# 입말 표기를 이용한 영어 단어 검색

## Retrieving English Words with a Spoken Word Transliteration

김 지 승(Ji-Seoung Kim)*
김 광 현(Kwang-Hyun Kim)**
이 준 호(Joon-Ho Lee)***

목 차



초 록

영어 사전 검색 서비스 이용자들은 원하는 영어 단어의 철자를 정확하게 기억하지 못하고 ,발음만을 기억하는 경우가 있다. 이러한 이용자들에게 도움을 주기 위해 본 연구에서는 입말 표기 ,즉 영어 단어 발음의 한글 표기를 이용하여 영어 단어를 효과적으로 검색할 수 있는 방법을 제안한다. 이를 위하여 코닉스(KONIX) 코드를 개발하며, 입말 표기와 영어 단어를 코닉스 코드들로 변환한다. 그리고 변환된 코닉스 코드들 사이의 음성적 유사도를 편집 거리 방법과 2-그램 방법을 이용하여 계산한다. 또한 제안한 방법이 입말 표기에 의한 영어 단어 검색에 매우 효과적임을 실험을 통하여 입증한다.

ABSTRACT

Users of searching Internet English dictionary sometimes do not know the correct spelling of the word in mind, but remember only its pronunciation. In order to help these users, we propose a method to retrieve English words effectively with a spoken word transliteration that is a Korean transliteration of English word pronunciation. We develop KONIX codes and transform a spoken word transliteration and English words into them. We then calculate the phonetic similarity between KONIX codes using edit distance and 2-gram methods. Experimental results show that the proposed method is very effective for retrieving English words with a spoken word transliteration.

키워드: 정보 검색, 영어 사전 검색, 음성적 유사도
Information Retrieval, English Dictionary Search, Phonetic Similarity

* 숭실대학교 정보과학대학 컴퓨터학과 박사과정(jskim89@naver.com)
** 숭실대학교 정보과학대학 컴퓨터학과 박사과정(iamkkh@naver.com)
*** 숭실대학교 정보과학대학 컴퓨터학과 조교수(joonho@comp.ssu.ac.kr)
논문접수일자 2005년 8월 15일
게재확정일자 2005년 9월 14일

# 1. 서 론

인터넷의 사용과 보급이 급격히 증가함에 따라 인터넷 검색 포탈들은 다양한 검색 서비스들을 제공하고 있다. 이들 중 영어 사전 검색 서비스는 원하는 단어에 대한 빠른 접근과 더불어 철자 오류 교정, 근접어 검색, 영문 뉴스 예문 제시 등의 유용한 기능들을 지원하기 때문에 인터넷 이용자들에게 널리 사용되고 있다. 따라서, 인터넷 검색 포탈들은 보다 편리한 영어 사전 검색 서비스를 제공하기 위하여 많은 노력을 기울이고 있다.

한편, 이용자들은 검색을 원하는 영어 단어의 철자를 정확하게 기억하지 못하고, 발음만을 기억하는 경우가 종종 있다. 이러한 이용자들에게 도움을 주기 위하여 일부 영어 사전 검색 서비스에서는 입말 표기, 즉 영어 단어 발음의 한글 표기에 의한 영어 단어 검색을 지원한다. 예를 들어, "retrieval"의 입말 표기 "리트리벌"을 검색창에 입력하면, 영어 사전 검색 서비스는 "retrieval", "retriever", "retrievable" 등을 검색 결과로서 제공한다.

입말 표기에 의한 영어 단어 검색을 위해서는 한글 입말 표기와 영어 단어 사이의 음성적 유사도 계산이 수행되어야 한다. 지금까지 단어들 사이의 음성적 유사도를 계산하기 위하여 사운덱스(SOUNDEX) 알고리즘(Hall and Dowling 1980), 포닉스(PHONIX) 알고리즘(Gadd 1988; Gadd 1990), 코덱스(KODEX) 알고리즘(강병주,최기선 1999) 등이 개발되었다. 그리고 일반적으로 문자열이 유사한 단어들은 음성적으로도 유사하기 때문에, 문자열 유사도를 계산하는 편집 거리(Edit Distance) 방법(Damerau 1964; Pollock and Zamora 1984)과 $n$-그램($n$-gram) 방법(Ukkonen 1992; Zamora and Pollock 1981)도 단어들 사이의 음성적 유사도 계산에 널리 사용되고 있다(Zobel and Dart 1996; Pfeifer, Poersch and Fuhr 1996).

