글자 및 발음 기반 영-한 음차표기 모델*

오종훈°, 배선미, 최기선
한국과학기술원 전산학과/전문용어언어공학연구센터/언어자원은행
{rovellia°, sbae, kschoi}@world.kaist.ac.kr

An English-to-Korean Transliteration Model based on Character and Pronunciation

Jong-Hoon Oh°, Sun-Mee BAE, Key-Sun Choi
Department of EECS
Korea Advanced Institute of Science and Technology/KORTERM/BORA

요 약

음차표기란 외국어의 발음을 자국어로 표기하는 것으로 정의된다. 영-한 자동 음차표기 방법에는 직접방식, 피봇방식, 혼합방식이 있다. 기존의 영-한 음차표기 연구들은 직접방식에 기반한 연구들이 대부분이었다. 하지만, 음차표기는 직접방식에서 사용하는 단순한 자소 대 자소변환 작업이라기보다는 자소의 음성적 변환 작업이라고 할 수 있다. 따라서 자소뿐만 아니라 음소 등 음성적 정보가 매우 중요하다. 본 논문에서는 이러한 특성을 이용하여 자소 정보뿐만 아니라 음소 정보를 이용한 음차표기 기법을 제안한다. 주어진 자소와 음소 및 자소와 음소의 문맥정보를 이용하여 한국어 음차표기를 생성하는 본 논문의 기법은 약 60%의 단어정확도를 나타내었다

1. 서론

음차표기란 외국어의 발음을 자국어로 표기하는 것으로 정의된다[1]. 전문분야문서와 같이 외국어원의 용어가 많이 포함된 문서를 처리할 때 음차표기를 올바르게 파악하고 효과적으로 처리하는 것은 매우 중요하다. 이는 전문용어의 많은 부분이 외국어에 기원을 두고 있고, 음차표기되거나 원어 그대로 사용되는 경우가 많기 때문이다. 전문분야 문서에서 음차표기와 원어의 혼재는 정보검색과 같은 자연언어응용에서 단어불일치 문제[1]를 야기한다. 음차표기로 인한 단어 불일치 문제를 해결하기 위한 방법으로 영-한 음차표기 대역쌍을 포함하는 사전을 이용하는 방법이 있다. 하지만 음차표기되는 용어들이 대부분 사전에 등재되지 않는 경우가 많기 때문에 이를 처리하기 위한 방법으로 자동 음차표기에 대한 연구가 활발히 진행되어 왔다[1,2,3,4,5,6].

영-한 음차표기에서 영어단어는 크게 입말 표기 (spoken word transliteration)와 글말표기 (written word transliteration)의 방식으로 음차표기된다. 글말표기는 영어단어의 글자를 이용하여 음차표기하는 방법이며, 입말표기는 영어단어의 발음을 이용하여 음차표기하는 방법이다. 이재성[1]은 이러한 입말표기와 글말표기를 처리하기 위한 음차표기 모델로 직접방식, 피봇방식 그리고 이들을 혼합한 혼합방식을 제안하였다.

직접방식은 영어단어를 한국어 음차표기로 직접 변환하는 방법으로 글말표기를 효과적으로 처리하기 위한 모델이다. 직접방식은 발음지식이 필요하지 않기 때문에, 발음지식을 필요로 하는 피봇방식에 비해 비교적 간단하게 음차표기를 생성할 수 있다는 장점이 있다. 즉 영-한 자소 변환 규칙만으로 음차표기를 할 수 있는 장점이 있다. 이러한 특성으로 인하여 기존의 영-한 음차표기 연구[1,2,3,4,5,6,7]에서는 간단한 직접방식에 기반한 연구들이 주로 이루어져 왔다. 이들 연구들로는 확률기반[1,4,5], 결정트리기반[2,3], 음차표기 네트워크 기반 방법[6], 문맥정보를 이용한 최대엔트로피 모델[7]이 있다. 하지만 직접방식은 글말표기에 초점을 맞춘 방법이기 때문에 입말표기되는 음차표기를 제대로 생성하지 못하는 단점이 있다.

