효과적인 외래어 이형태 생성을 위한 확률 문맥 의존 치환 방법

A Probabilistic Context Sensitive Rewriting Method for Effective Transliteration Variants Generation

이재성
충북대학교 사범대학 컴퓨터교육과

Jae Sung Lee(jasonl@cbu.ac.kr)

요약

완전 일치 방법을 주로 사용하는 정보 검색 시스템에서 외래어 이형태를 검색할 수 있도록 위해서는 외래어 이형태를 자동 생성하는 전처리나 질의어 확장이 필요하다. 본 연구에서는 하나의 외래어가 입력되면, 이를 근거로 실제 사용될 만한 외래어 이형태들을 효과적으로 생성하기 위한 방법을 제안한다. 혼동 자소를 단순하게 치환하는 방법은 불필요한 이형태를 과도하게 생성하므로, 본 연구에서는 실제 문서에 사용된 외래어 이형태들로부터 혼동 패턴을 학습하고, 이를 확률로 계산하여 생성 순서를 조절하였다. 특히, 혼동 패턴에서 좌우문맥을 고려하고 지역 치환 확률과 전역 치환 확률을 계산하여 조기에 많이 사용하는 이형태를 생성하도록 하였다. KT SET 2.0에서 추출한 이형태 데이터에 대해 실험한 결과, 상위 20개의 생성으로도 평균 80% 이상 찾아내어 이 방법이 매우 효과적임을 보였다.

■ 중심어 : | 외래어 이형태 | 질의어 확장 | 정보 검색 | 혼동 패턴 |

Abstract

An information retrieval system, using exact match, needs preprocessing or query expansion to generate transliteration variants in order to search foreign word transliteration variants in the documents. This paper proposes an effective method to generate other transliteration variants from a given transliteration. Because simple rewriting of confused characters produces too many false variants, the proposed method controls the generation priority by learning confusion patterns from real uses and calculating their probability. Especially, the left and right context of a pattern is considered, and local rewriting probability and global rewriting probability are calculated to produce more probable variants in earlier stage. The experimental result showed that the method was very effective by showing more than 80% recall with top 20 generations for a transliteration variants set collected from KT SET 2.0.

■ keyword : | Transliteration Variants | Query Expansion | Information Retrieval | Confusion Pattern |

I. 서 론

기술의 발달로 생성되는 전문 분야의 용어들이 비영어권 국가에 도입될 때 전문 용어나 고유 명사 등에 대해 음차 표기가 사용되고 있다[1-3]. 하나의 외국어 단어에 대한 음차 표기는 개인차 및 표준 외래어 표기법

* 본 논문은 2005년도 충북대학교 학술연구지원사업의 연구비지원에 의하여 연구되었습니다.

접수번호 : #070122-001
접수일자 : 2007년 01월 22일
심사완료일 : 2007년 02월 07일
교신저자 : 이재성, e-mail : jasonl@cbu.ac.kr

인식 부족 등을 원인으로 매우 다양하게 나타나며 [4][5], 이러한 다양한 음차 표기를 같은 개념으로 인식하는 것은 정보 검색 성능 향상에 주요 요소가 된다 [1][5].

그러나 아직 대부분의 정보검색 시스템이나 데이터베이스 시스템 등에서는 다양한 외래어 음차표기를 하나의 색인어로 처리하지 않고 있으며, 따라서 이형태의 외래어가 질의어로 들어왔을 경우, 완전 일치가 되지 않는 단어는 전혀 검색을 못하는 경우가 많이 있다. 따라서, 이러한 시스템에서 이형태를 찾기 위한 음운 매칭 알고리즘이나 이형태 생성 기법이 필요하다[1][5-7].

본 논문에서는 단일어 정보검색 시스템에서 입력된 '음차표기 외래어'(이하 '외래어'로 약칭함)를 보다 실제 사용할 가능성이 있는 다양한 이형태로 확장하기 위한 방법을 제시한다. 이 방법은 사람들이 혼동하여 쓰는 외래어 이형태에는 일정한 패턴이 있다고 가정하고, 이를 본 논문에서 제안한 "확률 문맥 의존 치환 방법(Probability Context Sensitive Rewriting, 이하 PCSR)"으로 학습시켜 실험함으로써 검증한다.

정보검색 시스템에서 이형태를 찾는 방법으로 1. 외래어 단어와 찾고자 하는 이형태 단어(target word)의 편집 거리(edit distance)나 n-그램을 계산하여 유사 외래어를 찾는 방법[8], 2. 외래어 단어와 이미 문서에 존재하는 이형태 단어를 음성적 유사도에 기반한 코드로 변환하거나 원어 단어(예를 들어 영어)로 복원(back transliteration)하여 대조(match)하는 방법[1][6], 3. 외래어 단어(source word)에서 가능한 이형태 단어를 생성한 후 찾고자 하는 단어(target word)와 일치(match)되는지 비교하는 방법[9-11] 등 크게 세 가지가 있다. 1과 2의 방법은 비교하기 전에 대상 단어(이형태)를 알아내서 편집거리, n-그램수를 계산하거나, 코드 변환 혹은 원어 복원 등을 해야 하므로 대개 시스템내의 비교 모듈이 수정되어야 한다. 그러나 3의 방법의 경우, 전처리 과정으로 대상 단어들을 생성하여 제공하는 것이므로 시스템 내의 비교 모듈은 변경하지 않고 검색을 수행할 수 있는 잇 점이 있다. 따라서 이런 전처리 생성 기법은 주 시스템의 수정 없이 통합하는 것이 가능하다.

그동안 연구되어진 외래어의 이형태 생성 기법 [6][7][9]들은 순서에 대한 고려 없이 단순한 교환 규칙을 사용하였으며, 규칙 역시 수동으로 구축되었다. 따라서 실제 사용될 가능성이 있는 외래어가 생성되기 보다는 너무 많은 불필요한 외래어가 생성되어 실제 시스템에서 전처리기로 사용하기에는 비효율적이었다. 이 논문에서 제안하는 PCSR 기법은 이형태 목록으로부터 규칙을 자동으로 학습하고 실제 사용되는 이형태를 확률 순으로 조정하여 보다 효과적으로 생성할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 외래어 이형태 생성의 관련 연구를 기술하고, 3장에서는 본 논문에서 제안하는 확률 의존 문맥 치환 방법(PCSR)에 대해 설명하며, 이를 위한 효과적인 확률 계산식을 제시한다. 4장에서는 치환 규칙을 실제 사용 예에서 자동으로 추출하는 방법과 경계 문맥과 빈 문자 치환 등의 특별한 경우에 대해서 설명을 한다. 5장에서는 제시된 기술에 대한 이형태 생성 성능을 실험하고 그 결과를 분석하며, 끝으로 6장에서는 연구 내용을 정리하여 결론을 맺는다.