사운덱스와 포닉스 알고리즘은 "smith", "smyth" 등과 같이 철자는 다르지만 발음이 동일한 영어 단어들을 하나의 그룹으로 분류하며, 코덱스 알고리즘은 "디지탈", "디지털", "디지틀" 등과 같은 다양한 외래어 표기들을 하나의 그룹으로 분류한다. 즉, 사운덱스와 포닉스 알고리즘은 영어 단어들 사이의 음성적 유사도를 계산하고, 코덱스 알고리즘은 한글 단어들 사이의 음성적 유사도를 계산한다. 그리고 편집 거리 방법과 $n$-그램 방법은 영어 단어들 사이 또는 한글 단어들 사이의 음성적 유사도 계산에 활용될 수 있다.

지금까지 연구된 방법들은 입말 표기에 의한 영어 단어 검색에 직접적으로 적용될 수 없으며, 한글 입말 표기와 영어 단어 사이의 음성적 유사도를 계산하기 위해서는 입말 표기와 영어 단어를 동일한 문자 집합으로 표현하는 과정이 선행되어야 한다. 이를 위하여 본 연구에서는 코닉스(KONIX) 코드를 개발하며, 입말 표기와 영어 단어를 코닉스 코드 변환표에 근거하여 코닉스 코드들로 변환한다. 그리고 변환된 코닉스 코드들 사이의 음성적 유사도 계산을 위하여 기존의 편집 거리 방법과 $n$-그램 방법의 검색 결과들을 결합한다.

본 연구의 구성은 다음과 같다. 2장에서는 관련 연구로써 편집 거리와 $n$-그램을 이용한 문자열 유사도 계산에 대해 기술하고, 3장에서는 입말 표기를 이용한 영어 단어 검색 방법에 대해 기술한다. 4장에서는 3장에서 기술한 방법의 성능을

평가하고, 마지막으로 5장에서는 결론을 맺는다.

## 2. 관련 연구

### 2.1 편집 거리 방법

편집 거리는 한 단어를 다른 단어로 변환할 때 수행되는 편집 연산의 수를 의미하며, 철자를 추가하는 삽입, 철자를 제거하는 삭제, 기존의 철자를 다른 철자로 변환하는 교체, 인접한 두 철자의 순서를 서로 뒤바꾸는 전치가 편집 연산에 포함된다(Damerau 1964). 예를 들어, 〈그림 1〉은 영어 단어 "school"을 "shower"로 변환하는 다양한 방법들 중에서 2개를 보여 준다. 이 그림으로부터 방법 1을 사용하여 단어를 변환할 경우 편집 거리는 5가 되고, 방법 2를 사용하여 단어를 변환할 경우 편집 거리는 4가 됨을 알 수 있다. 이처럼 두 단어들 사이의 편집 거리는 단어를 변환하는 방법에 의존적이다.

한편, 편집 거리 방법에서 단어 $s$와 $t$사이의 문자열 유사도 계산은 다음과 같은 과정으로 수행된다.

1. $m$개의 문자로 구성된 단어 $s$를 $n$개의 문자로 구성된 단어 $t$로 변환하기 위한 최소 편집 거리 $MinEdit_{s,t}$를 다음의 순환 방정식 $edit(m,n)$을 사용하여 계산한다.

$$
\begin{aligned}
&edit(0,0) = 0\\
&edit(i,0) = i\\
&edit(0,j) = j\\
&edit(i,j) = min\{\, edit(i-1,j)+1,\\
&\qquad\qquad edit(i,j-1)+1,\\
&\qquad\qquad edit(i-1,j-1)+d(s_i,t_j),\\
&\qquad\qquad edit(i-2,j-2)+d(s_{i-1},t_j)+d(s_i,t_{j-1})+1\}
\end{aligned}
$$

여기에서 $s_i$와 $t_j$는 각각 단어 $s$의 $i$번째 문자와 단어 $t$의 $j$번째 문자를 의미한다. 그리고 함수 $d(s_i,t_j)$는 두 문자 사이의 거리를 계산하며, $s_i$와 $t_j$가 동일한 철자이면 0, 다른 철자이면 1의 값을 반환한다. 예를 들어 영어 단어 "school"을 "shower"로 변환하기 위한 최소 편집 거리 $MinEdit_{school,shower}$는 $edit(6,6)$을 계산함으로써 얻을 수 있으며, 그 값은 4가 된다.

