학습가능한 영어단어의 한글조사 생성방법[1]

이재성, 이운재, 최기선
한국과학기술원 전산학과 / 인공지능연구센터
대전시 유성구 구성동 373-1, 우: 305-701
{jslee, wjlee, kschoi}@world.kaist.ac.kr

A Trainable Hangul Josa (particle) Generation Method for English Words

Jae Sung Lee, Woon Jae Lee and Key-Sun Choi
Dept. of Computer Science / Centre of Artificial Intelligence Research
Korea Advanced Institute of Science and Technology
373-1 Kusong-dong, Yusong-gu, Taejon, 305-701

요약

한국어에서 많이 사용되는 조사는 앞 명사의 종성에 따라 그 형태가 다르게 표기된다. 그런데 앞의 단어가 한글이 아닌 영어나 특수기호 등일 경우, 그 단어의 종성을 알아내는 일은 단순하지 않다. 본 논문에서는 영어가 명사로 쓰였을 경우, 그 단어 종성의 종류를 파악하여 적합한 형태의 조사를 자동으로 붙이는 방법을 제안하고, 실험한다. 제안된 방법은 기본적으로 결정트리(decision tree)와 같은 방법으로 영어 단어의 뒷글자부터 차례로 결정의 단서로 사용하였고, 분야에 맞도록 학습가능하다. 1,500단어로 학습한 후, 미학습 데이터에 대해 실험한 결과, 약 96%의 정확도를 보였고, 학습 데이터가 많아지면, 정확도가 더 증가할 것으로 보인다.

1 서론

외국과의 문화적 기술적 교류가 활발해 짐에 따라, 한글 문서내에서도 많은 외래어와 외국어가 사용되고 있다. 순수한 한글 사용을 국가적으로 권장하고 있지만, 현실적으로는 많은 사람들이 외국어 및 특수 기호 등을 한글과 혼용하여 사용하고 있다. 예를 들면, 영어의 약자를 그대로 사용하거나, 영어 이름이나 전문 용어를 한글 문장 중에 그대로 사용하기도 하고, 특수 기호나 숫자 등을 그대로 섞어 쓰기도 한다. 특히 컴퓨터에서는 다양한 형태의 메시지를 자동 생성함에 따라, 화일 이름 등으로 사용되고 있는 영어 단어와 한글 메시지를 혼합하여 사용하는 경우가 많다. 이러한 영어 단어들은 대개 명사로서 사용되며, 그 뒤에 조사를 붙이는 경우가 많다. 다음과 같은 예를 보자.

1) IBM은 오늘 새로운 제품을 발표했다.
2) Internet을 이용한 국제전화 서비스를 개시한다.
3) 무게가 70KG을 초과했다.
4) 구내전화는 4321로 해 주십시오
5) source를 destination으로 복사합니다.
6) file을 destination으로 복사합니다.

1)의 경우, IBM은 "아이비엠" 으로 읽어서 조사 "은" 이 붙었다. 2)의 경우, Internet을 "인터넷" 으로 읽어 "을" 이라는 조사가 붙었다. 만약, "인터네트" 로 읽었다면, 조사 "를" 을 붙여야

[1] 본고는 정보통신부의 지원을 받아 수행중인 [지능형 멀티미디어 통합 정보베이스] 과제의 일환으로 이루어졌다.

올바른 표기가 된다. 3)의 경우, "70KG"을 "칠십킬로그램"으로 읽어서 조사 "을"이 사용되었다. 만약 "칠십킬로그래므"로 잘못 읽는다면, 조사 "를"을 사용해야 올바른 것이 된다. 4)의 경우, 숫자를 "사삼이일"로 읽어서 조사 "로"가 사용되었다. 또한 5)와 6)의 경우, 주로 컴퓨터의 메시지로 자주 나오는 것으로 영어로 쓰인 부분은 사용자가 지정하는 화일 이름이나 디렉토리 이름이므로 그 뒤의 조사가 경우에 따라 가변적으로 붙여져야 한다. 5)에서 "source (소스)" 단어 대신 "file (화일)" 이라는 단어를 넣을 경우, 조사가 "를" 에서 "을" 로 바뀌어 6)처럼 되어야 좀 더 자연스런 메시지가 된다. 실제로 한글판 컴퓨터 프로그램에서는 많은 경우, 앞에 어떤 단어가 올 지 모르기 때문에 두 가지의 가능성을 모두 고려하여 "X을(를) Y으로(로) 복사합니다."와 같이 괄호를 사용하는 메시지가 표시되기도 한다. 이 경우, 앞의 조사가 뒤의 괄호 안의 조사로 대신 쓰일 수 있음을 나타내고 있지만, 이는 매우 어색한 표현이다. 만약 앞의 단어를 동적으로 분석하여 끝 발음을 알아내면, 좀 더 자연스런 메시지를 생성할 수 있을 것이다.

