한국어 정보 처리 시스템의 전처리를 위한 미등록어 추정 및 철자 오류의 자동 교정

박 봉 래† · 임 해 창††

요 약

본 논문은 한국어 정보 처리 시스템의 성능 향상을 위하여 입력 문서에 존재하는 미등록어를 인식하고 철자 오류(띄어쓰기 오류 포함)를 자동으로 교정하는 방법을 제안한다. 동일한 미등록어 후보가 포함된 둘 이상의 형태적 유사 어절을 비교 분석함으로써 입력 문서에 존재하는 미등록어를 인식하고, 오류 어절과 코퍼스내에 존재하는 교정 어절 사이의 형태적 및 문맥적 유사성에 근거하여 대량의 원시 코퍼스로부터 자동으로 오류 교정용 어휘 규칙을 생성한 후에 이를 이용하여 입력 문서에 존재하는 띄어쓰기 및 철자 오류를 교정한다. 실험 결과에 따르면 제안한 방법으로 구현된 시스템은 약 98.9%의 정확도로 미등록어를 인식할 수 있고, 98.1%와 97.1%의 정확도로 띄어쓰기 오류와 철자 오류를 각각 교정할 수 있다.

Recognizing Unknown Words and Correcting Spelling Errors as Preprocessing for Korean Information Processing System

Bong-Rae Park† · Hae-Chang Rim††

ABSTRACT

In this paper, we propose a method of recognizing unknown words and correcting spelling errors(including spacing errors) to increase the performance of Korean information processing systems. Unknown words are recognized through comparative analysis of two or more morphologically similar eojeols(spacing units in Korean) including the same unknown word candidates. And spacing errors and spelling errors are corrected by using lexicalized rules which are automatically extracted from very large raw corpus. The extraction of the lexicalized rules is based on morphological and contextual similarities between error eojeols and their correction eojeols which are confirmed to be used in the corpus. The experimental result shows that our system can recognize unknown words in an accuracy of 98.9%, and can correct spacing errors and spelling errors in accuracies of 98.1% and 97.1%, respectively.

1. 서 론

한국어 정보 처리를 위한 대부분의 연구들은 입력 문서에 대해 두 가지의 가정을 세워놓고 있다. 문서 내의 모든 어휘는 기계 가독형 어휘 사전에 등록되어 있다는 가정과 문서 내의 각 어절에는 띄어쓰기 오류나 철자 오류가 존재하지 않는다는 가정이다. 이러한 가정은 한국어 정보 처리 연구가 체계적이고 언어 이론에 집중할 수 있는 바탕을 제공한다. 그러나 실세계의 문서들에는 다양한 유형의 미등록어들이 자주 나타나고 띄어쓰기 오류나 철자 오류가 있는 어절들이 많이 존재한다[5,7,13]. 따라서 위의 가정에 기반하여 개

* 본 연구는 96년 교육부 학술연구조성비 자유공모과제의 지원을 받은 것입니다.
† 준 회 원 : 고려대학교 컴퓨터학과
†† 종신회원 : 고려대학교 컴퓨터학과 교수
논문접수 : 1998년 2월 25일, 심사완료 : 1998년 7월 13일

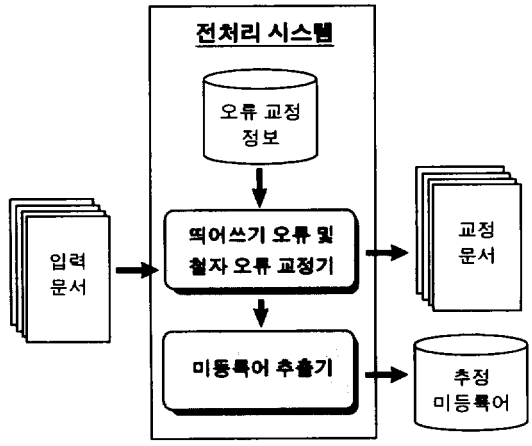
발된 한국어 정보 처리 시스템들은 실세계의 문서들에 적용시켰을 경우에 기대만큼의 성능을 볼 수 없다. 그러므로 한국어 정보 처리 시스템들의 실용성을 높이기 위해서는 입력 문서들에 존재하는 미등록어와 오류 어절에 대한 전처리가 필요하다.

최근 들어, 이러한 인식이 확산되면서 일부 한국어 정보 처리 시스템들이 미등록어를 처리할 수 있도록 개발되고 있지만 이들 시스템들은 크게 세 가지의 문제점을 가지고 있다. 첫번째는 띄어쓰기 오류나 철자 오류가 발생한 어절에 대해 미등록어 추정을 수행한다는 점이고, 두번째는 미등록어를 포함한 어절을 발견하기가 쉽지 않다는 점이다. 그리고 세번째는 미등록어가 포함된 어절에서 미등록어를 정확하게 분리하기가 어렵다는 점이다. 미등록어를 포함한 어절을 발견하는 기존 방법들은 주로 미등록어가 포함된 어절은 형태소 분석에 실패하는 경향이 있음을 이용한다. 그러나 오류가 발생한 어절도 형태소 분석에 실패할 수 있고 미등록어를 포함한 많은 어절들이 형태소 분석시에 오분석될 수 있기 때문에 형태소 분석 가능 여부만을 이용할 경우에 비어를 미등록어로 잘못 추출하거나 미등록어를 포함한 어절 자체를 발견하지 못할 수 있다[1,6,9]. 그리고 발견된 미등록어 포함 어절에서도 형태소를 분리할 때 발생하는 중의성으로 인해 다른 구성 형태소들과 구별하여 미등록어만을 정확하게 분리하기가 쉽지 않다[6,17].