2. 다음의 유사도 계산식을 사용하여 단어 $s$와 $t$ 사이의 문자열 유사도를 계산한다. 이때 계산된 결과 값이 음수일 경우, 문자열 유사도는 0이 된다.

$$Sim(s,t) = \frac{Len(s) - MinEdit_{s,t}}{Len(s)}$$

여기에서 $Len(s)$는 단어 $s$의 길이, 즉 단어 $s$를 구성하는 문자들의 수를 의미한다. 예를 들어, $Len(school)$의 값은 6이며, 따라서 "school"과 "shower" 사이의 문자열 유사도 $Sim(school, shower)$은 2/6가 된다.

### 2.2 n-그램 방법

$n$-그램은 인접한 $n$개의 문자들을 의미한다. 예를 들어, 영어 단어 "sports"의 2-그램들은 "sp", "po", "or", "rt", "ts"이며, 3-그램들은 "spo", "por", "ort", "rts"이다(Ukkonen 1992; Zamora and Pollock 1981). 일반적으로 다양한 길이의 $n$-그램을 사용하여 두 단어들의 문자열 유사도를 계산할 수 있으나, 다수의 연구에서 2-그램을 사용한 문자열 유사도가 가장 높은 검색 효과를 제공함을 입증하였다(Zobel

| 방법 1 | | | 방법 2 | | |
|---|---|---|---|---|---|
| school | shower | 편집 연산 | school | shower | 편집 연산 |
| school | s | | school | s | |
| school | s | 삭제 c | school | s | 삭제 c |
| school | sh | | school | sh | |
| school | sho | | school | sho | |
| school | show | 삽입 w | school | show | 삽입 w |
| school | showe | 삽입 e | school | showe | 교체 o → e |
| school | shower | 교체 o → r | school | shower | 교체 l → r |
| school | shower | 삭제 l | | | |

〈그림 1〉 단어 변환 방법

and Dart 1995). 이러한 $n$-그램 방법은 철자 오류가 빈번한 영어 인명 또는 영어 단어 검색뿐만 아니라, 사용자 질의에 적합한 문서를 검색하는 일반적인 정보 검색 시스템에서도 사용된다 (Lee 1999). $n$-그램 방법은 다음과 같은 과정을 거쳐 2개의 단어 $s$와 $t$ 사이의 유사도를 계산한다.

1. 단어의 첫 철자와 마지막 철자의 중요성을 강조하기 위하여, 단어 $s$, $t$의 양쪽에 공백 문자 "_"를 추가한다. 예를 들어, 영어 단어 "word"와 "world"가 주어졌을 경우, 이들은 "_word_"와 "_world_"로 변환된다.

2. 공백 문자가 추가된 단어들로부터 $n$-그램들을 생성한다. 예를 들어, 2-그램 방법을 적용할 경우, "_word_"와 "_world_"로부터 각각 "_w", "wo", "or", "rd", "d_"와 "_w", "wo", "or", "rl", "ld", "d_"가 생성된다.

3. 다음과 같은 식을 사용하여 단어 $s$, $t$ 사이의 문자열 유사도를 계산한다.

$$Sim(s,t) = \frac{|N_s \cap N_t|}{|N_s \cup N_t|}$$

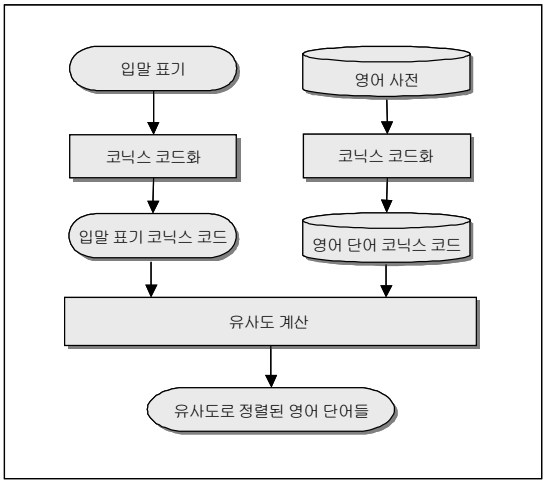
여기에서 $N_s$, $N_t$는 각각 단어 $s$, $t$로부터 생성된 $n$-그램들의 집합을 의미한다. 따라서, 2-그램 방법을 적용할 경우, 단어 "word"와 "world" 사이의 문자열 유사도는 4/7가 된다.

