

## 모바일 환경을 고려한 규칙기반 음성인식 오류교정

김진형\*, 박소영\*\*

# Rule-based Speech Recognition Error Correction for Mobile Environment

Jin Hyung Kim\*, So-Young Park \*\*

### 요약

본 논문에서는 모바일 환경에서 음성인식한 결과에 포함된 오류를 교정하는 규칙기반 접근방법을 제안한다. 제안하는 방법은 처리시간이나 메모리에 제약을 받는 모바일 환경을 고려하여 다음과 같이 구성된다. 오류 교정 속도를 최소화하기 위해서, 음절 해체 및 조합 과정이나 형태소 분석 등의 처리를 줄이고, 최장일치 규칙 선택기준을 바탕으로 오류 발생 추정 지점에서 교정 후보도 하나만 생성한다. 제안하는 방법은 메모리를 효율적으로 사용하기 위해서, 어절사전이나 형태소분석기를 사용하지 않고, 규칙도 유형별로 따로 구분하지 않고 통합하여 저장한다. 제안하는 방법은 모델의 수정 및 유지보수가 용이하도록, 오류교정규칙을 학습말뭉치에서 자동으로 추출하여 구축한다. 실험결과 제안하는 방법은 음성인식 결과에 대하여 정확률을 5.27% 정도 재현율을 5.60% 정도 개선하였다.

▶ Keywords : 텍스트 오류 교정, 음성인식 후처리, 규칙 자동 구축

### Abstract

In this paper, we propose a rule-based model to correct errors in a speech recognition result in the mobile device environment. The proposed model considers the mobile device environment with limited resources such as processing time and memory, as follows. In order to minimize the error correction processing time, the proposed model removes some processing steps such as morphological analysis and the composition and decomposition of syllable. Also, the proposed model utilizes the longest match rule selection method to generate one error correction candidate per point, assumed that an error occurs. For the purpose of deploying memory resource, the proposed

• 제1저자 : 김진형 • 교신저자 : 박소영

• 투고일 : 2012. 06. 12. 심사일 : 2012. 07. 19. 게재확정일 : 2012. 08. 08.

\* 상명대학교 디지털미디어학부(Dept. of Digital Media, SangMyung University)

\* 상명대학교 게임모바일콘텐츠학과(Dept. of Game Design and Development, SangMyung University)

※ 본 연구는 2012학년도 상명대학교 교내연구비를 지원받아 수행하였음.

model uses neither the Eojeol dictionary nor the morphological analyzer, and stores a combined rule list without any classification. Considering the modification and maintenance of the proposed model, the error correction rules are automatically extracted from a training corpus. Experimental results show that the proposed model improves 5.27% on the precision and 5.60% on the recall based on Eojeol unit for the speech recognition result.

▶ Keywords : Text Error Correction, Speech Recognition Postprocessing, Automatic Rule Construction

## I. 서론

기술이 점점 발전함에 따라 사람들은 좀 더 편한 사용법을 선호하게 된다. 이러한 흐름을 고려하여 많은 전자 제품들의 사용법이 개선되고 있다. 그런 변화의 흐름 속에 최근 화제가 되고 있는 것은 음성인식으로 전자제품을 조작하는 것이다. 이런 흐름 속에서 모바일 폰도 예외가 아니다. 모바일 폰의 조작 방법이 버튼을 누르는 방식에서 터치스크린을 이용하는 방식, 그리고 이제 음성인식기술을 바탕으로 음성으로 조작하는 방식으로 넘어가고 있는 시점이다. 스마트 폰이 대중화가 되어가는 현재, 스마트 폰 시장에서는 음성 인식에 대한 관심도는 날마다 높아지고 있다. 애플의 시리(Siri)를 출발선으로 하여 삼성전자와 팬택, 그리고 구글에서도 음성인식 기술을 모바일 폰에 도입하는데 열중하고 있다. 이런 음성인식 기술에 대해 무엇보다 중요한 것은 정확도이다. 음성인식 기술이 실용적으로 사용이 되기 위해서는 응용 시스템이 만족하는 수준의 정확도가 요구되고 있다. 그러나, 실제 발화 환경에서는 잡음 등으로 인해 음성 인식 시스템의 인식률이 다소 떨어질 수 있고, 음성 신호를 문자열로 변환하는 과정에서 다양한 유형의 오류들이 발생할 수 있다[1-2].

음성인식 오류 유형은 크게 띄어쓰기 오류 유형과 철자 오류 유형으로 나누어 볼 수 있다. 특히, 띄어쓰기 오류 교정과 관련된 연구는 많이 진행되어 높은 성능을 보이고 있다 [1][3]. 이러한 높은 성능을 바탕으로 띄어쓰기 후 철자오류를 교정하는 방법이 제안되고 있다[4]. 그러나, 두 가지 오류를 차례대로 교정하는 방법은 앞선 교정의 오류가 전파 될 수 있다. 이러한 오류전파를 차단하기 위해서, 이 두 가지 오류 교정을 동시에 수행하는 확률기반 접근방법[5]이 제안되었는데, 두 가지 오류를 모두 고려하여 오류 교정 후보를 생성하므로, 후보의 수가 지나치게 많아져 성능이 오히려 떨어질 수 있다[4].