띄어쓰기 오류와 철자 오류를 교정하기 위한 방법들도 많은 문서 처리 시스템들과 함께 개발되어 왔지만 정보 처리 시스템의 전처리로서 오류를 자동으로 교정하는 데에 이용하기에는 부적합하다. 그 이유는 오류 어절에 대한 대체어를 정확하게 결정하기가 어렵기 때문이다. 띄어쓰기 오류 어절의 대체어는 공백을 추가하거나 삭제한 후에 형태소 분석 가능 여부를 통해 선정하고, 철자 오류 어절의 대체어는 음소를 추가, 삭제, 또는 교환한 후에 형태소 분석 가능 여부를 통해 선정한다[2,4,8,10,11,16]. 그러나 이 과정만으로는 선정 가능한 후보가 여러 개 존재할 수 있으므로 오류 어절을 자동으로 교정하기 어렵다. 최근 들어, 연어정보나 말뭉치 분석 정보를 이용하는 방법과 문맥에 대한 통계 정보를 이용하는 방법들이 제시되고 있지만[3,12,14], 이들 방법들도 충분한 연어정보나 말뭉치 분석 정보를 구축하기가 어렵고 데이터 부족으로 통계정보를 신뢰하기 어려운 문제점이 있다.

본 논문에서는 기존의 미등록어 인식 방법의 세 가지 문제점을 최소화하기 위해 미등록어 인식 전에 오류 교정을 수행하며 형태소 분석 결과에만 의존하지 않고 동일한 미등록어 후보가 포함된 둘 이상의 어절들을 비교하여 미등록어를 인식하는 방법을 제시한다. 그리고 대량의 코퍼스로부터 띄어쓰기 및 철자 오류를 교정하기 위한 어휘 규칙을 오류 어절과 교정 어절의 형태적 및 문맥적 유사성을 이용하여 구축하는 방법을 제시한다. 띄어쓰기 오류의 경우 형태적 유사성을 이용하고 철자 오류의 경우 형태적 유사성과 함께 앞 또는 뒤 어휘와의 공기 관계를 고려하여 교정 규칙을 생성한다. 이와 같이 단순히 형태적 및 문맥정 유사성에 근거한 교정 규칙은 시스템의 부하를 최소화하면서 고빈도의 오류를 처리하는 데 유리하다.

제안한 방법으로 구현된 시스템은 (그림 1)에 나타난 바와 같이 '띄어쓰기 오류 및 철자 오류 *교정기*'와 '*미등록어 추출기*' 및 '*오류 교정 정보*' 데이터베이스로 구성된다. '*오류 교정 정보*' 데이터베이스는 대량의 학습 코퍼스로부터 미리 추출한 띄어쓰기 및 철자 오류 교정을 위한 어휘 규칙들로 구성되며 이를 이용하여 '*띄어쓰기 오류 및 철자 오류 교정기*'가 입력 문서 내에 존재하는 띄어쓰기 및 철자 오류를 교정하고, '*미등록어 추출기*'가 오류가 교정된 어휘들을 비교분석하여 미등록어를 추출한다. 전처리가 끝나면 띄어쓰기 및 철자 오류가 교정된 문서와 이 문서로부터 추출한 미등록어들이 출력된다.

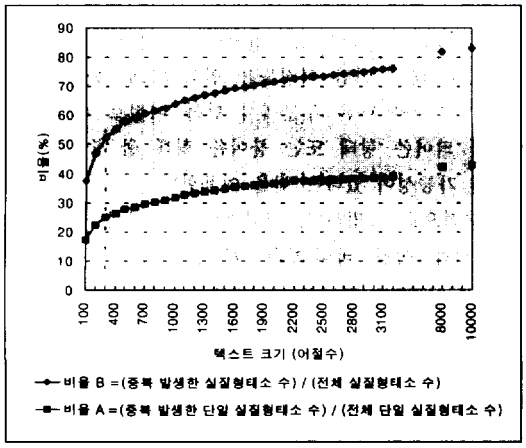


(그림 1) 시스템 구성도
(Fig. 1) System Configuration

이어지는 2장부터 4장에서 미등록어의 추정 방법, 띄어쓰기 오류의 자동 교정을 위한 어휘 규칙의 생성 방법, 철자 오류의 자동 교정을 위한 어휘 규칙의 생성 방법을 각각 자세히 설명한다. 그리고 5장과 6장에서 실험 결과와 결론 및 향후 연구를 제시한다.

2. 형태적 유사 어절의 비교 분석을 통한 미등록어 추정

태깅된 약 100만 어절의 코퍼스를 분석한 결과에 따르면 동일한 실질형태소가 텍스트의 크기에 따라 두 번 이상 발생할 확률은 (그림 2)와 같다. 그림에 따르면 300어절 크기의 텍스트 내에 존재하는 단일한 실질형태소들 중 약 25%(비율 A)가 두번 이상 발생하고 이들이 전체 발생한 실질형태소의 52%(비율 B)를 차지한다. 그리고 텍스트 크기가 커질수록 두번 이상 발생하는 단일 실질형태소의 비율이 높아져서 텍스트 크기가 10,000어절이 되면 약 42%(비율 A)의 단일 실질형태소가 두번 이상 발생하고 이들이 전체 실질형태소의 83%(비율 B)를 차지하게 된다.



