DOI: 10.3745/KIPSTB.2011.18B.2.087

한국어 자모 Viable Prefix를 이용한 외래어 표기 교정 기법

권 순 호[†] · 권 혁 철[††]

요 약

한국어 문서에서 외래어 표기는 한 단어에 대해 한 개만 존재하는 것이 아니라 여러 개의 다른 표기로 사용되고 있다. 이러한 표기상 불일치는 하나의 단어가 다른 개념으로 인식되어 정보 검색 시스템의 성능 저하의 원인이 된다. 따라서 정보 검색 시스템에서는 다양한 외래어 표기에 대해 같은 개념으로 인식하여 검색할 수 있도록 외래어 표기법에 맞는 외래어 표기로 교정하는 전처리가 필요하다. 본 논문에서는 질의어로 외래어가 입력되면, 이를 근거로 외래어 표기법에 맞는 외래어 표기로 교정해주는 방법을 제안한다. 제안하는 기법은 한국어 자모의 viable prefix를 이용하여 후보 외래어 표기를 생성하는 가상 트리를 작성하고 불필요한 외래어를 가지치기함으로써 검색 정확도를 높이고 속도를 개선한다.

키워드 : 외래어 표기, 정보 검색, Viable Prefix, 음성적 유사도

Transliteration Correction Method using Korean Alphabet Viable Prefix

Soonho Kwon[†] · Hyuk-Chul Kwon[††]

ABSTRACT

In Korean documents, there are diverse spellings of transliterated foreign loanwords. This fact diminishes the performance of information retrieval systems in that a foreign word can be recognized differently, which is to say, as two or several different words. Thus, information retrieval systems require preprocessing to correct nonstandard loanword spellings prior to searching and recognizing corresponding equivalent words. This paper proposes a method that improves precision and processing efficiency using the Korean alphabet's viable prefix, which prunes a virtual tree from which candidate loanwords are created.

Keywords : Writing of Loanword, Information Retrieval, Viable Prefix, Phonetic Similarity

1. 서 론

최근 한국어 문서에는 한국어뿐만 아니라 영어 등의 외국어 문자 표기와 그에 해당하는 외래어 표기 등이 혼합되어 사용되고 있다. 특히 외래어 표기는 한 단어에 대해 한 개만 존재하는 것이 아니라 개인차 및 외래어 표기법 인식 부족 등을 원인으로 여러 개의 다른 외래어 표기가 통용되는 것이 보편적이다. 예를 들어 영어 'data'에 대해 '데이터', '데이타' 등의 표기는 모두 같은 개념을 표현하지만, 표기상 불일치로 말미암아 다른 개념으로 인식되어 정보검색 시스템의 성능 저하가 일어난다. 이러한 다양한 외래어 표기를 같은 개념으로 인식하는 것은 정보검색 성능 향상에 주요 요소가 된다[3]. 그러나 아직 대부분의 정보검색 시스템이나 데이터베이스 시스템 등에서는 다양한 외래어 표기를 하나의 색인어로 처리하지 않아 다른 형태의 외래어가 질의어로 들어왔을 경우, 정확히 일치하지 않는 단어는 전혀 검색을 못 하는 경우가 많이 있다[7]. 따라서 정보검색 시스템의 성능 향상을 위해 여러 외래어 표기를 같은 개념으로 인식하는 시스템이 필요하다.

기존의 유사 외래어를 검출하는 방법으로 n-gram과 edit distance를 이용하여 두 외래어 표기의 유사도를 측정하는 방법[8], 외래어를 음성적 유사도에 기반을 둔 코드로 변환하여 음성적 유사도를 구하는 방법[1][2], 외국어 표기(영어, 일본어 등) 또는 외래어 표기를 일정한 패턴으로 외래어 표기를 확장하는 방법[7], 외래어 표기의 자모별 혼동행렬을 이용하여 확장된 n-gram 방법으로 외래어 표기의 유사도를

※ 이 논문은 2010년도 정부(교육과학기술부)의 재원으로 한국연구재단의 자원을 받아 수행된 연구임(No, 2010-0028784).
† 준 회 원 : 부산대학교 컴퓨터공학과 석사과정
†† 정 회 원 : 부산대학교 정보컴퓨터공학부, 인지과학협동과정 교수
논문접수 : 2010년 7월 13일
수 정 일 : 1차 2010년 9월 6일, 2차 2010년 10월 12일
심사완료 : 2010년 10월 14일

구하는 방법[3] 등이 있다.