## 3. 입말 표기를 이용한 영어 단어 검색

〈그림 2〉는 입말 표기를 이용하여 영어 단어를 검색하는 시스템의 구조를 보여준다. 영어 단어의 코닉스 코드화 모듈은 영어 사전에 수록된 영어 단어들을 본 연구에서 개발된 코닉스 코드로 변환한다. 입말 표기의 코닉스 코드화 모듈은 사용자가 질의로서 입력한 입말 표기를 코닉스 코드로 변환한다. 그리고 유사도 계산 모듈은 입말 표기의 코닉스 코드와 영어 단어의 코닉스 코드들 사이의 유사도를 계산하여, 유사도가 높은 영어 단어들을 우선적으로 출력한다. 다음에서는 이러한 세가지 모듈에 대하여 자세히 기술한다.

### 3.1 영어 단어의 코닉스 코드화

본 연구에서는 사용자 질의로서 주어진 영어 단어의 입말 표기에 대하여, 이 입말 표기를 생성할 수 있는 영어 단어들을 검색하고자 한다. 입말 표기는 영어 단어의 철자가 아닌 발음 기




〈그림 2〉 입말 표기를 이용한 영어 단어 검색 시스템 구조

호로부터 생성되며, 따라서 본 연구에서는 각각의 발음 기호에 〈표 1〉의 코드를 부여함으로써 영어 단어를 코닉스 코드로 변환한다. 〈표 1〉로부터 'l'과 'r', 'f'와 'p' 등과 같이 입말 표기 시 동일한 음소로 표기되는 발음 기호들에 동일한 코드가 부여됨을 알 수 있다. 또한 이 표에서는 입말 표기가 불가능한 장모음 기호 ':'와 강세 기호 'ˊ'가 생략되었다. 예를 들어, 영어 단어 "retrieval"의 발음 기호는 [ritríːvəl]이며, 이러한 발음 기호에 〈표 1〉의 코드표를 적용함으로써 영어 단어 "retrieval"은 코닉스 코드 "litlibcl"로 변환된다.

〈표 1〉 영어 발음 기호의 코닉스 코드표

| 발음 기호 | 코드 | 발음 기호 | 코드 |
|---|---|---|---|
| b v | b | t | t |
| d ð | d | z ʒ ʤ | z |
| f p | f | a | a |
| g | g | æ | @ |
| h | h | e ɛ | e |
| j | j | ə ʌ | c |
| k | k | i | i |
| l r | l | o ɔ | o |
| m | m | *r* | r |
| n | n | u | u |
| ŋ | $ | w | w |
| s ʃ θ | s | | |

### 3.2 입말 표기의 코닉스 코드화

한글에서 음절은 초성, 중성, 종성으로 구성되며, 종성은 생략될 수 있다. 초성으로는 기본 자음 14자와 쌍자음 5자가 사용되고, 중성으로는 기본 모음 10자와 복모음 11자가 사용되며, 종성으로는 기본 자음 14자와 쌍자음 2자, 복자음 11자가 사용된다. 그러나, 영어 단어의 입말 표기에는 초성과 중성에 동일한 음소들이 사용될 지라도, 종성에는 "ㄱ, ㄴ, ㄹ, ㅁ, ㅂ, ㅅ, ㅇ, ㅋ, ㅌ, ㅍ"의 10개의 자음들만이 사용된다. <표 2>는 입말 표기에 출현하는 초성 19자, 중성 21자, 그리고 종성 10자에 대한 코닉스 코드를 정의하며, 이를 위하여 다음과 같은 사항들이 고려되었다.