(그림 2) 실질형태소 중복 발생 통계
(Fig. 2) Repeated Occurrence of Lexical Morphemes

동일한 텍스트에서 두번 이상 발생하는 실질형태소는 단독 또는 다른 형태소들과 결합하여 종종 서로 다른 어절들을 형성하는데, 본 논문에서는 동일한 실질형태소를 포함한 어절을 형태적 유사 어절이라고 명명한다. 제안하는 방법은 이러한 형태적 유사 어절이 미등록어(미등록 실질형태소)에 대해서 발생할 때, 이들 어절들로부터 일관성있게 추출되는 미등록어 후보를 실제의 어휘로 간주하는 방법이다. 즉, 미지의 문자열이 두 어절 이상에서 서로 다른 기등록 형태소들과 결합되어 나타나는 경우의 대부분은 이 미지의 문자열이 실제 단어일 때라는 가정하에 이 미지의 문자열을 미등록어로 추출한다. 이 방법은 형태소 분석이 가능한 어절에도 적용되고 기등록어의 분리가 미등록어를 발견하는 과정에서 수행되는 장점이 있다.

예를 들어, 동일한 미등록 명사 '*이순신*'을 포함한 형태적 유사 어절 '*이순신의*'와 '*이순신이*'를 고려해 보자. 두 어절은 각각 다음과 같이 분석 또는 추정될 수 있다[1]:

E_i: 이순신의
  $I_{i,1}$: 이순/N+신의/N
  $I_{i,2}$: 이순신의/N?
  $I_{i,3}$: 이순신/N?+의/J
E_j: 이순신이
  $I_{j,1}$: 이순신이/N?
  $I_{j,2}$: 이순신/N?+이/J.

대부분의 형태소 분석기들은 첫번째 어절 '*이순신의*'를 '*이순*/N+*신의*/N'($I_{i,1}$)로만 형태소 분석함으로써 미등록어 '*이순신*'의 존재를 발견하지 못하며, 두번째 어절 '*이순신이*'로부터는 두 개의 추정결과 '*이순신이*/N?'($I_{j,1}$)와 '*이순신*/N?+*이*/J'($I_{j,2}$)를 출력함으로써 미등록어가 '*이순신이*'인지 '*이순신*'인지 결정할 수가 없다. 그러나 제안한 방법은 두 어절로부터 '*이순신*/N?'이 일관성있게 추출될 수 있기 때문에 이를 미등록 명사로 추정한다.

그러나, 이상의 방법만으로는 미등록어가 포함되어 있지 않은 형태적 유사 어절들로부터도 미등록어 후보를 추출하게 되는 문제점이 존재한다. 예를 들면, 두 어절 '*학교에서도*'와 '*학교에서만*'은 다음과 같이 분석 또는 추정될 수 있는데, 두 어절로부터 비단어 '*학교에*'와 '*학교에서*'가 일관성있게 분석되므로 이들을 미등록어로 잘못 추출하게 된다.

E_k: 학교에서도
  $I_{k,1}$: 학교/N+에서도/J

1) E_i와 I_{ij}: E_i는 텍스트내 i번째 어절의 문자열이고, I_{ij}는 E_i의 j번째 형태소 분석 결과임.
'+'와 '?': '+'는 형태소간의 경계 표시이고 '?'는 추정된 형태소 표시 N과 J: 각각 명사와 조사를 나타냄.

$I_{k,2}$: 학교에/N?+서도/J
$I_{k,3}$: 학교에서/N?+도/J
$I_{k,4}$: 학교에서도/N?
E_i: 학교에서만
$I_{i,1}$: 학교/N+에서만/J
$I_{i,2}$: 학교에/N?+서만/J
$I_{i,3}$: 학교에서/N?+만/J
$I_{i,4}$: 학교에서만/N?

이렇게 비단어들이 미등록어로 추정되는 경우를 분석해보면 일관성있는 분석 결과가 하나 이상 존재할 때이다. 미등록어 '*이순신*'의 형태적 유사어절들 '*이순신의*'와 '*이순신이*'의 각 분석 결과에 일관성있는 공통의 실질형태소는 '이순신/N?'뿐이지만 위의 형태적 유사어절 E_k와 E_i의 각 분석 결과의 일관성있는 공통의 실질형태소는 '학교/N', '학교에/N?', '학교에서/N?'로 여러개이다. 따라서 비단어들이 미등록어로 추정되는 것을 막기 위해서 <알고리즘 1>에서와 같이 형태적 유사어절들에서 일관성있는 공통의 실질형태소가 해당 미등록어뿐인 경우에만 제안한 방법을 적용한다.

<알고리즘 1> 형태적 유사 어절의 비교분석을 통한 미등록어 인식

1. 텍스트내 모든 어절들에 대해 모든 미등록어 후보들을 추출
2. 동일한 미등록어 후보가 추출된 둘 이상의 형태적 유사 어절 수집
3. 형태적 유사어절들에서 일관성있게 분석된 유일한 공통의 실질형태소가 해당 미등록어 후보뿐인 경우 이를 추출

3. 띄어쓰기 오류 교정을 위한 어휘 규칙의 생성

다양한 분야와 다양한 사람들에 의해 작성된 코퍼스에는 오류 어절이 존재하면 그 교정 어절도 함께 존재할 가능성이 높다. 이 가능성을 띄어쓰기 오류에 적용하여 교정 어절이 코퍼스에 함께 존재하는 띄어쓰기 오류를 띄어쓰기 오류 용례로 수집하고 이를 토대로 띄어쓰기 오류 자동 교정을 위한 어휘 규칙을 생성한다.