II. 관련 연구

외래어 처리에 대한 연구는 그동안 영어에서 외래어로 표기하는 방법을 중심으로 주로 연구되었다. 이러한 연구로는 수동으로 구축한 규칙에 의한 방법[12], 자동 정렬과 마코프 모델을 이용한 방법[5][13], 신경망을 이용한 방법[14], 결정트리를 이용한 방법[15], 마코프 모델을 확장한 방법[16], 음운패턴을 네트워크로 묶어 변환하는 방법[17], 결정트리 및 메모리기반 학습을 혼합하여 사용한 방법[18] 등 다양하다. 또한, 역으로 외래어에서 영어로 복원하는 방법[1][13][15][19]도 연구되었다. 그러나, 주어진 한글 외래어로부터 자동으로 다양한 이형태의 외래어를 생성하는 연구는 현재까지 조사한 바로는 거의 없었다. 따라서, 유사한 기존 연구들인 "영어에서 외래어로 표기하면서 동시에 다양한 외래어 이형태를 생성할 수 있는 방법들"을 살펴보고, 이 방법

들을 본 논문에서 제안한 방법과 비교한다.

초기의 외래어 이형태 생성은 [9]의 연구에서 찾아볼 수 있다. 이 연구에서는 병렬 트라이 구조 사전을 만들어 두고, 영어 혹은 외래어(한국어)에 대한 엔트리를 검색한 후, 그 병렬 트라이에서 해당되는 외래어나 원어(영어)를 찾아낸다. 사전에는 한 가지 외래어 표기가 들어 있으므로 다양한 외래어 표기를 인식할 수도 없고, 또 생성해 낼 수도 없다. 이런 문제를 해결하기 위해 표준화되지 않은 다양한 외래어 이형태를 처리하여 인식할 수 있도록 혼동 자소 쌍에 대한 치환 테이블을 만들어 두었다. 즉, 혼동되는 자소가 치환 테이블에 존재하는 경우 대응되는 자소를 치환함으로써 이형태가 생성된다. 예를 들어, 'ㅏ'는 'ㅓ', 'ㅐ', 'ㅔ', 'ㅗ'로, 'ㅗ'는 'ㅏ', 'ㅓ', 'ㅗㅗ', 'ㅗㅜ'로 각각 치환된다. 한 예로 '코코아(cocoa)'는 [그림 1]과 같이 124개의 후보를 생성하게 된다. 하지만, 이 방법은 조합폭발로 인해 이형태가 과생성(over generation) 될 수 있으며 별로 잘 쓰이지 않는 이형태 외래어도 많이 생성된다. 또한, 치환 테이블은 새롭게 만들어지는 다양하고 방대한 이형태를 다루기에는 충분치 않다는 문제점이 있다. 또한 치환 테이블 작성도 수작업으로 이루어지므로 새로운 환경에 적용하기위해서는 비용이 많이 든다.

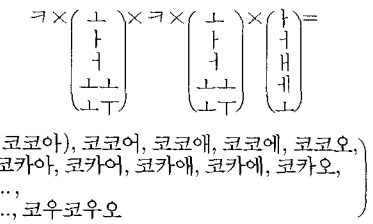

그림 1. 치환테이블에 의해 생성된 이형태

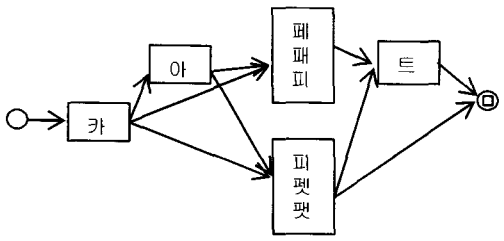

그림 2. 유한 상태 그래프를 이용한 이형태 표현

[10]의 연구에서는 영어 어휘를 올바르게 공부하기 위해, 외래어에 대응되는 영어 원어를 찾을 수 있도록 했다. 이 과정에서 다양한 외래어 이형태를 처리하기 위해 다양한 외래어 표기들을 유한 상태 그래프(finite state graph)로 표기하여 저장하였다. 이 과정은 역으로 하나의 이형태를 찾은 후, 이를 확장하여 가능한 모든 이형태를 생성할 수도 있다. 이형태 내의 각 문자들은 한 개 또는 몇 개씩 그룹지어 유한 상태 그래프의 노드가 된다. [그림 2]는 'carpet'의 외래어 표기에 대한 이형태를 유한 상태 그래프로 표현한 것을 나타낸다. 유한 상태 그래프는 모든 이형태 목록을 조사함으로써 수동으로 구성된다. 이는 이미 알려진 이형태에 대해서는 매우 효과적이나, 알려지지 않은 이형태나 조사하지 않은 단어에 대해서는 전혀 생성을 할 수 없다. 또한, 유한 상태 그래프에서 경로의 수가 많아질 경우에는 치환 테이블 방법과 유사하게 외래어 생성시 조합 폭발의 가능성이 있으나, 이를 조절할 수 있는 우선순위 등의 방법이 없다. 특히, 이 방법은 수동으로 새로운 단어를 찾아 유한 상태 그래프로 표시해야 하므로 쉽게 새로운 환경에 적용하기 어렵고, 규칙 구성비용도 높아진다.

[7]에서는 통계적으로 영어에서 한국어로 음차 표기하는 방법을 사용하였다. 주어진 영어 및 외래어 단어 쌍으로부터 자동으로 발음 단위로 정렬(alignment)을 한 후, 이를 통계적으로 학습하여 외래어 표기를 할 수 있도록 하였다. 이 과정에서 하나의 영어 발음 단위가 확률적으로 여러 한글 발음 단위로 표기 될 수 있기 때문에 이를 확률적으로 조절하여 다양한 음차 표기를 우선순위에 따라 생성해 낼 수 있다. 이 방법은 영어를 한글로 된 외래어로 음차 표기하는데 이용된 것이며, 아직 언어적 차이 등으로 인해 그 정확도가 떨어진다. 또한 이것은 영어로부터 다양한 외래어(한국어)를 생성하는 것이므로 외래어로부터 직접 이형태 외래어를 생성하기 위한 다른 방법이 필요하다.