한글 조사는 주로 그 앞 단어의 종성에 따라 선택된다. 종성의 종류는 다음과 같이 크게 받침이 있고 없음에 따라 유종성과 무종성으로 구분되며, 유종성은 다시 "ㄹ종성" 과 "ㄹ이외의 유종성" 으로 구분된다. 편의상 이 글에서는 "ㄹ이외의 유종성" 을 단순히 "유종성" 으로 부른다. 따라서 조사 선택을 위한 종성의 종류를 크게 3가지로 구분하며, 각각의 예는 다음과 같다.

종성종류	단어 예	붙는 조사 형태
유종성	한국, 뮤직	은, 이, 을, 과, 으로
무종성	나라, 박스	는, 가, 를, 와, 로
ㄹ종성	하늘, 화일	은, 이, 을, 과, 로

예를 들어, 주격 조사 "은" 은 유종성이나 ㄹ종성의 단어 뒤에, "는" 은 무종성 단어 뒤에 붙여지고, 부사격 조사 "으로" 는 유종성 단어 뒤에, "로" 는 무종성과 ㄹ종성 단어 뒤에 붙여 진다.

본 논문에서는 숫자나, 특수 기호, 영어 단어의 뒤에 한글 조사를 자동으로 선택하여 붙이는 방법을 제안하고 실험한다. 주로 일반적인 숫자[2]나 특수 기호 등도 영어 단어의 처리와 유사하므로 본 논문에서는 영어 단어를 중심으로 설명한다.

영어 단어에 한글 조사를 붙이기 위해서는 그 단어의 발음을 한글로 표기한 다음, 그 단어의 종성을 확인한 후, 적절한 조사를 선택하여 붙이면 된다. 영어 단어에서 한글로의 표기는 [1, 2]등에서 연구되어져 있다. 그러나, 이 경우에는 영어 단어 전체를 한글로 표기한 후, 끝 소리를 분석하여 조사를 붙이므로 불필요하게 앞부분까지 분석을 하게 된다. 조사를 붙이기 위해 영어 단어의 뒷부분의 일부만을 분석하고서 발음의 종성을 확인할 수 있다면, 이것이 더 효율적이다.

본 논문에서는 영어단어가 한가지의 표준발음으로 읽힌다는 가정하에, 영어 단어의 끝 부분만을 부분적으로 분석하여 종성을 추출해 내는 방법을 연구한다. 특히, 조사 생성을 위해서 앞에서 제시한 3가지 종성 종류로만 구분함으로써, 좀 더 단순한 종성 분류 규칙을 만들어 낸다.

2 규칙의 추출과 적용

종성 규칙은 영어와 한글표기 쌍에서 자동으로 학습될 수 있다. 학습을 위해, 우선 한글 표기로부터 각 영어 단어의 종성 종류를 분리해 낸 후, 다음의 알고리즘을 적용한다. 이때, 학습을 효과적으로 하기 위해 학습데이터를 단어의 뒷글자순서로 정렬하면 학습데이터를 한번만 읽고 전체 규칙을 만들어 낼 수 있다.

알고리즘

1. id_string을 null로 초기화한다.
2. 각 단어의 끝 글자 x를 취하고 x를 id_string에 추가한다.
3. id_string으로 끝나는 단어집합 S(id_string)을 찾아낸다.
4. S(id_string)내의 단어들이 모두 같은 종성 타입을 갖는지 비교한다.

5-1. 종성 타입이 같으면,
id_string을 그 종성 타입으로 정하고 id_string에 마지막 추가된 글자를 다른 단어의 끝 글자 x'로 대체하고 더 이상 다른 끝 글자가 없을 때까지 3부터 반복한다.