3.1 띄어쓰기 오류의 용례 수집

일반적으로 띄어쓰기 오류 어절은 대부분의 경우에 비어로서 형태소분석에 실패한다는 점에 근거하여 탐색하고, 교정 어절은 탐색된 오류 어절의 앞뒤 어절 사이의 공백을 제거하거나 탐색된 오류 어절내 각 음절 사이에 공백을 추가하여 만들어진 어절이 형태소 분석에 성공하는지 여부에 따라 생성한다. 그러나, 형태소 분석에 실패하는 어절은 띄어쓰기 오류가 발생한 어절 외에 철자오류가 발생한 어절이거나 미등록어를 포함한 어절일 수도 있으므로 형태소 분석에 실패한 어절을 모두 띄어쓰기 오류 어절로 간주할 수는 없다. 그리고 공백을 제거하거나 추가하여 교정 후보를 생성하는 방식에도 두 가지의 문제점이 존재한다. 첫번째는 둘 이상의 교정 후보가 생성될 수 있다는 점이고, 두번째는 한국어의 경우에 하나의 어절을 분리하여 생성된 두 어절이 상호 문맥에는 맞지 않지만 형태소 분석에 성공할 수 있다는 점이다.

제안한 방법은 띄어쓰기 오류의 탐색과 교정 후보의 생성에 있어서는 기본적으로 일반적인 방법과 같지만, 생성된 교정 후보가 실제로 사용되는지 여부를 대량의 코퍼스에서 확인함으로써 잘못 탐색된 띄어쓰기 오류 어절을 배제하고 교정후보를 단일화한다. 이러한 교정 후보의 확인 작업은 띄어쓰기 오류 형태와 그 교정 형태는 띄어쓴 형태의 두 어절 사이에 존재하는 공백을 제거하면 형태적으로 일치한다는 점에 근거한다. 즉, 코퍼스내의 인접 어절들을 인위적으로 붙인 형태가 코퍼스내의 다른 부분에서 단독 어절로 존재하는 경우에 띄어쓴 형태 또는 붙여쓴 형태 둘 중 하나는 오류일 가능성이 있다는 점을 이용한다.

기술적으로 설명하면, 코퍼스로부터 어절 유니그램($<E_i>$)과 어절 바이그램($<E_j, E_{j+1}>$)을 수집하여 어절 바이그램의 두 어절을 결합한 형태가 어절 유니그램에 존재하는지 확인함으로써 띄어쓰기 오류 용례를 수집한다. 그리고 각 어절의 형태소 분석 가능 여부에 따라 다음과 같이 띄어쓰기 오류 용례를 띄붙 오류 용례와 붙띄 오류 용례로 분류한다[2]. 즉, 붙여쓴 형태가 형태소 분석에 실패하는데 띄어쓴 형태의 두 어절은 모두 형태소 분석에 성공하는 경우에 이를 띄붙 오류 용례로 수집하고 반대의 경우에 붙띄 오류 용례로 수집한다.

- 띄붙 오류 용례: $\langle E_j \rangle \rightarrow \langle E_j, E_{j+1} \rangle$

2) 띄어쓰기 오류는 크게 두 종류로 분류된다. 두 어절이 불법으로 결합하여 발생하는 띄붙오류와 하나의 어절이 불법으로 분리되어 사용되는 붙띄오류이다.

· 붙띄 오류 용례: $\langle E_i, E_{i+1}\rangle \rightarrow \langle E_i\rangle$

그러나, 붙여쓴 형태와 띄어쓴 형태 모두가 형태소 분석에 실패하는 경우나 모두 형태소 분석에 성공하는 경우의 띄어쓰기 오류 용례는 배제한다. 왜냐하면 한국어 복합명사의 경우 띄어쓸 수도 붙여쓸 수도 있고, 관련된 명사가 기등록어이면 띄어쓰나 붙여쓰나 형태소 분석이 가능하지만, 미등록어인 경우에는 모두 형태소 분석에 실패할 가능성이 높기 때문이다.

3.2 띄어쓰기 오류의 교정 규칙 생성

한국어는 교착어로서 하나 이상의 형태소가 결합하여 띄어쓰기 단위인 어절을 구성한다. 따라서 띄어쓰기 오류는 앞 어절의 끝 형태소와 뒤 어절의 앞 형태소간에 발생한다. 이러한 관점에서, 위의 방법으로 수집된 띄어쓰기 오류 용례들로부터 일반화된 교정 규칙을 생성한다. 예를 들어, 다음과 같은 띄붙 오류 용례를 고려해 보자.

<학교에서공부를> → <학교에서, 공부를>

이 띄붙 오류는 앞 어절의 끝 형태소 '*에서*'와 뒤 어절의 첫 형태소 '*공부*'간에 발생하였다. 따라서 다음과 같은 일반화된 규칙을 유도한다.

〈*에서공부*〉 → 〈* 에서 공부*〉
조사 명사

이 규칙은 위의 띄어쓰기 오류 용례에 나타난 오류 뿐만 아니라 오류 어절 '*학교에서공부를*'과 유사한 오류 어절 '*학교에서공부만*'이나 '*도서관에서공부를*'과 같은 오류 어절들을 교정하는 데에도 이용할 수 있다.