반대로, 피봇방식은 영어단어의 발음을 이용하여 한국어 음차표기를 생성하는 방법으로 입말표기를 효과적으로 처리하는 모델이다. 피봇방식에서는 주어진 영어단어에 대한 발음을 파악하는 과정과 발음을 한국어 음차표기로 변환하는 과정으로 이루어진다. 피봇방식은 직접 방식에 비하여 발음지식이 필요하다는 점과 자소 정보를 고려하지 않는 음차표기를 수행한다는 단점이 있다. 또한 두 단계 과정을 거치기 때문에 첫 번째 단계의 오류가 두 번째 단계로 파급된다.

혼합방식은 직접방식과 피봇방식의 결과 중 올바른 결과로 판단되는 결과를 생성하는 방식이다. 따라서 혼합방식은 직접방식과 피봇방식에 대한 결과를 모두 생성한 후, 두 결과 중에서 가장 적합한 결과를 선택하는 방법이다. 혼합방식이 직접방식과 피봇방식을 따로 사용한 경우보다 좋은 결과를 나타내지만, 직접방식과 피봇방식 모두에서 올바른 음차표기를 생성하지 못하는 경우에는 올바른 결과를 생성할 수 없다. 예를 들어, */M AE G N AH S AY T/*으로 발음되는 *magnesite*의 올바른 음차표기는 '마그네사이트'이다. 하지만 글말표기에 기반한 직접방식에 의해 '마그네시테'라고 음차표기될 수 있으며, 입말표기에 기반한 피봇방식에 의해 '매그너사이트'라고 음차표기될 수 있기 때문에 기존의 혼합방식으로는 올바른 음차표기를 생성하기 어렵다.

본 논문에서는 기존의 직접방식, 피봇방식, 혼합방식의 문제점을 해결하기 위한 방법으로 새로운 혼합방식을 이용한 음차표기 방법을 제안한다. 본 논문의 기법은 기존의 방식과 다음과 같은 차이점이 있다.

첫째, 기존의 혼합방식은 직접방식의 결과와 피봇방식의 결과 중 하나를 선택하는 방식인데 반하여, 본 논문의 기법은 직접방식의 규칙과 피봇방식의 규칙을 통합적으로 이용하여 음차표기를 생성한다. 따라서 기존의 혼합방식과는 달리 직접방식 또는 피봇방식에 의존적인 방법이 아니다. 즉 기존의 직접방식 및 피봇방식이 자소정보 또는 음소정보만을 사용하였으며, 이를 이용하는 기존의 혼합방식은 자소 또는 음소만을 이용한 음차표기 결과를 생성한다. 하지만 본 논문의 기법은 음소정보뿐만 아니라 자소정보에 기반한 규칙을 통하여 한국어 음차표기를 생성한다. 따라서 본 논문에서 제안하는 기법은 입말표기 및 글말표기를 통합적으로 처리할 수 있는 모델이라 할 수 있다. 둘째, 기존의 혼합방식에서는 피봇방식의 결과를 이용하기 때문에 직접방식의 음차표기와 피봇방식의 음차표기를 모두 수행하여야 한다. 이 때, 피봇방식에서는 사전에 기반하여 영어발

*본 논문은 과학기술부, 과학재단, 한국과학기술원 BK21 정보기술사업단의 지원에 의해 이루어짐.

[1] 단어 불일치 문제란 같은 의미의 단어가 다른 형태로 나타날 경우 다른 용어로 취급하는 문제를 지칭한다.

음을 생성하고 생성된 발음을 표준 외래어 표기법[8]에 기반하여 음차표기를 생성하였다. 본 논문의 기법은 기존의 피봇방식과는 달리 발음으로 한국어 음차표기를 생성할 때, 기계학습을 이용하여 규칙을 자동으로 학습한다. 따라서 본 논문의 기법은 기존의 방식들에서 나타난 한계점을 보완하고 해결함으로써, 효과적으로 음차표기를 수행하는 모델이라 할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 자소 및 음소정보를 이용한 음차표기 방법에 대하여 기술한다. 3장에서는 실험 및 결과를 기술하고 4장에서는 결론을 맺는다.

