노드들이 된다. 그림 2의 예제에서는 ‘남산’과 ‘나무’가 후보 단어들이 되며 그 순위는 각 단어의 가중치가 결정한다. 만약 사용자가 “ㄴ ㅏ ㅇ”이라고 입력한다면 트라이에서 입력자

소에 해당하는 노드의 탐색이 실패하게 된다. 입력자소의 탐색이 실패하면 사용자 입력자소와 가장 유사한 노드를 탐색한다. 이 때, 입력자소와 비교하여 편집거리가 가장 적은 노드가 가장 유사한 노드가 된다. 그림 2에서는 '나', '낙', '난', '남' 등이 입력자소에 대해 가장 적은 편집거리를 갖는 노드들이다. 입력자소의 탐색이 성공했을 때와 마찬가지로 이들 노드들의 자손노드가 단어후보가 된다. 즉 '낙동강', '나는', '남산', '나무' 등이 단어후보들이 되며 이들 단어의 가중치로 순위를 결정한다.

4. 사전 및 평가 집합 구축

4.1 사전 구축

n그램 언어 모델을 위한 기본적인 사전의 항목은 트위터에서 수집된 약 60GB의 데이터에서 추출한 엔트리와 해당 엔트리의 빈도수로 구성된다. 단, 다음의 엔트리들에 대해서는 반자동으로 사전에서 제외한다.

- 비속어 및 저속표현
- 완성되지 않은 자/모음을 갖고 있는 엔트리

기본 사전 이외에 바이그램 사전도 구축하는데 트위터 데이터에서 추출한 고빈도 바이그램에 대해 동일한 과정을 거쳐 사전에 추가한다.

4.2 평가집합 구축

평가집합은 단어 자동완성 기능의 성능을 평가하며 동시에 이 기능의 성능향상을 위한 지표 및 타시스템과의 성능비교에 사용될 수 있다. 기본적으로 평가집합은 사용자가 입력한 키입력에 대해 완성된 어절로 표시한다. 평가집합은 문맥 정보가 있는 집합과 문맥 정보가 없는 집합으로 구성하였다. 트위터 말뭉치에서 무작위로 추출한 9,943개의 어절로 구성되어 있으며 가능한 모든 키입력의 개수는 53,785개이다.

평가집합의 형식은 다음과 같이 기술한다.

{ C1 키입력 } → { 완성어절 }

심볼 '→' 왼편에는 사용자의 키입력정보를, 오른편에는 키입력에 대해 추정된 단어이다. 키입력 정보에서 앞에 있는 C1은 사용자의 키입력이 입력되기 이전에 이미 입력된 바로 앞 어절을 의미한다. 키입력은 N개의 자소로 구성된 어절에서 1~(N-1)개의 자소로 구성된 바이그램 및 유니그램 입력을 생성한다.

다음은 바이그램 "위치추적어플 진돗개"와 유니그램 "같습니다"에서 생성 가능한 평가집합 예제들이다.

위치추적어플 ㅈ → 진돗개
위치추적어플 ㅈㅣ → 진돗개
위치추적어플 ㅈㅣㄴ → 진돗개
위치추적어플 ㅈㅣㄴㄷ → 진돗개
위치추적어플 ㅈㅣㄴㄷㅗ → 진돗개
위치추적어플 ㅈㅣㄴㄷㅗㅅ → 진돗개
위치추적어플 ㅈㅣㄴㄷㅗㅅㄱ → 진돗개

ㄱ → 같습니다
ㄱㅏ → 같습니다
ㄱㅏㅌ → 같습니다
ㄱㅏㅌㅅ → 같습니다
ㄱㅏㅌㅅㅡ → 같습니다
ㄱㅏㅌㅅㅡㅂ → 같습니다
ㄱㅏㅌㅅㅡㅂㄴ → 같습니다
ㄱㅏㅌㅅㅡㅂㄴㅣ → 같습니다
ㄱㅏㅌㅅㅡㅂㄴㅣㄷ → 같습니다

4.3 오탈자 포함 평가 집합 구축

오탈자 평가집합은 4.2절에서 구축한 평가집합 중 아주 짧거나 아주 긴 키입력을 제외하여 구축하였다. N개의 자소로 구성된 어절에서 키입력이 4개 이상 $\frac{2}{3}N$개 미만의 자소 (*4~floor(⅔N-1)*)인 경우로만 구성하였으며 총 26,489개이다. 이들 정상 키입력에 대해서 무작위로 오탈자를 발생시켰다. 오탈자의 종류는 삽입(insert), 삭제(delete), 치환(substitute), 자리바꿈(transpose)이다. 삽입과 치환의 오탈자는 쿼티 자판에서 정상 자소에 인접한 자소로만 제한하였다.