- 영어 발음 [g], [d], [b], [s], [z]를 입말 표기하기 위하여 단자음 "ㄱ, ㄷ, ㅂ, ㅅ, ㅈ" 또는 "ㄲ, ㄸ, ㅃ, ㅆ, ㅉ"이 선택적으로 사용된다. 예를 들어, 영어 단어 "gas", "service", "jazz"는 "가스", "서비스", "재즈" 또는 "까스", "써비스", "째즈"로 표기된다. 이러한 사항을 반영하기 위하여 <표 2>에서 단자음과 이에 대응하는 쌍자음을 동일한 코닉스 코드로 변환하였다.
- 영어 단어 cake, pop, set은 각각 "케익", "팝", "셋"으로 입말 표기될 수 있으며, 이로부터 종성에서 "ㅋ, ㅍ, ㅌ"을 대신하여 각각 "ㄱ, ㅂ, ㅅ"로 표기되는 경우가 있음을 알 수 있다. 따라서 <표 2>에서 "ㅋ"과 "ㄱ", "ㅂ"과 "ㅍ", "ㅅ"과 "ㅌ"을 동일한 코닉스 코드로 변환하였다.
- 영어 단어 "apple", "coin"의 입말 표기 "애플", "코인"에서 알 수 있듯이, 입말 표기의 초성에 자음 "ㅇ"이 추가될 수 있다. 또한 영어 단어 "club", "school"의 입말 표기 "클럽", "스쿨"에서 알 수 있듯이, 입말 표기의 중성에 모음 "ㅡ"가 추가될 수 있다. 따라서 <표 2>는 초성의 자음 "ㅇ"과 중성의 모음 "ㅡ"에 대한 코닉스 코드를 정의하지 않고 있다.

본 연구에서는 입말 표기를 구성하는 각각의 음소에 <표 2>에 정의된 코드를 부여함으로써 입말 표기를 코닉스 코드로 변환한다. 예를 들어 입말 표기 "리트리벌"은 <표 2>에 의해 코닉스 코드 "litlibcl"로 변환되며, 이는 영어 단어 "retrieval"의 코닉스 코드 "litlibcl"과 동일함을 알 수 있다.

### 3.3 유사도 계산

영어 사전에 수록된 단어들과 사용자 질의로 주어진 입말 표기가 모두 코닉스 코드로 변환된 이후, 입말 표기의 코닉스 코드와 영어 단어의 코닉스 코드 사이의 유사도는 2장에서 기술한 편집 거리 방법 또는 2-그램 방법을 사용하여 계산될 수 있다. 그러나, 편집 거리 방법과 2-그램 방법은 서로 다른 검색 결과를 생성한다. 즉, 질의에 따라서 편집 거리 방법의 검색 효과가 우수하기도 하고, 이와는 반대로 2-그램 방법의 검색 결과가 우수한 경우도 발생한다.

지금까지 정보 검색 분야에서 서로 다른 검색 결과를 결합하는 데이터 퓨젼(Fox and Shaw 1993; Lee 1995)에 대한 많은 연구들이 수행되었으며, 많은 경우에 데이터 퓨젼이 검색 효과를 개선함이 입증되었다. 따라서 본 연구에서

〈표 2〉 입말 표기 음소의 코닉스 코드표

| 자음 | | | | 모음 | |
|---|---|---|---|---|---|
| 초성 | 코드 | 종성 | 코드 | 중성 | 코드 |
| ㄱ ㄲ | g | ㄱ ㅋ | k | ㅏ | a |
| ㄴ | n | ㄴ | n | ㅐ | @ |
| ㄷ ㄸ | d | ㄹ | l | ㅓ | c |
| ㄹ | l | ㅁ | m | ㅔ | e |
| ㅁ | m | ㅂ ㅍ | f | ㅗ | o |
| ㅂ ㅃ | b | ㅅ ㅌ | t | ㅜ | u |
| ㅅ ㅆ | s | ㅇ | $ | ㅣ ㅢ | i |
| ㅇ | | | | ㅑ | ja |
| ㅈ ㅉ | z | | | ㅋ | jc |
| ㅊ | ts | | | ㅖ | je |
| ㅋ | k | | | ㅒ | j@ |
| ㅌ | t | | | ㅛ | jo |
| ㅍ | f | | | ㅠ | ju |
| ㅎ | h | | | ㅘ | wa |
| | | | | ㅝ | wi |
| | | | | ㅙ ㅞ | w@ |
| | | | | ㅝ | wc |
| | | | | ㅞ | we |
| | | | | ㅡ | |

는 편집 거리 방법의 검색 결과와 2-그램 방법의 검색 결과를 결합함으로써 입말 표기의 코닉스 코드와 영어 단어의 코닉스 코드 사이의 유사도를 계산한다.