그동안 연구된 유사 외래어 표기 검출 기법들은 실제 사용될 가능성이 있는 외래어 뿐만 아니라 너무 많은 불필요한 외래어가 생성되어 정보검색 시스템의 전처리기로 사용하기에 비효율적이었다. 또한, n-gram을 사용하는 기법들은 유사 외래어를 검출하는 속도가 느려 정보검색 시스템에서 질의어를 확장하기 위한 방법으로 적합하지 않다. 본 논문에서 제안하는 기법은 한국어 자모의 viable prefix를 이용하여 후보 외래어 표기를 생성하는 가상 트리를 작성하고 부적합한 외래어를 가지치기함으로써 검색 성능 향상과 속도를 개선한다.

본 논문의 구성은 다음과 같다. 2장에서는 유사 외래어 표기 검출의 관련 연구를 살펴보고 3장에서는 본 논문에서 제안하는 유사 외래어 표기 검출 기법을 자세히 소개한다. 4장에서는 기존의 방법들과 제안한 방법을 실험하고 비교하여 성능을 평가한다. 마지막으로 5장에서는 연구 내용을 정리하여 결론을 맺는다.

2. 관련 연구

기존의 유사 외래어를 검출하는 방법으로 먼저, n-gram과 edit distance를 이용한 방법이 있다[5][8]. N-gram 방법[9]는 두 문자열의 n-gram들이 많이 일치할수록 더 유사하다고 판단하는 방법이다. Edit distance[9]는 두 문자열을 같게 만들기 위해 삽입, 삭제, 대체, 전위의 수를 최소한의 비용으로 계산하는 철자교정 알고리즘이다. 만약 하나의 문자열이 임계값 이하의 비용으로 다른 문자열로 변환된다면, 두 문자열은 같은 문자열로 판단한다. 그러나 한글의 경우 자음과 모음의 조합으로 하나의 글자가 이루어지는 방식이기 때문에 n-gram과 edit distance를 적용하려면 전처리과정이 필요하다[6]. [8]은 단어를 구성하는 음절을 음소 단위로 변환하였고, [5]는 단어를 특정 코드로 치환하여 n-gram과 edit distance로 유사도를 측정하였다.

KODEX 알고리즘[1]은 영어 단어의 음성적 유사도를 구하는 Soundex 알고리즘[9][11][12]을 한글에 적용한 것으로, 초성과 종성에 코드번호를 부여하여 동일하게 생성된 코드에 대하여 유사 외래어로 인식한다. KODEX 알고리즘은 초성 이응 제거, 중복 종성 제거, 초성 대표 자음화, 코드 치환, 연속 중복 코드 제거의 다섯 단계의 처리 과정으로 구성된다.

CKODEX 알고리즘[2]은 KODEX 알고리즘에 기반을 둔 것으로 첫음절과 마지막 음절의 모음 정보를 추가하고 Metaphone 알고리즘[10]의 개념을 도입하여 KODEX 보다 세분화한 규칙을 적용함으로써 한국어 외래어 음차표기의 유사도 비교 성능을 향상했다. 그러나 KODEX, CKODEX 알고리즘은 모든 음절의 모음 정보를 사용하지 않아 낮은 정확도를 보인다.

[7]은 좌우 문맥 정보에 기반하여 이형태를 치환하여 생성하는 확률 규칙을 제안하였다. 실제 사용되고 있는 문서에서 추출한 이형태 리스트로부터 치환 규칙을 자동으로 학습하고, 학습 된 규칙을 기반으로 혼동하여 표기하기 쉬운 이형태 외래어를 확률 순서로 생성한다. 치환 관계는 경우에 따라 생성 가능성은 있지만, 실제 사용되지 않는 이형태를 생성할 수 있어 성능을 떨어뜨리는 요인이 될 수 있다.

[3]에서는 자모 혼동행렬을 구성하고, 이를 이용하여 n-gram을 확장하여 외래어 표기의 유사도를 비교하여 유사 외래어를 검출하는 방법으로, 세 단계의 과정으로 구성된다. 1) 입력된 외래어를 대표 자모로 변환하고, 2) 자모 혼동행렬을 이용하여 대표 자모 n-gram을 확장하여 검색한다. 3) 검색 결과로 생성된 후보 외래어들을 유사도 재검증을 통해 최종 외래어 표기를 생성한다. 이 방법은 <표 1>과 같이 "팬더"를 "판다"로 교정하고자 22개의 n-gram을 확장한다. 이러한 n-gram 확장으로 말미암아 유사 외래어 표기를 검색하는 속도가 느려진다.