이외에도 영어에서 일본어로 음차 표기되는 방법 [2][11]이 있는데, 여기에서는 영어 단어를 일정 단위로 분리하여 가능한 모든 일본어 음차 표기(카타카나)를 규칙으로 만들어서 생성한다. 하지만, 이는 일본어에 대한 것일 뿐만 아니라, 수동으로 규칙을 생성해야하고,

우선순위에 의한 생성 방법도 명확하지 않다.

본 논문에서는 하나의 외래어에서 그와 같은 의미의 이형태 외래어를 생성하는 방법에 대해 논의하며, 외래어가 사용된 실제 용례를 근거로 사람들이 혼동하여 표기하기 쉬운 이형태 외래어를 확률 순서로 생성할 수 있는 방법을 제안한다. 이를 위해 문맥 정보를 이용한 확률 치환 규칙(Probabilistic Context Sensitive Rewriting)을 이용한다.

III. 확률 문맥 의존 치환 규칙 (PCSR)

이 장에서는 좌우 문맥 정보에 기반하여 이형태를 치환하여 생성하는 확률 규칙(PCSR)을 제안한다. 먼저 문맥 의존 치환(CSR: Context Sensitive Rewriting) 규칙은 다음 식 (1)과 같이 정의될 수 있다.

$$l\,C_i\,r \rightarrow l\,C_j\,r \tag{1}$$

이 때, l과 r은 터미널이며, C_i와 C_j는 빈(null) 문자도 포함한 치환문자열이다. 여기에서는 발생 가능성이 높은 이형태를 생성하기 위해 문맥 정보를 사용한다. 즉, 왼쪽 문맥 l과 오른쪽 문맥 r을 갖는 C_i가, 같은 좌우 문맥을 갖는 C_j로 대치될 수 있음을 나타낸다. CSR에서 문맥 정보를 조금 완화하여 아래의 (2),(3), (4) 식으로 더 만들 수 있다.

$$l\,C_i \rightarrow l\,C_j \tag{2}$$

$$C_i\,r \rightarrow C_j\,r \tag{3}$$

$$C_i \rightarrow C_j \tag{4}$$

식 (2)는 좌-문맥 의존 치환 (CSR-L), 식 (3)은 우-문맥 의존 치환 (CSR-R), 식 (4)는 문맥 자유 치환 (CFR) 규칙이라 할 수 있다. 문맥 의존 치환 규칙(CSR)은 왼쪽 문맥을 단어의 시작 부분으로, 오른쪽 문맥을 단어의 끝 부분으로 확장하면 유한 상태 그래프[10]의 접근 방법과 유사하게 된다. 또한 문맥 자유 치환 규칙(CFR)은 치환 테이블[9]과 같은 작용을 하여 한 특정 문자 단위에 대응되는 문자 단위로 치환하도록 한다.

생성 가능한 많은 이형태 중에서 실제 사용되는 이형태를 우선 생성하기 위해서는 이런 규칙에 대한 확률 계산이 필요하다. 치환 확률은 크게 두 가지 요소를 고려하여 계산될 수 있다. 첫째는 지역(local) 요소로 하나의 문자열이 어떤 치환 문자열로 바뀌는지를 결정하는 요소이다. 예를 들어 어떤 글자내의 'ㅏ'라는 자소가 'ㅓ'나 'ㅐ'로 바뀔 경우, 어떤 것을 우선적으로 바꿀지를 결정하는 요소이다. 둘째는 전역(global) 요소로서 단어에 있는 전체 글자 중 어느 문자열이 더 확률적으로 높게 치환 문자열로 바뀔 수 있는가를 나타내는 것이다. 예를 들어 '인터내셔날'이라는 단어가 '인터네셔날'과 '인터내셔널'로 바뀔 경우, '내'가 '네'로 바뀌는 것이 먼저인지 '날'이 '널'로 바뀌는 것이 우선인지를 결정하여 더 가능성 있는 단어를 판단하는 요소이다. 본 연구에서는 지역 요소와 전역 요소를 곱하여 치환확률을 계산하였다. 즉, 앞에서 정의한 (1), (2), (3), (4) 식에 대한 확률은 각각 아래 식 (5), (6), (7), (8) 과 같이 계산된다. (단, C_i와 C_j가 같을 경우에는 실제 변환이 일어나지는 않은 것으로 확률 값은 1로 주어진다. C_i와 C_j가 서로 다른 치환 문자열일 경우, 치환될 수 있는 확률은 1보다 작거나 같고, 아래와 같이 계산된다.)

$$P_{lr}(l, C_i, r \rightarrow l, C_j, r) = \frac{cnt(l, C_i, C_j, r)}{1+\sum_y cnt(l, C_i, C_y, r)} \times \frac{cnt(l, C_i, C_j, r)}{1+\sum_{x,y} cnt(l, C_x, C_y, r)} \tag{5}$$

$$P_l(l, C_i \rightarrow l, C_j) = \frac{cnt(l, C_i, C_j)}{1+\sum_y cnt(l, C_i, C_y)} \times \frac{cnt(l, C_i, C_j)}{1+\sum_{x,y} cnt(l, C_x, C_y)} \tag{6}$$

$$P_r(C_i, r \rightarrow C_j, r) = \frac{cnt(C_i, C_j, r)}{1+\sum_y cnt(C_i, C_y, r)} \times \frac{cnt(C_i, C_j, r)}{1+\sum_{x,y} cnt(C_x, C_y, r)} \tag{7}$$

$$P_f(C_i \rightarrow C_j) = \frac{cnt(C_i, C_j)}{1+\sum_y cnt(C_i, C_y)} \times \frac{cnt(C_i, C_j)}{1+\sum_{x,y} cnt(C_x, C_y)} \tag{8}$$

