정보검색을 위한 외래어 자동표기 모델
Automatic Foreign Word Transliteration Model for Information Retrieval

이재성, 최기선
한국과학기술원 전산학과
Jae Sung Lee, Key-Sun Choi
Computer Science Dept. Korea Advanced Institute of Science and Technology

조사에 따르면 한글 문서에서 사용되는 단어 중 외래어 또는 영어가 포함된 단어가 약 26%정도를 차지하고 있으며, 이는 정보검색의 중요 색인어로 사용된다(권윤형 1996). 그러나 이들 단어들은 서로 같은 단어인데도 영어로 표기되기도 하고 이형의 외래어들로 표기되기도 하여, 정보검색의 효율을 떨어뜨리고 있다. 본 논문에서는 영어 단어와 그에 대응되어 표기되는 외래어들을 찾기 위한 한 단계로서, 영어를 한글로 음차(transliteration)하여 자동표기하는 통계적 모델을 제안하고 실험한다.

제안된 모델은 통계적 기계번역 방식과 그의 한 방법인 문서 정렬(text alignment) 방식에 근거하고 있다. 특히 이 모델에서는 효과적으로 발음의 단위를 분리한 다음 정렬을 하여, 전체적인 계산량을 줄이고 성능도 향상시켰다. 음차표기는 피봇방식과 직접방식의 두가지로 구현하였다. 피봇방식은 영어에서 발음을 생성한 후, 그 발음을 다시 한글로 표기하는 방식이고, 직접방식은 직접 영이 단어에서 한글 표기로 표기하는 방식이다. 두 방식을 제안된 모델을 이용하여 비교 테스트한 결과 직접방식이 보다 정확하게 표준 외래어로 표기하였다.

1. 서론

외국과의 교류가 빈번해 지면서 한글 문서 내에 많은 고유명사와 새로운 전문용어들이 음차되어 표기되고 있다. 권윤형(1996)에 따르면, KTSET(한국통신의 정보검색용 테스트용 문서 (김재군 1994)) 내에서 임의로 문서를 선택한 후 외래어와 영어를 조사해 본 결과, 색인용 단어 토큰의 17.6%가 외래어이거나 외래어와 합쳐진 복합어이었으며, 8.8%는 영어(외국어)로 쓰여있다고 했다. 이는 전체의 약 26%로 매우 많은 양의 외래어 및 외국어가 우리 문서에 쓰이고 있고 주요 키워드로 많이 사용되고 있음을 나타낸다.

현재의 정보검색시스템에서는 대부분 이들 외래어 및 외국어는 같은 뜻의 단어임에도 불구하고 어휘의 차이에 의해 다른 단어로 취급하고 색인한다. 외래어는 한글로 표기되므로 그 자소가 일부 틀릴 경우, 같은 단어로 간주하여 처리하는 것이 가능하다. 예를 들면, 단어사이의 틀린 자소 갯수의 비율이 적을 경우, 이를 같은 단어로 취급하거나, 틀린 자소의 종류에 따라 유사한 정도의 차이를 다르게 계산할 수 있다(Zobel 1996).

외국어의 경우 글자가 원어로 표기되어 직접 한글과 비교할 수 없으므로, 외국어를 우선 한글로 표기(transliteration)할 필요가 있다. 외국어의 음차표기는 사전에 정의하여 변환할 수도 있지만, 실제 많이 음차되어 표기되는 고유명사나 새로운 전문용어 등은 사전에 등록되지 않는 경우가 많다. 따라서 단어로부터 직접 음차표기를 할 수 있는 시스템이 필요하다. 이러한 시스템은 한국어 문서내에서의 정보검색의 효율을 높여 줄 수 있을 뿐만아니라 다

국어사이의 정보검색에서는 질의어 번역의 한 부분으로 사용될 수도 있을 것이다.

외국어를 외래어로 표기한 다음 이형표기의 외래어를 찾기 위한 방법은 두가지를 생각할 수 있다: 1. 한 외국어에서 가장 정확한 한 외래어로 변환한 다음, 그 외래어에서 자소 등을 변형하여, 이형외래어 그룹을 생성한다; 2. 한 외국어에 표현가능한 여러가지 외래어를 직접 변환하여 생성한다. 두가지 방법 모두, 우선은 가장 많이 쓰이는 형태의 외래어로 정확하게 변환하는 것이 필요하다. 본 논문에서는 정확한 표기를 할 수있는 일반 변환 모델을 통계적 기계번역 방법에 근거하여 제안하고, 그 모델을 이용하여 실제 문서에서 많이 사용되는 외래어 형태를 찾기 위한 방법들을 영어-한글에 대해 실험하고 비교해 본다.

