論文99-36C-4-6

형태소 분석 기법을 이용한 음성 인식 후처리

(Postprocessing of A Speech Recognition using the Morphological Analysis Technique)

朴 美 星 *, 金 美 辰 *, 金 桂 成 *, 金 城 圭 *, 李 文 熙 *,
崔 宰 赫 **, 李 相 祚 *

(Mi Sung Park, Mi Jin Kim, Kye Sung Kim, Sung Kyu Kim, Mun Hee Lee, Jae Hyuk Choi, and Sang Jo Lee)

요 약

연속 음성 인식 결과를 자연어 처리 기술과 접목시키기 위해 처리해야 할 두가지 문제점이 있다. 첫째는 말하는 단위와 문서의 띄어쓰기 단위가 일치하지 않는다는 것이고, 둘째는 발음시 형태소 내부 및 형태소 간에 음운 변동 현상이 생긴다는 것이다. 본 논문에서는 이 두가지 문제를 어절생성기와 음절복원기로 해결하고, 생성된 결과들을 형태소 분석하여 실패한 결과들은 교정기를 통해 교정하는 연속 음성 인식 후처리 시스템을 구현하였다. 제안한 시스템의 실험은 두 종류의 음성 말뭉치 즉, 교과서 음성 말뭉치와 사설 음성 말뭉치를 대상으로 수행하였다. 각 말뭉치에 대한 성공률은 각각 93.72%, 92.26% 였고, 이 실험으로 제안한 시스템은 음성 말뭉치의 종류에 민감하지 않는 안정된 시스템임을 알 수 있었다.

Abstract

There are two problems which will be processed to graft a continuous speech recognition results into natural language processing technique. First, the speaking's unit isn't consistent with text's spacing unit. Second, when it is to be pronounced the phonological alternation phenomena occur inside morphemes or among morphemes. In this paper, we implement the postprocessing system of a continuous speech recognition that above all, solve two problems using the eo-jeol generator and syllable recoveror and morphologically analyze the generated results and then correct the failed results through the corrector. Our system experiments with two kinds of speech corpus, i.e., a primary school text book and editorial corpus. The successful percentage of the former is 93.72%, that of the latter is 92.26%. As results of experiment, we verified that our system is stable regardless the sorts of corpus.

* 正會員, 慶北大學校 컴퓨터工學科
(Dept. of Computer Engneering, Kyungpook National University)
** 正會員, 新羅大學校 컴퓨터敎育科
(Dept. of Computer Education, Sila University)
※ 이 논문은 1997년 학술진흥재단의 공모과제 연구비에 의하여 연구되었음.
接受日字:1998年11月19日, 수정완료일:1999年3月23日

Ⅰ. 서 론

컴퓨터와 사용자의 의사 소통 수단으로 키보드를 이용하는 방법외에 문서 인식에 의한 방법과 음성인식에 의한 방법을 생각할 수 있다. 문서 인식을 통하는 방법은 문서를 스캐너로 인식해야 한다. 이는 마이크를 통해 보통의 상황과 같이 말하면 되는 음성인식에 비해서는 번거로움이 많다. 그런데 자연스로운 음성으로 문서를 입력하려면 연속적인 음성을 인식하여

문서화할 수 있는 STT(Speech To Text)시스템이 필요하다. 이러한 시스템을 위해서는 연속 음성 인식이 필수적이고 연속 음성 인식을 위해서는 인식된 결과 즉, 말하는 단위를 문서의 띄어쓰기 단위에 맞추어 주고, 발음시 일어나는 음운 변동 현상을 복원시켜 주는 후처리가 선행되어야 하고 이를 형태소분석, 구문분석, 의미분석까지 가능하게 하여야 한다. 이를 위해서는 자연어 처리 기술과의 통합이 필수적으로 요구된다[1,2,3,4,6].

지금까지 연속 음성 인식 결과를 자연어 처리 기술과 접목시켜 시도한 예는 많지 않지만 [1], [2], [3], [4]에서 이와 관련한 연구를 하였다. [1]에서는 다이폰 인식기를 기반으로 형태소 분석 결과를 출력하는 음성 언어 처리 모델을 제안하였고, [2]에서는 음성인식과 자연어 처리의 연결 구조로서 형태소 그래프를 정의하여 구축했고 한국어의 특성에 맞는 후처리와의 결합을 위해서 한국어 음운 현상을 선언적으로 모델링하고 형태접속과 음운 접속을 이용한 morpheme class pair 언어 모델을 제안했다. [3], [4]에서는 음성인식의 낮은 인식률로 한 음소에 대해 여러개의 후보를 제시하는 음성인식 결과에 대하여 어절간의 접속을 위한 확장된 접속검사, 음운변동을 고려한 음소열 사전, 음운 접속 정보를 사용하는 형태소 분석 방법을 제안했다. 이상의 방법들은 모두 음성인식의 결과를 음소열 사전, 발음열 사전, 형태 접속, 음운 접속 정보와 같은 정보를 미리 구축해서 이를 참조로 하여 인식된 음소열들을 형태소 분석하는 음성 언어 처리 모델들이다. 이와 관련한 세부 연구 분야 중 음운 경계 설정에 관한 기존 연구로는 주로 문서의 띄어쓰기에 관련된 연구인데 음성문자열의 경계에 대한 연구로는 [5]가 있다. 이는 대용량의 음성을 음소열로 바꾸어 연속적으로 입력하면서 다음에 올 음소를 예측하고, 이 예측률이 저조한 위치를 음운의 경계로 보고 분절하였다. 음운현상 복원에 관한 연구로는 [3], [6], [7], [8]이 있는데 [3]은 음성인식의 결과로 나온 음소열을 표제어로 하여 그 음소열이 발음될 수 있는 모든 단어를 사전에 수록하였다. 이는 확장할 때 사전량이 매우 커지고 구축하는데 시간도 많이 걸린다. [6]도 [3]과 마찬가지로 발음규칙을 기반으로 하여 발음열 사전을 미리 만들어 놓고 형태소 분석에 이용하므로 사전량이 커지는 단점이 있다. [7]은 읽기 규칙을 역으로 적용한 방식으로 이는 규칙 적용 순서가 복잡하고 예외 처리가 너무 많으며 [8]은 자소 단위 사전을 이용하여 형태소 단계에서 음운변동현상을 처리한 방식으로 음운변동현상 모두를 수용하지 못했으며 사전 검색 횟수가 많다는 단점이 있다.

본 논문에서는 음성 인식 후처리에서 처리해야 할 문제점[4,6] 즉, 말하는 단위와 문서의 띄어쓰기 단위가 일치하지 않는다는 것과 발음시 각 형태소들이 형태소 내부에서뿐만 아니라 형태소와 형태소 사이에 음운 변동 현상 일어나는 것을 기존 방법과 달리 어절생성기와 음절복원기를 통해 해결하고 이 결과들을 양방향 최장 일치 형태소 분석을 하여 실패한 결과들은 오류 유형에 따라 교정을 행하고 이를 다시 복원하여 재분석하는 음성 인식 후처리 시스템을 구현하였다. Ⅱ장에서는 음성 인식 후처리 시스템에 관한 전반적인 설명을 하고, Ⅲ장에서는 시스템 적용 예를, Ⅳ장에서는 실험 및 결과 분석을 Ⅴ장에서는 결론을 맺는다.

Ⅱ. 음성 인식 후처리 시스템

1. 시스템 구성도

연속적으로 인식된 한국어 음성 문자열을 입력으로 받아들여 먼저, 문서의 띄어쓰기 단위인 어절 단위로 끊어 주기 위하여 어절의 경계로 추정되는 위치를 설정한다. 다음으로 잘려진 어절을 입력으로 받아 각종 복원규칙에 의거하여 복원을 수행하고, 복원된 결과를 형태소 분석하여 최종결과를 생성한다. 이 때 형태소 분석에 실패한 어절들은 교정기를 통해 오류 유형에 따라 교정을 한 후 다시 복원하여 형태소 분석을 재시도하였다. 전체 시스템 흐름도는 그림 1과 같다.