각 식에서 cnt는 치환 패턴의 숫자를 세는 함수이다. 또, 각 식의 첫 번째 인수 (예를 들어 식(8)에서

$\frac{cnt(C_i,C_j)}{1+\sum_y cnt(C_i,C_y)}$)는 전체 이형태 리스트에서 C_i에 대응되는 여러 혼동 문자열 쌍 중 C_i로 바뀌는 확률을 나타내는 지역 요소를 계산한 것이고, 두 번째 인수 (예를 들어 식(8)에서 $\frac{cnt(C_i,C_j)}{1+\sum_{x,y} cnt(C_x,C_y)}$)는 전체 혼동 문자열 쌍 중 C_i가 C_j로 바뀌는 전역 요소를 계산한 것이다. 각 인수의 분모에 1을 더해 준 이유는 데이터가 적을 경우, 너무 큰 확률 값을 갖지 못하도록 유연화하기 위한 것이다. 이형태는 단지 하나의 문자열을 치환함에 의해서도 생성될 수 있기 때문에 상위 생성 단계에서 발생 가능성이 가장 높은 이형태를 생성하기 위해 이 두 가지 요소가 필수적이다.

문맥 의존 치환 규칙과 문맥 자유 치환 규칙을 식 (9)와 같이 4개의 파라미터를 이용하여 하나의 식으로 나타낼 수 있다. 이 식에서 α를 1.0으로 설정하고 β, γ, δ를 0으로 설정하면, 좌우 문맥을 모두 고려한 문맥 의존 규칙을 얻을 수 있다. δ를 1.0으로 설정하고 α, β, γ를 0으로 설정하면, 문맥을 고려하지 않은 문맥 자유 규칙을 얻을 수 있다. 문맥 정보를 이용하면 보다 정확한 이형태 결과를 얻을 수지만, 자료 희귀성 (data sparseness) 문제가 있을 수 있으므로, 파라미터의 값을 적절히 조정함으로써 보다 재현율을 높이고 초기 단계에 보다 발생 가능성이 높은 이형태를 생성할 수 있다.

$$\begin{aligned} &P_m(l,C_i,r\rightarrow l,C_j,r) \\ &=\alpha\times P_{lr}(l,C_i,r\rightarrow l,C_j,r)+\beta\times P_l(l,C_i\rightarrow l,C_j) \\ &\quad+\ \gamma\times P_r(C_i,r\rightarrow C_j,r)+\delta\times P_f(C_i\rightarrow C_j) \end{aligned} \tag{9}$$

단, $\alpha+\beta+\gamma+\delta=1.0$

IV. PCSR 규칙 학습

음차 표기된 이형태 목록을 살펴보면 서로 치환된 문자열을 찾아 치환되는 패턴을 찾아 낼 수 있으므로, 이를 이용하여 PCSR 규칙을 추출하고 그 확률을 계산할 수 있다. 예를 들어, (10) (11)은 'radio'의 등가(equivalent) 외래어 표기들로 'ㅏ'를 'ㅔ이'로 치환하여 다른 변이체로 변환된 것으로 볼 수 있다. 또, (11)은 'raster'의 등가 외래어 표기들로 'ㅏ', 'ㅐ', 'ㅔ'는 서로 치환될 수 있음을 알 수 있다.

라디오 레이디오 (10)

라스터 래스터 레스터 (11)

치환 관계는 이항적(binary)이며, 등가 외래어 표기로부터 가능한 이형태 쌍(pair)을 추출할 수 있다. 예를 들어 (11)의 이형태 쌍은 (12), (13), (14)에서 화살표의 좌변과 같이 나타날 수 있다. 각 식의 우변은 이형태 쌍으로부터 추출한 치환 패턴이다. 변이가 일어나는 문자열(이 예에서는 1개의 자소)을 중심으로 하여 각각 왼쪽과 오른쪽의 자소를 나타냄으로써 좌우 문맥정보를 나타내었고 콜론(:)을 통하여 각 자소를 구분하였다.

라스터 래스터 → ㄹ:ㅏ:ㅅ ㄹ:ㅐ:ㅅ (12)

래스터 레스터 → ㄹ:ㅐ:ㅅ ㄹ:ㅔ:ㅅ (13)

라스터 레스터 → ㄹ:ㅏ:ㅅ ㄹ:ㅔ:ㅅ (14)

이들 치환 문자들은 서로 재귀적(reflexive), 대칭적(symmetric), 전이적(transitive)이며 따라서 원칙적으로 등가(equivalent) 관계에 있다. 재귀적이란 의미는 사실상 치환이 일어나지 않고 그대로 있는 것을 나타낸다. 대칭적이란 의미는 양방향의 치환이 가능하다는 것으로 (12)에서 'ㅏ'가 'ㅐ'로 바뀌는 것을 나타냄과 동시에 역으로 'ㅐ'가 'ㅏ'로 바뀔 수 있음을 나타낸다. 전이적이란 의미는 'ㅏ'가 'ㅐ'로 바뀌고, 'ㅐ'가 'ㅔ'로 바뀌면 'ㅏ'가 'ㅔ'로 바뀔 수 있음을 나타낸다. 즉 (12), (13) 규칙으로부터 (14)를 자동 생성해 낼 수 있음을 나타낸다. 따라서, 전이 관계를 이용하면 저장해야하는 규칙의 수를 줄일 수 있을 뿐만 아니라, 적은 수의 규칙에서 많은 새로운 규칙을 유도해 낼 수 있다. 그러나 전이 관계를 통해 유도된 규칙의 경우, 확률도 새로 유도해서 계산해야 하나, 그에 따른 부정확성이 발생할 수 있다. 본 논문에서는 보다 간단하게 실제적으로 사용할 수 있도록, 전이 관계를 이용한 유도 규칙은 무시하고, 실제 나타난 치환 패턴 규칙만을 고려하여 확률을 계산하였다.

즉, (11)에서 (12), (13) 뿐만 아니라 (14)도 치환 패턴으로 만들어내고, 실제 나타난 횟수를 이용하여 확률을 계산하였다.

치환 관계는 경우에 따라 생성 가능성은 있지만 실제 사용되지 않는 이형태를 생성할 수도 있다. 한 예로 (10)에서 문맥을 고려하지 않을 경우, 'ㅏ'가 'ㅔ이'로 치환될 수 있으므로, 이를 (11)의 '라스터'에 적용시키면 '레이스터'가 생성될 수 있다. 그러나 이 단어는 실제 사용되지 않고 있는 단어이므로 불필요한 생성이 되어 전체 성능을 떨어뜨리는 요인이 될 수 있으므로 이를 염두에 두어야 한다.