〈표 1〉 확장된 "팬더"의 3-gram

	3-gram
확장 전	$표ㅐ$_1$ 표ㅐㄴ$_2$ ㅐㄴㄷ$_3$ ㄴㄷㅏ$_4$ ㄷㅏ-$_5$ ㅏ-$$_6$
확장 후	$ㅂㅐ$_1$ ㅂㅐㄴ$_2$ $$표ㅏ$_1$ 표ㅏㄴ$_2$ ㅏㄴㄷ$_3$ 표ㅐㅁ$_2$ ㅐㅁㄷ$_3$ ㅁㄷㅏ$_4$ 표ㅐㅂ$_2$ ㅐㅂㄷ$_3$ ㅂㄷㅏ$_4$ 표ㅐ-$_2$ ㅐ-ㄷ$_3$ -ㄷㅏ$_4$ ㅐㄴㅌ$_3$ ㄴㅌㅏ$_4$ ㅌㅏ-$_5$ ㄴㄷㅐ$_4$ ㄷㅐ-$_5$ ㅐ-$$_6$ ㄷㅏㄹ$_5$ ㅏㄹ$$_6$

본 논문에서는 한국어 자모의 viable-prefix를 이용하여 입력된 외래어 표기를 외래어 표기법에 맞는 유사 외래어 표기로 확장하는 방법을 제안한다. 다음 장에서는 본 논문에서 제안하는 유사 외래어 표기 검출 기법을 자세히 소개한다.

3. 외래어 표기 교정 기법

본 논문에서 제안하는 한국어 자모 viable prefix를 이용한 외래어 표기 교정 기법을 설명하기에 앞서 두 가지 선행되어야 하는 과정이 있다. 첫째, 혼동하기 쉬운 자모를 하나의 그룹으로 묶어 대표 자모를 구성하는 단계, 둘째, 좌우 문맥을 고려하여 혼동되는 대표 자모의 변환 규칙을 생성하는 단계가 필요하다. 이 장에서는 대표 자모 구성 방법, 대표 자모 변환 규칙, 그리고 본 논문에서 제안하는 교정 알고리즘을 소개한다.

3.1 대표 자모 구성

본 논문에서는 [3]에서 사용된 대표 자모를 사용하였다. [3]은 우리말 배움터(http://urimal.cs.pusan.ac.kr)에서 서비스되고 있는 외래어-한글표기 상호변환기의 로그 데이터에서 1,662쌍의 유사 외래어 표기 데이터에서 길이가 같은 외래어 쌍을 자모 단위로 비교하여 혼동하기 쉬운 자모를 추출하여 표준 발음법, 외래어 표기법에 근거하여 대표 자모를 만들었다. <표 2>, <표 3>, <표 4>는 각각 초성, 중성, 종성의 대표 자모를 나타낸다.

〈표 2〉 초성 대표 자모

대표 초성	초성
ㄱ	ㄱ
ㄴ	ㄴ
ㄷ	ㄷ
ㄹ	ㄹ
ㅁ	ㅁ
ㅂ	ㅂ, ㅃ
ㅅ	ㅅ, ㅆ
ㅇ	ㅇ
ㅈ	ㅈ
ㅊ	ㅊ, ㅉ
ㅋ	ㅋ, ㄲ
ㅌ	ㅌ, ㄸ
ㅍ	ㅍ
ㅎ	ㅎ

〈표 3〉 중성 대표 자모

대표 중성	중성
ㅏ	ㅏ, ㅓ, ㅗ
ㅐ	ㅐ, ㅒ
ㅡ	ㅡ, ㅜ, ㅠ
ㅣ	ㅣ, ㅟ, ㅢ
ㅑ	ㅑ, ㅕ, ㅛ
ㅝ	ㅝ, ㅘ
ㅖ	ㅖ, ㅒ, ㅞ, ㅙ, ㅚ

〈표 4〉 종성 대표 자모

대표 종성	종성
ㄱ	ㄱ, ㄳ, ㄲ, ㅋ, ㄺ
ㄴ	ㄴ, ㄵ, ㄶ
ㄹ	ㄹ, ㄻ, ㄼ, ㄽ, ㄾ
ㅁ	ㅁ, ㄻ
ㅂ	ㅂ, ㅄ, ㅍ, ㄿ
ㅅ	ㅅ, ㅆ, ㄷ, ㅈ, ㅊ, ㅌ, ㅎ
ㅇ	ㅇ
- (종성 없음)	- (종성 없음)

3.2 대표 자모 변환 규칙

대표 자모만으로는 "팬더(판다)", "동키호테(돈키호테)"와 같은 외래어를 교정할 수 없으므로 혼동되는 대표 자모를 변환할 수 있는 규칙이 필요하다. 본 논문에서는 [3]에서 사용된 대표 자모 혼동행렬(Confusion Matrix)을 이용하여 자모 변환 규칙을 만들었다. 대표 자모 혼동행렬 C는 수식(1)과 같다.

$$C_{ij}=\begin{cases}1 & if\ (P(j|j)>\lambda_1 \text{ and } P(i|j)>\lambda_3)\\ & \text{or}\ (\lambda_1 \geq P(j|j)>\lambda_2 \text{ and } P(i|j)>\lambda_4)\\ & \text{or}\ (P(i|j)>\lambda_5)\\ 0 & otherwise\end{cases} \quad (1)$$