편집 거리 방법과 2-그램 방법은 서로 다른 범위의 유사도를 생성하기 때문에, 이들의 검색 결과를 결합하기 위해서는 유사도의 범위를 일치시키는 정규화 과정이 선행되어야 한다. 본 연구에서는 다음과 같이 최대 비정규화 유사도를 사용하여 유사도 정규화를 수행하였다.

$$\text{정규화 유사도} = \frac{\text{비정규화 유사도}}{\text{최대 비정규화 유사도}}$$

한편, 데이터 퓨젼을 위하여 다양한 결합 함수들이 사용되어 왔으며, 본 연구에서는 우수한 성능을 제공하는 것으로 알려진 선형 결합 함수를 사용하여 정규화된 유사도들을 결합하였다.

결합 유사도 = 2-그램 정규화 유사도 + 가중치 * 편집 거리 정규화 유사도

예를 들어, 2개의 코닉스 코드들 "litlibcl"과 "litribal" 사이의 유사도는 편집 거리 방법과 2-그램 방법에 의해 각각 0.8750, 0.7778이 된다. 그리고 각 방법의 최대 비정규화 유사도를 0.9로 가정하면, 이들의 정규화 유사도는 각각 0.9722, 0.8642이 되며, 가중치가 1 인 경우, 이들의 결합 유사도는 1.8364가 된다.

## 4. 성능 평가

### 4.1 성능 평가 방법

본 장에서는 3장에서 기술한 입말 표기를 이용한 영어 단어 검색 방법의 성능을 평가한다.

이러한 성능 평가를 위하여 본 연구에서는 중고등학생들을 위한 영어 단어장에 수록된 7,628개의 입말 표기와 약 95,000여 개의 영어 단어가 수록된 영어 사전을 사용하였다. 영어 단어장에는 "[vəkæbjuleri]보캐뷸레리", "[dʒuvənail]쥬버나일", "[θəːrzdei]서즈데이" 등과 같이 영어 단어의 발음을 한글로 표기한 입말 표기들이 포함되어 있으며, 본 연구에서는 영어 단어장의 입말 표기를 본 연구에서 제안한 방법의 성능 평가를 위한 질의로서 사용하였다. 즉, 본 연구에서는 영어 단어장의 입말 표기로 영어 사전을 검색한 후, 검색 결과 중에서 적합한 영어 단어의 순위를 조사함으로써 검색 결과의 질을 평가하였다.

본 연구에서는 검색 결과의 질을 평가하기 위한 척도로서 다음에서 정의되는 역순위 평균(Mean Reciprocal Rank)을 사용하였다(Voorhees, Ellen and Tice 2000). 이는 질의의 검색 결과 중에서 적합한 문서의 순위를 계산하고, 이 순위의 역수를 평균한 값이다.

$$\text{역순위평균} = \frac{\sum_{i=1}^{N} \frac{1}{rank_i}}{N}$$

이 식에서 $N$은 전체 입말 표기 질의들의 수로서 본 실험에서는 7,628이며, $rank_i$는 $i$번째 입말 표기 질의의 검색 결과 중에서 질의에 적합한 영어 단어의 순위를 의미한다. 예를 들어, 3개의 질의에 대하여 적합한 영어 단어의 순위들이 각각 2위, 8위, 5위일 경우, 이들의 역순위는 각각 1/2, 1/8, 1/5이며, 역순위 평균은 0.275이다. 또한, 본 연구에서는 검색 결과의 상위 10위 이내에 적합한 영어 단어가 검색된 입말 표기 질의의 수를 측정하였다.