2. 어절생성기

어절생성기는 연속 음성 문자열의 분리 위치를 3단계로 추정한다. 1단계에서는 휴리스틱 정보를 이용하여 1차 분리 위치를 추정하여 반드시 분할되어야 하는 부분을 설정한다. 2단계에서는 음성 어절 분할 정보 사전을 이용하여 2차 분리 위치를 설정한다. 이 때 최장 조사·어미에 우선 순위를 부여하여 분리 위치를 설정한다. 3단계에서는 한 음절을 사이에 두고 분리 위치가 추정되었을 경우에 두 분리 위치 중 더 정확한 분리 위치를 결정해야 한다. 왜냐하면 한 어절은 조사·어미로 추정되는 1음절로 구성될 가능성이 희박하기 때문이다. 그러므로 한 음절을 사이에 둔 두 분리 위치에서 2음절 이상의 최장 조사·어미가 있으면 우

선적으로 3차 분할 위치로 선택되는데 만약 두 음절 모두 1음절 조사·어미로 추정된 경우에는 음절 분리 가능도를 이용하여 3차 분할 위치를 설정한다.

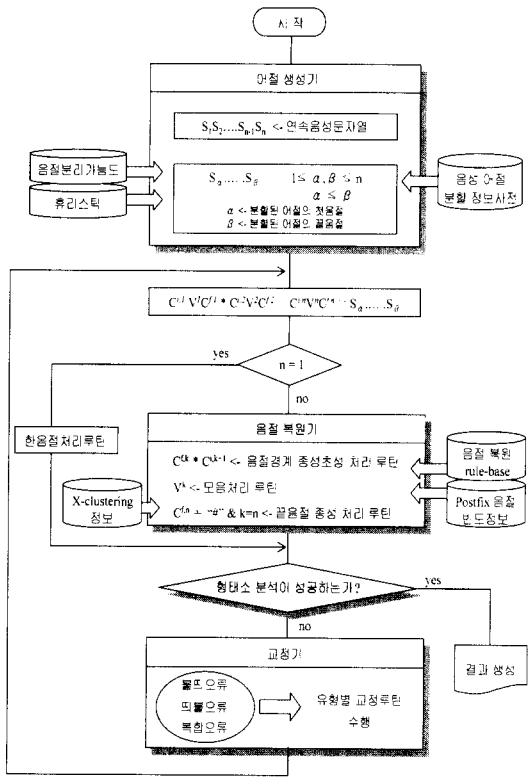


그림 1. 시스템 흐름도
Fig. 1. Flowchart of the system.

1) 휴리스틱 정보

보통 자연어 처리 응용시스템, 즉 교정 시스템이나 띄어쓰기 시스템[9,10]은 말뭉치에서 사용빈도가 높으면서 여러 어절에 걸쳐 사용되는 아래의 예들이 문서상에서 잘 못 사용되었을 때, 즉시 교정이 되거나 띄어쓰기 처리가 가능하다. 이들에 대해서는 선언적으로 정의하여 프로그램에서 처리를 해주기 때문이다.

- ㄹ 수(+있/없), ㄹ 때
- ㄴ 것, ㄹ 것, ㄴ 체, ㄴ 뒤, ㄴ 후 …
- -에 대한/대하여, -에 관한/관하여, -에 의한/의하여
- -에 따라/따른, -기 위한/위하여, -를 위한/위해
- -기 때문에
- 등, 및 …
- 할/한(하다+'ㄹ/ㄴ')
- 될/된(되다+'ㄹ/ㄴ'),적,히,화
- 는, 를, 은, 을(어절의 첫음절이 되지 않도록)

이와 유사하게, 본 논문에서는 위의 예들이 음성문자열에서 어떻게 나타나는 가를 조사하고, 각각의 경우마다 앞 음절의 음운이 어떻게 변동되는 가를 예측하여 가능한 조건들을 모두 고려한 정보를 추출하였다. 이 정보를 휴리스틱 정보라 하고 음성문자열의 어절을 생성하는 데 이용한다. 음성문자열에서 추출한 휴리스틱 정보의 일부를 나타내면 아래와 같다.

- -ㄴ+[채], -ㄴ+[뒤], -ㄴ+[지], -ㄴ+[거/건] …
- -ㄹ+쑤+[인/일/엄/업-] …
- -ㄹ+[쭈/쭐], -ㄹ+[꺼/껃] …
- [-기/끼/키] 위한- …
- [-기/끼/키] 때무네 …
- -에(게,네,레,메,베·ㆍ) 대한- …
- -에(게,네,레,메,베·ㆍ) 의한- …
- 는, 를

본 논문에서는 위와 같이 추출된 휴리스틱 정보를 1차 어절 분할 위치를 추정할 정보로 이용한다. 위의 "는", "를"은 음성 문자열에서 어절의 첫음절로 나타날 수 없으므로 어절의 첫음절이 되지 않도록 "는", "를" 바로 뒤에서 분할 위치를 결정한다. "는"과 "를"은 문자 "은"과 "을"이 앞 음절의 종성 "ㄴ"과 "ㄹ"이 연음되어 나타난 경우도 포함된 것이다.

예를 들어 다음 (a.1)의 입력 음성 문자열에 대해서 1차 분할 위치 설정을 위해 휴리스틱 정보를 이용하면 결과 문장 (a.2)가 생성된다. (a.2)에서 "#0"는 휴리스틱 정보 참조로 분할된 위치를 나타낸다.

(a.1) "모두가자기에이를잘해낼쑤업쓸꺼심니다"
(a.2) "모두가자기에이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

2) 음성 어절 분할 정보사전

음성 어절 분할 정보 사전은 음성문자열의 2차 분리 위치 추정 정보로 사용하는 주요 사전으로, 한국어 어절 구성의 특징과 음운변동을 고려하여 만든 사전이다.

한국어 어절은 보통 '체언+조사'나 '용언+어미'로 결

합되어 있는 경우가 80%이상이며[11], 관형사, 부사 등은 주로 단독 어절을 형성한다. 그러므로 조사, 어미, 관형사, 부사 등의 음절은 어절의 끝을 추정하는 데 이용될 수 있다.

또한 한국어는 발음할 때 음소 결합의 제약성, 발음의 편의, 말의 청취 효과에 따른 명확성 등의 이유로, 한 음절의 초성과 중성, 중성과 종성, 종성과 다음 음절의 초성 사이에 음운 변동이 일어난다. 다음의 예를 살펴보자.

(b.1) 바블 먹꼬 (b.2) 공부하고 (b.3) 놀지 안코

(b.1)~(b.3)에서 "먹+고", "공부하+고", "않+고"가 발음된 형태로, 모두 어미 "고"가 포함된 어절이지만 앞 음절 종성에 따라 다르게 발음되고 있음을 알 수 있다.

따라서 어절의 분할 정보로 사용하기에 충분한 이러한 두 가지 특징들을 고려하여 조사·어미, 부사, 관형사 등의 음절에 대해 각각의 음운 변동 조건과 그에 따라 변동된 음운이 수록된 6000개의 음성 어절 분할 정보 사전을 구축하였다. 음성 어절 분할 정보 사전의 예를 보이면 표 1과 같다.

표 1. 음성 어절 분할 정보 사전
Table 1. Speech eo-jeol's division information dictionary.

음성문자열	앞음절의 종성
가	#
:	:
고	#,ㄹ
고는	#,ㄹ
:	:
과	ㄴ,ㄹ,ㅁ,ㅇ
:	:
기에	#,ㄹ
:	:
꼬	ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ
꼬는	ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ
:	:
꽈	ㄱ,ㄷ,ㅂ
꾸나	ㄱ,ㄷ,ㅂ
:	:
자	#,ㄹ
:	:
한테	#,ㄴ,ㄹ,ㅁ,ㅇ

('#': 앞음절 종성이 비어있음(fillcode)을 의미함)

표 1의 음성문자열 "꼬", "꼬는"의 경우에 현재 위치의 음절이 "꼬" 라고 할 때, 앞 음절의 종성이 {ㄱ, ㄴ,ㄷ,ㄹ,ㅁ,ㅂ} 중의 한 원소이면 "꼬" 뒤에서 분리 위치가 설정될 가능성을 나타낸다. 만약 입력 음성 문자열의 "꼬" 뒤에 "는" 음절이 이어 나타난다면 2음절 이상의 최장 조사·어미가 우선 선택되므로 "는" 뒤에서 분리 위치가 설정된다. 위의 (a.2)의 결과에 음성 어절 분할 정보 사전을 이용하면 다음의 결과 (a.3)이 생성된다. (a.2)의 "가", "자", "기에" 세 음절들이 음성 어절 분할 정보 사전 검색에서 발견되고 조건에 부합되므로 2차 분할 위치로 결정된 것이다.