좌우 문맥 정보를 고려할 경우, 처음 글자와 마지막 글자는 좌 문맥과 우 문맥이 사실상 없다. 그러나 처음 글자와 마지막 글자는 그 위치에 의해 생성 확률에 더 영향을 줄 수도 있으므로, 이를 고려하여 단어 앞과 뒤에 각각 문맥을 나타내는 기호를 추가하여 규칙 학습에 사용하였다. (15)는 '컷'과 '커트'에서 추출한 치환 패턴이다. 여기에서 '>'는 단어 끝을 나타내는 기호이며 치환 패턴의 문맥으로 사용되었다.

컷 커트 → ㅓ:ㅅ:> ㅓ:ㅌㅡ:> (15)

치환 관계에 빈 문자가 포함될 경우, 학습과 생성시 주의가 필요하다. 빈 문자가 포함되는 경우는 두 가지로 나눌 수 있다. 치환하여 빈 문자로 치환되는 경우(생략 치환)와 빈 문자에서 새 문자로 치환되는 경우(추가 치환)이다. 다음의 (16), (17)은 생략 치환과 추가 치환의 예를 각각 나타내며, '$'는 빈 문자를 나타내는 기호이다.

키이 키 → ㅣ:이:> ㅣ:$:> (16)

키 키이 → ㅣ:$:> ㅣ:이:> (17)

생략 치환의 경우는 이형태 생성 과정에 빈 문자 기호를 일단 생성한 후, 최종 이형태 단어를 출력할 때, 빈 문자 기호를 없애면 된다. 추가 치환의 경우는 좌우 문맥이 맞으면 새로운 글자가 계속 무한히 추가될 가능성도 있기 때문에 이를 방지하기 위한 방법이 필요하다. 특히, 문맥정보를 사용하지 않고 치환할 경우, 생성의 정확성도 떨어질 뿐만 아니라 더 쉽게 무한 루프에 빠질 수 있다. 이를 위해 본 연구에서는 추가 치환은 좌우 문맥을 고려한 경우에만 이루어지도록 했고, 또, 같은 위치의 문맥에서 1번 이상 반복되지 않도록 제한하였다.

전체적인 규칙 학습 과정을 정리하면 다음과 같다. 우선 같은 단어를 음차 표기한 이형태들을 하나의 등가 외래어 표기로 구분한다. 이 등가 외래어 표기로부터 이항적 관계를 찾아내기 위해 이형태 쌍(pair)을 추출한다. 이형태 쌍은 단어들로 되어 있으므로, 이 단어에서 치환이 일어난 문자열들의 패턴을 찾아 좌우 문맥을 포함하여 출력한다. 이 치환 패턴은 수동으로 추출할 수도 있지만, 빠르고 정확하게 처리하기 위해 본 연구에서는 문자 정렬프로그램[20]을 수정하여 이형태 쌍으로부터 자동으로 출력되도록 하였다. 이러한 치환 패턴 목록을 이용하여 식(5), (6), (7), (8), (9)를 기반으로 치환 규칙의 확률을 계산할 수 있다.

V. 실 험

실험 데이터는 KTSET 2.0[21]에서 추출한 외래어 이형태 모음을 사용하였다. 이 모음은 2개 이상의 변이가 있는 단어들만 모은 것이며 전체 등가 외래어 그룹 수는 277개이고, 변이 단어들의 평균 개수는 2.18개이다. 실험을 위해 전체 그룹에서 27개 그룹(약 10%)을 임의로 추출하고 각 그룹에서 다시 임의로 한 단어씩을 추출하여 미학습 데이터(unseen data) 테스트용으로 사용하였고, 나머지 90%를 학습 데이터로 사용하였다. 또, 같은 방법으로, 학습에 사용한 데이터 중에서 27개(전체의 약 10%) 그룹을 임의로 뽑고 각 그룹에서 한 단어를 임의로 추출하여 학습 데이터(seen data) 테스트에 사용하였다. 평가는 재현율, 즉, 테스트 데이터로 선정한 데이터 중 몇 %를 생성 프로그램이 찾아 내는가로 계산하였다. 생성 프로그램은 확률 순위에 따라 단어를 생성하므로 30개까지 만을 생성하여 전체 성능과 각 단계별 변화 과정을 측정하였다.