수식(1)의 $P(i|j)$는 자모 i를 자모 j로 혼동한 확률을 나타내는 값으로, 3.1절의 데이터를 이용하였다. 확률값이 수식(1)의 조건을 만족할 때, 자모 i를 자모 j로 혼동한 것으로 가정한다. [3]에서 $\lambda_1=0.94$, $\lambda_2=0.8$, $\lambda_3=0.006$, $\lambda_4=0.033$, $\lambda_5=0.09$일 때, 실험적으로 가장 적합한 조건이다. 수식(1)에서 C_{ij}는 자모 i를 자모 j로 혼동한 수치로 C_{ij}가 1이면 자모 j를 자모 i로 확장한다. (그림 1)은 대표 중성의 혼동행렬이다.

(그림 1)에서 대표 중성 'ㅡ'와 같이 혼동되는 자모의 개수가 1개일 경우는 변환할 자모가 바로 결정되지만, 대표 중성 'ㅏ'와 같이 혼동되는 자모의 개수가 2개 이상일 경우에는 어떤 대표 자모로 변환할지 좌우 문맥을 보고 결정되

$$C=\begin{array}{c} \begin{matrix}\text{ㅏ} & \text{ㅐ} & \text{ㅡ} & \text{ㅣ} & \text{ㅑ} & \text{ㅝ} & \text{ㅖ}\end{matrix}\\ \begin{pmatrix}0&1&1&0&1&0&0\\1&0&0&1&0&0&0\\0&0&0&1&0&0&0\\0&1&0&0&0&0&0\\1&0&0&0&0&0&0\\1&0&0&0&0&0&0\\0&1&0&0&0&0&0\end{pmatrix}\begin{matrix}\text{ㅏ}\\\text{ㅐ}\\\text{ㅡ}\\\text{ㅣ}\\\text{ㅑ}\\\text{ㅝ}\\\text{ㅖ}\end{matrix}\end{array}$$

(그림 1) 대표 중성 혼동행렬

〈표 5〉 초성 대표 자모 변환 규칙

대표초성	변환 규칙	예
ㄱ	ㅋ: default	헬리곱터->헬리콥터
ㄴ	ㅇ: default	컨베이너->컨베이어
ㄷ	ㅌ: default	카달로그->카탈로그
ㅂ	ㅍ: default	트램블린->트램펄린
ㅅ	ㅈ: default	스타워스->스타워즈
ㅇ	ㄴ: default	레몬에이드->레모네이드
ㅈ	ㅅ: default	베이직->베이식
ㅊ	ㅈ: default	뱃치->배지
	ㅌ: 앞 음절 종성 'ㄴ' or 중성 'ㅡ'	센치->센티 센츄럴->센트럴
ㅋ	ㄱ: default	타킷->타깃
ㅌ	ㄷ: default	쿠테타->쿠데타
ㅎ	ㅍ: default	훼밀리->패밀리

〈표 6〉 중성 대표 자모 변환 규칙

대표 중성	변환 규칙	예
ㅏ	ㅐ: default	로즈마리->로즈메리
	ㅡ: 마지막 음절 and (초성 'ㄷ' or 'ㅌ' or 종성 없음)	샌달->샌들
	ㅑ: 초성 'ㅅ' or (마지막 음절 and 초성 'ㅇ' and 종성 없음)	그루지아->그루지야
ㅐ	ㅏ: default	게스->가스
	ㅣ: 종성 'ㅅ' or (종성 없음 and (뒤 음절 초성 'ㄷ' or 'ㅂ' or 'ㅋ' or 'ㅌ' or 'ㅍ')	초콜렛->초콜릿 코메디언->코미디언
ㅡ	ㅣ: default	플렉스블->플렉시블
ㅣ	ㅐ: default	오린지->오렌지
ㅑ	ㅏ: default	캬바레->카바레
ㅝ	ㅏ: default	제스춰->제스처
ㅖ	ㅐ: default	훼이크->페이크

〈표 7〉 종성 대표 자모 변환 규칙

대표종성	변환 규칙	예
ㄱ	- (종성 없음): 뒤 음절 초성 'ㄱ' or 'ㅋ'	팩킷->패킷
ㄴ	ㅇ: default	바톤->바통
ㄹ	- (종성 없음): 뒤 음절 초성 'ㄹ'	하일라이트->하이라이트
ㅅ	- (종성 없음): 뒤 음절 초성 'ㅅ' or 'ㅌ'	컷트->커트
ㅇ	ㄴ: default	콩크리트->콘크리트
- (종성 없음)	ㄹ: 뒤 음절 초성 'ㄹ'	포크레인->포클레인

어야 한다. <표 5>, <표 6>, <표 7>은 (그림 1)과 같은 대표 자모 혼동행렬을 이용하여 각각 초성, 중성, 종성의 대표 자모 변환 규칙을 생성한 것이다. 변환 규칙에서 default는 좌우문맥에 관계없이 변환됨을 나타낸다. (그림 1)에서 'ㅏ'는 혼동되는 자모의 개수가 3개이므로, <표 6>에서 좌우 문맥에 따라 3가지로 변환됨을 보인다.