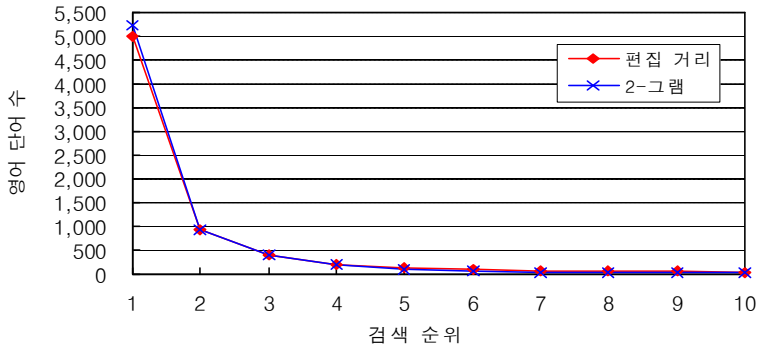
한편, 본 연구에서 제안한 방법들은 주어진 입말 표기 질의에 대하여 다수의 영어 단어들에 동일한 유사도를 부여할 수 있다. 예를 들어 영어 단어 "rain", "reign", "lane" 등은 모두 동일한 코닉스 코드 "lein"을 지니기 때문에, 이들은 입말 표기 질의에 대하여 모두 동일한 문자열 유사도를 지니게 된다. 이러한 경우 본 연구에서는 동일한 유사도를 갖는 영어 단어들을 사전 순서로 정렬하였다.

### 4.2 성능 평가 결과

〈표 3〉은 편집 거리 또는 2-그램 방법을 사용하여 유사도를 계산할 경우, 검색 효과를 보여준다. 이 표로부터 2-그램 방법이 편집 거리 방법보다 다소 우수한 검색 효과를 제공함을 알 수 있다. 그리고 이 표는 편집 거리 방법과 2-그램 방법이 각각 7,034개(92.2%), 7,125(93.4%) 개의 질의에 대하여 상위 10위 이내에 적합한 영어 단어를 검색하고 있음을 보여주고 있다. 한편, 〈그림 3〉은 질의에 적합한 영어 단어의 검색 순위별 분포를 보여준다. 이 그림으로부터 대다수의 적합한 영어 단어들이 1, 2위로 검색됨을 알 수 있다.

〈표 3〉 입말 표기를 이용한 영어 단어 검색 결과

| | 역순위 평균 | 상위 10위 |
|---|---|---|
| 편집 거리 | 0.7539 | 7,034 |
| 2-그램 | 0.7825 | 7,125 |




〈그림 3〉 검색 순위별 질의에 적합한 영어 단어 수

〈표 4〉는 편집 거리 방법의 검색 결과와 2-그램 방법의 검색 결과를 3.3절에서 기술된 결합 유사도 식을 사용하여 결합한 결과를 보여준다. 이 표로부터 가중치 1.6 이상에서 가장 높은 검색 효과를 제공하고, 또한 결합된 검색 결과가 편집 거리 방법과 2-그램 방법보다 우수한 검색 효과를 제공함을 알 수 있다. 이러한 성능 평가 결과는 본 연구에서 제안한 방법이 입말 표기를 이용한 영어 단어 검색에 매우 효과적임을 의미한다. 한편, 가중치가 1.6 이상이 되면 편집 거리 방법이 검색 결과에 미치는 영향이 매우 커지게 된다. 따라서 가중치가 1.6 이상일 경우의 검색 결과는 편집 거리 방법으로 검색을 수행한 후, 동일한 유사도의 단어들을 2-그램 방법의 유사도 순으로 재정렬할 경우와 유사하다.

〈표 4〉 데이타 퓨전 방법을 이용한 영어 단어 검색 결과

| 가중치 | 역순위 평균 | 상위 10위 |
|---|---|---|
| 0.0 | 0.7825 | 7,125 |
| 0.1 | 0.7885 | 7,154 |
| 0.2 | 0.7937 | 7,175 |
| 0.3 | 0.7966 | 7,192 |
| 0.4 | 0.7992 | 7,210 |
| 0.5 | 0.8016 | 7,225 |
| 0.6 | 0.8035 | 7,232 |
| 0.7 | 0.8052 | 7,238 |
| 0.8 | 0.8063 | 7,239 |
| 0.9 | 0.8066 | 7,244 |
| 1.0 | 0.8071 | 7,247 |
| 1.1 | 0.8073 | 7,252 |
| 1.2 | 0.8076 | 7,254 |
| 1.3 | 0.8081 | 7,255 |
| 1.4 | 0.8086 | 7,258 |
| 1.5 | 0.8088 | 7,260 |
| **1.6** | **0.8090** | **7,261** |
| 1.7 | 0.8090 | 7,261 |
| 1.8 | 0.8090 | 7,261 |
| 1.9 | 0.8090 | 7,261 |
| 2.0 | 0.8090 | 7,261 |