본 논문에서는 음성인식 오류를 교정하기 위한 규칙기반 접근방법을 제안한다. 제안하는 방법은 처리시간이나 메모리에 제약을 받는 모바일 기기 환경[2][6]을 고려하여 띄어쓰

기 오류와 철자 오류를 한 번에 교정한다. 음성인식 오류 교정 후보수가 늘어나지 않도록, 오류의 발생 추정 위치에서 최장일치 규칙 하나를 선택하여 적용한다.

앞으로, 2장에서는 음성인식 결과의 오류 유형과 텍스트 오류 교정과 관련된 기존연구에 대해서 살펴본다. 그리고, 3장에서는 제안하는 규칙 기반 음성인식 오류 교정 모델에 대하여 설명하고, 4장에서는 제안하는 모델을 실험 및 평가한다. 마지막 5장에서는 결론을 맺는다.

## II. 관련 연구

### 1. 음성인식 결과의 오류 유형 분석

음성인식 오류 유형에 대해 분류하면 [표1]과 같다. 먼저, 띄어쓰기 오류유형은 정답과 음성인식결과의 띄어쓰기가 다른 경우를 나타낸다. 정답 '최소화한다'와 음성인식결과 '최소화 한다'와 같이, 띄어쓰기 오류만 나타난 경우 응용분야에서 활용하는데 크게 영향을 주지 않을 수도 있지만, 정답 '45분'과 음성인식결과 '4 10 5분'과 같이 추후 응용 분야에서 활용할 때 문제가 발생할 수 있으므로 오류교정이 필요하다.

삽입 오류 유형은 정답에 없던 문자열이 추가된 경우를 나타낸다. 정답 '문자나'와 음성인식 결과 '문자나라'와 같이 오류 예제를 살펴본 결과 음성인식기에서 사용한 학습 데이터에 자주 나타난 단어에 영향을 받은 것으로 추정된다. 반대로, 삭제 오류 유형은 정답에 있던 문자열을 제거한 경우를 나타내는데, 실제 발화 환경에 잡음이 많아서 일부 문자열을 인식하기 못하였기 때문으로 추정된다.

치환 오류 유형은 한글 단어를 다른 한글 단어로 잘못 인식된 경우가 대부분이지만, 정답 'TV'와 음성인식 결과 '비리'처럼 영어 단어를 한글 단어로 잘못 인식하거나, 정답 '내달로'와 음성인식 결과 'that allow'처럼 한글 단어를 영어 단어나 숫자로 잘못 인식한 경우도 있었다.

[표1]에서 제시된 바와 같이, 대부분의 음성인식 오류는

표 1. 음성인식 오류 예제  
Table 1. Speech Recognition Error Examples

오류 유형	정답	인식결과
띄어 쓰기 오류	두께를 최소화한다 8시 45분에 출발하는 향후 다양한 자연색상을	두께를 최소화 한다 8시 4 10 5분에 출발하는 짱구 태양 한자 연 책상에
삽입 오류	문자나 인터넷 서핑을 하면서도 전체적으로 곡면이 하드웨어 면에서도 기존 제품	문자나 인터넷서핑을 하면서도 전체적으로 옥곡면이 하드웨어하면에서도 기존 제품
삭제 오류	전자는 3일 영국 런던에서 한편 이날 행사장에는 자연의 색상과 질감에	전자는 영국 런던에서 이날 행사다니는 자연 이 시간에
치환 오류	늘어난 배터리 용량으로 후속작이란 점에서 스마트 화면유지 기능이	늘어난 배터리 증량으로 소속작이란 점에서 스마트폰은 유지기능이
영어-한글	슬립하는 tv 같은 다른 제품과 음성인식기능 엑스보이스도 다양한 모션인식 기능도	슬림어는 비리 같은 다른 분들과 음성인식 기능해보이도 다양한 부산인식 기능도
한글-영어	이라고 밝혔다 국내 출시는 내달로 예정했다 전자는 경쟁사들의 견제로	이라고 back to da 국내 출시는 that allow예정했다 전자는 경쟁사there 0
한글-숫자	판매 1위를 굳히겠다고 선언했다 두 회사가 스마트폰 1위를 두고 아이폰에 적용될 칩도	판매 11 19지겠다고 선언했다 2 회사가 스마트폰 1 1 1 2 go 아이폰에 적용된 19

두 가지 이상의 오류 유형이 함께 나타난다. 예를 들어, 정답 '스마트 화면유지'와 음성인식결과 '스마트폰은 유지'는 두 번째 어절의 앞부분에 있던 '화면'을 '폰은'으로 잘못 인식하면서 첫 번째 어절의 뒷부분으로 띄어쓰기가 바뀌었다. 정답 '향후 다양한 자연색상'과 음성인식결과 '짱구 태양 한자 연 책상'에는 '향후'를 '짱구'로, '다양'을 '타양'으로, '색'을 '책'으로 잘못 인식하면서 동시에 띄어쓰기 오류가 함께 나타났다. 정답 '경쟁사들의 견제로'와 음성인식결과 '경쟁사there 0'는 '들'의 'there'로, '견제로'를 '0'으로 잘못 인식하였고 띄어쓰기 오류도 함께 나타났다.