오탈자 평가집합의 키입력은 오탈자 정보를 포함하도록 다음과 같이 기술한다.

{발생횟수 정상자소 오류자소 (종류, 발생위치, 정상자소)+}

다음은 '이벤트'와 '있습니다'의 키입력에 대한 오탈자 발생 키입력의 예이다.

- 이벤트

0 ㅇ	ㅇ	
0 ㅇㅣ	ㅇㅣ	
0 ㅇㅣㅂ	ㅇㅣㅂ	
1 ㅇㅣㅂㅖ	ㅇㅣㅖ	(deletion, 2, ㅂ)
1 ㅇㅣㅂㅖㄴ	ㅇㅂㅣㅖㄴ	(transpose, 1, ㅣ)

- 있습니다

0 ㅇ	ㅇ	
0 ㅇㅣ	ㅇㅣ	
0 ㅇㅣㅆ	ㅇㅣㅆ	
1 ㅇㅣㅆㅅ	ㅇㅣㅅㅆ	(transpose, 2, ㅆ)
1 ㅇㅣㅆㅅㅡ	ㅇㅣㅆㅅ	(deletion, 4, ㅡ)
1 ㅇㅣㅆㅅㅡㅂ	ㅇㅣㅆㅅㅣㅂ	(substitution, 4, ㅡ)
1 ㅇㅣㅆㅅㅡㅂㄴ	ㅇㅣㅆㅅㅡㅂㅊㄴ	(insertion, 6, ㅊ)

5. 실험

5.1 성능 평가

다음 **Table 1**은 문맥 정보가 없는 평가집합에 대해 유니그램 사전의 종류에 따른 단어 자동완성 시스템의 MRR(Mean Reciprocal Rank)과 재현율을 나타낸다. MRR

은 올바른 후보단어가 나타났을 때의 점수를 후보단어의 순위로 나눈 값의 평균이며, 일반적으로 순위의 역수의 평균값으로 계산할 수 있다. 재현율과의 비교를 위해 백분율로 나타내었다. 평가 대상 시스템은 사용자의 키입력에 대해 최대 15개의 완성후보를 출력한다. 이 때 MRR과 재현율을 측정하였다.

Table 1: MRR and recall of the autocomplete system for unigram dictionaries

dictionary	MRR(%)	recall(%)
100k unigram	31.5	56.8
250k unigram	36.8	62.0
500k unigram	37.9	64.2

Table 1에서와 같이 유니그램 사전의 크기가 커질수록 시스템의 성능이 향상되었으나 사전의 크기의 증가량에 비해 성능향상의 폭이 그리 크지 않았다.

다음 **Table 2**는 문맥 정보가 있는 평가 집합에 대해 바이그램 사전의 종류에 따른 단어 자동완성 시스템의 평균 정확률과 재현율을 나타낸다.

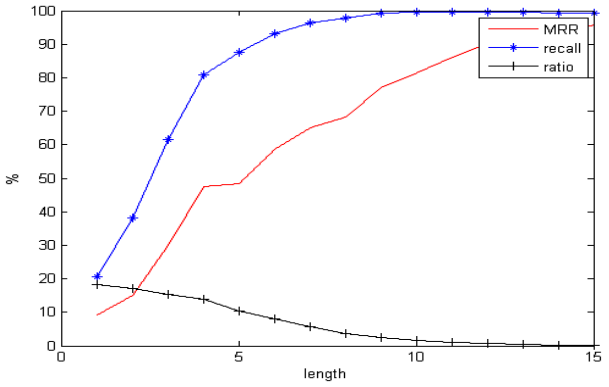
Table 2: MRR and recall of the autocomplete system for bigram dictionaries

dictionary	MRR(%)	recall(%)
100k bigram	36.8	60.3
250k bigram	50.8	72.7
500k bigram	51.9	74.3
2M bigram	**54.5**	**77.2**