2. 이형 표기의 원인

현재의 외래어 표기법에서는 표준외래어를 용례집으로 만들어 발표하고 있으며, 이 용례집에는 주로 사람들이 많이 사용하는 외래어 표기를 표준으로 하는 것을 원칙으로 만들어지고 있다. 하지만, 새로운 단어나 고유명사 등에 대한 것은 모두 포함하지 않고 있고, 그 많은 단어들을 모두 포함할 수도 없으므로 외래어 원어의 발음으로부터 직접 표기를 할 수 있는 표기 규칙(외래어 표기 세칙)을 정하여 표준을 유도하고 있다. 이 표기 규칙은 기본적으로 원어의 발음에 충실하게 표기하는 것을 원칙으로 한다. 이러한 표준의 이중성은 외래어가 여러가지로 다양하게 표기되어 사용되는 한 원인으로 작용하고 있다.

현재 사용되고 있는 외래어의 실태를 파악하기 위해 간단한 조사를 했다. 즉, "data" 및 "digital"의 외래어 표기에 대해 그 사용형태를 KT SET 문서내에서 탐색해 본 결과는 다음과 같다. (Unix 의 grep 명령을 사용하여 단어를 탐색하여, 그 단어가 포함된 줄 수를 세었으며, 단어 앞에 *가 표시된 것은 현재의 표준어가 아님을 나타낸다.)

데이터: 61 줄(3%), *데이타: 1901 줄(97%)

디지털: 9 줄(0.1%), *디지틀: 321 줄(50%),
*디지탈: 308 줄(48%)

이 경우에서는 표준어보다는 비표준어가 대부분의 문서에서 쓰이고 있고, 그 비표준어도 한가지가 아니라 여러가지인 것을 알 수 있다. 이런 이유는 크게 두가지로 생각할 수 있다.

첫째로는 같은 단어나 철자의 발음이 여러가지일 수 있다. 표준 외래어 용어집에 없는 단어에 대해서는 원어사전에서 발음을 읽어오거나, 원어사전에도 없는 단어의 경우는 발음을 철자로부터 생성해내야 한다. 그 다음, 외래어 표기 규칙에 따라 변환을 하여 한글로 표기해야 한다. 그러나 영어 철자로부터 생성되는 발음이 불규칙하기 때문에 여러가지 발음이 있는 경우가 많고, 한 단어의 발음내에 생략가능한 음이 포함되어 있다든지, 같은 단어라도 나라마다 다른 발음기호로 표기하는 경우 등이 있다. 또한 똑같은 발음기호이더라도 나라에 따라 약간씩 다르게 발음을 하는 경우가 있다. 예를 들어 발음 [ɒ]는 미국식 영어에서는 "ㅓ" 발음에 가깝고, 영국식 영어에서는 "ㅏ"발음에 가깝다(이현복 1979). 이를 원음에 가깝게 표기할 경우, 어느 나라를 기준으로 하는가에 따라 표기가 다르게 된다.

둘째로는 사람들이 철자에서 직접 표기를 하는 경향이 있다는 것이다. 앞의 예에서 "data"의 발음은 [δειτ↔], [δΘτ↔], [δα:τ↔]이다. 이중 [δειτ↔]가 표준발음으로 채택되어 이를 표준 표기법에 따라 표기한 "데이터"을 표준어로 정했지만 "데이타"를 사람들이 더 많이 사용하고 있다. "digital"의 경우도 마찬가지이다. 이 단어의 발음은 [διδZιτλ]이며, 이를 표준 표기법으로 표기하면, "디지틀"이 된다. 그러나 "디지털"이 표준어로 채택되어 있고, 많은 사람들이 비표준어인 "디지틀"과 "디지탈"을 훨씬 더 많이 사용하고 있다.

"데이타"나 "디지탈"과 같은 표기가 사용되는 원인을 이현복(1979)은 사람들의 표기 습관에서 찾고 있다. 즉, 사람들이 단어의 발음기호로부터 복잡한 규칙을 적용하여 외래어 표기를 하기 보다는 단어의 철자를 보고 발음을 추측하여 바로 표기한다는 것이다. 이러한 현상을 눈말표기라고 하고 이와 반대되는 소리에 근거하여 표기하는 방식을 입말표기라고 했다. 예를 들어 "data"의 경우, 그림 1 과 같이 마지막 글자 "a"은 발음에 의하면 "ㅓ"으로 표기되어야 하지만, 글자 "a"의 추정된 발음인 "ㅏ"으로 표기되었다. 따라서 "데이타"는 눈말에 의해 표기된 것으로 볼 수 있다. "디지탈"의

경우도 이와 마찬가지로 "a"이 눈말로 표기된 것으로 볼 수 있다.