(a.2) "모두가자기에이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

(a.3) "모두가#1자#1기에#2이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

위 (a.3)에서 "#1"은 앞 한음절이 음성 어절 분할 정보 사전에 존재하여 분할된 위치를 의미하고, "#2"는 앞 음절이 사전에 존재하여 분할된 위치를 나타낸다.

3) 음절 분리 가능도

음절 분리 가능도는 음성문자열 3차 분리 위치 추정 정보로 사용된다. 한국어의 조사·어미 음절은 음절의 특성상 때때로 체언이나 용언의 일부가 될 수 있다. 이 때, 체언, 용언의 일부가 되는 음절이 조사·어미로 인식되어 분리 위치가 잘못 추정되거나 두 군데 이상의 분리점이 생겨 모호성을 발생시킬 수 있다. 이러한 모호성에 대한 해결책으로 각 음절의 특성을 고려한 음절 분리 가능도 $P(S_i)$를 다음과 같이 계산하여 이용한다.

(1) 음절 분리 가능도 $P(S_i) = \log Pe(S_i)/Pf(S_i)$

음절 분리 가능도 $P(S_i)$는 음절 S_i가 끝음절로 사용될 가능성을 나타낸다. $Pe(S_i)$는 S_i가 어절의 끝음절로 사용될 확률을 가리키는데 약 60만 어절의 말뭉치에서 추출하였다. $Pe(S_i)$는 다음과 같이 계산하여 구한다.

(2) $Pe(S_i)= S_if_i /S_1f_1+S_2f_2+\cdots+S_if_i+\cdots+S_nf_n$

(2)에서 S_1, S_2, S_i, S_n는 말뭉치에서 끝음절로 사용되는 각각의 음절을 나타내고, f_1, f_2, f_i, f_n는 말뭉치에서 끝음절로 사용된 각 음절의 빈도값이다. 그리고 $Pf(S_i)$는 S_i가 체언이나 용언의 첫음절로 사용될 확률을 나타내는 것으로 약 23만 단어가 수록된 어휘사전에서 모든 어휘를 음성문자열로 변환시킨 뒤 추출하였다. $Pf(S_i)$는 다음과 같이 계산하여 구한다.

(3) $Pf(S_i)= S_if_i' /S_1f_1' + S_2f_2' +\cdots+ S_if_i' +\cdots+ S_nf_n'$

(3)에서 f_1', f_2', f_i', f_n'는 어휘사전에서 첫음절로 사용된 각 음절의 빈도값이다. 구한 음절 분리 가능도의 개수는 약 1500개이고, 각 음절의 분리 가능도의 일부를 보이면 표 2와 같다.

표 2. 음절 분리 가능도
Table 2. Syllable division probability measure.

순위	음절	Pe(S_i)	Pf(S_i)	P(S_i)
1	를	6.997	0.001	3.844912
2	른	1.584	--	3.199755
3	는	9.195	0.006	3.185259
4	떤	0.912	--	2.959995
5	믈	1.683	0.002	2.924796
:	:	:	:	:
:	에	2.191	0.529	0.617210
:	:	:	:	:
:	가	3.277	1.222	0.465531
:	:	:	:	:
:	자	0.168	1.100	-0.815309
:	:	:	:	:

표 2에서 $Pf(S_i)$ = "--"인 음절은 어휘 사전에서 첫음절로 사용되지 않는 음절이므로 "0.001"을 기본값으로 주어 음절 분리 가능도를 계산하였다. 위 (a.3) 결과에 3차 분할 정보인 음절 분리 가능도를 적용한 결과는 다음 (a.4)와 같다.

(a.3) "모두가#1자#1기에#2이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

(a.4) "모두가#1자기에#2이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

위 (a.3)의 일부 "모두가#1자#1기에"에서 "자"가 한 음절로 분할된 경우이므로 이 때 음절 분리 가능도가 적용되는데, "자"를 기준으로 앞음절 "가"가 한 음절만 보고 분할된 경우 즉 "#1"이므로 이때 "가"의 분리 가능도와 "자"의 분리 가능도를 비교하게 된다. "가"의 분리 가능도가 "자"보다 높으므로 "가"뒤를 분할 위치로 결정하고 "자"는 다음 음절의 "기에"와 결합되어 "자기에"가 된다.

이상의 어절생성기에서 어절의 분할 위치를 추정하면 다음의 음절복원기에서는 추정된 어절을 기본 입력으로 받아 복원을 수행한다. 음성문자열은 여러 개의 어절로 이루어져 있고, 한 어절은 하나 이상의 음절로 구성된다. 그러므로 어절생성기를 통한 연속 음성 문자열 $S_1S_2\cdots\cdots S_{n-1}S_n$는 몇 개의 $S_\alpha\cdots S_\beta$ ($1 <= \alpha, \beta < n, \alpha<=\beta$) 크기로 분할이 되고, 분할된 각 어절 $S_\alpha\cdots S_\beta$는 각 음절이 복원기의 입력 패턴 초성(C^i)중성(V^i)종성(C^f)의 형태에 맞도록 대응되어 복원기로 입력된다. 복원기의 입력은 어절 단위이므로 어절생성기의 S_α가 항상 첫번째 음절로 대응되고 S_β는 n번째 음절로 대응된다. 즉 $S_\alpha=C^{i,1}V^1C^{f,1}$, $S_{\alpha+1}=C^{i,2}V^2C^{f,2}\cdots$, $S_\beta = C^{i,n}V^nC^{f,n}$ 으로 대응된다. 그러므로 복원기의 입력은 다음과 같이 표현할 수 있다.

(4) $S_\alpha\cdots S_\beta = C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*\cdots*C^{i,k}V^kC^{f,k}*\cdots *C^{i,n}V^nC^{f,n}$

(4)에서 i는 초성, f는 종성을 의미하며, 1,2,k,n은 한 어절내의 각 음절 위치 즉 첫번째, 두번째, k번째, n번째 음절을 의미한다. 그러므로 $C^{i,1}$는 첫음절 초성을, V^1는 첫음절 중성을, $C^{f,1}$는 첫음절 종성을 가리킨다.

3. 음절복원기

한국어가 연속적으로 발음될 때 여러 가지 음운변동이 일어난다. 음절복원기는 이러한 음운 변동이 반영된 음성문자열을 변동 이전의 문서 기반 문자열로 다시 복원시켜 주는 과정이다. 복원은 음절의 위치와 구성에 따라 정의한 각종 규칙에 의거하여 이루어진다. 복원 규칙은 크게 음절 경계 종성 초성 복원 규칙, 모음 처리 복원 규칙, 끝음절 종성 복원 규칙, 한 음절 처리 규칙으로 나누어진다. 위 (4)의 식에서 복원 규칙이 적용되는 조건은 다음과 같다.

① $1 < k < n$ 일 때, $C^{f,k}*C^{i,k+1}$에 대해서는 음절 경계 종성 초성 복원 규칙을 적용한다.

② $C^{f,k}$ = fill-code(#) and k=n 일 때, $C^{f,k}$ 에 대해서는 끝음절 복원 규칙을 적용한다.

③ $k = \{1,2,3 \cdot\cdot n\}$ 일 때, V^k 에 대해서는 모음 처리 복원 규칙을 적용한다.

④ $C^{f,k}$ = fill-code(#) and k=n=1일 때, $C^{i,k}V^kC^{f,k}$ 에 대해서는 한 음절 처리 규칙을 적용한다.

1) 음절 복원 규칙

(1) 음절 경계 종성 초성 복원 규칙

앞 음절의 종성과 다음 음절의 초성사이에 적용될 수 있는 규칙으로 보통 음성문자열에서 앞음절의 종성

으로 나타날 수 있는 값이 무종성("#" 또는 fill-code)과 7개의 받침 "ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅇ" 이므로 이를 기준으로 규칙을 정의한다. 여기에 해당하는 총 규칙 수는 86개의 주규칙과 231개의 부규칙이 있다.

표 3. 음절 경계 종성 초성 복원 규칙
Table 3. Syllable boundary final-consonant initial-consonant recovery rule.