대표 자모 변환 규칙뿐만 아니라 "윈도우" - "윈도", "로보트" - "로봇", "네트웍" - "네트워크"와 같이 길이가 다른 외래어의 검색을 위해 <표 8>과 같은 규칙을 추가하였다. 표 8의 'ㅡ', 'ㅇ' 제거 규칙은 viable prefix를 이용한 교정 알고리즘을 수행하기 전에 입력된 외래어 표기에 미리 적용한다.

〈표 8〉 'ㅡ', 'ㅇ' 제거 규칙

	규칙 ('*'는 종성, '-'는 종성 없음)
'ㅡ' 제거	-*크 -> ㄱ*, -*트 -> ㅌ*, ㅅ*트 -> ㅌ* -*르 -> ㄹ*, -*프 -> ㅂ*
'ㅇ' 제거	ㅗ우 -> ㅗ, ㅗ오 -> ㅗ, ㅖ이 -> ㅖ ㅠ우 -> ㅠ, ㅣ유 -> ㅠ, ㅖ인 -> ㅖㄴ* ㅣ엔 -> ㅖㄴ*, ㅗ올 -> ㅗㄹ*

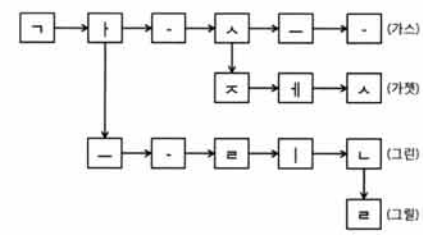
3.3 viable prefix를 이용한 교정 알고리즘

자모 변환 규칙을 이용하여 후보 어절을 생성하면 (그림 2)와 같이 다른 형태의 외래어가 매우 많이 생성될 수 있다. 이러한 방법은 유사 외래어 표기를 검출하는 속도를 느리게 하며, 검색 정확도를 떨어뜨린다. 그러므로 본 논문에서는 viable prefix를 이용하여 후보 외래어 표기를 생성하는 가상 트리에 가지치기함으로써 후보 외래어 생성을 최대한 제약한다. Viable prefix[4]란, 문자열에 대한 최장의 유효한 앞부분(prefix)을 말한다. 즉, 질의어와 사전 내의 단어가 최장으로 일치되는 앞부분까지는 올바른 외래어 표기로 가정한다.

예를 들어, (그림 2)와 같이 입력된 외래어가 '텀브링' 일 때, 정답인 '텀블링(tumbling)'을 찾기 위해 10,800가지의 후보 외래어 표기가 생성된다. 그러나 viable prefix를 이용하여 검색하게 되면, 'ㅌㅓㅁㅂㅡ'가 viable prefix가 되어 '-(종성 없음)'을 'ㄹ'로 변환하는 규칙 한 번만 적용하여 '텀블링'을 찾을 수 있다.

(그림 2) 자모 변환 규칙에 따라 생성된 외래어 표기

본 논문에서는 각 실험 데이터에 있는 표준 외래어 표기를 대표 자모로 변환하여 사전을 구성하였다. 이때, 사전은 viable prefix를 효과적으로 찾아내기 위해 trie 구조를 이용하여 구성한다. Trie 구조는 사전 탐색에서 매우 효율적인 자료구조로 적은 메모리를 사용하여 최대의 탐색 효과를 낼 수 있다. 특히, prefix 검색이 가능하여 viable prefix를 찾는 데 매우 효과적이다. (그림 3)은 한국어 자모 단위 trie 사전에 저장된 예를 보여준다.



(그림 3) 한국어 자모 단위 trie 사전의 예

```
VPSearch(query)
1  word = WordToRepKorAlpha(query)
2  p ← FindViablePrefix(word)
3  while(p ≠ NIL)
4    term = ChangeKorAlpha(word, p)
5    result = DictionarySearch(term)
6    if result ≡ NIL
7      then p ← p − 1
8  end while
9  Add result
10 rWord = ReverseString(word)
11 p ← FindViablePrefix(rWord)
12 while(p ≠ NIL)
13   term = ChangeKorAlpha(rWord, p)
14   result = DictionarySearch(term)
15   if result ≡ NIL
16     then p ← p − 1
17 end while
18 Add result
19 return result
```

(그림 4) viable prefix를 이용한 교정 알고리즘

한국어 자모 단위 trie 사전에서 최장으로 일치하는 위치인 viable prefix를 찾고, 찾아낸 viable prefix를 기준으로 자모를 대치하여 올바른 외래어 표기로 교정한다. 이 viable prefix를 이용하여 전체 알고리즘의 교정 속도를 향상하고 부적절한 대치어의 생성을 방지할 수 있다. (그림 4)는 viable prefix를 이용한 교정 알고리즘이다.