눈말표기로 된 외래어들이 일반적으로 많이 쓰임에 따라 표준어로 많이 채택되었다. 표준 외래어 어휘집의 단어를 살펴보면, 약 53%의 단어만이 표기 규칙에 따라 표기된 단어이다[1]. 이는 표준 외래어 단어들 중에 눈말표기로 된 외래어가 많이 포함되어 있음을 간접적으로 나타내 주는 수치이다. 또 이 조사 결과는 외래어의 발음으로부터 아무리 정확하게 표기를 하더라도 그 정확도에 한계가 있음을 나타낸다.

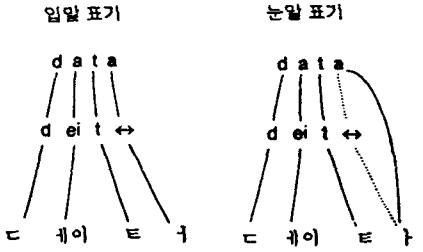

그림 1. 입말표기와 눈말표기

3. 표기 방식

사전에 없는 영어 단어로부터 외래어를 자동으로 생성하는 방법으로는 크게 2가지를 생각할 수 있다. 첫째는 외래어 표기법의 방식을 따르는 것이다. 즉, 영어 단어에서 영어 발음을 자동 생성해 내고, 외래어 표기 규칙에 따라 한글로 변환시키는 방법이다. 이 방법은 주로 입말표기를 하기 위한 방법이며, 이 글에서 피봇(pivot) 방식이라고 부른다. 여기에서 사용되는 발음기호는 일반적인 기계번역 방식에서 사용되는 중간언어(interlingua)의 역할과 비슷하다(Dorr 1993).

둘째는 영어단어에서 직접 한글로 된 외래어를 생성해 내는 것이다. 이를 직접(direct) 방식이라고 부르고, 피봇방식에 비해 눈말 표기의 규칙을 쉽게 학습해 낼 수 있을 것으로 생각된다. 두가지 표기방식의 단계는 그림 2와 3에 나타나 있으며, 두가지 모두 통계적 기계번역 방식에 근거한 모델을 사용하였다.

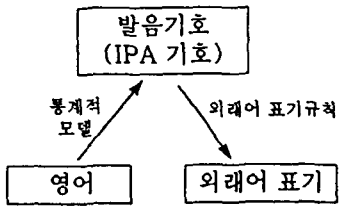

그림 2. 피봇방식의 외래어 표기방식

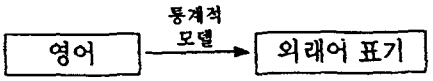

그림 3. 직접방식의 외래어 표기방식

4. 자동표기 모델

Brown(1990)은 한 언어의 문장이 다른 언어의 문장으로 번역되었을 때, 그 한 쌍의 문장 S, T에 대한 확률을 $p(T|S)$로 하였다. 이는 원문 S가 주어졌을 때, 번역문 T가 나타날 확률로 해석된다. 이를 다시 기계번역 문제로 바꾸면, 문장 T가 주어졌을 때, 문장 T를 생성해 낸 원문 S를 찾는 문제로 바꿀 수 있다. 베이스 정의(Bayes' theorem)를 이용하여 이를 수식으로 다시 쓰면 다음과 같다.

$$\arg\max_{s} p(S|T)$$

$$= \arg\max_{s} \frac{p(S)p(T|S)}{p(T)}$$

이 수식을 최대화하는 문장 S를 찾는 것이 기계번역의 문제이므로, T가 주어졌을 경우, 분모 $p(T)$는 상수가 되어 (S의 선택에 영향을 미치지 않으므로) 생략할 수 있다. 따라서 위의 식은 $p(S|T) = p(S)\,p(T|S)$로 다시 쓸 수 있다. 이때 $p(S)$는 언어확률 모델이고 원어의 구성이 잘 된 정도를 나타내는 확률이며, $p(T|S)$는 번

[1] 이희승(1994)의 표준어 어휘에서 원어의 발음을 사전에서 찾아낸 다음, 외래어 표기 규칙으로 변환하여 표준어와 비교해 본 결과이다. 참고로 자소수준의 정확도는 85.7%였다. (자소수준의 정확도 공식은 6장의 평가함수를 참조.)

역확률 모델이고 원어의 단어들 S가 번역문의 단어들 T로 번역될 수 있는 확률을 나타낸다.

위 수식에서 S와 T를 각각 원어 단어 및 역어 단어로 바꾸어, 각 단어 속에서 글자들의 번역문제로 바꾸면 자동표기 모델이 된다. 이 모델을 이용하면 영어 단어로부터 발음(또는 발음기호나 한글자소[2])을 생성해 내기 위해서는 단어를 구성하고 있는 글자로부터 발음을 생성해 내는 규칙을 통계학적으로 추출할 수 있다.