규칙 No	입력부 -> 출력부
Rule 001 Rule 002 :	$C^{f,k}\{\#\}*C^{i,k+1}\{ㄱ\}->C^{f,k}\{ㄱ,ㄲ,ㄺ,ㅋ^+\}*C^{i,k+1}\{ㅇ,ㅇ,ㅇ,ㅇ^+\}$ $C^{f,k}\{\#\}*C^{i,k+1}\{ㄲ\}->C^{f,k}\{ㄱ,ㄲ,ㅅ\}*C^{i,k+1}\{ㄱ,ㅇ,ㄱ\}$:
Rule 020 Rule 021 :	$C^{f,k}\{ㄱ\}*C^{i,k+1}\{ㄲ\}->C^{f,k}\{ㄱ,ㄲ,ㄺ,ㅋ\}*C^{i,k+1}\{ㄱ,ㄱ,ㄱ,ㄱ\}$ $C^{f,k}\{ㄱ\}*C^{i,k+1}\{ㄸ\}->C^{f,k}\{ㄱ,ㄲ,ㄲ,ㄺ\}*C^{i,k+1}\{ㄷ,ㄷ,ㄷ,ㄷ\}$:
Rule 026 Rule 027 :	$C^{f,k}\{ㄴ\}*C^{i,k+1}\{ㄱ\}->C^{f,k}\{ㄴ\}*C^{i,k+1}\{ㄱ\}$ $C^{f,k}\{ㄴ\}*C^{i,k+1}\{ㄲ\}->C^{f,k}\{ㄴ,ㄵ\}*C^{i,k+1}\{ㄱ,ㄱ\}$:
Rule 038 :	$C^{f,k}\{ㄷ\}*C^{i,k+1}\{ㄲ\}->C^{f,k}\{ㄷ,ㅅ,ㅆ,ㅈ,ㅊ,ㅌ\}$ $*C^{i,k+1}\{ㄱ,ㄱ,ㄱ,ㄱ,ㄱ,ㄱ\}$:
:	:
:	:
Rule 085 Rule 086	$C^{f,k}\{ㅇ\}*C^{i,k+1}\{ㅇ\}->C^{f,k}\{ㅇ\}*C^{i,k+1}\{ㅇ\}$ $C^{f,k}\{ㅇ\} * C^{i,k+1}\{ㅉ\}->C^{f,k}\{ㅇ\}*C^{i,k+1}\{ㅈ\}$

총 규칙 수에 자신의 값이 자신으로 복원되는 경우는 포함시키지 않았다. 각 종성별 규칙의 개수를 살펴보면, 먼저 무종성 (fill-code 또는 "#")에 관한 복원 규칙은 19개의 주규칙과 55개의 부규칙이 있고, ㄱ-종성에 관한 복원 규칙은 5개의 주규칙과 15개의 부규칙, ㄴ-종성에 관한 복원 규칙은 12개의 주규칙과 29개의 부규칙, ㄷ-종성에 관한 복원 규칙은 9개의 주규칙과 35개의 부규칙, ㄹ-종성에 관한 복원 규칙은 16개의 주규칙과 32개의 부규칙, ㅁ-종성에 관한 복원 규칙은 9개의 주규칙과 23개의 부규칙, ㅂ-종성에 관한 복원 규칙은 8개의 주규칙과 24개의 부규칙, ㅇ-종성에 관한 복원 규칙은 8개의 주규칙과 18개의 부규칙이 있다. 음절 경계 종성 초성 복원 규칙 정의의 일부를 보이면 표 3과 같다.

예를 들어 "닦다"의 음성문자열 "닥따"는 앞음절 종성($C^{f,k}=C^{f,1}$)이 "ㄱ"이고 다음 음절 초성($C^{i,k+1}=C^{i,2}$)이 "ㄸ"이므로 표 3의 Rule 021에 따라 4가지의 부규칙 {ㄱ, ㄷ}, {ㄲ, ㄷ}, {ㄳ, ㄷ}, {ㄺ, ㄷ}이 적용될 수 있다. 복원을 하면 "닥다", "닦다", "닧다", "닭다"가 된다. 그러나 세번째 후보 "닧다"는 2.3.2에 정의할 x-clustering 정보에 의해 복원이 미연에 방지되고, 첫번째, 네번째 후보는 형태소 분석에 실패하므로 최종 복원 결과로 "닦다"만 남게 된다.

(2) 모음 처리 복원 규칙

모음을 발음할 때 일어나는 음운 변동을 복원하기 위한 규칙으로 대부분의 모음 발음은 그대로 발음이 되고 "ㅢ" 와 "ㅕ", "ㅖ"가 발음의 편의상 경우에 따라 변동이 일어난다. 예를 들면 희망 [히망], 시계 [시게], 주의 [주이], 협의의 [혀비에] 와 같은 예들의 모음 발음 복원이 이에 해당된다. 모음 처리 복원 규칙은 표 4와 같다.

표 4. 모음 처리 복원 규칙
Table 4. Vowel-process recovery rule.

규칙No	입력부 -> 출력부
Rule 101	$V^k\{ㅏ,ㅐ,ㅑ,ㅒ,ㅓ,ㅔ,ㅕ,ㅖ,ㅗ,ㅘ,ㅙ,ㅚ,ㅛ,ㅜ,ㅝ,ㅞ,ㅟ,ㅠ,ㅡ,ㅣ\}$ $->V^k\{ㅏ,ㅐ,ㅑ,ㅒ,ㅓ,ㅔ,ㅕ,ㅖ,ㅗ,ㅘ,ㅙ,ㅚ,ㅛ,ㅜ,ㅝ,ㅞ,ㅟ,ㅠ,ㅡ,ㅣ\}$
Rule 102	if $C^{i,k}$ = {ㅈ,ㅉ,ㅊ} then $V^k\{ㅕ\}$ -> $V^k\{ㅓ\}$ | $V^k\{ㅕ\}$ else $V^k\{ㅕ\}$ -> $V^k\{ㅕ\}$
Rule 103	if k = 1 then $V^k\{ㅖ\}$ -> $V^k\{ㅔ\}$ | $V^k\{ㅖ\}$ else $V^k\{ㅖ\}$ -> $V^k\{ㅔ\}$ | $V^k\{ㅖ\}$ | $V^k\{ㅡ\}$
Rule 104	if $C^{i,1}$ = {ㅇ} then $V^k\{ㅣ\}$ -> $V^k\{ㅣ\}$ else $V^k\{ㅣ\}$ -> $V^k\{ㅣ\}$ | $V^k\{ㅡ\}$

(3) 끝음절 종성 복원 규칙

한 어절의 마지막 음절에 종성이 있을 경우에 적용되는 규칙으로 5개의 주규칙과 13개의 부규칙이 있다.

표 5. 끝음절 종성 복원 규칙
Table 5. Last syllable final-consonant recovery rule.

규칙 No	입력부 -> 출력부
Rule 201	$C^{f,k}\{ㄱ\}$ -> $C^{f,k}\{ㄲ,ㄳ,ㄺ,ㅋ\}$
Rule 202	$C^{f,k}\{ㄷ\}$ -> $C^{f,k}\{ㅅ,ㅈ,ㅊ,ㅌ\}$
Rule 203	$C^{f,k}\{ㄹ\}$ -> $C^{f,k}\{ㄽ,ㄾ\}$
Rule 204	$C^{f,k}\{ㅁ\}$ -> $C^{f,k}\{ㄿ\}$
Rule 205	$C^{f,k}\{ㅂ\}$ -> $C^{f,k}\{ㅄ,ㅍ\}$

예를 들어 새삶 [새삼], 여덟 [여덜] 과 같은 경우에 마지막 음절 종성의 복원을 위해 적용된다.

(4) 한 음절 처리 규칙

한 어절이 한 음절로 이루어져 있고 종성이 존재할 때 적용되는 규칙으로, 끝음절 종성 복원 규칙과 동일

한 표 5의 규칙을 적용하여 복원한다.

2) x-clustering 정보

x-clustering은 상용조합 2350자에서 "x 받침을 가질 수 있는 초성, 중성 쌍들의 집합"이라 정의하였는데, x는 무종성(#), ㄱ, ㄴ,…, ㅍ, ㅎ 중에 하나가 될 수 있다. 이 정보는 복원시 여러 개의 부규칙을 지닌 임의 주규칙에서 올바른 음절을 생성시킬 수 있는 부규칙만 적용하여 불필요한 복원 과정(path)을 거치지 않도록 하기 위해 사용된다.

x-clustering의 종류는 28가지인 데 일부를 나타내면 표 6과 같다.

표 6. x-clustering의 분류
Table 6. Classification of the x-clustering.

종류(갯수)	{초성,중성}쌍들의 집합
ㄱ-clustering(201)	{가개갸거겨고과괴구궈귀그 ..... 홰회후훼휘휴흐히}
ㄲ-clustering(13)	{겨까꺼나다더바부보서소어파}
ㅆ-clustering(6)	{너모바사써싸}
:	:

예를 들어 "바블"이란 음성 문자열을 복원하기 위해 앞 음절의 종성과 다음 음절의 초성 즉, "#"와 "ㅂ"이 입력 값이 되는데, 이 입력 값에 적용될 수 있는 주규칙은 Rule 008이다. 이 주규칙에 의하면 다음의 세가지 가능한 부규칙이 존재함을 볼 수 있다.