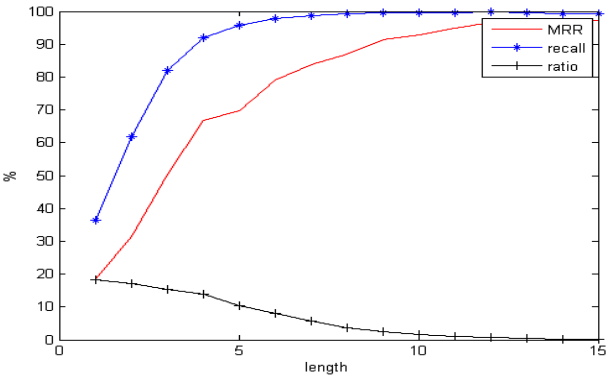
Table 2에서와 같이 문맥 정보를 사용하였을 때 **Table 1**의 문맥 정보를 사용하지 않았을 때보다 더 나은 성능을 보였다. 바이그램 사전의 경우에도 사전의 엔트리 개수를 늘여 나갈수록 MRR과 재현율이 증가하는 것을 알 수 있다. 그러나 모바일기기의 저장 공간 비용에 대한 효율을 고려할 때, 25만 엔트리의 바이그램 사전을 사용하였을 때의 성능과 2백만 엔트리의 바이그램 사전을 사용하였을 때의 성능 차이는 비용 대비 효율이 낮다고 볼 수 있으며 대용량의 바이그램 사전보다 사용자 이력을 문맥으로 반영하는 것이 비용 대비 시스템 효율을 높일 수 있을 것이다.

5.2 자소 길이에 따른 성능변화

다음 **Figure 3은** 입력자소 길이에 따른 시스템 성능변화를 각각 나타낸 그래프이다. 500k 유니그램 사전(a)과 2M 바이그램 사전(b)을 각각 사용했을 때의 성능변화이다. 입력자소의 길이가 6이상인 경우에 MRR은 약 50%이상이고 재현율도 약 90%이상을 상회할 정도로 시스템 성능이 급격히 좋아지는 것을 볼 수 있다. 입력자소의 길이가 6이상이라면 적어도 3음절 이상의 단어를 추정하는 것을 의미한다. 즉, 3음절 이상의 단어에 대해 사용자가 6개 이상의 키를 입력할 때, 시스템이 추정한 단어가 2순위(MRR이 50%이상인 경우) 이내로 90% 이상의 재현율로 나타나는 것을 의미한다. 그러나 전체 평가집합에서 각 입력자소 길이의 비율을 보면 평가집합의 과반수 이상이 6개 이하의 길이를 갖는 입력자소로 구성되어 있으며 이들이 시스템 성능 저하의 주요 원인이라 할 수 있다.



a) 500k unigram



a) 2M bigram

Figure 3: Performance with the length of an input strings

다음 **Table 3은** 500k 유니그램 사전을 사용하였을 때 자소길이별 성능을 나타낸 것이다.

Table 3: MRR and recall of the system with the length of input strings for the 500k unigram dictionary

input length t	avg(t).	MRR (%)	recall (%)	rate (%)
1 ⇐ t < 1/3N	1.5	1.9	8.1	15.7
1/3N ⇐ t < 2/3N	3.2	20.2	51.9	38.8
2/3N ⇐ t	5.8	**65.4**	**94.2**	45.5
Overall	4.1	37.9	64.2	100

Table 3에서 보는 바와 같이 입력자소가 목적단어 자소길이 N에 근접할수록 시스템의 성능이 더 좋아지는 것으로 나타난다. 그런데 MRR과 재현율만으로는 단어추정 시스템의 정확한 성능을 직관적으로 파악하기에 어려움이 있다. 예를 들어 '진돗개'의 경우, 키입력이 5개 이상이라면 'ㅈㅣ

ㄴㄷㅗ', 'ㅈㅣㄴㄷㅗㅅ', 'ㅈㅣㄴㄷㅗㅅㄱ' 등이 키입력인데 입력자소의 2/3 이상이 이미 입력이 되었기 때문에 후보 단어의 개수가 급격히 감소하게 된다. 이 경우 시스템 성능은 높아지지만 사용자 측면에서는 이미 2/3이상의 자소를 입력하였기 때문에 시스템의 추천 단어를 선택하기보다는 사용자가 스스로 남은 어절을 완성하는 편이 더 선호된다. 따라서 실제로 입력자소가 목적단어의 자소길이에 근접할 때까지 시스템이 올바른 단어를 추정하지 못한다면 시스템 성능이 좋지 않다고 할 수 있다. 따라서 이 특성을 평가하기 위한 다른 성능 평가 방법이 필요하다.