그러나, 이러한 일반화된 규칙은 띄어쓰기 오류 용례의 어절 바이그램내 앞 어절과 뒤 어절로부터 끝 형태소와 첫 형태소가 각각 모호성 없이 추출될 때에만 가능하다. 따라서, 앞 어절이나 뒤 어절이 중의적으로 형태소 분석되어 첫음절과 끝음절의 추출이 단일하게 이루어지지 않는 경우나 붙띄 오류 용례에서 앞 또는 뒤 어절이 형태소 분석에 실패하는 경우에 어절 자체를 이용하여 규칙을 생성할 필요가 있다.

<알고리즘 2>은 이상의 내용을 고려하여 생성한 띄어쓰기 오류 교정용 어휘 규칙에 대한 묘사이다.

<알고리즘 2> 띄어쓰기 오류의 교정 규칙 생성

정의: 불린 함수 $Uniq(M,E)$은 형태소 M이 어절 E로부터 단일하게 추출될 때에만 '참'값을 갖는 함수이다. 그리고 심볼 'H'와 'T'는 각각 어절의 첫번째 형태소와 마지막 형태소를 나타낸다. 'E_i'와 'E_{i+1}'은 두 인접 어절을 의미하며, 심볼 '*'는 하나 이상의 형태소를 나타낸다

1. 띄붙 오류의 교정 규칙 생성
   - $\langle *TH*\rangle \rightarrow \langle *T, H*\rangle$ if $Uniq(T,E_i) \wedge Uniq(H,E_{i+1})$
   - $\langle *TE_{i+1}\rangle \rightarrow \langle *T, E_{i+1}\rangle$ if $Uniq(T,E_i) \wedge \neg Uniq(H,E_{i+1})$
   - $\langle E_i H*\rangle \rightarrow \langle E_i, H*\rangle$ if $\neg Uniq(T,E_i) \wedge Uniq(H,E_{i+1})$
   - $\langle E_i E_{i+1}\rangle \rightarrow \langle E_i, E_{i+1}\rangle$ if $\neg Uniq(T,E_i) \wedge \neg Uniq(H,E_{i+1})$
2. 붙띄 오류의 교정 규칙 생성
   - $\langle *T, E_{i+1}\rangle \rightarrow \langle *TE_{i+1}\rangle$ if $Uniq(T,E_i) \wedge \neg Uniq(H,E_{i+1})$
   - $\langle E_i, H*\rangle \rightarrow \langle E_i H*\rangle$ if $\neg Uniq(T,E_i) \wedge Uniq(H,E_{i+1})$
   - $\langle E_i T, E_{i+1}\rangle \rightarrow \langle E_i TE_{i+1}\rangle$ if $\neg Uniq(T,E_i) \wedge \neg Uniq(H,E_{i+1})$
3. 심볼 '*'를 가진 규칙중 추출 빈도가 1이하인 규칙은 배제

<알고리즘 2>에서 붙띄 오류를 교정하기 위한 교정 규칙 중 첫번째와 두번째는 불필요하게 여겨질 수도 있다. 왜냐하면 띄붙 오류와 달리 붙띄 오류에서 떨어진 두 어절은 모두 비어절로서 형태소 분석에 실패할 가능성이 높기 때문이다. 그러나, 예를 들어, 어절 '*학교에서부터*'가 어절 '*학교에서*'와 비어절 '*부터*'로 분리되는 경우처럼 분리된 후에도 한 쪽 어절이 실제로 사용되는 어절일 수 있기 때문에 이러한 오류를 교정하기 위해 필요하다.

4. 철자 오류 교정을 위한 어휘 규칙의 생성

철자 오류에 대해서도 다양한 분야와 다양한 사람들에 의해 작성된 코퍼스에는 철자 오류 어절이 존재하면 그 교정 어절도 함께 존재할 가능성이 높다. 그리고 철자 오류와 교정 어절은 동일한 의미를 가지므로 주위 문맥도 유사할 가능성이 높다. 이러한 가능성들에 근거하여 대량의 코퍼스에서 각종 철자 오류 용례를 수집하고 이들로부터 철자 오류 교정 규칙을 생성한다.

4.1 철자 오류의 용례 수집

철자 오류는 크게 세 종류로 나뉜다. 첫번째는 타이

핑 오류로서 키보드내 인접키를 잘못 눌러서 발생하고, 두번째는 인식 오류로서 표준어에 대한 무지에서 발생한다. 그리고 세번째는 음운 오류로서 유사한 음가를 갖는 음소를 혼동하는 데서 발생한다[15]. 이러한 철자 오류는 결국 불법적으로 음소를 삽입, 삭제 또는 대체함으로써 비어를 생성하게 된다. 기존의 철자 오류 검사기들은 이러한 점에 근거하여 형태소 분석에 실패한 어절을 철자 오류 어절로 탐색한다. 그리고 철자 오류의 패턴에 따라서 교정 후보를 생성한다.

그러나, 형태소 분석에 실패하는 어절이 모두 철자 오류 어절은 아니며, 하나의 오류 어절로부터 생성할 수 있는 교정 어절은 보통 둘 이상일 수 있다. 따라서, 기존의 방법으로는 철자 오류를 자동으로 교정할 수도 없고 철자 오류 용례를 자동으로 수집할 수도 없다.