5-2. 종성 타입이 다르면,
단어의 그 다음 끝 글자를 id_string에 추가하고 3으로 가서 반복한다.

예를 들어 표1과 같이 "b"로 끝나는 단어들에 대해 종성 규칙을 추출한다면, 우선 id_string을 "b"로 하고 종성 타입을 조사한다. id_string으로 끝나는 단어들이 무종성과 유종성의 두가지 타입을 가지고 있으므로, id_string을 늘려서 다시 조사한다. 첫번째

[2] 숫자는 주로 끝자를 읽으면 되지만, "$100 (백달러)"와 같이 앞에 있는 특수 기호가 끝 발음을 결정하는 경우가 있으므로, 이러한 예외 처리를 하여야 한다.

표1. 규칙생성용 영어단어 예

slab(슬래브) 무종성
arab(아랍) 유종성
bulb(벌브) 무종성
job(잽) 유종성
club(클럽) 유종성
영어 단어의 역순 배열
bals 무
bara 유
blub 무
boj 유
bulc 유

단어의 끝에서 두번째 글자를 취하여 id_string을 "ab"로 하고 다시 이 id_string으로 끝나는 단어들을 조사한다. 이 경우도 역시 무종성 및 유종성 두 가지가 존재하므로 다시 id_string을 "lab" 로 취한다. 이때 id_string은 "slab"가 되어 S(id_string)은 한 단어가 된다. 따라서, 그 단어의 종성 종류인 무종성이 되어, "lab"로 끝나는 단어는 무종성으로 처리한다. 그 다음으로 id_string을 다음 단어에서 취하여 "rab"로 하면, 이 경우도 역시 한 단어이고 따라서 같은 종성 타입이 되어 "rab"로 끝나는 단어는 유종성으로 처리한다. 다음으로는 id_string을 "lb"로 하여 무종성 규칙을 찾고, 계속하여 "ob", "ub" 에 대해 유종성인 규칙을 찾아 낸다. id_string을 역으로 배열한 후, 학습된 규칙을 서술하면 다음과 같다.

bal: 무
bar: 유
bl: 무
bo: 유
bu : 유

이 규칙은 간단하게 결정트리 형태로 나타낼 수 있다. 즉, 각각의 id_string의 글자를 하나의 노드로 하여 나타내면 그림 1과 같다. 따라서 입력된 단어를 끝 글자부터 결정트리를 적용하여 따라가면 그 단어의 종성 타입을 결정할 수 있다.

그러나, 학습에 사용된 단어에 없는 종류의 끝 글자가 왔을 경우, 이를 처리하는 방법이 필요하다. 이를 위해, 결정트리를 따라가다 실패했을 경우, 그 위의 부모노드를 따라가 그 노드에 저장된 기정치 정보를 사용하도록 했다. 이를 위해 미리 필요한 부모노드에 기정치 값을 주어야 한다. 예를 들어, "b"로 끝나는 단어의 경우, 기정치는 유종성으로 정하였다 (학습 데이터에서 가장 많이 나타나는

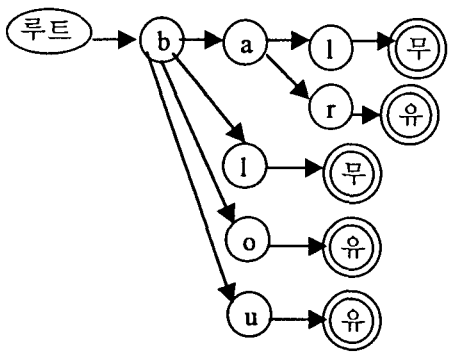

그림 1. "b"로 끝나는 단어의 결정트리 예
(기정치를 사용하지 않은 예)

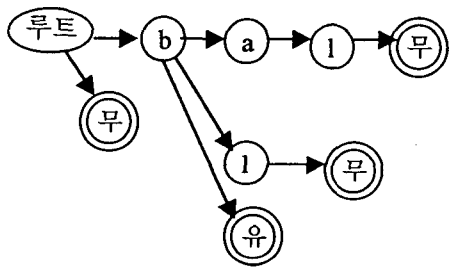

그림 2. "b"로 끝나는 단어의 결정트리 예
(기정치를 이용하여 축약한 예)

타입을 기정치로 하면 효과적일 것이다.) 만약 "bab"이라는 단어가 나타나면, 그림 1의 결정트리에 나타나 있지 않으므로 "유종성" 으로 판별한다. 이와 같이 기정치값을 정하면 결정트리를 더 축약할 수 있으며, 그림 2는 그림 1의 결정트리를 축약한 것이다.