비교를 위해, 확률 정보와 문맥 정보를 모두 사용하

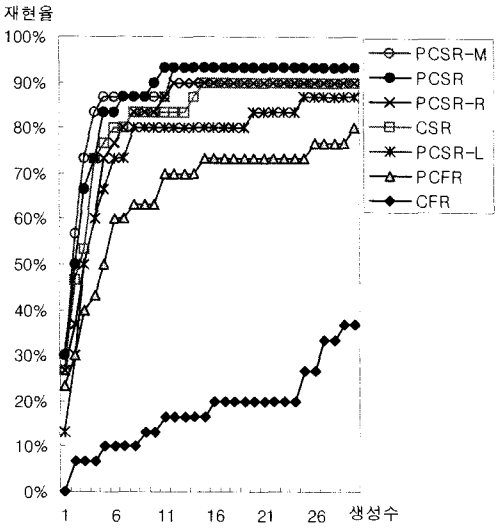

그림 3. 학습 데이터에 대한 재현률

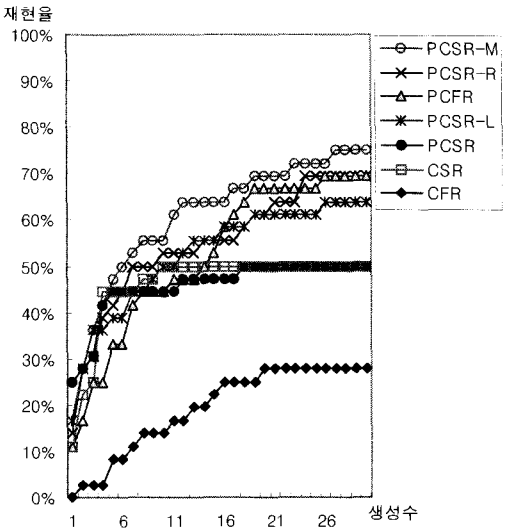

그림 4. 미학습 데이터에 대한 재현률

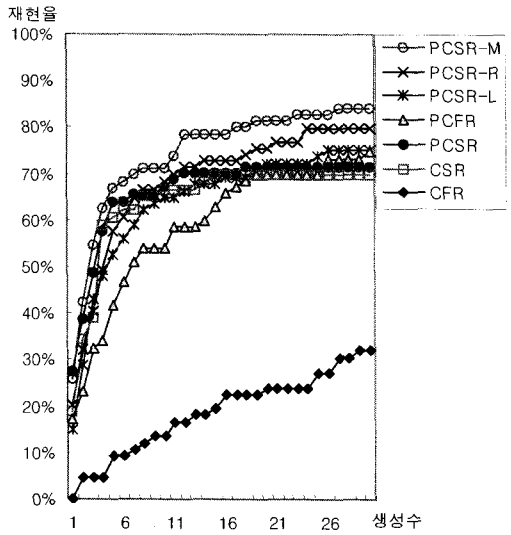

그림 5. 재현률 평균

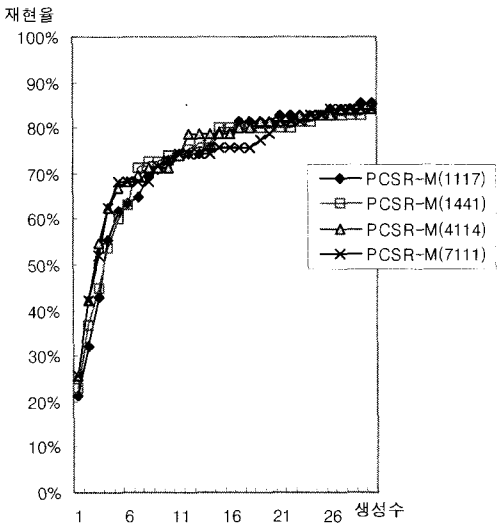

그림 6. 파라미터 값에 따른 PCSR-M 비교

지 않은 방법으로 문맥 자유 치환 방법(CFR: Context Free Rewriting)과, 확률 정보는 사용하지 않고 좌우 1글자 문맥 정보만 사용한 문맥 의존 치환 방법 (CSR: Context Sensitive Rewriting)을 추가하여 다른 기법들과의 성능을 비교하였다. 실험은 제 3장에서 기술된 각 기법을 이용하여 학습 데이터와 미학습 데이터에 대해서 수행하였다. 설명의 편리를 위해 각 기법의 약어를 이용하였다. 즉, 식(5)의 확률적 좌우문맥 의존 치환 기법인 P_{lr}은 PCSR로, 식(6)의 확률적 좌문맥 의존 치환 기법인 P_l은 PCSR-L, 식(7)의 확률적 우문맥 의존 치환 기법인 P_r은 PCSR-R, 식(8)의 확률적 문맥자유 치환 기법인 P_f는 PCFR, 식(9)의 확률적 문맥 의존 치환 기법을 파라미터를 이용하여 하나의 식으로 나타낸 P_m은 PCSR-M으로 나타낸다.

[그림 3]은 학습데이터에 대한 각 기법의 재현율을 나타낸 것이다. 문맥 정보나 확률을 사용하는 방법은

초기에 매우 빠르게 정답을 찾아냈고, 그 이후에는 변화가 매우 완만했다. 반면에 CFR은 처음부터 완만하게 증가했으며, 전체적인 성능도 다른 방법에 비해 떨어졌다. 30개까지 생성했을 때 PCSR-M과 PCSR은 93.3, PCSR-R과 CSR은 90.0, PCSR-L은 86.7, PCFR은 80.0, CFR은 36.7로 나타났다.

[그림 4]는 미학습 데이터에 대한 각 기법의 재현율을 나타낸 것이다. 전체적으로 비교적 완만하게 재현률이 증가했고, 학습데이터에 대한 실험에 비해 성능이 다소 떨어졌다. 문맥정보를 많이 사용하는 CSR이나 PCSR은 초기에 재현율이 높다가 나중에 완만하게 증가했으며, 문맥정보를 사용하지 않는 PCFR은 초기에 재현율이 낮았지만, 나중에까지 지속적으로 증가하였다. CSR은 PCSR과 거의 비슷한 수준으로 재현율을 나타낸 반면, CFR은 PCFR에 비해 매우 낮은 재현율을 보였다. 전체적으로 통합된 PCSR-M이 초기에서 부터 나중까지 가장 성능이 우수하였다. 30개 생성시, 각 방법의 재현율은 PCSR-M이 75.0, PCSR-R과 PCFR이 69.4, PCSR-L은 63.9, PCSR 및 CSR 은 50.0, CFR은 27.8이었다.

학습데이터와 미학습 데이터의 재현율을 평균으로 계산하여 나타낸 것이 [그림 5]이다. 현장에서 사용되는 데이터에 대해 평가를 할 경우, 일반적으로 학습데이터의 성능과 미학습 데이터에 대한 성능의 평균으로 수렴된다고 보면, 각 방법의 성능을 쉽게 비교할 수 있다. 이 결과에서 보면, 확률이나 문맥 정보를 사용한 방법(PCSR-M, PCSR-R, PCSR-L, PCFR, PCSR, CSR)이 그렇지 않은 방법(CFR)보다 훨씬 성능이 뛰어났다. 또, 문맥을 많이 사용한 방법(PCSR, CSR)이 초기에는 빠르게 재현율이 증가하다가 나중에는 문맥을 적게 사용하는 방법(PCSR-R, PCSR-L, PCFR)보다 성능이 떨어졌다. 여기에서도 마찬가지로 PCSR-M 방법이 초기에서부터 거의 최고 성능을 나타내었다. 30개 생성시 각 방법의 성능은 PCSR-M 84.2, PCSR-R 79.7, PCSR-L 75.3, PCFR 74.7, PCSR 71.7, CSR 70.0, CFR 32.2를 각각 나타내었다.