## 5. 결 론

인터넷 검색 포탈들에서 제공하는 영어 사전 검색 서비스는 영어 단어를 검색하고자 하는 이용자들에게 널리 사용되고 있다. 그러나 이용자들은 검색을 원하는 영어 단어의 철자를 정확하게 기억하지 못하고, 발음만을 기억하는 경우를 경험한다. 이러한 이용자들에게 도움을 주기 위하여 일부 영어 사전 검색 서비스에서는 입말 표기, 즉 영어 단어 발음의 한글 표기에 의한 영어 단어 검색을 지원하고 있으나, 지금까지 이에 대한 연구는 매우 미흡한 실정이다. 따라서 본 연구에서는 입말 표기를 이용하여 영어 단어를 효과적으로 검색할 수 있는 방법을 제안하였다.

입말 표기에 의한 영어 단어 검색을 위해서는 한글 입말 표기와 영어 단어 사이의 음성적 유사도 계산이 수행되어야 한다. 이를 위하여 본 연구에서는 코닉스 코드를 개발하였고, 입말 표기와 영어 단어를 코닉스 코드로 변환하였다. 그리고 이들 코닉스 코드들 사이의 유사도 계산을 위하여 편집 거리 방법과 n-그램 방법으로부터 생성된 검색 결과들의 결합을 제안하였다. 또한, 성능 평가를 통하여 제안한 방법이 입말 표기에 의한 영어 단어 검색에 매우 효과적임을 입증하였다.

## 참 고 문 헌


강병주, 최기선. 1990. 외국어 음차 표기의 음성적 유사도 비교 알고리즘. 『정보과학회 논문지(B)』, 26(10): 1237-1246.

Damerau, F. 1964. "A technique for computer detection and correction of spelling errors." *Communications of the ACM*, 7: 171-176.

Fox, E. and Shaw, J. 1993. "Combination of Multiple searches." In Harman, D., editor, Proc TREC, pages 35-44, Washington. National Institute of Standards and Technology Special Publication, 500-215

Gadd, T.. 1988. "'Fisching fore werds'. Phonetic retrieval of written text in information retrieval systems." *Program*, 22(3): 222-237.

Gadd, T. 1990. "PHONIX: The algorithm." *Program*, 22(4): 363-366.

Hall, P. and Dowling, G. 1980. "Approximate string matching." *Computing Surveys*, 12(4): 381-402.

Lee, J. 1995. "Combining Multiple Evidence from Different Properties of Weighting Schemes," ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, USA, 180-188.

Lee, J., Cho, h. and Park, H. 1999. "N-Gram-Based Indexing for Korean Text Retrieval," *Information Processing & Management*, 35(4): 427-441.

Pfeifer, U., Poersch, T., and Fuhr, N. 1996. "Retrieval effectiveness of proper name search methods." *Information Processing & Management*, 32(6): 667-679.

Pollock, J., and Zamora, A.. 1984. "Automatic spelling correction in scientific and scholarly text." *Communications of the ACM*, 27(4): 358-368.

Ukkonen, E. 1992. "Approximate string-matching with q-grams and maximal matches." *Theoretical Computer Science*, 191-211.

Voorhees, Ellen M. and Tice, D. 2000. "The TREC-8 Question Answering Track Evaluation." In Text Retrieval Conference TREC-8.

Zamora, E., Pollock, J., and Zamora, A. 1981. "The use of trigram analysis for spelling error detection." *Information Processing and Management*, 17(6): 305-316.

Zobel, J., and Dart, P. 1995. "Finding Approximate Matche in Large Lexicons." *Software-Practice and Experience*, 25(3): 331-345.

Zobel, J., and Dart, P. 1996. "Phonetic string matching: Lessons from information retrieval." In Proceedings of ACM SIGIR Conference on Information Retrieval, Zurich, Switzerland, 166-172.