2. 글자 및 발음 기반 영-한 음차표기 모델

2.1 시스템 구조도

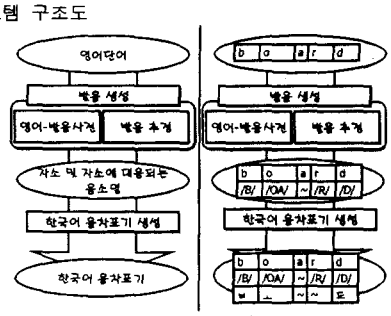

그림 1. 시스템 구조도

그림 1은 글자 및 발음 기반 영-한 음차표기 모델을 도식화한 것이다. 모델은 크게 발음 생성과 한국어 음차표기 생성의 두 단계로 구성된다. 발음 생성 단계에서는 주어진 영어 단어에 대한 발음을 생성한다. 발음 생성은 각 자소에 대응하는 음소를 파악하는 과정이다. 예를 들어, 영어 단어 *board*의 자소 *b, o, a, r, d*에 대하여 음소 *'/B/', '/OA/', '~'*[2], *'/R/', '/D/'*를 생성한다. 발음 생성 단계에서 발음은 발음사전을 이용하는 방법과 사전 미등록어에 대한 발음을 추정하는 두 가지 방법으로 생성된다. 두 번째 단계는 발음에 대한 음차표기 생성 단계이다. 음차표기 생성 단계에서는 발음 생성단계의 결과를 이용하여 한국어 음차표기를 생성한다. 예를 들어, 영어 단어 *board*와 *'/B/', '/OA/', '~', '/R/', '/D/'*에 대하여 각 음소와 자소에 대응하는 한국어 자소를 *'b-/B/-ㅂ', 'o-/OA/-오', 'a-~-', 'r-/R/-~', 'd-/D/-드'*와 같이 생성하고, 이를 통하여 한국어 음차표기 '보드'를 생성한다.

2.2 발음 생성 및 음차표기 생성

2.2.1. 발음 생성

발음 생성과정은 영어자소에 대응되는 음소를 할당하는 과정이다. 따라서 발음 생성의 결과는 각 자소와 각 자소에 대응되는 음소의 열로서 표현된다. 본 논문에서는 발음 생성의 결과를 *GP*로 표현한다. *E*를 영어단어, *P*를 *E*의 발음이라고 정의하면, *E*와 *P*를 $E=e_1,e_2,...,e_m$와 $P=p_1,p_2,...,p_n$로 정의할 수 있다. 여기에서 e_i는 *i*번째 영어자소를 나태내고 p_i는 e_i에 대응하는 음소를 나타낸다. 그러면 발음 생성의 결과 *GP*는 식 (1)과 같이 표현된다.

$$GP = gp_1, gp_2, ..., gp_n \quad (1)$$
$$where, gp_i = (e_i, p_i) \in Q \subset G \times \Phi^*, e_i \in G, p_i \in \Phi^*$$

여기에서 *G*는 영어자소의 집합을 나타내고, *Φ*는 음소의 집합을 나타낸다. 따라서 *Q*는 하나의 영어자소와 하나 이상의 음소로 구성할 수 있는 가능한 모든 <영어자소-음소> 쌍을 포함하는 집합을 나타낸다. gp_i는 집합 *Q*의 원소 중 영어 자소 e_i와 대응되는 음소 p_i로 구성된 것이다. 예를 들어 영어단어 *butyl*과 발음 *'/B Y UW T AH L/'*은 *GP=(b,/B/),(u, /Y UW/), (t,/T/), (y, /AH/), (l, /L/)*와 같이 표현할 수 있다.

이러한 발음 생성 결과를 얻기 위하여, 본 논문에서는 발음 사전 검색 방법과 발음 추정 방법을 사용한다. 발음 사전은 주어진 단어에 대하여 정확한 발음이 수록된 사전이다. 따라서 주어진 영어단어에 대한 발음을 생성할 경우 발음 사전을 우선적으로 검색한다. 사용한 발음 사전은 *CMU pronouncing dictionary*로 약 120,000개 영어단어에 대한 발음을 포함하고 있다. 하지만 발음사전이 모든 영어 단어에 대한 발음을 수록하고 있지 않기 때문에, 발음 사전에 등재되지 않은 단어에 대해서는 발음을 추정해야 한다. 본 논문에서는 발음 추정 문제를 음소할당문제로 변환하여 해결한다. 음소할당문제는 영어자소 e_i와 e_i의 문맥정보를 통하여 집합 *Q*에서 가장 적합한 gp_i를 생성하는 문제로 정의된다. 음소할당함수를 δ_p라고 하면 자소 e_i에 대한 음소할당은 다음과 같이 표현된다.