## 2. 기존 연구

효과적으로 텍스트에 나타난 오류를 올바르게 교정하려면 다양한 접근방법이 제안되었다. 이들을 교정 단위에 따라 다음과 같이 나누어 볼 수 있다. 첫째, 어절 단위 교정 방법은 틀린 어절에 대해 편집거리 등을 바탕으로 사전에서 가장 유

사한 어절을 찾아 교정하는 접근방법이다[7-10]. 이 접근방법은 어절이 사전에 없으면 무조건 틀린 어절로 판단하므로 신조어에 대해 취약하다[5]. 그리고 한국어는 여러 형태소가 결합하여 다양한 어절을 생성하므로, 모든 어절을 사전에 포함하기가 쉽지 않다[11].

둘째, 형태소 단위 교정 방법은 정확한 교정을 위해 형태소 분석기를 이용하는 접근방법이다[12-14]. 이 접근방법은 교정 후보마다 매번 형태소 분석을 수행하여 검증하므로 효율성이 떨어진다[11]. 그리고, 미등록어로 인해 올바른 어절이 형태소 분석에 실패하거나, 과분석으로 인해 틀린 어절이 형태소 분석에 성공할 수 있는 약점이 있다[3]. 또한, 형태소 분석기의 전처리로 사용된다면, 동일한 연산을 이중으로 한다는 문제점이 있다[5].

셋째, 음절 단위 교정 방법은 음절을 바탕으로 오류를 교정하는 접근방법이다[15]. 이 접근방법은 미등록 어절에 대해서도 견고하게 분석할 수 있다[3]. 특히, 한국어 띄어쓰기 오류교정 모델에서 음절 단위를 바탕으로 한 확률모델이 많이 제안되었다[1][3]. 오류 교정의 정확도를 높이기 위해서는 문맥을 충분히 고려해야 한다.

넷째, 자소 단위 교정방법은 자주 일어나는 철자 오류에 대해 자소 단위로 교정 후보를 생성하고 검증하는 접근방법이다[5][16-17]. 이 접근방법은 음절 단위보다 더 자세하게 교정할 수 있지만 생성하는 교정 후보가 많아지기 때문에 교정 복잡도가 크게 증가하고 최적 교정후보를 찾는 것을 방해한다[4]. 또한 주변 문맥을 충분히 고려하지 않으면 부정확한 교정을 할 수 있다.

다섯째, 단계별 교정방법[4]은 띄어쓰기 교정, 음절 단위 철자 교정, 어절 단위 철자 교정을 단계별로 수행하는 방법이다. 이 접근방법은 띄어쓰기 및 철자 오류 교정을 위해서 음절단위를 사용하므로, 다양한 오류를 견고하게 교정할 수 있다. 또한, 어절단위로 철자오류를 교정하므로, 정확하게 교정할 수 있다. 그러나, 다단계접근방법으로 단계마다 오류 교정 처리시간이 요구된다는 한계가 있다.

본 논문에서는 모바일 환경에서 음성인식 오류를 교정하기 위한 규칙기반 접근방법을 제안한다. 제안하는 방법은 처리시간이나 메모리에 제약을 받는 모바일 환경을 고려하여, 어절 사전이나 형태소분석기를 따로 사용하지 않는다. 또한, 자소 단위로 교정시 입력된 음절열을 자소열로 해체하였다가 교정 후 다시 음절열로 조합하는데 처리시간이 소요되므로, 자소단위 교정은 하지 않는다. 처리 속도 개선을 위해, 띄어쓰기 오류와 철자 오류를 한 번에 교정한다. 즉, 띄어쓰기 규칙, 음절 단위 교정규칙, 어절단위 교정규칙을 구분하지 않고 규칙사전

에 저장한다. 그리고, 오류 발생 추정 위치에 적합한 규칙중 가장 긴 문맥을 지닌 규칙을 규칙사전에서 선택하여 적용한다. 따라서, 오류 발생 추정위치에서 오류 교정후보가 1개만 생성되므로, 교정 복잡도가 증가하지 않는다.

### III. 규칙기반 음성인식 오류 교정

본 논문에서는 [그림 1]과 같이 규칙 자동 구축 단계와 음성인식 오류 교정 단계로 구성된 규칙기반 음성인식 오류교정 방법을 제안한다. 제안하는 방법은 모바일 환경에서 음성인식된 결과를 입력으로 받아 띄어쓰기 오류 및 철자 오류를 교정하고, 그 결과를 출력한다. 제안하는 음성인식 오류교정에 사용되는 규칙은 정답 텍스트와 이에 대한 음성인식 결과를 쌍으로 하는 음성인식 오류교정 말뭉치에서 자동으로 구축된다. 앞으로, 1절에서는 규칙 자동 구축 방법에 대해 설명하고, 2절에서는 규칙 적용 방법에 대해 기술한다.

#### 1. 규칙 자동 구축

규칙 자동 구축 단계는 [그림2]와 같이 규칙 후보 추출 단계와 규칙 후보 제거 단계로 구성된다. 정답과 음성인식결과 의 쌍으로 구성된 학습말뭉치가 주어지면, 규칙 후보 추출 단계에서는 학습말뭉치에서 오류교정과 관련된 모든 가능한 규칙 후보를 추출하고, 규칙 후보 제거 단계에서는 추출한 각 규칙에 대해 학습집합에 재적용하여 오류 교정 정확도가 낮은 규칙 후보를 제거한다.

먼저, 규칙 후보 추출 단계에서는 정답과 음성인식 결과를 비교하여 일치하지 않는 모든 부분을 오류교정 규칙으로 추출한다. 이 때, 불일치 부분뿐만 아니라, 주변 문맥도 함께 고려하여 오류교정 규칙을 추출한다. 이는 주변문맥을 고려하지 않고 불일치 부분만으로 오류교정규칙을 구축할 경우, 올바른 문자열을 잘못 교정하는 경우가 발생하기 때문이다.