영어 단어로부터의 발음 생성은 그 불규칙성으로 인해 규칙을 찾는 것이 어렵다고 알려졌다(Stanfill 1986). 이러한 이유는 어원이 다른 많은 언어를 영어가 포함하고 있기 때문이며, 이러한 단어들이 학습 데이터에 포함될 경우, 오히려 규칙 생성을 방해하여 전체 효율을 떨어뜨리고 있기 때문이다. 본 실험에서는 우선 일반적인 몇가지 가정을 하여, 그 규칙성을 쉽게 계산할 수 있도록 했다.

가정 1: 일부의 글자들은 한 단위로 붙어 쓰이면서 최소한 하나 이상의 발음 또는 한글 자소를 생성해 낸다. 이 단위를 발음단위(*pu: pronunciation unit*)로 부른다.

가정 2: 한 단어는 발음단위들로 구성된다.

가정 3: 하나의 발음단위가 여러가지 가능한 발음을 가질 경우, 그 선택은 앞에 나타난 n개의 발음단위에 의해 결정되어진다.

발음단위는 글자열로 구성되며 글자의 갯수가 k개일 경우 k-길이 발음단위라고 한다. 원어의 i번째 k-길이 발음단위를 sp_i^k로 하고 원어 단어 S가 $s_1s_2s_3 \ldots s_m$ 일 경우, 이는 $s_is_{i+1}\ldots s_{i+k-1}$의 글자열로 구성된다. 대상언어의 i번째 k-길이 발음단위 tp_j^k도 이와 같은 방법으로 표기된다.

언어확률 모델을 발음단위를 이용하여 표현하면 다음과 같이 전개된다. 수식 (1)에서 (2)로의 변환은 일반 문자열을 발음단위 문자열로 변환한 것이며, (3)은 수식 (2)를 조건확률로 바꾸어 표현한 것이고, 수식 (4)는 수식 (3)을 계산의 편의상 2-그램 조건확률로 변환시킨 것이다. 마지막의 수식 (5)는 가상의 발음단위를 초항으로 가정하여 수식을 간편하게 표시한 것이다.

$$p(S)$$

$$= p(s_1, s_2, \ldots s_m) \quad (1)$$

$$= p(sp_{k_1}^{k_2-k_1}, sp_{k_2}^{k_3-k_2}, \ldots sp_{k_l}^{m-k_l}) \quad (2)$$

$$= \prod_{i=1}^{l} p(sp_{k_i}^{k_{i+1}-k_i} | sp_{k_1}^{k_2-k_1}, \ldots sp_{k_{i-1}}^{k_i-k_{i-1}}) \quad (3)$$

$$\approx p(sp_{k_1}^{k_2-k_1}) \prod_{i=2}^{l} p(sp_{k_i}^{k_{i+1}-k_i} | sp_{k_{i-1}}^{k_i-k_{i-1}}) \quad (4)$$

$$\approx \prod_{i=1}^{l} p(sp_{k_i}^{k_{i+1}-k_i} | sp_{k_{i-1}}^{k_i-k_{i-1}}) \quad (5)$$

번역확률 모델도 마찬가지로 수식 (6), (7), (8)과 같이 대응되는 발음단위의 조건부 확률로 전개할 수 있다.

$$p(T|S)$$

$$= p(t_1, t_2, \ldots t_n | s_1, s_2, \ldots s_m) \quad (6)$$

$$= p(tp_{k_1}^{k_2-k_1}, tp_{k_2}^{k_3-k_2}, \ldots tp_{k_r}^{n-k_r} |$$

$$sp_{k_1}^{k_2-k_1}, sp_{k_2}^{k_3-k_2}, \ldots sp_{k_l}^{m-k_l}) \quad (7)$$

$$\approx \prod_{i=1}^{l} p(tp_{k_i}^{k_{i+1}-k_i} | sp_{k_i}^{k_{i+1}-k_i}) \quad (8)$$

전개 결과 얻어진 수식 (5)와 (8)의 확률을 추정하기 위해 다음과 같은 함수를 정의한다.

정의 1: $C(x)$ 는 x가 나타난 횟수이다.

정의 2: $FC(x, y)$ 는 x 다음에 y가 나타난 횟수이다.

정의 3: $C(x, y)$ 는 x와 y가 동시에 나타난 횟수이다.