우리는 입력자소의 길이에 대한 키입력 수익률(이하 수익률)과 목적단어의 자소 길이에 대한 복원율을 단어 추정 시스템의 새로운 성능평가 방법으로 제안한다. 수익률(profit rate)은 입력자소 길이에 대한 복원자소 길이의 비율이다. 시스템이 올바른 단어를 추정했을 때 다음 **식 (1)**과 **식 (2)**를 이용하여 복원자소 길이와 수익률을 각각 계산할 수 있다. 복원율은 채널 압축률과 유사한데 목적 자소 길이에 대한 복원 자소의 길이의 비율이다. 복원율은 **식 (3)**으로 계산할 수 있으며 직관적으로 시스템이 목적단어의 몇 퍼센트를 복원하는지를 나타낸다. *InputL*은 입력자소길이, *WordL*은 목적단어의 자소길이, *RecoveredL*은 복원한 자소길이를 각각 나타낸다. *Cost*는 후보단어의 선택을 위한 키입력 횟수인데 *Cost*는 모바일 기기의 화면의 크기를 감안하여 결정한다. 평가 시스템은 3순위까지는 한 번의 키입력으로 목적단어를 선택할 수 있고 3순위 이후의 순위의 선택을 위해서는 리스트를 보기 위한 키입력이 추가되어 결국 2번의 키입력이 추가로 필요하다.

$$RecoveredL = WordL - (InputL + Cost) \tag{1}$$

$$profit(\%) = \frac{RecoveredL}{InputL} \times 100 \tag{2}$$

$$recovery(\%) = \frac{RecoverdL}{WordL} \times 100 \tag{3}$$

수익률로 단어자동완성 시스템을 평가하면 입력자소가 목적 자소의 길이에 가까울수록 시스템의 유용성이 떨어지는 특성을 잘 반영할 수 있다.

다음 **Table 4**는 입력자소의 길이에 따른 평균 수익률과 평균 복원율을 MRR과 함께 나타낸 표이다. 수익률로 시스템을 평가하면, MRR이나 재현율과 달리 입력자소의 길이가 목적 자소 길이에 가까워질수록 시스템 성능이 더 저하되는 것을 볼 수 있다. 짧은 자소 입력에서 수익률이 약 28.1%라는 것은 직관적으로 설명하면, 사용자가 한 개의 자소를 입력했을 때 시스템을 사용하여 얻을 수 있는 자소의 이득이 약 0.281개라는 것을 의미한다.

Table 4: MRR, keystroke profit rate, and recovery rate of the system with the length of input strings

input length	MRR (%)	profit (%)	recovery (%)
1 ⇐ t < 1/3N	1.9	28.1	5.0
1/3N ⇐ t < 2/3N	20.2	50.2	19.1
2/3N ⇐ t	65.4	13.2	8.6
Overall	37.9	29.9	12.1

Figure 4는 입력자소의 길이에 따른 시스템 성능 변화를 살펴본 것이다. MRR이나 재현율과 달리 수익률은 입력자소의 길이가 길어질수록 점점 떨어지는 것을 볼 수 있다.

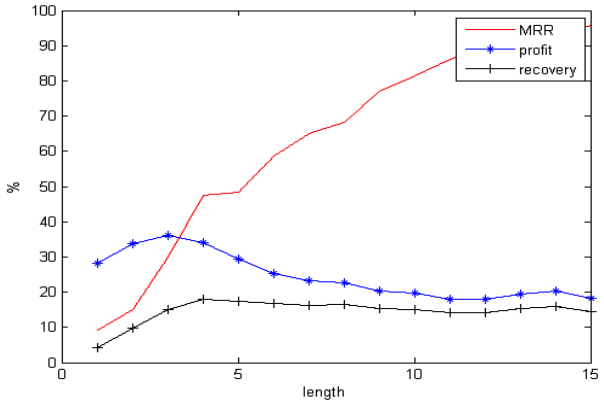


Figure 4: Performance with the length of input strings using the 500k unigram dictionary.

Table 4와 **Figure 4**에서와 같이 복원율은 입력자소 길이가 가장 짧을 때 가장 낮은 성능을 보였으나, 자소길이별 성능의 편차가 다른 세 지표보다 적은 특성을 보였다. 따라서 복원율은 다른 평가 지표와 달리 입력자소의 길이에 비교적 영향을 적게 받는 지표라 할 수 있다.

5.3 오탈자 평가

다음 **Table 5**는 오탈자 평가집합에 대한 유니그램 500k 사전을 사용한 시스템의 성능을 오탈자 종류에 따라 나타낸 것이다.

Table 5: Performance of the system with the type of input errors

type	MRR (%)	profit (%)	recovery (%)
delete	6.6	7.0	3.6
insert	0.6	0.1	0.1
substitute	41.9	19.6	12.4
transpose	2.6	1.4	0.8
multi	7.0	1.9	1.4
Overall	12.1	5.7	3.7

시스템은 오류를 포함한 입력자소가 주어졌을 때, 치환 오류에 대해서만 정상 자소입력보다 수익률은 약 10% 저조한 성능을 보였으나 MRR과 복원율은 비슷한 성능을 보였다. 그 외 삽입, 삭제, 자리바꿈 등의 오류 유형들에 대해서는 아주 저조한 성능을 보였다. 즉, 평가 시스템은 치환 오류 이외의 오타에 대해서는 단어 자동완성 기능을 거의 수행하지 못한다고 할 수 있다.