제안한 방법은 철자 오류의 용례를 수집할 때 오류 어절의 탐색과 교정 후보의 생성에 있어서 기존 방법을 이용한 후에, 생성된 교정 후보가 실제로 사용되는지 여부와 해당 철자 오류 어절의 문맥과 유사한 문맥에서 사용되는지 여부를 대량의 코퍼스에서 확인함으로써 잘못 탐색된 철자 오류 어절을 배제하고 교정후보를 단일화한다. 예를 들어, 어절 '*계열기업*'에 대한 오류 어절 '*개열기업*'에 대해서 교정 어절로서 올바른 어절 '*계열기업*' 이외에 잘못된 교정 어절 후보 '*개열기업*'이 함께 생성될 수 있으나[3], 코퍼스에 '*개열기업*'은 나타나지 않으므로 배제한다. 또한 어절 '*개열기업*'이 분리되어 '*개열 기업*'으로 존재하는 경우에도 '*개열*'이 '*기업*'과 인접하여 나타나지 않으므로 '*개열*'의 잘못된 교정 어절 후보 '*개열*'을 배제한다.

이와 같은 방법으로 대량의 코퍼스로부터 철자 오류와 그 교정 어절이 단일하게 결정되면 이들을 이용하여 다음과 같은 튜플 형태의 철자 오류 용례를 구축한다.

$$\langle E_L, E, E_R, C, Freq(E_L, C), Freq(C, E_R)\rangle$$

이 튜플은 형태소 분석에 실패한 철자 오류 후보 어절 E와 앞뒤 어절 E_L과 E_R, 및 오류 어절 E에 대한 교정 어절 후보 C, 그리고 어절 E_L과 교정 어절 후보가 인접하여 발생한 빈도 $Freq(E_L, C)$ 및 교정 어절 후보와 어절 E_R이 인접하여 발생한 빈도 $Freq(C, E_R)$로 구성된다. 수집된 용례들에서 $Freq(E_L, C)$ 또는 $Freq(C, E_R)$가 항상 1이상의 값을 갖는다.

3) '개열'이 사전에 등록된 명사이므로 '개열기업'이 '개열'과 '기업'의 복합명사로 분석된다.

4.2 철자 오류의 교정 규칙 생성

앞절에서 제시한 방법으로 수집된 철자 오류 용례들이 모두 신뢰할 만한 것은 아니다. 오류 어절과 교정 어절에서 일치하는 앞 또는 뒤 어절이 문맥을 형성하는 어절이 아닐 수 있기 때문이다. 그러나 앞뒤 어절이 모두 일치하는 경우나 일치하는 한 어절이 오류 어절 또는 교정 어절과 자주 나타나는 것으로 확인된 경우에는 일치하는 어절이 문맥을 형성하는 어절일 가능성이 높다. 오류 어절과 교정 어절 후보간에 문맥을 형성하는 어절이 일치하다는 의미는 교정 어절 후보가 해당 오류 어절에 대한 올바른 대체어일 가능성이 높음을 의미한다.

따라서, 수집된 철자 오류 용례들로부터 다음과 같은 규칙의 생성을 생각해 볼 수 있다.

- 철자 오류 용례:
  $\langle E_L, E, E_R, C, Freq(E_L, C), Freq(C, E_R)\rangle$
- 생성 규칙:
  - $E \rightarrow C / E_L\_$ if $(Freq(E_L, C) > 0) \wedge (Freq(E_L, C) + Freq(C, E_R) \geq 2)$
  - $E \rightarrow C / \_\ E_R$ if $(Freq(C, E_R) > 0) \wedge (Freq(E_L, C) + Freq(C, E_R) \geq 2)$

첫번째 규칙은 앞어절을 문맥으로 고려한 경우이고, 두번째 규칙은 뒤 어절을 문맥으로 고려한 경우이다. 각 규칙의 조건은 오류 어절과 교정 어절의 두 문맥 어절이 모두 일치하거나 일치하는 하나의 문맥 어절이 교정 어절과 함께 인접하여 나타난 빈도가 2이상이어야 함을 나타낸다. 이러한 조건은 규칙을 만드는 과정에서 신뢰도가 높은 철자 오류 용례만을 고려하기 위해 필요하다.

그러나 이러한 규칙은 해당 철자 오류 용례에 나타난 오류에만 적용될 수 있다. 예를 들어, 다음과 같은 철자 오류 교정 규칙이 존재할 경우에

가느→가는/학교에_

오류 어절 '*가느*'는 앞 어절이 '*학교에*'일 때에만 올바른 어절 '*가는*'으로 교정할 수 있다. 즉, 실질형태소 '*학교*'가 다른 형식형태소들과 결합한 '*학교도*', '*학교까지*', '*학교만*', '*학교를*' 등과 같은 어절들이 오류 어절 '*가느*' 앞에 나타난 경우에는 '*가느*'를 '*가는*'으로 교정할 수 없다. 그러나 이 어절들과 어절 '*학교에*'는 형식형태소

만 다를 뿐 실질형태소 '*학교*'가 동일하기 때문에 모두 유사한 의미를 가진다. 따라서 오류 어절 '*가느*'는 앞 어절이 '*학교에*'일 때 어절 '*가는*'으로 교정되기보다는 앞 어절에서 형식형태소를 제외한 부분이 '*학교*'일 때 '*가는*'으로 교정하는 것이 바람직하다.

제안한 방법은 이러한 사실을 토대로 일반화된 규칙을 생성하기 위해 다음과 같이 철자 오류 용례을 재구성한다.