이를 이용하여 수식 (5)와 (8)을 추정하는 식으로 전개하면 다음과 같다

[2] 한글의 단어는 음절 또는 음소 단위로 분리할 수 있다. 이 글에서는 한글의 음소(자소)를 영어글자나 발음기호와 마찬가지로 한 글자 단위로 취급한다.

$$p(sp_j^{k'} \mid sp_i^k) \approx \frac{FC(sp_i^k, sp_j^{k'})}{C(sp_i^k)} \quad (9)$$

$$p(tp_j^{k'} \mid sp_i^k) \approx \frac{C(tp_j^{k'}, sp_i^k)}{C(sp_i^k)} \quad (10)$$

식 (9) 및 (10) 을 실제로 계산할 경우, 많은 수의 발음단위를 생성하게 되어 계산이 매우 어렵게 된다. 예를 들어 3-길이 발음단위를 사용할 경우, 1-길이와 2-길이도 모두 포함하여 사용하기 때문에 영어 글자에 대해서만 $26 + 26^2 + 26^3 = 18{,}278$ 개의 발음단위가 만들어 진다. 사실상 실제로 코퍼스에서 나타나는 발음단위의 종류수는 이것보다는 훨씬 적지만, 여전히 계산하기에는 부담이 되는 숫자이다. (이 실험에서 사용된 1,500 단어에서 나타난 발음단위 종류는 2,459 개이다.)

이러한 계산의 부담은 몇 가지 제약을 가면서 줄일 수 있다. 첫째로, *sp* 가 *tp* 로 번역될 확률은 각 단어의 상대적 위치에 매우 영향을 받는다는 사실이다. 이를 반영하기 위해 새로운 함수를 다음과 같이 정의한다.

정의 4: *DC(x, y, d)*는 *x* 와 *y* 가 동시에 나타나되 상대적 위치차이(*dist(x, y)*)가 *d* 를 넘지 않고 나타나는 횟수이다.

정의 5: *dist(x, y)*는 *x* 가 포함된 단어내에서 *x* 의 상대적 위치와 *y* 가 포함된 단어내에서의 *y* 의 상대적 위치의 차이이다. 이 값은 $x=tp_j^{k'}$이고 $y=sp_i^k$이며, *T* 의 길이가 *n* 이고, *S* 의 길이가 *m* 일 때, 다음과 같이 계산된다:

$$dist(tp_j^{k'}, sp_i^k) = abs((j + \frac{k'}{2}) \times \frac{m}{n} - (i + \frac{k}{2})) \quad (11)$$

정의 4 와 5 에 의해 식 (10)을 다시 계산하면 다음과 같이 쓸 수 있다.

$$p(tp_j^{k'} \mid sp_i^k) \approx \frac{DC(tp_j^{k'}, sp_i^k, d)}{C(sp_i^k)} \quad (12)$$

두번째로 제약을 줄 수 있는 것이 발음단위의 결합이다. 대개의 글자는 그것이 자음과 모음 또는 반모음의 형태로 구분되어 있으며, 자음과 모음이 반복되어 하나의 음절을 이루게 된다. 이때 각 자음과 모음은 대개 같은 종류의 글자가 합쳐져서 하나의 음을 만들어 낸다. 이 성질을 이용하여 각 글자의 발음타입을 정하고 비슷한 타입의 글자끼리만 발음단위가 형성될 수 있도록 한다. 이러한 발음단위를 호환 발음단위 (*cu: compatible pronunciation unit*)라고 하고 이를 이용하면 식 (12)를 식 (13)과 같이 다시 쓸 수 있다. 이 식은 단순히 *sp* 와 *tp* 대신에 *sc* (원어 호환 발음단위)와 *tc* (대상어 호환 발음단위)로 바꾼 식이다.

$$p(tc_j^{k'} \mid sc_i^k) \approx \frac{DC(tc_j^{k'}, sc_i^k, d)}{C(sc_i^k)} \quad (13)$$

언어확률 모델은 다른 코퍼스에서 학습시켜서 만들어 낼 수도 있지만, 이 실험에서는 번역확률 모델과 같은 코퍼스를 사용하고, 또 발음단위도 번역확률 모델에서 선택되어진 것만을 사용하였다. 따라서 불필요한 발음단위를 미리 제거하여 언어확률 모델의 계산량을 줄였으며, 모델의 디코딩 계산량을 줄이기 위해 비터비(Viterbi) 알고리즘의 변형인 스택(stack)을 이용한 알고리즘을 사용하였다(Rabiner 1989).

5. 실험 및 결과

5.1실험 수행 방식 및 결과

위에서 제시한 수식을 이용하여 반복 학습을 통해 확률조정을 했다. 이 실험 과정은 다음과 같다.