문맥 정보를 많이 사용하는 방법(PCSR, CSR)이 미학습 데이터에 대해 초기에만 좋은 성능을 나타내고, 나중에 성능이 향상되지 않는 이유는 데이터 희귀성 때문이다. 충분히 많은 양의 학습 데이터가 있을 경우, 이 방법들이 우수할 수 있지만, 현실적으로 모든 경우에 대한 충분히 많은 학습데이터를 만들기는 어렵다. 따라서, 데이터 희귀성을 처리할 수 있도록 보완된 PCSR-M 방법이 필요하며, 실험 결과 항상 우수한 결과를 나타냈다.

PCSR-M은 각각의 파라미터를 조정하여 성능을 변화시킬 수 있다. 최적의 파라미터 설정을 위해 4개의 파라미터 값을 0.1단위로 나누어 여러 가지 조합에 대해 측정하였으나, 거의 비슷한 성능을 보였다. [그림 6]은 몇 가지 종류의 파라미터에 대한 PCSR-M의 성능을 보여주는 그래프이다. 예를 들어, PCSR-M(4114)에서 4114는 파라미터 값이 α=0.4, β=0.1, γ=0.1, δ=0.4 임을 나타낸다. 일반적으로 문맥정보가 많이 들어간 항(α)의 가중치를 높이면 초기에 조금 더 재현율이 높고, 반대로 문맥정보가 적게 들어간 항(δ)에 가중치를 많이 두면, 후반에 재현율이 약간 높아졌다. 따라서, 응용분야에 맞게 파라미터를 조절하면 보다 효율적으로 외래어 이형태를 생성할 수 있을 것이다. PCSR-M(4114)의 경우, 5개 생성시 약 67.0%, 10개 생성시 71.1%, 20개 생성시 81.4%, 30개 생성시 84.2%의 성능을 보여 적은 수의 생성으로도 많은 외래어 이형태를 찾아낼 수 있음을 알 수 있다.

표 1. 미학습 데이터 '인터내셔날(international)'에 대해 생성된 상위 10개의 이형태 목록

	CFR	CSR	PCFR	PCSR	PCSR-M
1	인타내셔날	인터내셔날드	**인터네셔널**	**인터내셔널**	**인터내셔널**
2	인태내셔날	**인터내셔널**	**인터내셔널**	인터내셔날드	인턴내셔날
3	인테내셔날	인터내셔널드	엔터내셔날	**인터내쇼날**	**인터내쇼날**
4	인텨내셔날	**인터내쇼날**	인타내셔날	인턴내셔날	이신터내셔날
5	인토내셔날	인터내쇼날드	인터내셔낼	이신터내셔날	인터내셔날드
6	인투내셔날	인터내쇼널	인터내셔나	앤터내셔날	엔터내셔날
7	인트내셔날	인터내쇼널드	인터나셔날	이신터내셔널	**인터네셔날**
8	인테르내셔날	엔터내셔날	인토내셔날	인턴내셔널	인타내셔날
9	인더내셔날	앤터내셔날드	인터내져날	인터내쇼날	인처내셔날
10	인다내셔날	앤터내셔널	인테내셔날	인터내셔널드	인터나셔날

[표 1]은 미학습 데이터인 '인터내셔날(international)'에 대하여 각 방법을 적용하여 생성된 결과 데이터와

정답 데이터(밑줄 친 굵은 글자로 표시)를 나타낸 것이다. '인터내셔날'의 이형태 정답으로는 '인터내셔널', '인터내쇼날', '인터네셔날'이다. CFR은 '인터내셔날'의 둘째 음절 '터'에 처음으로 치환규칙을 적용한 결과를 보여준다. 즉,'ㅓ'가 'ㅏ', 'ㅐ', 'ㅔ', 'ㅕ' 등으로 치환되는 규칙을 사용하여 규칙의 순서대로 생성된 것이다. 따라서, 생성하는 각각의 경우에 따라, 규칙의 적용순서가 성능에 영향을 미칠 수도 있지만, 모든 경우에 따라 규칙 적용순서를 동적으로 변경할 수 있는 구조가 없으므로 전체적인 평균 성능은 떨어진다. 이 경우에도 가능성이 있는 이형태를 생성했지만, 실제 사용한 이형태(정답)를 10개 내에서는 하나도 찾지 못했다. CSR의 경우, 좌우 문맥을 고려하여 치환할 경우, 더 가능성 있는 이형태를 생성하여 2번째와 4번째 정답 이형태를 찾아내었다.

PCFR의 경우, 가능성 있는 이형태를 확률로 계산하여 높은 확률의 이형태를 초기에 생성함으로써 CFR에 비해 성능이 월등히 높아졌다. 즉, 1, 2번 생성시 정답 이형태를 생성하였다. PCSR의 경우도 정답 가능성이 있는 이형태들을 확률적으로 계산하여 우선 생성함으로써 CSR에 비해 더 빠르게 정답을 찾아내었다. 실제로 CSR에서 생성되었던 이형태가 확률 순서에 의해 재정렬되어 재현율이 높아진 것을 볼 수 있다. 이 예에서는 PCFR이 1, 2 번에서 정답을 미리 생성하여 PCSR보다 조금 빠르게 재현율이 높아졌다. 하지만, 이와는 다르게 [그림 5]에서 보면 PCSR이 PCFR보다는 평균적으로 초기에 훨씬 더 성능이 좋다. PCSR-M은 실제 정답 이형태 3개를 1, 3, 7번에서 모두 생성하여 다른 모든 방법에 비해 재현율이 비교적 빨리 올라갔고, 전체적인 성능도 우수하였다.

빈 문자에 대한 추가 치환 규칙이 이형태 생성에 정확성을 떨어뜨리고 있다. 예를 들어 맥도널'와 '맥도널드'에서 '드'가 추가 치환 규칙으로 학습되었는데, 이 규칙이 '인터내셔날'의 끝 문맥에 적용되어'인터내셔날드'를 생성했다. 또, 표에 나타난 단어들 중 일부 단어들이 비록 정답은 아니더라도 문서에서 사용될 가능성이 있는 단어들이다. 표1의 PCSR-M의 결과에 대해 실제 인터넷[22, 23]에서 같은 의미로 쓰인 단어가 사용되는지 검색하였다. 검색 첫 페이지 문서에서만 정답 외에 추가로 '인타내셔날', '인처내셔날', '인터나셔날' 등이 실제 사용되고 있음을 확인했다.