또, 여기에서 추출된 "lab"과 같은 규칙은 "slab (슬래브)"라는 단어에도 적용되지만, "lab (랩)"이라는 한 단어에 대해서도 적용이 된다. 이를 구분하기 위해 학습 데이터에 "lab"이라는 단어를 넣을 경우, id_string이 단어의 첫글자까지 가더라도 "slab (슬래브)"와 종성타입이 다른 형태로 나타난다. 이 문제를 해결하기 위해 단어 앞에 특수한 목적으로 "$"를 넣어 "$lab"과 같이 표시하여, 알고리즘을 그대로 적용할 수 있도록 했다.

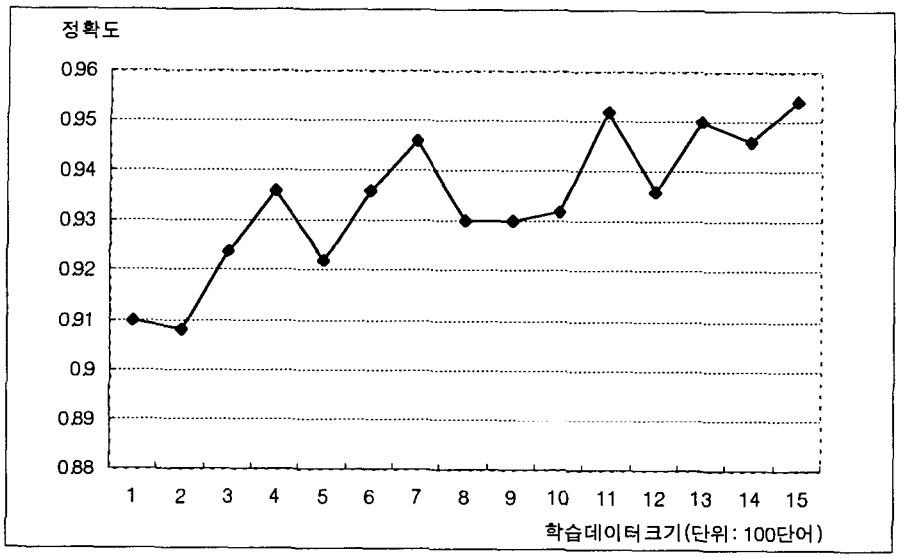

그림 3. 학습 데이터 크기 변화에 따른 정확도

3 실험 및 결과

실험을 위해 이미 [1, 2]의 연구에서 구축된 외래어 1,687단어를 이용하였다. 즉, 영어와 한글 표기가 포함된 집합에서 한글의 종성 타입을 구분해 내고, 그 타입을 이용하여 학습하였다. 앞 절에서 제시된 알고리즘은 학습된 데이터에 대해서는 100% 정확한 결과를 내놓는다. 미학습된 데이터에 대한 정확성을 실험하기 위하여, 테스트 데이터용으로 150단어를 무작위 추출해 내었고, 나머지 1,537단어만을 학습에 사용하였다. 또, 학습데이터의 영향을 측정하기 위해 1,537단어에서 무작위로 100개부터 1,500개까지 100개 단위로 늘려가면서 학습데이터를 추출하여 학습에 사용하였다. 그 결과는 그림 3과 같다. 그림에서 보듯이, 초기에는 매우 불안정하게 변화가 발생하다가 점차적으로 학습데이터의 크기에 비례하여, 정확도도 향상되어 갔고, 1,500개의 학습데이터에서는 약 96%의 정확도를 나타내었다. 적은 학습데이터를 사용했을 경우, 정확도가 불규칙적으로 변화하였는데 이는 적은 데이터에 대해서는 학습데이터 양이 늘어도 반드시 그 정확도가 증가하지 않을 수도 있음을 보여준다. 예를 들어, 다음과 같은 학습데이터1에 대해 앞의 알고리즘을 적용하면, 규칙1이 생성된다. 이에 대해 테스트 데이터를 분석하면, 모두 맞는 결과인 ㄹ종성으로 분석된다. 하지만, 데이터를 늘려 학습데이터2를 이용하면 규칙2가 생성되고, 그 결과 테스트 데이터는 모두 틀린 결과인 무종성으로 분석된다. 이는 학습데이터가 많다고 반드시 미학습 데이터에 대한 정확도가 높아지지는 않음을 보여주는 예이다. 그러나 전술한 바와 같이 많은 양의 학습데이터를 사용할 경우, 전반적으로 정확도가 향상되었다.