$$\delta_p(e_i, context(e_i)) = gp_i \quad (2)$$

식 (1)과 (2)에 의하여, 발음추정에 의한 발음 생성은 식 (3)과 같이 표현될 수 있다.

$$GP = \delta_p(e_1, context(e_1)), ..., \delta_p(e_n, context(e_n)) \quad (3)$$

음소할당함수 δ_p의 학습을 위하여 메모리기반 학습[9]과 결정트리 기법[10]을 사용하였다. 학습데이터는 *CMU* 사전의 영어-발음에 대한 영어자소-음소간 대응된 데이터를 사용하였으며, 학습을 위한 특성자질(feature)은 영어자소의 문맥정보를 사용하였다. 본 논문에서는 문맥의 크기를 3으로 한정하였다. 예를 들어, *board*에 대한 발음추정에서 영어자소 *b*는 *b*가 생성하는 모든 가능한 음소집합 중에서 문맥정보 $'L_1=\$', 'L_2=\$', 'L_3=\$', 'R_1=o', 'R_2=a', 'R_3=r'$[3]에 의하여 가장 적합한 음소 *'/B/'*를 음소할당함수 δ_p에 의해 생성한다. 영어자소 *'o', 'a', 'r', 'd'*도 마찬가지로 *'/OA/', '~', '/R/', '/D/'*를 각각 생성한다.

2.2.2 한국어 음차표기 생성

한국어 음차표기 생성작업은 주어진 영어자소와 음소정보를 이용하여 한국어 음차표기를 생성하는 작업으로 정의된다. 본 논문에서는 음차표기 생성 결과를 *GPK*로 표현한다. *E*를 영어단어, *P*를 *E*의 발음 그리고 *K*를 *E*와 *P*에 대응하는 한국어 음차표기라고 정의하면, *E, P, K*는 각각 $E=e_1,e_2,...,e_m$와 $P=p_1,p_2,...,p_n$, $K=k_1,k_2,...,k_n$로 정의될 수 있다. 여기에서 e_i는 *i*번째 영어자소를 나타내고 p_i는 e_i와 대응되는 음소를 그리고 k_i는 e_i, p_i와 대응되는 한글자소를 각각 나타낸다. 그러면 음차표기 생성 결과 *GPK*는 식 (4)와 같이 표현될 수 있다.

$$GPK = gpk_1, gpk_2, ..., gpk_n$$
$$where, gpk_i = (e_i, p_i, k_i) \in R \subset G \times \Phi^* \times KG^* \quad (4)$$
$$e_i \in G, p_i \in \Phi^*, k_i \in KG^*$$

여기에서 *G*는 영어자소의 집합을 나타내고, *Φ*는 음소의 집합을 그리고 *KG*는 한국어 자소의 집합을 나타낸다. 따라서 *R*은 하나의 영어자소, 하나 이상의 음소, 하나 이상의 한국어 자소로 구성할 수 있는 가능한 모든 <영어자소-음소-한국어자소>을 포함하는 집합을 나타낸다. gpk_i는 집합 *R*의 원소 중 영어자소 e_i와 음소 p_i에 대응되는 한국어 자소 k_i로 구성된 것이다. 예를 들어, 영어단어 *butyl*, 발음 *'/B Y UW T AH L/'*, 한국어 '부틸'에 대하여, *GPK=(b,/B/, ㅂ),(u, /Y UW/, 우), (t,/T/, ㅌ), (y, /AH/, ㅣ), (l, /L/, ㄹ)*와 같이 표현할 수 있다.

본 논문에서는 음차표기 생성 문제를 한글자소 할당문제로 변환하여 해결한다. 한글자소 할당문제는 영어자소 e_i와 음소 p_i 그리고 이들의 문맥정보를 이용하여 집합 *R*에서 가장 적합한

[2] 본 논문에서는 '~'를 묵음(silence)으로 정의하고, 하나의 음소로 취급한다.