"#" + "ㅂ" -> ① "ㅂ" + "ㅇ"
-> ② "ㅄ" + "ㅇ"
-> ③ "ㅍ" + "ㅇ"

여기서 정의한 x-clustering 정보를 사용하지 않는다면 위 세 가지 경우의 부규칙으로 모두 복원시켜 본 후 ①의 경우만 올바른 문자가 생성되고 ②, ③의 경우는 올바른 문자가 생성될 수 없음을 알게 되어 ②, ③의 복원 후보는 버리게 된다. 하지만, x-clustering 정보(ㅂ-clustering,ㅄ-clustering,ㅍ-clustering)를 사용한다면 "바"의 복원 받침으로 ①의 "ㅂ"만 가능하고 ②의 "ㅄ" 과 ③의 "ㅍ"은 적용할 필요가 없음을 미리 알게 되어 ①번 부규칙만 적용하게 된다. 이로 인해 좀 더 효과적인 복원이 가능하게 된다.

3) 후위 음절 빈도 정보

규칙에 의해 복원된 후보는 한 개부터 여러 개까지 나타났다. 후보들을 조사해 보니 한 음절 한 음절은 문자로 만들어질 수는 있지만 그 음절들이 모여서는 의미를 갖지 못하는 후보가 많이 나타났다. 이처럼 많은 후보가 나올 경우에 모두를 형태소 분석한다면 형태소 분석기의 처리시간이 증가하게 되어 형태소 분석기에 부담을 주게 된다. 그러므로 형태소 분석기의 입력 후보를 제한하는 것이 필요하다. 그래서 단어로 생성될 수 없는 후보를 미리 제한하기 위한 방안으로 임의의 한 음절 뒤에 바로 나타날 수 있는 문자의 빈도값을 말뭉치로부터 구해서 이용한다. 말뭉치로 사용된 것은 신문사설, 초등학교 교과서, 소설로 약 60만 어절이다. 빈도값은 상용조합 2350자를 기준으로 하여 말뭉치로부터 획득하였으며 2350자 중 1777자가 자주 사용되는 문자로 나타났다. 예를 들어 살펴보면, 음성문자열 "감끼는"의 복원 후보는 "감^끼는", "갊^기는", "감기는" 세가지 경우인데 후위 음절 빈도 정보의 참조로 첫번째, 두번째 후보는 결과 후보에서 제외된다. 즉, " ^ "를 기준으로 바로 앞과 뒤 음절은 나란히 함께 나타날 수 없는 음절이라 간주하여 형태소 분석 후보 대상에서 제외시킨다.

4. 형태소분석

어절을 기본 처리 단위로 하여 그 어절을 구성하는 형태소들을 찾아주는 형태소 분석 단계를 거친다. 음성인식 후처리의 형태소 분석은 복원 단계에서 나온 복원 결과들에 대해서 형태소들이 결합하여 올바른 어절을 구성할 수 있는지를 검사하여 비 문법적 후보 어절을 필터링 해준다. 예를 들어 "먹어라"의 음성문자열 "머거라"는 3가지 후보 {머거라|먹을아|먹어라}로 복원이 되어 형태소 분석기로 넘어온다. 각 후보를 형태소 분석하면, "머거라"와 "먹을아"는 분석에 실패하고, "먹어라"만 먹(어간)+어라(어미)로 형태소 분석에 성공한다. 그러므로 최종 결과는 "먹어라"가 생성된다. 본 논문에서는 양방향 최장일치 형태소 분석 방법[11]을 이용한다.

5. 교정기

분할된 어절을 규칙에 의거하여 복원을 하고 이를 형태소 분석하는데, 이 때 한 어절에 대해 형태소 분석이 성공한 후보가 하나도 존재하지 않을 때 어절 분할이 잘못된 것으로 간주하고 교정을 하게 된다.

1) 오류의 유형 및 분석

교정의 대상이 되는 어절 분할 오류는 보통 3가지로 나눌 수 있다. 첫 째는 붙어야 할 위치에서 분할이

이루어져 발생한 오류(이하 붙띄오류)이다. 이 오류는 주로 2음절 이상의 조사·어미가 한 어절을 구성할 때 나타난다. 둘째는 분할이 되어야 하는 위치에서 제대로 분할되지 못하고 붙어서 발생한 오류(이하 띄붙오류)이다. 이 오류는 주로 1음절 명사/의존명사가 포함된 어절에서 많이 나타난다. 셋째는 붙띄오류와 띄붙오류가 복합적으로 나타난 오류(이하 복합오류)이다. 이 세가지 오류의 유형별 백분율은 다음 그림 2와 같다.

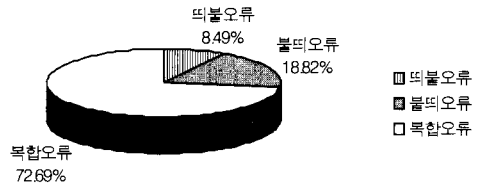


그림 2. 오류 유형별 백분률
Fig. 2. Percentage of each error pattern.

오류 유형 중 복합오류가 72.69%로 가장 많이 나타났음을 알 수 있다. 복합오류가 많은 비중을 차지하는 이유는 체언, 용언의 일부가 되는 음절이 조사·어미 음절로 오인식되어 생긴 모호성 때문이다. 그리고 붙띄오류가 18.82% 나타났는데, 이 오류는 보통 두 어절을 결합시켜 재분석하면 교정이 가능하지만 결합시에 제약을 두지 않으면 과분석을 초래할 수 있다. 그러므로 과분석을 방지하기 위해 결합 음절수에 제약을 두었다. 적절한 음절수를 구하기 위해 60만 어절의 말뭉치를 분석했는데, 얻어진 어절의 길이별 빈도는 표 7과 같다.

표 7. 어절의 길이 빈도
Table 7. Length frequency of the eo-jeol.

어절길이	빈도	어절길이	빈도
1음절	7.3%	7음절	1.0%
2음절	27.6%	8음절	0.28%
3음절	34.8%	9음절	0.15%
4음절	18.2%	10음절	0.06%
5음절	8.0%	:	:
6음절	2.5%	:	:

표 7에 의하면 한 어절을 구성하는 음절수는 대부분이 1~6음절 사이에 존재함을 알 수 있다. 따라서 붙띄오류의 두 어절을 결합하기 위한 조건으로 앞, 뒤 어절의 음절수를 합하여 6음절 이하일 때만 결합하도록 제약을 두었다. 띄붙오류는 8.49%로 가장 적게 나타난 오류인데 이는 뒷어절의 첫음절을 분리해 냄으로 대부분 교정이 가능하다. 오류 유형별 예와 교정되어야 할 결과를 살펴보면 (5), (6), (7)과 같다. E_b는 붙띄오류를, E_t는 띄붙오류를, E_{pb}는 복합 오류중 붙띄오류를, E_{pt}는 복합 오류 중 띄붙오류를 나타낸다.

(5) 붙띄오류(E_b)
강화도 에는 -> 강화도 에는

(6) 띄붙오류(E_t)
걷가타요 -> 걷∨가타요

(7) 복합오류(E_{pb}와 E_{pt})
① 새로 운정채글 -> 새로 운∨정채글
② 되어나 가야 -> 되어∨나 가야
③ 일부기 득꿔늬 -> 일부∨기 득꿔늬
④ 나라 의역싸가 -> 나라 의∨역싸가
⑤ 구겨너 얻씀니다 -> 구겨∨너 얻씀니다

2) 오류 유형별 교정방법

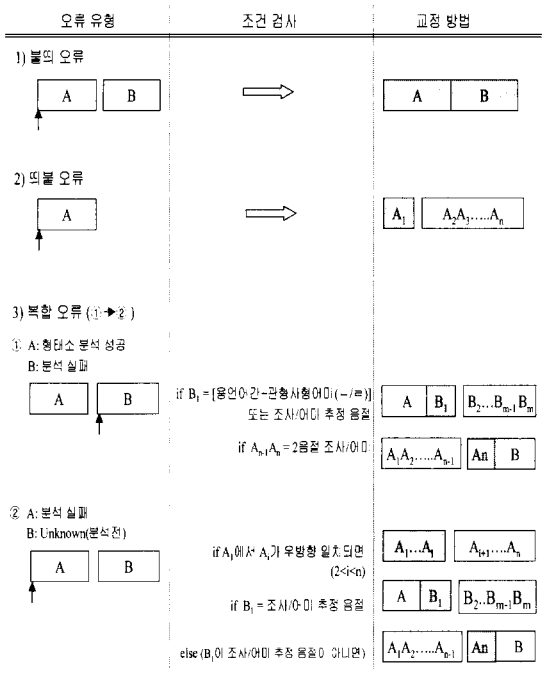


그림 3. 오류 분석 및 교정 방법
Fig 3. The error analysis and correction method.