1. 단어 단위로만 정렬된 코퍼스로부터 각 단어에서 모든 조합 가능한 발음단위에 대해 수식 (10), (12), (13)을 이용하여 번역 모델 확률의 초기값을 구한다. (사실상 대상 발음단위(*tp*)와 원어 발음단위(*sp*)의 공기정보를 구하여 확률 값으로 만든 것이다.)
2. 구해진 번역 모델 확률을 이용하여 단어쌍의 최적 정렬을 한다. 이 결과 글자단위로 정렬된 새로운 코퍼스가 만들어 진다.
3. 새로 만들어진 정렬 코퍼스에 대해 다시 번역 모델의 확률을 계산한다.
4. 2, 3 과정을 정해진 숫자 만큼 반복한다.
5. 번역 모델에서 나타난 발음단위만을 이용하여 언어모델의 확률을 계산한다.

6. 구해진 번역 모델 확률과 언어 모델 확률을 이용하여 실험 데이터를 디코딩(decoding)하여 외래어 표기의 결과를 내놓는다.
7. 피봇 방식의 경우는 결과로 나온 발음기호를 외래어 표기 규칙 프로그램으로 변환하여 한글로 표기한다.

평가함수로는 글자수준의 정확도를 사용했고, 이를 수식으로 표현하면 다음과 같다 (Parfitt 1991, Lucas 1991).

$$\text{글자수준의 정확도} = (L - i - d - s)/L$$

L은 원어의 길이, i는 삽입(insert), d는 삭제(delete), s는 교체(substitute)의 횟수이다. 이 값이 음수가 되면, 정확도는 0으로 본다. L을 두 단어 중 길이가 긴 것으로 하는 경우도 있지만(Lucas 1991), 여기에서는 원래의 단어를 기준으로 계산하여 L을 원어의 길이로 했다.

전체 코퍼스는 약 1,700개의 단어쌍으로 구성되었으며, 이중 1,500개의 단어쌍을 학습용 단어로 사용하였다. 학습용 단어 중 다시 150 단어쌍을 학습 데이터(seen data) 테스트용으로 사용하고, 학습에 사용하지 않은 단어 중 150 단어쌍을 미학습 데이터(unseen data) 테스트용으로 사용하였다.

테스트는 1-길이, 2-길이, 3-길이 발음단위에 대하여 각각 해보았고, 상대위치 차이 값도 0.1 에서 1 까지 여러가지로 바꾸어 보면서 했다. 그 결과 2-길이이고 상대 위치 차이 값이 0.3 내지 0.4 부근에서 좋은 성능을 보였다.

정렬시 사용된 휴리스틱인 상대위치 제약과 발음단위의 결합 제약이 표기 방식이나 테스트 데이터에 관계없이 모두 성능향상에 기여했다. 이 결과로 호환발음단위를 사용하고, 상대위치 차이를 고려한 수식 (13)의 성능이 가장 좋았고, 그 다음으로 상대위치 차이만을 이용한 수식 (12), 그 다음으로 아무런 제약이 없는 수식 (10)의 순이었다.

두가지 표기방식의 성능을 비교한 도표는 그림 (4), (5)에 있다. 학습 데이터에 대해서는 직접방식이나 피봇방식이 비슷한 정확도를 보였으나, 미학습 데이터에 대해서는 직접방식이 월등한 성능을 보였다. 이 시스템이 새로운 단어에 대한 표기에 주로 쓰일 것을 고려하면, 당연히 직접방식을 사용하는 것이 훨씬 효과적일 것임을 알 수 있다. 또 그림 (6)에서 보듯이 학습용 데이터의 양을 늘려 나감에 따라 미학습 데이터에 대한 정확도가 증가해 갔다. 이는 이 시스템이 학습의 효과가 있으며, 많은 양의 데이터가 있을 경우, 더 정확도가 증가할 수 있음을 보여준다.