예를 들어, 정답 '느낌이었다면'과 음성인식 결과 '느낌이었

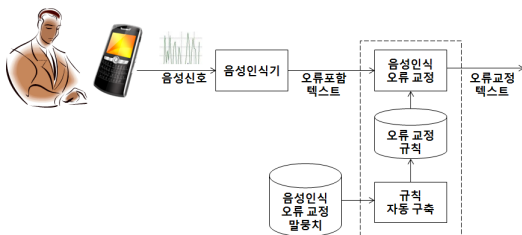


그림 1. 규칙기반 음성인식 오류 교정  
Fig. 1. Rule-based Speech Recognition Error Correction

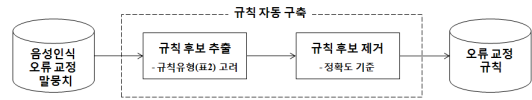


그림 2. 규칙 자동 구축  
Fig. 2. Automatic Rule Construction

다면'이 주어졌을 때, 주변 문맥을 고려하지 않고 불일치 부분만으로 오류교정 규칙 '였 → 었'을 추출한다면, 이 오류교정 규칙은 올바른 어절 '성장기였다면'에 적용되어 '성장기였다면'으로 잘못 교정할 수 있다. 따라서, 음성인식 오류가 발생한 위치의 앞뒤 주변 문맥도 함께 고려해야 오류교정을 보다 정확하게 할 수 있는 규칙을 추출 할 수 있다.

이와 같이, 제안하는 규칙기반 음성인식 오류교정 방법에서는 오류음절열의 앞뒤 문맥을 고려한 5가지의 규칙 유형을 바탕으로 규칙을 추출하여 적용한다. 예를 들어, 신문기사 문장 "전작인 갤럭시S, 갤럭시S2가 각진 직사각형 느낌이었던 이날 공개된 갤럭시S3는 네 모서리 부분에서 곡선이 두드러지는 디자인이 돋보인다."에서 어절 '느낌이었다면'을 '느낌이었다면'으로 잘못 음성인식하였다면, [표2]과 같이 앞뒤 문맥을 바탕으로 5가지 유형을 바탕으로 오류교정 규칙을 추출할 수 있다.

즉, 정답 어절 '느낌이었다면'과 '음성인식 오류어절 '느낌이었다면'에 대해 오류음절 '였'만 고려한 규칙 '였 → 었', 어절 내 앞 음절열을 문맥으로 고려한 규칙 '느낌이었 → 었', 어절 내 뒤 음절열을 문맥으로 고려한 규칙 '였다면 → 었', 어절내 앞뒤 음절열을 문맥으로고려한 규칙 '느낌이었다면 → 었', 어절내 앞뒤 음절열 및 앞뒤 어절을 문맥으로 고려한 규칙 '직사각형 느낌이었던 이날 → 었'을 추출한다.

표 2. 규칙 유형 설명  
Table 2. Rule Type Description

규칙유형	규칙 설명
오류	오류음절열 → 교정음절열 예) 었 → 었
앞 오류	앞문맥 오류음절열 → 교정음절열 예) 느낌이였 → 었
오류 뒤	오류음절열 뒤문맥 → 교정음절열 예) 었다면 → 었
앞 오류 뒤	앞문맥 오류음절열 뒤문맥 → 교정음절열 예) 느낌이였다면 → 었
앞 오류 뒤	확장 앞문맥 오류음절열 확장 뒤문맥 → 교정음절열 예) 직사각형 느낌이였다면 이날 → 었

또한, 정답 문장 “외관과 색상 모두 기존과 다르다”와 음성 인식 오류문장 “왜 반 까지 비상모드 기존과 다르다”는 철자 오류와 띄어쓰기 오류가 함께 나타는데, 이러한 경우 오류음절열을 기준으로 오류교정 규칙 ‘왜 반 까지 비상모드 → 외관과 색상 모두’을 추출한다. 세 어절 전체가 오류 음절열에 해당되므로, 어절내 앞 음절열을 문맥으로 고려한 규칙, 어절내 뒤 음절열을 문맥으로 고려한 규칙, 어절내 앞뒤 음절열을 문맥으로 고려한 규칙은 추출할 수 없다. 그리고, 오류음절열만 고려한 규칙 ‘왜 반 까지 비상모드 → 외관과 색상 모두’이 세 어절 이상으로 구성되어 이미 충분히 문맥을 확보하고 있으므로, 어절내 앞뒤 음절열 및 앞뒤 어절을 문맥으로 고려한 규칙은 추출하지 않는다.

규칙 후보 추출 단계에서 다양한 문맥을 고려한 오류 교정 규칙을 추출하면, 규칙 후보 제거 단계에서는 정확도가 떨어지는 일부 오류교정 규칙을 제거한다. 이를 위해 추출한 모든 오류교정 규칙을 학습집합에 재적용하고, 적용된 횟수에 올바르게 오류를 교정한 횟수를 정확도로 계산하여, 규칙의 정확도가 임계값이하로 떨어지는 규칙은 제거한다.