세 가지 오류 즉, 붙띄오류, 띄붙오류, 복합오류의 유형별 교정방법을 살펴보면 그림 3과 같다. 먼저 교정시 교정 대상이 되는 두 어절을 A와 B라 하고, A, B는 각각 m, n어절로 구성되어 있다고 할 때, A는

$(A_1A_2...A_{n-1},A_n)$로, B는 $(B_1B_2...B_{m-1},B_m)$로 나타낸다. 그리고 ↑는 현재 분석 위치를 나타낸다.

그림 3에서 붙띄오류는 A, B중 하나 또는 모두가 형태소 분석에 실패하고, 두 어절 길이의 합이 6어절 이하(n+m<=6) 일 때 두 어절을 결합시켜 교정한다. 띄붙오류는 여러 어절이 한 어절로 구성되어 있는 경우로 한 음절씩 잘라서 분절해야 적절한 위치를 찾을 수 있다. 복합오류는 조건에 따라 앞 어절의 맨 마지막 음절을 잘라서 뒷 어절에 결합시키거나, 뒷어절의 첫음절을 잘라서 앞어절의 뒤에 붙여나가는 식으로 교정이 이루어진다.

Ⅲ. 적용예

본 시스템에서 제안한 어절생성기, 음절복원기, 형태소분석, 교정기를 통하여 처리되는 과정을 다음의 간단한 예를 통해 살펴보자.

Input :

$S_1S_2S_3S_4S_5S_6S_7S_8S_9S_{10}S_{11}S_{12}S_{13}S_{14}S_{15}S_{16}S_{17}S_{18}$

<-- 어절생성기 -->

$S_1S_2\cdots S_{17}S_{18}$ ← 모두가자기에이를잘해낼쑤업쓸꺼심니다

(a.1) "모두가자기에이를잘해낼쑤업쓸꺼심니다"

(a.2) "모두가자기에이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

: #0 - 휴리스틱참조

(를, ㄹ+쑤+ [인/읻/염/업-], -ㄹ+ [꺼/껃])

(a.3) "모두가#1자#1기에#2이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

: #1, #2 - 음성 어절 분할 정보 사전 참조

("가 : #", "자 : #, ㄹ", "기에 : #,ㄹ")

(a.4) "모두가#1자기에#2이를#0잘해낼#0쑤#0업쓸#0꺼심니다"

: 음절 분리 가능도 비교 참조

(가 : 0.465531 > 자 : -0.815309)

↓

$S_{\alpha1}S_{\beta1}$ = $S_1S_2S_3$ = 모두가

$S_{\alpha2}S_{\beta2}$ = $S_4S_5S_6$ = 자기에

$S_{\alpha3}S_{\beta3}$ = S_7S_8 = 이를

$S_{\alpha4}S_{\beta4}$ = $S_9S_{10}S_{11}$ = 잘해낼

$S_{\alpha5}S_{\beta5}$ = S_{12} = 쑤

$S_{\alpha6}S_{\beta6}$ = $S_{13}S_{14}$ = 업쓸

$S_{\alpha7}S_{\beta7}$ = $S_{15}S_{16}S_{17}S_{18}$ = 꺼심니다

<-- 음절복원기 -->

$C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*\ \cdot\cdot*C^{i,k}V^kC^{f,k}*\ \cdot\cdot*C^{i,n}V^nC^{f,n}$

← $S_{\alpha1}S_{\beta1}|S_{\alpha2}S_{\beta2}|S_{\alpha3}S_{\beta3}|S_{\alpha4}S_{\beta4}|S_{\alpha5}S_{\beta5}|S_{\alpha6}S_{\beta6}|S_{\alpha7}S_{\beta7}$

① $S_1S_2S_3 \rightarrow C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*C^{i,3}V^3C^{f,3}$

= {ㅁㅗ#ㄷㅜ#ㄱㅏ#}

② $S_4S_5S_6 \rightarrow C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*C^{i,3}V^3C^{f,3}$

= {ㅈㅏ#ㄱㅣ#ㅇㅔ#}

③ $S_7S_8 \rightarrow C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}$

= {ㅇㅣ#ㄹㅡㄹ}

④ $S_9S_{10}S_{11} \rightarrow C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*C^{i,3}V^3C^{f,3}$

= {ㅈㅏㄹㅎㅐ#ㄴㅐㄹ}

⑤ $S_{12} \rightarrow C^{i,1}V^1C^{f,1}$

= {ㅆㅜ#}

⑥ $S_{13}S_{14} \rightarrow$ ㅊ $C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}$

= {ㅇㅓㅂㅆㅡㄹ}

⑦ $S_{15}S_{16}S_{17}S_{18} \rightarrow C^{i,1}V^1C^{f,1}*C^{i,2}V^2C^{f,2}*C^{i,3}V^3C^{f,3}*C^{i,3}V^3C^{f,3}$

= {ㄲㅕ#ㅅㅣㅁㄴㅣ#ㄷㅏ#}

↓

< -- ① 의 적용 rule -->

(r.1) $C^{f,2}*C^{i,3}$: $C^{f,k}\{\#\}*C^{i,k+1}\{ㄱ\}$

→ $C^{f,k}\{ㄱ,ㄲ,ㄺ,ㅋ^+\}*C^{i,k+1}\{ㅇ,ㅇ,ㅇ,ㅇ^+\}$

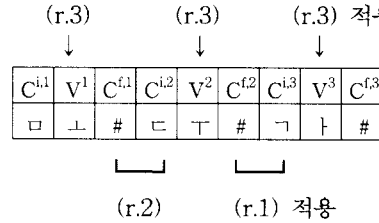
(r.2) $C^{f,1}*C^{i,2}$: $C^{f,k}\{\#\}*C^{i,k+1}\{ㄷ\}$

→ $C^{f,k}\{ㄷ,ㅅ,ㅈ,ㅊ,ㅌ\}*C^{i,k+1}\{ㅇ,ㅇ,ㅇ,ㅇ,ㅇ\}$

(r.3) V^1,V^2,V^3 : $V^k\{ㅏ,ㅐ,ㅓ,ㅔ,ㅜ,ㅟ,ㅡ,ㅑ,ㅒ,ㅕ,ㅖ,ㅘ,ㅙ,ㅛ,ㅝ,ㅞ,ㅠ,ㅢ\}$

→ $V^k\{ㅏ,ㅐ,ㅓ,ㅔ,ㅜ,ㅟ,ㅡ,ㅑ,ㅒ,ㅕ,ㅖ,ㅘ,ㅙ,ㅛ,ㅝ,ㅞ,ㅠ,ㅢ\}$

① "모두가" 복원과정



(r.3)(r.2) (r.3)(r.1)
모두 -> self : 모두 -> self : 모두가(o)
-> ㄱ+ㅇ : 모^둑아(p)
-> ㄲ+ㅇ : x
-> ㄷ+ㅇ : x
-> ㅋ+ㅇ : x
-> ㄷ+ㅇ : x
-> ㅅ+ㅇ : 못우 -> self : 못^우가(p)
-> ㄱ+ㅇ : 못^욱아(p)
-> ㄲ+ㅇ : x
-> ㄷ+ㅇ : 못^옭아(p)
-> ㅋ+ㅇ : x
-> ㅈ+ㅇ : x
-> ㅊ+ㅇ : x
-> ㅌ+ㅇ : x

(x ; x-clustering참조로 비적용되는 path
p ; ^ 위치에서 후위음절 빈도 정보 참조로 제거되는 후보
o ; 최종 복원 후보)