[3] 본 논문에서는 문맥정보를 L_i, R_i로 표현한다. L_i는 *i*번째 왼쪽 문맥 정보를, R_i는 *i*번째 오른쪽 문맥정보를 각각 나타낸다.

gpk_i를 생성하는 문제로 정의된다. 한글자소 할당함수를 δ_k라고 하면 $gp_i=(e_i,p_j)$에 대한 한글자소 할당은 식 (5)와 같이 표현된다. 식 (4)와 (5)에 의하여, 한글자소 할당은 식 (6)과 같이 표현될 수 있다.

$$\delta_k(gp_i, context(gp_i)) = gpk_i \quad (5)$$

$$GPK = \delta_k(gp_1, context(gp_1)),...,\delta_k(gp_n, context(gp_n)) \quad (6)$$

한글자소 할당함수 δ_k는 발음추정함수 δ_p와 마찬가지로 메모리기반 학습방법과 결정트리기법을 이용하여 학습한다. 학습데이터는 영어자소-음소-한글자소 정렬 데이터를 사용한다. 한글자소 할당함수 δ_k의 학습을 위하여 영어자소, 음소, 일반화된 영어자소, 일반화된 음소를 사용하였다. 예를 들어, 영어단어 *board*와 발음 *'/B OA R D/'*에 대한 음차표기 생성에서 영어자소 *b*와 자소에 대응되는 음소 *'/B/'*가 생성하는 모든 한글자소 집합에 대하여, 한글자소 할당함수 δ_k는 문맥정보 $L_1, L_2, L_3, R_1, R_2, R_3$를 이용하여 가장 적합한 한글자소 'ㅂ'을 생성한다. 마찬가지 방법으로 'ㅗ', '~', '~', 'ㄷ'가 생성되어 영어 단어 *board*와 발음 *'/B OA R D/'*에 대하여 음차표기 '보드'를 생성한다.

3. 실험

3.1 실험환경

본 논문에서는 실험 및 평가를 위하여 두 가지 실험집합을 사용하였다. *Test Set I*[1]은 1,650개 영어-한국어 음차표기 쌍으로 구성된 실험집합으로 기존의 여러 연구에서 평가를 위한 실험집합으로 사용되었다[1,2,3,4,5,6]. 본 논문에서는 [1,2,3,4,5,6]과의 비교평가를 위하여 *Test Set I* 실험집합을 사용하였다. *Test Set I* 중 1,500쌍은 학습데이터로 사용하고, 150쌍은 시험데이터로 사용하였다. *Test set II*[2,3,11]은 7,185개 영어-한국어 음차표기 쌍으로 구성되어 있으며, 6,185쌍은 학습데이터로 1,000쌍은 시험데이터로 사용하였다. *Test set II*는 [2,3,6]에서 사용한 실험집합으로 본 논문의 기법과 [2,3,6]의 성능평가를 위하여 사용한다.

평가 방법은 음차표기 평가 방법으로 널리 사용되는 단어 정확도(W.A.)와 글자 정확도(C.A.)를 이용한다. 단어 정확도와 글자 정확도는 식 (7)과 같이 표현된다.

$$W.A. = \frac{\#of\ correct\ words}{\#of\ generated\ words}, C.A. = \frac{L-(i+d+s)}{L} \quad (7)$$

여기에서 L은 원문자열의 길이를 나타내며, i,d,s는 각각 원문자열에서 목표문자열로 변환하기 위해 필요한 삽입, 삭제, 치환의 개수를 나타낸다. 만약 $L < (i+d+s)$이면 *C.A.*는 0으로 판단한다[12].

3.2 실험결과

표 1. 영-한 음차표기 실험결과

방법	*TestSet I*		*TestSet II*	
	C.A.	*W.A.*	*C.A.*	*W.A.*
[1,5]	69.3%	40.7%	N/A	N/A
[4]	79.0%	35.1%	N/A	N/A
[2,3]	78.1%	37.6%	81.8%	48.7%
[6]	86.5%	55.3%	89.05%	57.2%
Dtree	89.35%	57.33%	89.56%	58.4%
MBL	89.51%	60.67%	89.29%	59.9%