(그림 4)의 알고리즘은 7가지 단계로 볼 수 있다. 1) 입력된 외래어 표기를 대표 자모 표기로 변환한다. 2) viable prefix를 구한다. 3) viable prefix 위치 이후의 자모를 자모 변환 규칙을 통해 변환한다. 4) 변경된 문자열로 검색한다. 5) 검색된 결과가 없다면 viable prefix 위치를 이전 자모로 이동한다. 6) '3-5'의 과정을 반복 수행한다. 7) 첫음절의 초성에 잘못된 표기가 있으면 viable prefix가 NULL이 되므로 질의어를 역방향으로 바꾸고 '2-6'의 과정을 수행한다.

본 논문에서 제안한 VP 교정 알고리즘에서 6번째 과정이 많이 일어나면 사전 검색 횟수가 증가하여 시스템의 성능이 나빠질 수 있으므로 실험 데이터 1을 이용하여 사전 접근 횟수를 세어보았다. 실험 데이터 1은 4장에서 자세히 설명한다. <표 9>는 실험 데이터 1을 이용하여 사전 접근 횟수를 세어 그 데이터의 비중을 나타낸 것으로 평균 2.46번의 접근 횟수를 보였다. 이에 반해 자모 혼동행렬을 이용한 확

장 n-gram 방법[3]의 평균 사전 접근 횟수는 평균 26.58번으로 10배 이상 차이가 났다.

〈표 9〉 VP 교정 알고리즘의 사전 접근 횟수에 따른 데이터 비중

사전 접근 횟수(회)	1	2	3	4	5
데이터 비중(%)	48.9	11.8	13.0	11.0	7.3
사전 접근 횟수(회)	6	7	8	9	10 이상
데이터 비중(%)	4.1	2.5	0.9	0.3	0.2

4. 실험 및 성능평가

제안 기법의 성능 평가를 위해 <표 10>에 나타난 두 가지 실험 데이터를 이용하였다. 실험 데이터 1은 타 알고리즘과 비교 평가하기 위해 [3]에서 사용한 실험 데이터를 대상으로 실험하였다. 실험 데이터 1은 총 834개의 외래어 표기와 355개의 유사 외래어 표기 집합으로 구성되어 있다. 실험 데이터 2는 국립국어원에서 제공하는 데이터로 71,005개의 외래어 표기와 22,281개의 유사 외래어 표기 집합으로 구성되며 본 논문에서 제안한 알고리즘의 속도와 성능을 평가하기 위해 사용되었다. 유사 외래어 표기 집합은 다음과 같이 한 개의 정답 외래어 표기와 여러 개의 비표준 외래어 표기로 이루어져 있다. 유사 외래어 표기 집합의 예는 다음과 같다.

예) duet: 듀엣(O), 두엣(X), 두에트(X), 듀에트(X)

〈표 10〉 실험 데이터

1	IT 외래어 표기 용례집	한국정보통신기자협회(2002. 12)
	깁고 더한 우리말의 바른 표기와 표준어 지도자료(2661어) 중 VI. 외래어의 새 표기법	경상남도 교육청
	이런말실수 저런글실수 중 부록 2. 기본외래어표기	문화관광부
2	국립국어원 찾기 마당, 외래어 표기법, 용례 찾기	국립국어원 홈페이지

본 논문에서는 n-gram 방법, CKODEX 알고리즘[7], 자모 혼동행렬을 이용한 확장 n-gram 방법[3]과 제안한 기법(VP)의 성능을 비교 실험하였다. 추가로 본 논문에서 제안한 기법으로 검색하여 결과가 없을 때, 자모 혼동행렬을 이용한 n-gram 확장 방법을 이용하는 복합형 방법도 실험하였다. 평가 방법으로는 정확도(Precision)와 재현율(Recall)의 조화 평균을 이용하는 F1-measure 값을 사용한다.

$$\text{정확도} = \frac{\text{올바르게 교정한 외래어 표기 개수}}{\text{시스템이 교정한 외래어 표기 개수}} \quad (2)$$

$$\text{재현율} = \frac{\text{올바르게 교정한 외래어 표기 개수}}{\text{전체 정답 개수}} \quad (3)$$

$$F1 = \frac{2 \times Recall \times \text{Precision}}{Recall + \text{Precision}} \quad (4)$$

<표 11>은 실험 데이터 1을 이용하여 각 알고리즘의 성능을 비교한 것이다. 본 논문에서 제안한 기법이 자모 혼동행렬을 이용한 확장 n-gram 방법보다 F1-measure 값이 작지만, 정확도에서는 많이 앞서는 것을 볼 수 있다. 혼합형 방법의 경우, 정확도와 재현율에서 모두 높은 값을 보였다.