예를 들어, 정답 ‘사용자의 얼굴과 눈을’과 음성인식결과 ‘사용자에 얼굴과 눈을’에서 추출된 오류교정 규칙 ‘에 → 의’ 이 오류에 적용되어 올바르게 교정할 수도 있지만, 오히려 ‘시연에서는’, ‘홈쇼핑에서’, ‘젊은이에게’와 같은 올바른 음절열에 적용되어 ‘시연의서는’, ‘홈쇼핑의서’, ‘젊은이의게’처럼 틀리게 교정할 수도 있다. 또 다른 예로, 정답 ‘패널 공급사가 이를’과 음성인식결과 ‘패널 공급자가 이를’서 추출된 오류교정 규칙 ‘자가 → 사’는 ‘사용자가 갤럭시로’, ‘삼성전자가 유립을’ 등의 올바른 음절열에 적용되어 ‘사용사가 갤럭시로’, ‘삼성전사가 유립을’ 로 잘못 교정할 수 있다. 이러한 규칙은 학습집합에 재적용하였을 때 정확도가 낮게 평가되므로 제거된다.

반면에, 규칙 후보 제거 단계에서는 충분한 문맥을 확보하여 정확하게 교정할 수 있는 규칙은 오류 교정 규칙으로 남겨둔다. 예를 들어, 어절내 앞뒤 음절열뿐만 아니라 앞뒤 어절도 문맥으로 활용하는 오류 교정 규칙 ‘직사각형 느낌이었다면 이날 → 있’는 학습집합에 재적용하면, 이러한 문맥을 가진 경우는 한 번만 나타나고, 이 경우 오류 교정 규칙을 적용했을 때 올바르게 교정하므로, 규칙의 정확도는 100%로 계산된다. 이와 같이 규칙의 정확도가 임계값 이상인 규칙은 추후 규칙 적용 단계에서 오류교정 규칙으로 활용하게 된다.

## 2. 음성인식 오류 교정

음성인식 오류 교정 단계에서는 규칙 자동 구축 단계에서 생성된 음성인식 오류교정 규칙을 음성인식 결과에 적용하여

교정한다. 즉, 음성인식 결과의 각 위치에서 규칙집합에 있는 오류교정규칙을 하나씩 비교하고, 문맥 및 오류 음절열이 일치하는 경우가 있으면, 음성인식 결과의 그 위치에 오류교정 규칙을 적용하여 교정을 시도한다. 이 때, 적용할 수 있는 규칙이 여러 개 있을 경우에는 길이가 긴 규칙이 선택된다. 또한, 길이가 동일한 규칙이 둘 이상 있을 경우 규칙의 정확도가 더 높은 규칙이 선택된다.

예를 들어, 음성인식 결과 “배터리 종양으로 인해”에서 두 번째 어절 ‘종양’을 기준으로 적용할 수 있는 규칙은 어절내 앞뒤 음절열 및 앞뒤 어절을 문맥으로 고려한 규칙 ‘배터리 종양으로 인해 → 용량’, 어절내 뒤 음절열을 문맥으로 고려한 규칙 ‘종양으로 → 용량’ 및 ‘종양으로 → 불량’이 있다. 어절내 뒤 음절열을 문맥으로 고려한 규칙은 두 가지 모두 앞뒤 문맥의 길이가 동일한데, 학습집합에 재적용하였을 때 규칙의 정확도가 ‘종양으로 → 불량’이 ‘종양으로 → 용량’보다 높다면, ‘종양으로 → 불량’가 선택될 수 있다. 그러나, 이 예제의 경우 오류 교정 규칙 ‘배터리 종양으로 인해 → 용량’의 문맥 길이가 더 길게 나타났으므로, 최종적으로 규칙 ‘배터리 종양으로 인해 → 용량’이 적용된다.

## IV. 실험 및 평가

제안하는 규칙기반 음성인식 오류교정 방법의 성능을 평가하기 위해서, 안드로이드 플랫폼에서 사용하는 구글 음성인식 API를 이용하여 신문기사에 대해 음성인식한 데이터를 수집하였다. 개방된 장소에서 음성인식을 수행하여, [표3]와 같이 음성인식결과에 일부 오류가 포함되었다. 이러한 음성인식 오류에 대하여 제안하는 방법을 적용하여 교정하였고, 오류교정 결과에 대해 수식 (1), (2), (3)과 같이 정확률, 재현율, f-척도를 사용하여 평가하였다.

$$\text{정확률} = \frac{\text{모델이 생성한 올바른 어절수}}{\text{모델이 생성한 어절수}} \quad (1)$$

$$\text{재현율} = \frac{\text{모델이 생성한 올바른 어절수}}{\text{전체 올바른 어절수}} \quad (2)$$

$$f\text{-척도} = \frac{2 \times \text{정확률} \times \text{재현율}}{\text{정확률} + \text{재현율}} \quad (3)$$

음성인식에 사용된 신문기사는 [표3]와 같이 794개 문장과 3,289개의 어절로 이루어져있는데, 이를 음성인식한 결과 794개의 문장과 3,337개의 어절이 인식되었다. 정답 “대검찰청은

표 3. 평가 말뭉치 분석  
Table 3. Evaluation Corpus Analysis

항목	정답	인식결과
문장수	794	794
어절수	3,289	3,337
오류 어절수	-	610

오늘 중수부를"과 인식결과 "대검찰청은 안 오네 중수부를"처럼 한 어절이 두 어절 이상으로 인식되는 경우가 있기 때문에 문장수는 동일하지만 어절수가 다르게 나타났다. 이렇게 준비된 신문기사와 음성인식 결과의 문장쌍 794개에 대해 90%인 715개에서 규칙을 학습하고, 10%인 79개에 규칙을 적용하여 평가하였다.