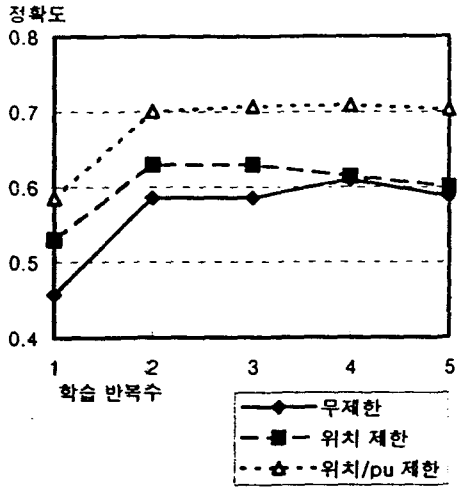

그림 4. 직접방식에서 미학습 데이터에 대한 각 수식의 정확도

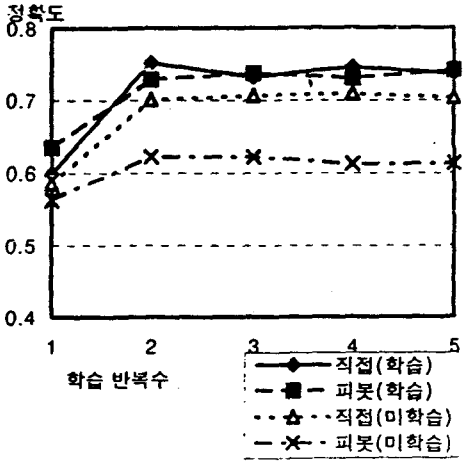

그림 5. 직접방식과 피봇방식의 정확도

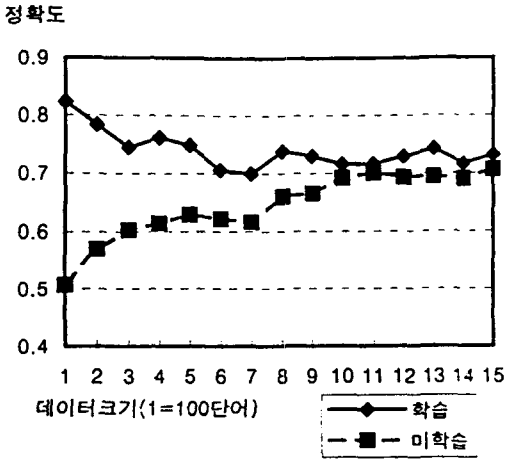

그림 6. 학습용데이터양 변화에 따른 정확도 (직접방식, 위치 및 *pu* 제한, 학습 3 회반복)

다양한 표기를 찾기 위한 테스트는 우선 사람들이 사용하고 있는 다양한 표기 형태를 수집하여야 한다. 현재 이러한 자료가 준비되지 않아서, 우선 간단한 테스트에 의해 그 결과를 예측해 보았다. 즉, 영어 "media"에 대해 직접방식과 피봇 방식으로 가능한 표기 5 가지만을 생성해 내도록 했다. 그리고, 5 명의 사람들에게 "media"의 가능한 표기를 적어보도록 하여 나타난 모든 표기어를 다음과 같이 나열하였다. (피봇방식에서 "미디어"는 두 번 생성되었는데, 이는 발음기호는 다르더라도 같은 한글 자소로 표기될 수 있기 때문이다.)

직접방식: 메디어, 메디아, 미디어,
미디아, 마디어
피봇방식: 메디어, 머디어, 미디어,
미디어, 메이디어
사람: 미디어, 미디아, 매디아

"media"의 표준 표기는 "미디어"이지만, 이 경우에도 사람들은 "a"의 발음인 [↔]보다는 "ㅏ"를 많이 사용하여 눈말표기를 하였음을 알 수 있다. 직접방식은 이러한 눈말표기를 반영하여 "매디아"와 "미디아"를 생성해 냈다. 하지만, 피봇방식은 이를 찾지 못하여 결국은 직접방식이 2/3 를 찾았고, 피봇방식은 1/3 을 찾았다. 이 실험에서 사람들이 "메디아"를 나열하지는 않았지만, 직관적으로 사용할 가능성이 많은 단어이다. 따라서, 좀더 정확한 평가는 나중에 내릴 수 있지만, 직접방식이 좀더 유리할 것이라는 것은 추측할 수 있다.

6.토의 및 앞으로의 연구

실험 결과에 알 수 있듯이 직접방식이 정확도에서 훨씬 우수한 성능을 발휘했고, 다양한 표기어에 대한 커버율도 더 높을 것이라도 추정된다. 직접 방식은 피봇방식에 비해 처리과정도 단순하므로 외래어 표기를 위한 방식으로 적합하다고 생각된다.

이러한 두가지 접근 방식은 이와 비슷한 다른 시스템에서도 테스트가 가능할 것이다. 예를 들면, Decision tree(Bahl 1991), Neural Net(Lucas 1991), Markov Model(Parfitt 1991), MBRtalk(Memory Based Reasoning)(Stanfill 1986), NETtalk 등이 그것들이다. 우리 시스템을 이들 중 일부 시스템과 단순한 정확도의 성능만을 비교해 보면, 표 1 과 같다. (각 시스템 사이에 평가함수가 일부 조금씩 다른 것도 있다.)

표 1. 텍스트-음성기호 변환시스템의 성능비교

	학습데이터	미학습데이터
NETtalk	94%	78%
MBRtalk	100%	86%
Neural Net	67.8%	66.2%
피봇방식	74.4%	63.1%

우리 시스템은 영어에서 한글로 표기하는 것이므로 직접 방식은 비교하기가 곤란하다. 따라서 피봇 방식에서 첫번째 단계인 영어에서 발음기호 생성과정의 결과를 가지고 비교했다. 물론 이 방식이 직접 방식에 비해 성능이 훨씬 낮은 것이다.