〈표 11〉 실험 데이터 1을 이용한 알고리즘의 성능 비교

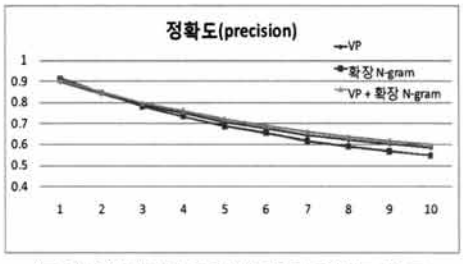
	정확도 (Precision)	재현율 (Recall)	F1-measure
N-gram	0.894	0.553	0.683
CKODEX	0.865	0.689	0.767
확장 N-gram	0.914	0.786	0.854
VP	0.965	0.711	0.818
VP + 확장 n-gram	0.963	0.772	0.857

데이터의 크기 변화에 따른 성능을 평가하기 위해 실험 데이터 2를 크기에 따라 10단계로 준비하였다. 즉, 전체 외래어 목록을 10등분 하여 1/10, 2/10, …, 10/10의 크기별로 10개의 데이터 집합을 준비하였다. <표 12>는 크기별로 분류된 10개의 데이터 집합을 나타낸다.

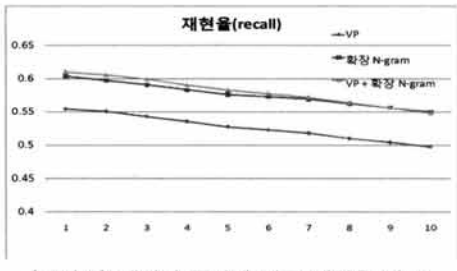
〈표 12〉 실험 데이터 2를 크기에 따라 10단계로 분류

	외래어 표기 집합 개수	질의어 개수
1	2,275	4,957
2	4,516	9,903
3	6,763	14,852
4	8,996	19,726
5	11,232	24,583
6	13,469	29,546
7	15,683	34,333
8	17,905	39,259
9	20,094	43,939
10	22,281	48,724

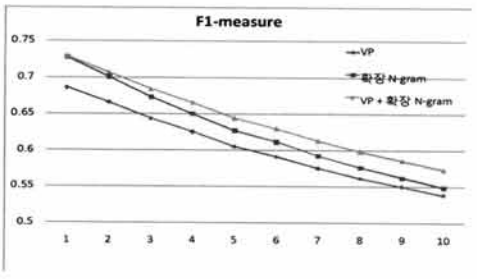
10개의 데이터 집합을 이용하여 본 논문에서 제안한 기법, 자모 혼동행렬을 이용한 확장 n-gram 방법과 두 기법을 혼합한 방법의 성능을 평가하였다. 각 알고리즘의 정확도, 재현율, F1-measure, 검색 속도는 (그림 5), (그림 6), (그림 7), (그림 8)에 나타내었다.



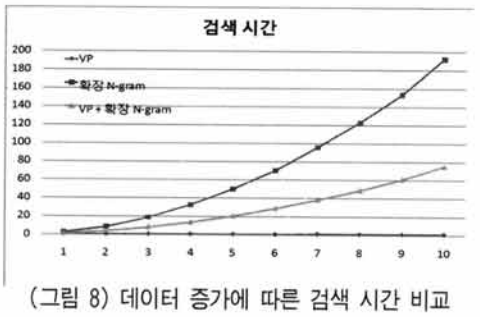
(그림 5) 데이터 증가에 따른 정확도 비교



(그림 6) 데이터 증가에 따른 재현율 비교



(그림 7) 데이터 증가에 따른 F1-measure 비교



(그림 8) 데이터 증가에 따른 검색 시간 비교

(그림 5)와 (그림 7)에서 확장 n-gram 방법이 본 논문에서 제안한 기법보다 데이터 증가에 따라 정확도가 낮아지는 폭이 커져서 대량의 데이터에서는 F1-measure 값이 점차 비슷해지는 것을 볼 수 있다. 대량의 데이터에서 성능이 나빠지는 원인은 실제 정답보다 더 유사한 외래어 표기가 생기기 때문으로 보인다. 예를 들어, '기부스'에 대한 올바른 교정으로 '깁스(Gips)'이지만, '기브스(Gibbs): 인명'으로 잘못 교정된다. 앞으로 대량의 데이터에서도 성능을 높일 수 있는 방법에 대한 추가 연구가 필요하다.