VI. 결 론

본 논문에서 한국어 음차표기에 따른 여러 발생 가능한 이형태들을 생성하기 위한 확률 문맥 의존 치환 기법(PCSR)을 제안하였다. 실제 사용되고 있는 문서에서 추출한 이형태 리스트로부터 치환 규칙을 자동적으로 학습하고, 이를 이용하여 실험한 결과, 문맥 정보와 지역 및 전역 확률 정보를 사용하면 실제로 사용되는 이형태를 초기에 생성해 낼 수 있음을 보였다. 실험 결과 가장 우수한 성능을 나타내는 PCSR-M 방법은 5개 생성시 약 67%, 10개 생성시 약 71%, 20개 생성시 약 81%, 30개 생성시 약 84%를 생성하여 실제 사용되는 이형태를 효과적으로 찾아내었다. 이는 기존의 치환 테이블 방법[9]이나 유한 상태 그래프 방법[10]의 문제를 해결한 것으로, 보다 실제 사용될 가능성이 있는 이형태를 초기에 생성해 냄으로써 효과적인 검색을 가능하게하고, 또, 미등록 외래어에 대해서도 처리도 가능하다. 이 방법은 정보검색이나 데이터베이스 시스템 등의 전처리 프로그램에서 질의어 확장용으로 사용되거나 유사 외래어 용어 추출용 프로그램 등으로 사용될 수 있을 것이다.

참 고 문 헌

[1] K. S. Jeong, H. Myaeng, J. S. Lee, and K. Choi, "Automatic Identification and Back-Transliteration of Foreign Words for Information Retrieval," Information Processing and Management, Vol.35, No.4, pp.523-540, 1999.

[2] H. Yuichi and Y. Issei, "A Method for Transliterating the Spelling of English Words into Katakana Using the Rewrite Rules,"

Natural Language Processing, Vol.79, No.1, pp.1-8, 1990.

[3] W. Gao, K. Wong, and W. Lam, "Improving Transliteration with Precise Alignment of Phoneme Chunks and Using Contextual Features," In proceedings of Asia Information Retrieval Symposium, pp.63-70, 2004.

[4] 이희승, 안병주, *한글 맞춤법 강의-고친판*, 신구문화사, 1994.

[5] J. S. Lee and K. Choi, "English to Korean Statistical Transliteration for Information Retrieval," Computer Processing of Oriental Languages, Vol.12, No.1, pp.17-37, 1998.

[6] 강병주, 이재성, 최기선, "외국어 음차 표기의 음성적 유사도 비교 알고리즘", 정보과학회 논문지 (B), 제26권, 제10호, pp.1237-1246, 1999.

[7] J. S. Lee and K. Choi, "A Statistical Method to Generate Various Foreign Word Transliterations in Multilingual Information Retrieval System," In Proceedings of the 2nd International Workshop on Information Retrieval with Asian Languages-1997, pp.123-128, Oct. 1997.

[8] K. Jeong, Y. Kwon, and S. H. Myaeng, "Construction of Equivalence Classes of Foreign Words through Automatic Identification and Extraction," Natural Language Processing Pacific Rim Symposium, pp.335-340, 1997.

[9] SERI/KIST, *지능형 정보처리기의 개발에 관한 연구*, 제1차년도 최종보고서, 과학기술처, 1995.

[10] S. M. Cheon, *Construction of English Loanwords Contents for the Development of Educational Tools: a Step Towards the Prosperity of CALL Courseware*, Ph. D dissertation. Hankuk University of Foreign Studies, 2005.

[11] M. Mettler, "TRW Japanese Fast Data Finder," TIPSTER Text Program Phase I Proc., pp. 113-116, Sep. 1993.

[12] 김병혜, *영어단어의 알파벳표기로부터 한글표기로의 자동변환*, 서강대학교 공공정책대학원 석사학위논문, 1991.

[13] 이재성, *다국어 정보검색을 위한 영한 음차 표기 및 복원 모델*, 한국과학기술원 박사학위논문, 1999.

[14] 김정재, 이재성, 최기선, "신경망을 이용한 발음 단위 기반 자동 영한 음차 표기 모델", 한국 인지과학회 춘계 학술대회, pp.147-252, 1999.

[15] 강병주, *한국어 정보검색에서 외래어와 영어로 인한 단어 불일치문제의 해결*, 한국과학기술원 박사학위논문, 2001.

[16] S. Y. Jung, S. L. Hong, and E. Pack, "An English to Korean Transliteration Model of Extended Markov Window," In Proceedings of 18th International Conference on Computational Linguistics, pp.383-389, 2000.

[17] 강인호, 김길창, "복수 음운 정보를 이용한 영한 음차표기", 제11회 한글 및 한국어 정보처리 학술발표 논문집, pp.50-54, 1999.

[18] 오종훈, 최기선, "자소 및 음소 정보를 이용한 영어-한국어 음차표기 모델", 정보과학회 논문지: 소프트웨어 및 응용, 제32권, 제4호, pp.312-326, 2005.

[19] 강병주, 최기선, "한-영 자동 음차 복원", 제11회 한글 및 한국어 정보처리 학술 발표 논문집, pp.63-69, 1999.

[20] W. A. Gale and K. W. Church, "A Program for Aligning Sentences in Bilingual Corpora," In Using Large Corpora (ed. Armstrong, S.) The MIT Press, Cambridge, Massachusettes, London, England, pp.75-102, 1994.

[21] 김재군, 김영환, 김성혁, "한국어 정보검색연구를 위한 시험용 데이터 모음(KTSET) 개발", 제6회 한글 및 한국어 정보처리 학술 발표 논문집, pp. 378-385, 1994.

[22] http://www.naver.com
[23] http://www.google.co.kr

저 자 소 개

이 재 성(Jae Sung Lee) 정회원

- 1983년 2월 : 서울대 컴퓨터공학과(학사)
- 1985년 2월 : KAIST 전산학과(석사)
- 1999년 2월 : KAIST 전산학과(박사)
- 1985년 ~ 1988년 : 큐닉스컴퓨터(주) 과장
- 1988년 ~ 1993년 : 마이크로소프트 차장
- 1999년 ~ 2000년 : 전자통신연구원 팀장
- 2005년 ~ 2006년 : 미국 아리조나 대학 방문교수
- 2000년 9월 ~ 현재 : 충북대 컴퓨터교육과 부교수

<관심분야> : 정보검색, 자연언어 처리, 컴퓨터교육