$\langle E_L, E, E_R, C, Freq(E_L, C),\ Freq(C, E_R),\ Freq(H_L, C), Freq(C, H_R)\rangle$

이 용례에서 H_L와 H_R은 각각 앞 어절과 뒤 어절에서 형식형태소를 제외한 부분이고, $Freq(H_L, C)$와 $Freq(H_L, C)$는 각각 H_L와 H_R이 교정 어절 C와 함께 인접한 어절에 나타난 빈도이다. 이러한 두 빈도는 H_L와 H_R이 어절 E_L와 E_R로부터 각각 단일하게 분석될 때에만 측정된다.

그리고 다음 <알고리즘 3>에 따라 새로운 철자 오류 용례들을 이용하여 일반화된 철자 오류 교정용 어휘 규칙을 생성한다.

<알고리즘 3> 철자 오류의 교정 규칙 생성

정의: <알고리즘 2>의 *Uniq* 함수.

입력: 철자 오류 용례

$\langle E_L, E, E_R, C, Freq(E_L, C), Freq(C, E_R), Freq(H_L, C), Freq(C, H_R)\rangle$

생성 규칙:

- $E \rightarrow C / H_L^* \_$ if $Uniq(H_L, E_L) \wedge (Freq(H_L, C) > 0) \wedge (Freq(H_L, C) + Freq(C, H_R) \geq 2)$
- $E \rightarrow C / \_\ H_R^*$ if $Uniq(H_R, E_R) \wedge (Freq(C, H_R) > 0) \wedge (Freq(H_L, C) + Freq(C, H_R) \geq 2)$
- $E \rightarrow C / E_L \_$ if $\neg\ Uniq(H_L, E_L) \wedge (Freq(E_L, C) > 0) \wedge (Freq(E_L, C) + Freq(C, E_R) \geq 2)$
- $E \rightarrow C / \_\ E_R$ if $\neg\ Uniq(H_R, E_R) \wedge (Freq(C, E_R) > 0) \wedge (Freq(E_L, C) + Freq(C, E_R) \geq 2)$

5. 실험 결과

제안한 방법을 평가하기 위하여 신문, 소설, 대본 등에서 추출한 약 7백만 어절 코퍼스를 이용하였는데, 이중 약 7만 어절 코퍼스는 테스트용으로 이용하였고 나머지는 학습용으로 이용하였다.

<표 1>은 띄어쓰기 오류 및 철자 오류의 교정 규칙을 학습 코퍼스로부터 생성한 결과이다. 띄어쓰기 오류에 대하여 생성된 교정 규칙의 경우, 띄붙 오류의 교정 규칙이 붙띄 오류의 교정 규칙보다 더 많은 것은 띄붙 오류가 붙띄 오류에 비해 더 자주 발생하기 때문이다. 그리고 철자 오류를 교정하는 규칙은 띄어쓰기 오류를 교정하는 규칙보다 상대적으로 매우 작은데 이는 두 가지 이유에 기인한다고 볼 수 있다. 첫번째는 띄어쓰기 오류가 철자 오류보다 많이 발생한다는 점이고, 두번째는 제안한 방법이 단지 앞뒤 어절만을 고려함에 따라 발견되지 못한 철자 오류가 많이 존재한다는 점이다. 생성된 규칙을 분석해 보면 띄어쓰기 오류의 경우 관형사와 명사 사이 및 부사와 용언 사이의 띄붙 오류와 복합조사의 구성 조사들 사이의 붙띄 오류에 대한 교정 규칙들이 가장 많이 발생하였고, 철자 오류의 경우에는 사투리 및 비표준 외래어에 대한 교정 규칙들이 가장 많이 발생하였다.

<표 1> 띄어쓰기 및 철자 오류의 교정 규칙 생성
<Table 1> Generation of Spacing and Spelling Error Correction Rules

띄어쓰기 오류 교정 규칙		64,141개
	띄붙 오류 교정	53,128개
	붙띄 오류 교정	11,013개
철자 오류 교정 규칙		3,038개

<표 2>는 실험 코퍼스내 형태소 분석에 실패한 어절들 1,285개의 형태소 분석 실패의 원인별 분포를 나타내고 있다. 표에 따르면 미등록어가 가장 많이 형태소 분석 실패의 원인이 되고 있고, 다음으로 띄어쓰기 오류와 철자 오류이다. 항목 '*기타*'는 주로 형태소분석기가 문제되는 경우이다.

<표 2> 테스트 코퍼스내 형태소 분석 실패 어절의 분포
<Table 2> Distribution of Test Corpus Eojeols Failing in Morphological Analysis

띄어쓰기 오류	철자 오류	미등록어	기타
20.8%	9.6%	67.3%	2.3%

<표 3>은 테스트 코퍼스내 미등록어를 추출하고 철자 및 띄어쓰기 오류를 교정한 결과이다. 정확도는 모두 높은 편이며, 미등록어 추출과 띄어쓰기 오류 교정에 있어서는 재현율도 높다. 그러나 철자 오류 교정은 재현율이 매우 낮음을 알 수 있다. 이는 문맥을 고려함에 따라 데이터 부족 문제가 심각해지기 때문이다.