표 1은 *Test set I*과 *Test set II*에 대한 기존연구와의 성능비교 실험 결과를 각각 나타낸다. 실험 결과에서 본 논문의 기법은 결정트리를 이용한 방법(DTree)과 메모리학습을 이용한 방법(MBL) 모두에서 기존의 방법보다 높은 성능을 나타낸다. 또한 *Test set I*에 대해서는 약 10%~70%의 W.A.의 성능향상을 나타내었으며, *Test set II*에 대해서는 약 5%~ 23%의 W.A.의 성능향상을 나타내었다. 그리고 본 논문의 기법에서 메모리 학습방법이 결정트리 방법보다 좋은 성능을 나타낸다. 이는 결정트리 기법이 각 특성자질에 의한 분기 방법인데 비해, 메모리기반 방법은 전체 특성자질 간의 유사도를 비교하여 결과를 생성하기 때문으로 분석된다. 즉, 음차표기가 각 특성자질인 음소 또는 자소만의 변환 작업이 아니라 음소 및 자소의 통합적인 변환 작업이라는 측면에서 전체 특성자질을 비교하는 메모리기반 학습방법이 음차표기에 보다 유용함을 나타낸다. 실험결과는 직접방식에 기반한 기존의 자소 변환 방법보다 본 논문에서 제안한 자소 및 자소에 대응하는 음소 정보를 통합적으로 이용한 변환 방법이 보다 효과적으로 한국어 음차표기를 생성함을 알 수 있다.

4. 결론

본 논문에서는 자소 및 음소정보를 이용한 영-한 음차표기 모델을 제안하였다. 본 논문의 기법은 기존 연구와 달리 자소 정보 뿐만 아니라 음소정보를 사용하였다. 즉, 직접방식과 피봇 방식을 통합한 기법으로 높은 성능을 나타내었다. 또한 기존 방법보다 최고 70%의 성능향상을 나타내었다.

하지만 본 논문의 기법은 성능향상의 여지가 많이 남아 있다. 첫째, 영어의 어원, 영어의 단어 형성과 같은 언어적 지식은 음차표기에 중요하게 사용될 수 있으므로, 이를 이용한 연구가 추가적으로 수행되어야 할 것이다. 둘째, 신경망, 최대엔트로피 모델 등의 다른 기계학습의 적용에 대한 추가적인 연구가 필요하다. 본 논문의 기법은 정보검색, 기계번역, 이중언어 사전 구축, 음성 인식 등 여러 자연언어응용에 유용하게 사용될 수 있을 것으로 기대된다.

참고문헌

[1] 이재성, 1999, 다국어 정보검색을 위한 영-한 음차표기 및 복원 모델, 박사학위논문, 한국과학기술원 전산학과

[2] Kang B.J. and K-S. Choi, 2000, "Automatic Transliteration and Back-transliteration by Decision Tree Learning", In Proceedings of LREC'2000, Athens, Greece.

[3] 강병주, 2001, 한국어 정보검색에서 외래어와 영어로 인한 단어불일치문제의 해결, 박사학위논문, 한국과학기술원 전산학과

[4] 김정재, 이재성, 최기선, 1999, 신경망을 이용한 발음단위 기반 자동 영-한 음차 표기 모델, 1999년도 한국인지과학회 춘계 학술대회 발표논문집, 서울, 1999. 5, pp. 247-252.

[5] Lee, J. S. and K. S. Choi, 1998, "English to Korean Statistical transliteration for information retrieval" Computer Processing of Oriental Languages, 12(1):17-37.

[6] Kang I.H. and G.C. Kim, 2000, "English-to-Korean Transliteration using Multiple Unbounded Overlapping Phoneme Chunks", In Proceedings of the 18th International Conference on Computational Linguistics.

[7] Goto I., N. Kato, N. Uratani and T. Ehara (2003) Transliteration Considering Context Information Based on the Maximum Entropy Method, In Proceedings of MT-Summit IX

[8] 문화부, 1995, 표준 외래어 표기법

[9] Daelemans W., Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch, 2002, "TiMBL: Tilburg Memory Based Learner, version 4.3 Reference Guide", ILK Technical Report 02-10, 2002.

[10] Quinlan, J.R., 1993, "C4.5: Programs for Machine Learning", Morgan Kauffman.

[11] 남영신, 1997, 최신 외래어 사전, 국어사전 별책부록, 서울: 성안당 출판사

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