NETtalk 는 7 개의 윈도우를 사용하여 많은 양의 문맥정보를 이용하여 처리했고, MBRtalk 도 마찬가지이다(Stanfill 1986). MBRtalk 에서는 학습데이터에 대해 정확도가 100%인 이유는 모든 학습 데이터를 메모리에 저장해 두고 가장 가깝게 일치하는 것을 찾아내기 때문이다. 순수한 데이터에서 자동으로 규칙을 추출해내는 점에서 Neural Net 방식이 우리 모델과 비슷하다. 대개의 학습시스템에서 학습 데이터가 충분히 많을 경우, 학습 데이터 테스트 결

과와 미학습 데이터 테스트 결과는 그 평균으로 수렴한다. 따라서 Neural Net 방식의 경우, 평균 67%이고, 우리 모델의 경우, 68.7%이므로 약간 더 우수하다고 볼 수 있다. (Neural Net의 노드 수를 늘리더라도 평균 정확도는 67%였다.) 또, 우리 모델에 사용된 데이터가 여러 어원의 영어를 사용한 점과 코퍼스의 크기도 비교적 작은 1,500 단어를 사용했다는 점에서 고려하면, 우리 시스템의 성능이 우수함을 알 수 있다.

현재의 시스템은 아직 정확률면에서 향상시켜야 될 점이 많다. 앞으로 정확한 표기를 위해 기본 모델을 좀더 확장하여 좌우의 문맥정보를 더 많이 이용하고, 발음단위 생성 규칙을 좀더 자세하게 연구할 필요가 있다. 또한 다양한 표기에 대해 다른 방법(Mettler 1993, SERI 1995)과의 비교 평가가 필요하다. 또, 현재의 시스템이 영어에서 한글로 표기하는 시스템이므로, 한글에서 영어로 표기하는 시스템에 대한 연구와 다른 나라 언어 사이의 음차 표기에 관한 연구로 확대할 수도 있을 것이다.

7. 참고문헌

권윤형, 정길순, 맹성현 (1996), "정보검색 효과의 개선을 위한 외래어 처리기," 과학기술정보워크샵.

김재군, 김영환, 김성학 (1994), "한국어 정보검색연구를 위한 시험용 데이터 모음(KTSET) 개발," 제 6 회 한글 및 한국어 정보처리.

이현복 (1979), "외래어 표기법 개정 시안의 문제점," 어학연구 15. 1.

이희승, 안병희 (1994), "고친판 한글 맞춤법 강의," 신구문화사.

최기선 (1992), "한국어정보처리와 한글코드," 한국어 정보처리, 제 1 권 제 2 호.

SERI/KIST (1995), "지능형 정보처리기의 개발에 관한 연구," 제 1 차년도 최종보고서, 과학기술처.

L. R. Bahl, F. Jelinek, R L. Mercer (1983), "A Maximum Likelihood Approach to Continuous Speech Recognition," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-5, No. 2, March.

L. R. Bahl, P. V. de Souza, P. S. Gopalakrishnan, D. Nahamoo, M.A. Picheny (1991), "Decision Trees for Phonological Rules in Continuous Speech," IEEE ICASSP.

P. F. Brown, and et al. (1990), "A Statistical Approach to Machine Translation," Computational Linguistics, Vol 16, Num. 2, June.

P. F. Brown, and et al. (1993), "The Mathematics of Statistical Machine Translation: Parameter Estimation," Computational Linguistics, Vol 19, Num. 2.

B. J. Dorr (1993), "Machine Translation: A View from the Lexicon," MIT Press, pp1-13.

S. M. Lucas and R. I. Damper (1991), "Syntactic neural networks for bi-directional text-phonetics," pp 127-141 in Talking Machines, ed G. Bailly and C.Benoit, North Holland.

M. Mettler (1993), "TRW Japanese Fast Data Finder," TIPSTER Text Program Phase I Proc., Sep. pp113-116.

S. H. Parfitt and R.A. Sharman (1991), "A Bi-directional Model of English Pronunciation," EuroSpeech.

L. Rabiner (1989), "Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE vol 77, no 2. pp257-286.

R. A. Sharman (1994), "Syllable-based Phonetic Transcription by Maximum Likelihood Methods," COLING 94.

C. Stanfill and D. Waltz (1986), "Toward Memory-Based Reasoning," Communication of the ACM, Vol 29, No. 12, December, pp 1213-1228.

J. Zobel, P. Dart (1996), "Phonetic String Matching: Lessons from Information Retrieval," ACM SIGIR Conference on R & D in information Retrieval (SIGIR 96).