②,③,④,⑤,⑥,⑦도 ①과 같이 복원 수행

== 복원결과 ==
① {모두가}
② {자기에|자기의}
③ {이를|일을|잃을}
④ {잘해낼}
⑤ {수}
⑥ {없을}
⑦ {것입니다}

<-- 형태소분석 -->
① {모두가} : 모두(명사/부사) + 가 : success
② {자기에|자기의} : 자(동사) + 기에 : success
자기(명사) + 의 : success
③ {이를|일을|잃을}: 이(명사) + 를 : success
일(명사) + 을 : success
잃(동사) + 을 : success
④ {잘해낼} : fail
⑤ {수} : 수 : 명사
⑥ {없을} : 없(형용사) + 을 : success
⑦ {것입니다} : 것(명사) + 입니다 : success

== 형태소분석결과 ==
{모두가}{자기에|자기의}{이를|일을|잃을}{F}{수}{없을}{것입니다}

<-- 교정기 -->
{F}->{잘해낼}
->{F}의 앞, 뒤어절 모두 성공
->{오류판명 : 띄붙오류 -> E_t }

$A = \{A_1A_2...A_{n-1},A_n\} = A_1A_2A_3$ = {잘해낼}
if Error = E_t then $A \leftarrow A_1$
$B \leftarrow A_2A_3$
$A \leftarrow$ {잘} $B \leftarrow$ {해낼}
A와 B를 다시 복원후 형태소분석

== 실패어절 복원기 재실행 결과 ==
④-1 {잘}
④-2 {해낼}

== 실패어절 형태소분석 재실행 결과 ==
④-1 {잘} : 자(동사) + ㄹ : success
잘(부사) : success
④-2 {해낼} : 해내(동사) + ㄹ : success

<-- 최종결과 -->
{모두가}{자기에|자기의}{이를|일을|잃을}{F}{수}{없을}{것입니다}

{F} ← {④} : {④-1} {④-2}

Result :
{모두가}{자기에|자기의}{이를|일을|잃을}{잘}{해낼}{수}{없을}{것입니다}

Ⅳ. 결과 분석

1. 실험결과 및 분석

본 실험은 연속 음성 문자열 9,820어절을 대상으로 이루어졌는데 서로 다른 두 종류의 말뭉치, 즉 교과서 음성 말뭉치 3,671어절과 사설 음성 말뭉치 6,139어절

에 대해 각각 행해졌다. 실험 방법은 시스템 구성 모듈 중 교정기를 수행하기 전과 수행한 후의 결과에 대한 성공률 향상을 각각 비교 분석하였다. 그리고 각 모듈별 수행 단계에서 발생되는 실패율과 그 원인을 분석해 보았다. 먼저 교과서 음성 말뭉치와 사설 음성 말뭉치를 각각 A와 B라 하였다. 그리고 실험1은 어절 생성기와 음절복원기를 거쳐 형태소 분석을 수행한 결과이고, 실험2는 실험 1의 결과에 교정 단계를 거쳐 나온 결과이다. 두 가지 실험의 성공률과 실패율을 비교하면 그 결과는 표 8과 같다.

표 8. 각 말뭉치의 성공률과 실패율
Table 8. Success percentage and Fail percentage of the each Corpus.

말뭉치 A	어절생성기	음절복원기 (2-1)	(2-2)	형태소분석 (3-1)	(3-2)
실험1	66.64%	76.36%	23.19%	91.06%	8.94%
실험2	93.72%	79.08%	20.47%	93.72%	6.28%
향상률	27.08%	2.72%	-2.72%	2.66%	-2.66%

말뭉치 B	어절생성기	음절복원기 (2-1)	(2-2)	형태소분석 (3-1)	(3-2)
실험1	68.83%	81.06%	18.34%	90.09%	9.91%
실험2	92.26%	81.08%	18.32%	92.26%	7.74%
향상률	23.43%	0.02%	-0.02%	2.17%	-2.17%

표 8에서 각각 다른 두 말뭉치 A, B에 대해 실험1과 실험2의 결과를 비교해 보면 본 시스템은 말뭉치의 난이도와는 관계 없이 결과가 비슷하다는 것을 알 수 있다. 이는 시스템이 말뭉치의 종류에 민감하지 않는 안정성이 있는 시스템임을 의미한다고 할 수 있다.

어절생성기의 실험1과 실험2의 값은 전체 입력 어절에 대해 올바르게 분리된 어절의 수를 백분률로 계산하여 나타낸 것으로, 실험1과 실험2의 값을 비교해 보면 실험2의 결과가 말뭉치 A에 대해서는 27.08%, 말뭉치 B에 대해서는 23.43%가 향상되었음을 볼 수 있다. 이것은 어절생성기에서 교정단계가 매우 필요한 부분임을 시사한다. 음절복원기의 성공률은 올바르게 분리된 어절에 대해 복원이 올바르게 된 경우를 백분율로 계산하여 구하였으며 (2-1)과 (2-2)의 경우로 나누어 보였다. (2-1)은 올바르게 생성된 어절에 대해 하나의 복원 후보가 생성된 경우의 결과를 나타내고, (2-2)는 복원 후보가 2개 이상 나타난 경우를 나타낸다. 음절복원기의 전체 성공률은 말뭉치 A가 99.5%, 말뭉치 B가 99.4%이므로 실패율은 0.5~0.6%정도이다. 음절복원기는 실험1과 실험2의 결과에 큰 차이가 없으므로 교정 단계를 거쳐도 큰 영향을 받지 않음을 알 수 있다. 그리고 말뭉치 A, B 모두에 대해서 복원 후보가 하나로 결정되는 경우는 약 80%정도이고, 20%정도는 복원 후보가 복수개 생성된 경우이다. 복수개 생성된 이유로는 음성문자열에 대한 동음이형어 즉, 문서상에서 문자의 형태는 다르나 같은 발음을 가지는 경우로, 이 때 복원 후보가 복수개가 나타나는 경우가 많았다. 형태소 분석 단계의 결과값은 올바르게 복원된 결과에 대해 형태소 분석이 성공한 경우(3-1)와 잘못 분리되었지만 올바르게 복원되어 형태소 분석에 성공한 경우(3-2)로 나누어 결과를 구하였다. 말뭉치에 관계없이 (3-1)의 경우가 90%~93%를 차지하고 (3-2)의 결과가 7%~10%정도인데, (3-2)의 경우가 교정을 어렵게 하여 전체 시스템의 성능을 떨어뜨리는 주요 요인이 됨을 알 수 있었다. 형태소 분석에도 말뭉치 A와 B에 대해서 실험 1과 실험 2의 결과가 약간 향상되기는 했으나 큰 차이는 없었다. 실험을 통해 보았듯이 어절 생성기의 성공과 교정의 여부에 따라 전체 시스템의 성능이 좌우되므로 음성 인식 후처리 시스템에 교정기를 시스템의 한 모듈로 포함시켜 전체 시스템을 구현하였다. 그리고 본 시스템의 각 모듈은 독립적으로 수행되고 그 결과를 순차적으로 다음 단계에 넘겨주므로 각 단계별 오류가 다음 단계에 영향을 미친다. 그러므로 각 단계에서의 실패 요인을 최대한 줄여야 한다.

그리고 다음은 전체 시스템 결과에서 실패로 분석된 요인을 몇 가지로 나누었는데, 첫째는 잘못 분리된 연속 어절들이 형태소 분석에 모두 성공한 경우로 분석 결과 (3-2)에 해당하는 경우이다. 이는 교정을 통해 올바른 분석 후보를 생성할 수도 있지만 성공한 어절에 대해서도 모두 교정을 수행하면 과분석을 초래하므로 이 경우는 교정을 수행하지 않았다. 둘째는 어절생성기의 실패율에 해당하는 경우로, 어미 "-ㄴ데"와 의존명사 "데"의 구분이 모호하여 잘못 분리되어 생긴 오류로 이는 의존 명사와 형태가 같은 조사, 어미, 접미사 등을 구분하여야 해결할 수 있는데, 이것은 부분 파싱이나 의미정보 등의 추가로 해결이 가능하다. 그리고 셋째는 합성어에만 해당되는 규칙을 전체 규칙으로 일반화하므로 일부 복원 후보에 영향을 미쳐 엉뚱한

복원 후보가 발생된 경우이다. 이는 복원한 후보들에 대해 약간의 후처리를 하므로 처리가 가능하다. 넷째는 음절복원기의 0.5~0.6%의 실패율에 해당하는 경우로 어절간에 발생될 수 있는 규칙의 미비로 복원이 안된 경우로 어절간 발음에 발생되는 규칙을 좀 더 세밀하게 구분하여 추가함으로써 처리가 가능하리라 본다.

2. 기존 시스템과 비교

다음은 서론에서 언급한 기존의 연구방식과 본 논문의 방식을 몇 가지 관점에서 살펴본다.