〈표 3〉 미등록어 추출 및 오류 교정 실험 결과
〈Table 3〉 Experimental Result of Unknown Word Extraction and Spelling Error Correction

항목	재현율	정확도
미등록어 추출	87.3%	98.9%
띄어쓰기 오류 교정	85.7%	98.2%
철자 오류 교정	42.0%	97.2%

6. 결론 및 향후 연구

본 논문에서는 한국어 정보 처리 시스템의 전처리로서 미등록어와 띄어쓰기 및 철자오류로 인한 정보처리 시스템의 성능 저하를 최소화하기 위해, 입력 문서에 존재하는 미등록어를 추출하고 띄어쓰기 및 철자 오류를 자동으로 교정하는 방법을 제안하였다. 미등록어는 오류 교정 후에 미등록어 후보가 포함된 둘 이상의 형태적 유사 어절을 비교 분석하여 인식함으로써 형태소 분석 가능 여부에만 의존하는 기존 방법의 세 가지 문제점인 오류어절의 오인식, 미등록어 포함 어절의 오분석 및 미등록어 후보의 과다 생성 문제를 극복할 수 있었다. 그리고 띄어쓰기 및 철자 오류는 대량의 원시 코퍼스로부터 오류 어절과 교정 어절 사이의 형태적 및 문맥적 유사성에 근거하여 자동으로 구축한 어휘 규칙을 이용함으로써 오류 어절에 대한 정확한 대체어를 제시할 수 있게 되었다. 실험 결과는 제안한 방법으로 구현된 시스템이 약 98.9%의 정확도로 미등록어를 추출할 수 있고, 98.1%와 97.1%의 정확도로 띄어쓰기 오류와 철자 오류를 각각 교정할 수 있음을 보여주었다.

향후에는 음절 및 음절 바이그램 정보를 이용하여 미등록 외래어를 구별하면서 띄어쓰기 오류를 자동으로 교정하는 방법을 연구할 계획이며, 낮은 재현율을 개선하기 위해 고빈도 철자 오류(특히, 비표준 외래어)의 경우 코퍼스로부터 앞뒤 1음절 이상의 어절들을 문맥으로 고려하여 추출하고 입력 문서에서 교정할 때에는 문맥을 고려하지 않고 적용하는 방법을 연구할 계획이다.

참 고 문 헌

[1] 강승식, "음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석," 서울대학교 컴퓨터공학과 박사학위 논문, 1993.

[2] 강재우, "접속정보를 이용한 한글 철자 및 띄어쓰기 검사기의 설계 및 구현," 한국과학기술원 석사학위논문, 1989.

[3] 김병희, 임권묵, 송만석, "말뭉치를 기반으로 한 한국어 철자 교정기의 구현," 한글및한국어정보처리 학술발표논문집, pp.285-293, 1993.

[4] 김재원, "한글 맞춤법 오류의 교정 기법에 관한 연구," 부산대학교 석사학위 논문, 1992.

[5] 미승우, 새 맞춤법과 교정의 실제, 지학사, 1993.

[6] 박봉래, 임해창, "띄어쓰기 오류교정을 위한 용례 사전의 자동구축," 인공지능학회 추계학술발표 논문집, pp.87-93, 1994.

[7] 원영섭, 띄어쓰기·맞춤법 용례, 세창, 1993.

[8] 이종현, 오상헌, "N-GRAM 한글 사전을 이용한 오인식 단어의 교정 알고리즘," 한글및한국어정보처리 학술발표논문집, pp.271-283, 1993.

[9] 임희석, "어절의 중의성 유형 분류에 근거한 한국어 형태소 분석기," 고려대학교 전산과학과 석사학위 논문, 1993.

[10] 조영환, "한글 맞춤법 교정기의 설계 및 구현," 한국과학기술원 석사학위 논문, 1990.

[11] 최재혁, "양방향 최장일치법을 이용한 한국어 띄어쓰기 자동 교정 시스템," 한글및한국어정보처리 학술발표논문지, pp.145-151, 1997.

[12] A. R. Golding and Y. Schebes, "Combining Trigram-based and Feature-based Methods for Context-Sensitive Spelling Correction", 34th ACL, pp.71-78, 1996.

[13] B. Park, H. Rim, "A Korean Corpus Refining System based on Automatic Analysis of Corpus," NLPRS, pp.89-94, 1995.

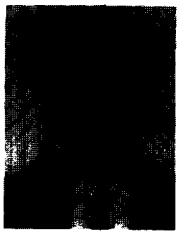
[14] C. Sim, M. Kim, H. Kwon, "Automatic Revision

of Korean Texts by Collocation Words," ICCPOL, pp.280-284, 1994.

[15] K. Karen, "Techinques for Automatically Correcting Words in Text," ACM Computing Survey, Vol.24, No.4, December, 1992.

[16] K. Oflazer, "Error-tolerant Finite-state Recognition with Applications to Morphological Analysis and Spelling Correction," Computational Linguistics, Vol.22, No.1, pp.73-89, 1996.

[17] S. Myaeng, Y. Kwon, K. Jeong, "Foreign Word Identification Using a Statistical Method for Information Retrieval", ICCPOL, pp.675-680, 1997.



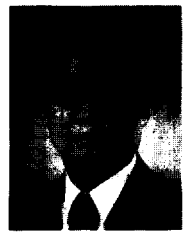
박 봉 래

1993년 고려대학교 전산과학과 졸업(학사)

1995년 고려대학교 전산과학과 대학원 졸업(석사)

1995년~현재 고려대학교 컴퓨터학과 박사과정

관심분야 : 한국어정보처리, 자연어처리, 정보검색 등.



임 해 창

1979년 2월 고려대학교 졸업.

1983년 12월 University of Missouri-Columbia, 전산학 석사.

1990년 12월 Univeristy of Texas at Austin, 전산학 박사.

1991년 3월~8월 전남대학교 강사.

1991년 9월~1994년 고려대학교 컴퓨터학과 조교수.

1994년 9월~현재 고려대학교 컴퓨터학과 부교수.

관심분야 : 한국어정보처리, 정보검색, 인공지능.