규칙 유형별 성능을 평가하기 위해서, 각 유형을 바탕으로 추출한 규칙들로 음성인식결과를 오류교정하고, 그 결과에 대해 [표4]와 같이 어절단위로 정확도와 재현율을 평가하였다. 앞뒤문맥을 더 많이 고려할수록 학습집합의 성능이 올라갔는데, 어절내 앞뒤음절열뿐만 아니라 앞뒤 어절까지 고려하는 앞' 오류 뒤' 규칙유형은 정확율이 98.39%이고 재현율이

98.44%로 나타났다. 충분한 문맥을 고려함에도 불구하고 앞' 오류 뒤' 규칙유형에서 1.5%내외의 오류가 발생한 이유는 복합 명사는 띄어 써도 되고 붙여 써도 되기 때문이다. 예를 들어, 복합명사 '아이폰 4s'와 '아이폰4s'는 띄어쓰기 맞춤법 규정상 둘다 맞으므로, 띄어서 쓴 신문기사도 있고 붙여 쓴 신문기사도 있다. 반면에 제안하는 음성인식 오류교정 방법에서는 규칙 '애플 아이폰 4S → 폰'을 적용하여 일관성있게 항상 붙여쓰도록 교정하므로 정확률과 재현율이 1.5%정도 떨어지게 나타났다.

먼저, 오류음절열만 고려하는 규칙 오류의 경우 교정 전보다 오히려 정확률이 2.28%, 재현율이 1.77% 떨어졌다. 이는 "차지하고 읽는 만큼"에서 추출된 오류교정규칙 '읽 → 있'이 올바른 어절 "책을 읽더라도 전자책을"에 적용되어 오히려 틀리게 교정하는 경우가 있었기 때문이다.

한편, 어절내 앞뒤문맥을 고려하는 앞' 오류 뒤' 규칙 유형보다 어절내 앞뒤문맥과 함께 앞뒤 어절까지 고려하는 앞' 오류 뒤' 규칙 유형의 정확률 및 재현율이 더 높게 나타났다. 이는 한국어에서 어절의 평균 길이가 3.17음절이므로[18], 어절내 음절만 규칙의 문맥으로 활용하기에는 너무 짧은 것을 보여준다. 반면에 앞뒤 어절까지 포함할 경우 규칙은 충분한

표 4. 규칙 패턴에 따른 어절 단위 성능  
Table 4. Eojeol based Performance According to Rule Patterns

규칙패턴	규칙 수	실행 속도 (msec.)	메모리 (MB)	학습						실험					
				정확률		재현율		f-척도		정확률		재현율		f-척도	
교정 전	-	-	-	81.72	-	80.85	-	81.28	-	79.77	-	79.65	-	79.71	-
① 오류	326	5.41	1.390	88.83	△ 7.11	88.39	△ 7.54	88.61	△ 7.53	77.49	▽ 2.28	77.88	▽ 1.77	77.68	▽ 2.03
② 앞 오류	387	4.60	1.394	83.58	△ 1.86	85.22	△ 4.37	84.39	△ 3.11	76.90	▽ 2.87	77.58	▽ 2.07	77.24	▽ 2.47
③ 오류 뒤	383	4.10	1.395	89.28	△ 7.56	89.18	△ 8.33	89.23	△ 7.95	83.92	△ 4.15	84.07	△ 4.42	83.99	△ 4.28
④ 앞 오류 뒤	451	5.66	1.393	92.37	△10.64	92.70	△11.85	92.53	△11.25	84.16	△ 4.39	84.37	△ 4.72	84.26	△ 4.55
⑤ 앞 오류 뒤	600	5.25	1.395	98.39	△ 16.67	98.44	△ 17.59	98.41	△17.13	85.04	△ 5.27	85.25	△ 5.60	85.14	△ 5.43
④+⑤	670	9.30	1.404	93.93	△12.21	94.75	△13.90	94.34	△13.06	84.75	△ 4.98	85.55	△ 5.90	85.15	△ 5.44
①+③+⑤	685	9.07	1.404	90.58	△ 8.86	90.88	△10.03	90.73	△ 9.45	83.58	△ 3.81	84.37	△ 4.72	83.97	△ 4.26
①+③+④+⑤	899	10.39	1.403	88.10	△ 6.38	89.15	△ 8.30	88.62	△ 7.34	81.82	△ 2.05	82.60	△ 2.95	82.21	△ 2.50
①+②+③+④+⑤	1,046	9.94	1.403	82.75	△ 1.03	85.92	△ 5.07	84.31	△ 3.02	80.35	△ 0.58	81.71	△ 2.06	81.02	△ 1.31

앞뒤 문맥을 고려할 수 있으므로 성능이 더 높게 나타났다. 교정 전과 비교해 볼 때, 앞 오류 뒤 규칙 유형으로 음성인식 교정한 후 정확률은 5.27%, 재현율에서는 5.60% 상승하였다.