방식 / 비교 관점	본 시스템	[4] 이원일	[6] 정민화
음운규칙	사용	사용안함	사용
사전 (형태소분석 사전 제외)	사용안함	음소열사전	발음열 사전
사전검색횟수	적음	빈번	약간 빈번
사전크기	적음	크다	크다
사전구축시간	불필요	많이필요	많이필요
시스템안정성	도메인에 independent	도메인에 dependent	도메인에 dependent
처리방식 (형태소분석)	순차적	복합적	복합적
교정단계	필요	불필요	불필요
형태소분석기	기존 것 이용	변형요구	변형요구
extra정보	많이 사용	약간사용	약간사용

본 논문의 처리 방식은 알고리즘의 모듈이 여러 단계로 나누어져 있고, 각 단계별로 필요한 정보를 이용하여 순차적으로 수행이 이루지는 반면에, [4]와 [6]의 시스템은 음소열사전이나 발음열 사전을 기반으로 하여 형태접속정보 및 음운 접속 정보를 이용하여 형태소 분석을 수행하는 복합적인 방법이라 할 수 있다. 본 시스템은 규칙을 기반으로 하기 때문에 처리가 명료하고 알고리즘이 간결하나 잘못 분석되었을 때 교정단계를 거쳐야 하는 단점이 있다. 그리고 [4]와 [6]에 비해 사전 구축 시간이 필요 없고 사전 크기가 작고 형태소분석기를 그대로 이용할 수 있다는 장점이 있다.

V. 결 론

본 논문에서는 연속 음성 인식 결과를 자연언어 처리 기술과 접목시키기 위해 처리하여야 하는 두 가지 문제점 즉 말하는 단위와 문서의 띄어쓰기 단위간의 불일치 문제, 발음시 형태소 내부 및 형태소 간에 발생하는 음운 변동 현상 처리 문제를 어절생성기와 음절복원기를 통해 해결하고, 이 결과들을 형태소 분석하여 실패한 결과들은 교정기를 통해 교정하는 시스템을 구현하였다.

먼저, 어절생성기에서는 연속된 음성문자열을 문서의 띄어쓰기 단위로 분할하기 위하여 3단계를 거쳐 분할 위치를 추정한다. 1단계에서는 휴리스틱 정보를 이용하여 반드시 분할되어야 하는 분할 위치를 결정한다. 2단계에서는 약 6000개의 분할정보가 담긴 음성 어절 분할 정보 사전을 이용하여 어절의 경계를 추정한다. 3단계에서는 한 음절을 사이에 두고 분할된 경우에 좀 더 정확한 분리 위치를 추정하기 위하여 2음절 이상의 최장 조사·어미 우선 선택 정책과 약 1500개 음절의 음절 분리 가능도를 이용하여 최종 분리 위치를 결정한다.

음절복원기에서는 어절생성기에서 추정되어 나온 어절을 입력으로 하여 음운 변동 현상이 반영되기 전의 문자열로 복원하기 위해 음절 복원 규칙들을 제안하여 사용했다. 제안한 규칙은 크게 음절 경계 종성 초성 복원 규칙(86개의 주규칙과 231개의 부규칙), 모음 처리 복원 규칙(4개의 주규칙과 24개의 부규칙), 끝음절 종성 복원 규칙(5개의 주규칙과 13개의 부규칙), 한음절 처리 복원 규칙(5개의 주규칙과 13개의 부규칙)으로 나누어 정의하였다. 이 때 좀 더 효과적인 복원을 위하여 x-clustering정보와 후위 음절 빈도 정보를 정의하여 이용하였다. 음절복원기를 거쳐 나온 결과들은 양방향 최장 일치 형태소 분석을 행하고 형태소 분석에 실패한 어절들은 교정기를 통해 오류 유형에 맞는 교정을 하여 최종 문서 기반 문자열들을 생성한다.

제안한 시스템의 실험은 두 종류의 음성 말뭉치 즉, 교과서 음성 말뭉치 3,671어절과 사설 음성 말뭉치 6,139어절을 대상으로 수행하였다. 각 말뭉치에 대한 성공률은 각각 93.72%, 92.26%였고, 실패율은 각각 6.28%, 7.74%였다. 이 실험으로 제안한 시스템은 음성 말뭉치의 종류에 민감하지 않는 안정된 시스템임을 알 수 있었다.

그리고 실패한 결과에 대한 주요 요인으로는 몇 가지가 나타났는데, 첫째는 잘못 분리된 연속 어절들이 형태소 분석에 모두 성공한 경우, 둘째는 연속 어절 분리시 어미 "-ㄴ데"와 의존명사 "데"의 구분이 모호

하여 잘못 분리된 경우, 셋째는 합성어에만 적용되는 복원 규칙을 일반화하므로 비합성어에 영향을 미쳐 잘못된 복원후보가 생성된 경우 넷째는 어절간에 발생될 수 있는 규칙의 미비로 복원이 잘못된 경우였다.

이상의 4가지 문제점에 관한 연구가 좀 더 세밀하게 이루어져야 하고, 더불어 이 시스템의 결과 문자열에 대한 구문분석과 의미분석이 이루어진다면 이 시스템은 음성을 통한 문서 입력과 같은 STT(Speech-To-Text)시스템에서 뿐만 아니라 상호 회화에 의한 통역전화, 강연의 동시 통역, 국제전화교환 등의 음성 번역 시스템과 같은 응용시스템에도 유용하게 사용될 수 있을 것이다.

참 고 문 헌

[1] 김경희, 이근배, 이종혁, "한국어 음성언어 처리를 위한 음소 단위 인식과 형태소 분석의 결합", 정보과학회 논문지(B), 제22권 제10호, 1995년

[2] 김병창, 이원일, 이근배, 이종혁, 이영직, "형태소 그래프를 이용한 한국어 연속음성인식과 형태소분석의 통합", 한국정보과학회 가을 학술발표논문집, pp549-552, 1996

[3] 이원일, "신경망과 CYK-table을 이용한 음성언어의 분석", 석사학위논문, 포항공과대학, 1992

[4] 이원일, 이근배, 이종혁, "한국어 음성인식 결과의 선언적 형태소 분석", 제6회 한글 및 한국어 정보처리 학술대회 , p322-325, 1994

[5] 이찬도, "음성인식합성을 위한 한국어 운율단위 음운론위 계산적 연구: 음운단위에 따른 경계의 발견", 정보처리논문지 제4권 제1호, pp. 280-287, 1997

[6] 정민화, "한국어 연속음성인식을 위한 자연언어처리 기술의 적용방법", '98 지능기술 튜토리얼, pp27-55, 1998

[7] 서상현, "한글 음운 규칙에 기반한 음절 복원기 구현", 경북대학교, 석사학위논문, 1997

[8] 이근용, 이기오, 안동언, 이용석, "한국어 음성인식 후처리를 위한 음운변이 처리", 한국정보과학회 봄 학술발표논문집, pp 927-930, 1996

[9] 김계성, "음절 정보를 이용한 한국어 띄어쓰기 시스템의 구현", 경북대학교, 석사학위논문, 1998

[10] 최재혁, "양방향 최장일치법을 이용한 한국어 띄어쓰기 자동 교정 시스템", 제9회 한글 및 한국어 정보처리 학술대회 발표 논문집, pp.145-151, 1997

[11] 최재혁, "양방향 최장일치법에 의한 한국어 형태소 분석기의 구현", 경북대학교, 박사학위논문, 1993

[12] 허웅, "국어음운학", 정음사, 1982

저 자 소 개

朴 美 星(正會員) 第 36卷 C篇 第 3號 參照

金 美 辰(正會員) 第 36卷 C篇 第 3號 參照

金 桂 成(正會員) 第 34卷 C篇 第 5號 參照

金 城 圭(正會員)

1975년 10월 17일생. 1998년 2월 경북대학교 컴퓨터공학과 공학사. 1999년 현재 경북대학교 컴퓨터공학과 석사과정 재학. 주관심분야는 정보검색, 인식후처리

李 文 熙(正會員)

1974년 6월 10일생. 1998년 2월 경일대학교 컴퓨터공학과 공학사. 1999년 현재 경북대학교 컴퓨터공학과 석사과정 재학. 주관심분야는 자연어처리, 음성인식 후처리, 문자인식 후처리

崔 宰 赫(正會員) 第 36卷 C篇 第 3號 參照

李 相 祚(正會員) 第 33卷 B篇 第 4號 參照