6. 결론 및 향후 과제

우리는 모바일기기에 사용되는 A사의 단어 자동완성 시스템의 성능을 평가하였다. 사전의 크기와 언어모델 사용에 따른 성능 평가를 위해 약 60GB 크기의 트위터 데이터로부터 유니그램과 바이그램 사전을 각각 구축하였으며 자소의 키입력으로 구성된 평가집합과 오탈자 평가집합을 각각 구축하였다. 사전의 크기가 커질수록 시스템의 성능은 향상되었으며 유니그램보다 바이그램 사전을 사용하였을 때 더 나은 성능을 보였다. 모바일기기의 저장용량을 고려할 때 바이그램 사전보다는 사용자의 입력 이력을 문맥으로 이용하는 편이 시스템의 효율을 높일 수 있을 것이다. 우리는 기존 단어 자동완성 시스템의 평가 지표로 주로 사용되는 MRR과 재현율 이외에 입력자소에 대한 수익률과 목적자소에 대한 복원율을 단어 자동완성 시스템의 새로운 평가지표로 제안하였다. 키입력 수익률과 복원율은 단어 자동완성 시스템의 성능을 직관적으로 이해할 수 있도록 해주는 장점이 있다. 키입력 오류에 대한 시스템 성능도 측정하였는데 치환오류 이외의 다른 입력 오류에 대한 시스템 성능은 저조한 것으로 분석되었다. 단어 자동완성 시스템이 사용자에게 유용한 도구가 되려면 사용자의 입력이력과 도메인에 따라 단어를 추정할 수 있는 방법이 연구되어야 할 것이다. 향후에 추천단어의 순위를 향상 시킬 수 있는 알고리즘의 개발이 필요하다. 이러한 단어 자동완성 시스템은 해상에서 긴급 상황이 발생하였을 때 신속한 메시지를 작성하도록 도와 줄 것이다.

후 기

이 논문은 2013년도 한국교통대학교 교내학술연구비의 지원을 받아 수행한 연구임.

References

[1] E. Fredkin, “Trie memory,” Communications of the ACM, vol. 3, no. 9, pp. 490-499, 1960.

[2] K. J. Lee and S. W, Lee, “Error-driven noun-connection rule extraction for morphological analysis,” Journal of the Korean Society of Marine Engineering, vol. 36, no. 8, pp. 1123-1128, 2012 (in Korean).

[3] A. Acharya, H. Zhu, and K. Shen, “Adaptive algorithms for cache-efficient trie search,” Proceedings of the International Workshop on Algorithm Engineering and Experimentation, pp. 296-311, 1999.

[4] J. Aoe, K. Morimoto, M. Shishibori, and K. H. Park, “A trie compaction algorithm for a large set of keys,” IEEE Transactions on Knowledge and Data Engineering, vol. 8, no. 3, pp. 476-491, 1996.

[5] A. Nandi and H. V. Jagadish, “Effective phrase prediction,” Proceedings of Very Large Data Bases 2007, pp. 219-230, 2007.

[6] M. Arias, J. M. Cantera, J. Vegas, P. del la Feunte, J. C. Alonso, G. G Bernardo, C. Llamas, and A. Zubizarreta, “Context-based personalization for mobile web search,” Proceedings of Personalized Access, Profile Management, and Context Awareness in Databases, pp. 33-39, 2008.

[7] Z. Bar-Yossef and N. Kraus, “Context-sensitive query auto-completion,” Proceedings of WWW, pp. 107-116, 2011.

[8] H. Bast and I. Weber, “Type less, find more: fast autocompletion search with a succinct index,” Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 364-371, 2006.

[9] H. Bast and I. Weber, “When you’re lost for words: Faceted search with autocompletion,” Proceedings of the SIGIR 2006 Workshop on Faceted Search, pp. 31-35, 2006.

[10] H. Bast, C. W. Mortensen, and I. Weber, “Output-sensitive autocompletion search,” String Processing and Information Retrieval, vol. 11, no. 4, pp. 269-286, 2008.

[11] A. van den Bosch, “Effects of context and recency in scaled word completion,” Computational Linguistics in the Netherlands Journal, vol. 1, pp. 79-94, 2011.

[12] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” Journal of Molecular Biology, vol. 48, no.3, pp. 443-453, 1970.