학습데이터 1

stroke	스트로크	무
cable	케이블	ㄹ

추출된 규칙 1

ek	무
el	ㄹ

테스트 데이터

scale	스케일	ㄹ
sale	세일	ㄹ

분석결과 1

scale	ㄹ (맞음)
sale	ㄹ (맞음)

학습데이터 2

stroke	스트로크	무
finale	피날레	무
cable	케이블	ㄹ

추출된 규칙 2

ek	무
ela	무
elb	ㄹ

테스트 데이터

scale	스케일	ㄹ
sale	세일	ㄹ

분석결과 2

scale	무 (틀림)
sale	무 (틀림)

1,500개의 학습데이터에 대한 학습결과 161개의 규칙이 생성되었으며, 표 2는 그 규칙의 일부로서 "e"로 끝나는 단어들에 대한 것이다. 표에서 첫 열의 글자는 역순으로 나타낸 id_string이고, 둘째열은 영어단어의 종성 타입을 나타내며, "무"는 무종성, "유"는 유종성, "ㄹ"은 ㄹ종성을 각각 뜻한다. id_string은 가장 긴 것이 5글자이었고, 가장 짧은 것은 1글자이었다. 1글자인 것들은 "a", "f", "m", "o", "u", "y", "z"이며, 이중 "m"으로 끝나는 글자는 유종성이고, 그 이외는 모두 무종성이다.

표2. 추출된 규칙 예 (e로 끝나는 단어)

ek	무
elac	ㄹ
elan	무
elg	ㄹ
elib	무
elif	ㄹ
elim$	ㄹ
elimi	무
elim	ㄹ
elip	ㄹ
elit	ㄹ
eli	무
el	ㄹ
em	유
enol	유
enome	무
enomr	유
enom	무
enoz$	유
enozn	무
en	유
epo	무
epy	유
ettes	무
etteu	유
euqil	무
euqin	유

4 맺음말

영어 단어가 문맥 내에서 애매성없이 정확하게 한가지로 읽힌다는 가정하에 한글 조사를 붙이는 방법에 대하여 설명하였다. 본 시스템은 영어와 함께 한글이 사용되는 분야, 예를 들면, 기계번역에서의 언어 생성, 프로그램에서의 메시지 생성, 철자교정 등에 다양하게 이용될 수 있을 것이다

본 시스템은 기본적으로는 학습 데이터에 의존한다. 따라서, 각각의 분야에서 다른 발음으로 읽고 다른 조사를 붙일 경우에도, 본 프로그램을 다르게 학습시켜 사용할 수 있다. 마찬가지로, 새로운 단어가 나왔을 경우에도 원래 규칙과 다르면, 그 단어를 학습데이터에 넣어 새롭게 학습시켜, 새로운 규칙을 능동적으로 생성해 나갈 수 있을 것이다. 본 시스템은 현재 일반 단어에 대해서만 실험하고 테스트하였지만, 약어나 한글자 단위로 쓰여진 영어 알파벳, 일반적인 숫자나 특수 기호 등에 대해서도 읽는 방법만 제공하여 학습데이터에 넣게 되면, 그 처리가 가능하다.

참고문헌

[1] 이재성, 최기선, "정보검색을 위한 외래어 자동표기 모델," 제4회 한국 정보관리학회 학술대회 논문집, 1997

[2] Jae Sung Lee and Key-Sun Choi, "English to Korean statistical transliteration for information retrieval," *Computer Processing of Oriental Languages*, Vol. 11, No. 4, 1998. (to appear)