또한, 앞문맥을 고려하는 앞 오류 규칙 유형보다는 뒤문맥을 고려하는 오류 뒤 규칙 유형의 성능이 더 높게 나타났는데, 이는 한국어에서 어절의 앞쪽에 있는 실질형태소보다 어절의 뒤쪽에 있는 형식형태소가 오류교정의 문맥정보로 더 유용하다는 것을 보여준다. 예를 들어, 오류 뒤 규칙 '였다 → 있'은 오류음절 '였'을 포함하는 "뚜껑을 열어보니 예상 밖이었다"나 "세계 최강 스페인이었다"에 모두 적용될 수 있는 반면에, 앞 오류 규칙 '느낌이었 → 있'의 경우 실험집합에서 적용되는 경우가 없었다.

다양한 오류 교정 규칙 유형을 함께 사용하는 경우 어절내 앞뒤문맥뿐만 아니라 앞뒤 어절까지 고려하는 앞' 오류 뒤' 규칙 유형에 비해 대부분 성능이 떨어지는 것을 확인할 수 있다. 어절내 앞뒤문맥을 고려하는 앞 오류 뒤 규칙 유형과 앞' 오류 뒤' 규칙 유형을 조합하였을 때만 재현율이 약간 상승하였다. 이는 앞뒤 문맥정보를 충분히 확보하는 매우 중요하다는 것을 나타낸다.

제안하는 음성인식 오류교정 방법의 실행환경을 알아보기 위해서, intel(R) core(TM) i7 920 cpu, 6기가 램, Windows 7 Enterprise K 64bit 환경에서 문장당 처리속도 및 메모리 크기를 측정하여 평균값으로 계산하였다. [표4]에 제시된 바와 같이, 실행속도는 문장당 5밀리초 이내로 실행되고, 추출된 규칙 중 정확도가 현저히 떨어지는 규칙은 제거하였기 때문에 추출

된 규칙 수는 326개에서 1,046개로 많지 않았고, 메모리는 1.4MB정도 소요되었다.

[표5]는 [표4]의 어절단위 성능에 대하여 문장단위 성능을 분석한 결과이다. 문장내 여러 어절에서 한 어절이라도 틀리면 틀린 문장으로 인식되므로, 어절 단위 성능에 비해 문장 단위 성능은 매우 낮게 나타났다. 규칙의 문맥이 늘어날수록 학습집합의 정확률은 92.44%까지 상승되었지만, 재현율은 60.76%를 넘지 못하였다.

### IV. 결론

본 논문에서는 처리시간이나 메모리에 제약을 받는 모바일 환경에서 음성 인식한 결과에서 나타나는 오류를 교정하기 위한 규칙기반 접근방법을 제안한다. 제안하는 방법의 특징은 다음과 같다.

첫째, 제안하는 방법은 음성인식결과에 나타난 오류를 교정하여 최종 음성인식결과와 정확도를 높였다. 실험집합에서 어절단위 정확률이 79.77%이고 재현율이 79.65%인 음성인식 결과에 대하여 제안하는 규칙기반 오류교정 방법은 정확률을 5.27% 정도, 재현율을 5.60% 정도 개선하였다.

둘째, 제안하는 방법은 불필요한 처리를 최소화하여 처리속도를 줄였다. 제안하는 방법은 음절단위 및 어절단위를 바탕으로 교정을 수행하므로, 자소단위 교정시 필요한 음절해체 및 음절조합을 위한 처리시간을 절약할 수 있고, 형태소 단위 교정시 필요한 형태소 분석 시간을 줄일 수 있다. 또한, 어절 단위 문맥 규칙과 음절단위 문맥 규칙을 구분하지 않고 한 번에 적용하여 교정시간을 줄였다. 특히, 각 오류 발생 추정 위치에서 가능한 모든 교정후보를 고려할 경우 처리시간이 길어지므로, 최장일치 규칙 하나를 선택적용하여 교정후보를 하나만 생성한다.

셋째, 제안하는 방법은 불필요한 리소스를 최소화하여 메모리를 효율적으로 사용한다. 먼저, 오류 교정 규칙의 수를 줄이기 위해서, 띄어쓰기 오류 교정 규칙과 철자 오류 교정 규칙을 구분하지 않고, 음절단위 문맥 규칙과 어절단위 문맥 규칙을 구분하지 않는다. 그리고, 리소스를 줄이기 위해 어절 사전이나 형태소 분석기를 사용하지 않는다. 앞에서 언급한 바와 같이 각 오류 발생 추정 위치에서 최장일치 규칙을 사용하여 교정후보를 하나만 생성하므로, 중간결과물의 생성을 최소화할 수 있다.

넷째, 제안하는 방법은 학습말뭉치에서 음절단위를 위주로 규칙을 자동으로 구축하므로, 새로운 유형의 오류교정도 유연하게 적용할 수 있다. 즉, 학습집합만 주어지면 오류 교정

표 5. 문장 단위 성능  
Table 5. Sentence based Performance

규칙패턴	학습	실험
교정 전	44.58	43.04
① 오류	62.09	54.43
② 앞 오류	44.71	53.16
③ 오류 뒤	62.47	54.43
④ 앞 오류 뒤	70.15	54.43
⑤ 앞' 오류 뒤'	92.44	60.76
④+⑤	75.57	54.43
①+③+⑤	66.50	55.70
①+③+④+⑤	58.19	48.10
①+②+③+④+⑤	44.58	45.27

규칙을 자동으로 구축하여 오류를 교정할 수 있으므로, 모델의 수정 및 유지보수 비용을 절감할 수 있다. 또한, 오류교정 규칙이 형태소단위나 어절단위가 아닌 음절단위를 중심으로 구성되어 있으므로, 신조어에도 비교적 견고하게 오류를 교정할 수 있다.