(그림 8)은 질의어를 처리하는 데 걸린 시간을 나타낸 것이다. 확장 n-gram 방법은 데이터가 작을 때에는 초당 2,000개의 질의어를 처리하였지만, 점차 데이터가 많을수록 초당 처리할 수 있는 질의어 수가 줄어들어 10번째 데이터 집합에서는 초당 250개의 질의어를 처리하였다. 이에 반해 본 논문에서 제안한 기법은 데이터 크기에 관계없이 초당 40,000개의 질의어를 처리하였다. 이와 같은 결과는 VP 교정 알고리즘이 확장 n-gram 방법에 비해 사전 접근 횟수가 매우 적기 때문으로 보인다.

5. 결 론

본 논문에서는 한국어 자모 viable prefix를 이용한 외래어 표기 교정 기법을 제안하였다. 한국어 자모의 viable prefix를 이용하여 후보 외래어 표기를 생성하는 가상 트리를 작성하고 부적합한 외래어를 가지치기함으로써 교정 기법의 속도 및 정확도를 높일 수 있었다. 특히 대용량의 데이터에서 기존 확장 n-gram 방법에 비해 F1-measure 값은 비슷하지만, 처리 속도가 매우 빨라 정보검색 시스템의 전처리기로 사용하기에 적합하다. 또한, 본 논문에서 제안한 기법과 확장 n-gram 방법을 같이 이용하는 혼합형 방법은 높은 정확도와 재현율을 보였다. 따라서, 본 논문에서 제안하는 방법을 이용하여 정보검색이나 데이터베이스 시스템 등의 전처리 프로그램에서 질의어 확장용으로 사용되거나 유사 외래어 추출 프로그램 등으로 응용할 수 있다는 데에 본 연구의 의의가 있다.

참 고 문 헌

[1] 강병주, 이재성, 최기선, "외국어 음차 표기의 음성적 유사도 비교 알고리즘", 정보과학회 논문지(B), 제26권 제10호, pp.1237-1246, 1999.
[2] 고숙현, 이재성, "문맥을 고려한 유사 외래어 검출 알고리즘의 성능 향상", 한국정보과학회 언어공학연구회 학술발표 논문집, pp.114-121, 2007.
[3] 권순호, 권혁철, "한국어 자모 혼동행렬 기반 유사 외래어 표기 검출 기법", 한국정보처리학회 춘계학술발표대회 논문집, 제17권 1호, pp.433-436, 2010.
[4] 김민정, 권혁철, "언어적, 경험적 제약을 이용한 한국어 문자 인식 후처리 기법", 정보과학회논문지(B), 제24권 제1호, pp.25-31, 1997.
[5] 김지승, 김광현, 이준호, "입말 표기를 이용한 영어 단어 검색", 한국문헌정보학회지, 제39권 제3호, pp.93-103, 2005.
[6] 윤태진, 조환규, "한글 자소정렬을 이용한 온라인 욕설 필터링 시스템", 한국정보과학회 학술발표논문집(C), 제36권 제2호, pp.194-198, 2009.
[7] 이재성, "효과적인 외래어 이형태 생성을 위한 확률 문맥 의존 치환 방법", 한국콘텐츠학회논문지, 제7권 제2호, pp.73-83, 2007.
[8] 정길순, 권윤형, 맹성현, "외래어와 영어처리를 통한 검색 효과 향상", 한국정보과학회 학술발표논문집, 제24권 제2호, pp.189-192, 1997.
[9] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, 'Introduction to Infomation Retrieval', pp.58-59, Cambridge University Press, 2008.
[10] Lawrence Phillips, Hanging on the Metaphone, Computer Language, Vol.7, No.12, pp.39-43, 1990.
[11] S. M. Chaware and S. Rao, "Phonetic Matching through Writing Style", International Conference and Workshop on Emerging Trends in Technology, pp.541-543, 2010.
[12] Victoria J. Hodge and Jim Austin, "An Evaluation of Phonetic Spell Checkers", Technical Report YCS 338, Department of Computer Science of the University of York, 2001.

권 순 호
e-mail : soonhok7@pusan.ac.kr
2009년 부산대학교 정보컴퓨터공학부(학사)
2009년~현 재 부산대학교 대학원 컴퓨터공학과 석사과정
관심분야 : 정보검색, 자연언어처리

권 혁 철
e-mail : hckwon@pusan.ac.kr
1982년 서울대학교 컴퓨터공학과(학사)
1984년 서울대학교 대학원 컴퓨터공학과(공학석사)
1987년 서울대학교 대학원 컴퓨터공학과(공학박사)
1992년~1993년 (미) Stanford 대학교 CSLI 방문 교수
1987년~현 재 부산대학교 정보컴퓨터공학부, 인지과학협동과정 교수
관심분야 : 인간언어공학, 정보검색, 인공지능