## 참고문헌

- [1] Dong-Hee Lim, Seung-Shik Kang, Du-Seong Chang, "Word Spacing Error Correction for the Postprocessing of Speech Recognition", Proceedings of Korea Computer Congress, Vol. 33, No. 1, pp. 25 - 27, Jun. 2006.
- [2] Myung-Won Kim, Young-Jin Kim, Eun-Ju Kim, "User Adaptive Post-Processing in Speech Recognition for Mobile Devices", Journal of KIISE, Vol. 13, No. 5, pp. 338 - 342, Oct. 2007.
- [3] Do-Gil Lee, Sang-Zoo Lee, Heui-Seok Lim, Hae-Chang Rim, "Two Statistical Models for Automatic Word Spacing of Korean Sentences", Journal of KIISE : Software and Applications, Vol. 30, No. 4, pp. 358-371, Apr. 2003.
- [4] Jeung-Hyun Byun, So-Young Park, Seung-Wook Lee, Hae-Chang Rim, "Three-Phase Text Error Correction Model for Korean SMS Messages", IEICE TRANSACTIONS on Information and Systems, Vol. E92-D, No. 5 pp. 1213-1217, May. 2009.
- [5] Hun-Gjong Noh, Jeong-Wong Cha, Geun-Bae Lee, "A joint statistical model for word spacing and spelling error correction", Journal of KIISE : Software and Applications, Vol. 34, No. 2, pp. 131-139, Feb. 2007.
- [6] So-Young Park, Jeung-hyun Byun, Hae-Chang Rim, Do-Gil Lee, Heuseok Lim, "Natural language-based user interface for mobile devices with limited resources", IEEE Transactions on Consumer Electronics, Vol. 56, No. 4, pp. 2086-2092, Nov. 2010.
- [7] AiTi Aw, Min Zhang, Juan Xiao, Jian Su, "A Phrase-Based Statistical Model for SMS Text Normalization". Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 33 - 40, Jul. 2006.
- [8] Eric Mays, Fred J. Damerau and Robert L. Mercer "Context Based Spelling Correction", Information Processing and Management, Vol. 27, No. 5, pp. 517-522, 1991.
- [9] Bong-Rae Park, Hae-Chang Rim, "Recognizing Unknown Words and Correcting Spelling Errors as Preprocessing for Korean Information Processing System", Journal of Information Processing Systems, Vol. 5, No. 10, pp. 2591-2599, Oct. 1998.
- [10] Eric Brill, "Transformation-based error-driven learning and natural language processing: A case study in part-of-speech tagging", Computational Linguistics, Vol. 21, No. 4, pp. 543-565, Dec. 1995
- [11] Young-Sin Lee, Young-Ja Park, Man-Suk Song, "Spelling Correction in Korean Using the 'Eojeol' Generation Dictionary", The KIPS Transactions : Part B, Vol. 8B No. 1, pp. 98-104, Feb. 2001.
- [12] Masaaki Nagata, "Context-Based Spelling Correction for Japanese OCR", Proceedings of the COLING, pp. 806-811, Aug. 1996.
- [13] Jin-Hee Yoo, Jong-Hyeok Lee, Geun-Bae Lee, "Post - Processing for Character Recognition Using Morphological Analysis and Linguistic Evaluation", Journal of KIISE, Vol. 22, No. 6, pp. 880-891, Jun. 1995.
- [14] Won-Il Lee, Nam-Hee Hong, Jong-Hyuk Lee, Geun-Bae Lee, "Design and Implementation of Korean Spelling Corrector based on Morphological Analysis and Binary N-gram Method", Journal of KIISE Vol. 20, No. 1, pp. 813-816, Apr. 1993.
- [15] Jeung-Hyun Byun, "Automatic Extraction of Spelling Correction Rule using Corrected Corpus", Master Thesis, Korea University, Feb. 2008
- [16] Han-Kyu Lim, Ung-Mo Kim, "A Spelling Correction System Based on Statistical Data of Spelling Errors", Journal of information

- Processing Systems, Vol. 2, No. 6, pp. 839-846, Nov. 1995.
- [17] Han-Min Jung, Geun-Bae Lee, Jong-Hyeok Lee. "An Implementation of Neuro-Fuzzy Korean Spelling Corrector Using Keyboard Arrangement Characteristics", Proceedings of the 5th Annual Conference on Human and Cognitive Language Technology, pp. 317-328, Oct. 1993.
- [18] Kwang-Seob Shim, Jae-Hyung Yang. "High Speed Korean Morphological Analysis based on Adjacency Condition Check", Journal of KIISE : Software and Application, Vol. 31 No. 1, pp. 89-99, Jan. 2004.

## 저 자 소개



**김진형**  
 현재 : 상명대학교 디지털미디어학부  
 재학중  
 관심분야 : 컴퓨터공학  
 Email : ghostliving@naver.com



**박소영**  
 1997 : 상명대학교 전자계산학과  
 이학사  
 1999 : 고려대학교 컴퓨터과학과  
 이학석사  
 2005 : 고려대학교 컴퓨터과학과  
 이학박사  
 현재 : 상명대학교  
 게임모바일콘텐츠학과 조교수  
 관심분야 : 컴퓨터공학  
 Email : ssoya@smu.ac.kr

