어절 생성 사전을 이용한 한국어 철자 교정

이 영 신† · 박 영 자†† · 송 만 석†††

요 약

본 논문에서는 어절 생성 사전을 이용한 한국어 철자 교정을 제안한다. 어절 생성 사전은 두 문자열 간 음절 특성이 고려된 편집 거리 계산을 기반으로 탐색되어 언어와 오류 유형에 의존적인 정보를 이용하지 않고 오류 어절에 대한 후보 어절을 생성한다. 또한 교정된 어절들의 가능한 형태소 분석들을 산출하여 후보들 간의 순위 계산 시에 재차 형태소 분석을 수행하지 않고 언어 정보를 적용할 수 있다. 본 논문에서 제안하는 철자 교정은 두 단계로 구성된다. 첫째, 오류 어절로부터 가능한 오류 정정 어간들을 계산한다. 둘째, 계산된 어간들로부터 어절 생성 사전을 탐색하여 원형 후보 어절들을 생성한다. 또한 품사 태깅과 공기 정보를 사용하여 오류 수정된 결과의 순위를 매긴다. 본 시스템의 자동 철자 교정 성능을 평가한 결과 3,000개의 어절에서 시험한 결과 단어 수준으로 93%가 옳게 교정되었다.

Spelling Correction in Korean Using the '*Eojeol*' generation Dictionary

Young-Sin Lee† · Young-Ja Park†† · Man-Suk Song†††

ABSTRACT

In this paper, we propose an algorithm for Korean spelling correction using the eojeol generation dictionary. Since the eojeol generation dictionary is searched by edit distance based on the syllabic properties of target languages, it generates candidate eojeols independently of languages and error patterns. In addition, it provides morphological analyses of generated eojeols, and all of candidates can be effectively ranked on the basis of linguistic information without re-morphological analyses. The proposed correction algorithm consists of two steps : First, it computes all possible corrected stems from incorrect eojeols. Second, we use the eojeol generation dictionary and the stems to generate candidates for original eojeols. We combined the model with POS tagging and co-occurrence information to rank the recovered eojeols. In order to evaluate the performance of the automatic spelling correction, we have conducted an experiment on 3,000 eojeols and accuracy is around 93% at the word level.

키워드 : 철자교정, 사전, 편집거리(edit distance), 음절특성, 후보 어절

1. 서 론

철자 교정이란 텍스트에서 틀렸다고 판단되는 단어가 발견되었을 때, 이에 대해 주어진 단어들의 집합, 즉 단어 사전으로부터 가장 유사한 단어들을 찾아 수정 혹은 수정 후보를 제시하는 것이라 정의할 수 있다. 실용적인 측면에서 볼 때, 철자 검사 및 교정기는 워드프로세서와 같은 응용 프로그램뿐만 아니라 문자 인식이나 음성 인식기의 인식률을 높이기 위한 후처리로 이용되기도 한다[6, 12].

영어와 같은 비교착어에서는 이러한 사전을 트라이(trie), 유한 상태 오토마타(finite state automata) 등으로 표현한 효율적인 사전 검색 기법에 대한 여러 연구가 있었다[21]. 그러나 비교착어에 적용되는 이와 같은 방법은 교착어인 한국어의 오류를 수정하는데 적용하는 것은 적합하지 않은데[2, 19], 이는 주로 교착어에서는 한 어근이 여러 개의 형식형태소들과 결합되어 다양한 단어들을 생성하기 때문이다.

따라서 한국어에 대한 기존의 대다수 철자 교정 방법들은 형태소 분석기를 이용하고 있다[4, 5, 8]. 또 철자 검사 사전은 형태소들과 각 형태소의 결합 관계를 표시하는 접속 정보로 이루어져 있으며, 이를 바탕으로 형태소 분석의 성공 여부에 따라 철자의 오류 여부를 판단한다. 즉, 철자 검사는 사전에서 형태소들을 찾아 기술된 형태소 결합 규칙에 의해 주어진 어절이 분석 가능한지 검사함으로써 수행된다. 이를 바탕으로 하는 철자 교정 시스템들에서는 사전 검색이나 형태소들간의 결합 관계를 검사할 때 실패한 위치 정보를 기록하고 철자 검사 시 계산된 결과를 철자 교정의 단계에서 이용하는 것이 일반적인 방법이다. 또 생성된 후보 어절들 중에서 최적의 해를 구하기 위해 품사

† 정 회 원 : LG전자/정보통신 네트워크연구소 연구원
†† 정 회 원 : Text Analysis and Natural Language Engineering IBM T.J. Watson Research Center
††† 정 회 원 : 연세대학교 기전공학부 교수
논문접수 : 2000년 8월 4일, 심사완료 : 2001년 1월 31일

태깅과 분석된 형태소들의 공기 정보 혹은 연어 정보(collocation information)가 이용되기도 했다[5, 6]. 그러나 기존의 사전 검색 방법은 계산된 위치 정보를 이용하여 틀린 어절에 대한 후보를 생성할 때 대상 언어와 오류 유형에 의존적인 정보를 사용하며, 후보를 검증하기 위해 재차 형태소 분석을 수행해야 한다.

또, 문자 또는 음성 인식의 후처리로 철자 교정을 이용하지 않는 경우에는 인식기의 계산 결과를 이용하는데 생성된 후보 문자들 중에는 의도된 문자가 없을 수도 있고, 철자 교정으로 형태소 분석만을 이용하는 경우에 한 문자에 대해 여러 개의 후보 문자를 생성하면 후보의 검증을 위해 필요한 형태소 분석의 횟수가 기하급수적으로 증가되게 된다. 예를 들어 길이가 m인 어절에 대해 각 음절마다 k개의 후보 음절을 생성하면 가능한 어절의 개수는 $O(m^k)$이다.

본 논문에서는 문자 인식이나 음성 인식의 후처리를 주 대상으로 하는 철자 검사기에 대해 논하며 위에 제시된 단점을 보완하기 위해 형태소들간의 편집 거리 계산을 통해서 어절을 생성할 수 있는 사전을 철자 교정 사전으로 이용하는 방법을 제안한다. 제안된 철자 교정 사전은 철자 교정 시 편집 거리를 기반으로 탐색되며, 비교착어를 교정하기 위해 연구된 사전 검색 기법들을 이용함으로써 본 시스템은 휴리스틱에 의존하지 않고 후보를 생성할 수 있다는 장점을 가진다. 또 본 연구는 한국어의 음절 특성을 이용하여 편집 거리를 계산하는 방법을 제시한다. 또한 철자 교정기가 오류 어절에 대한 후보 어절들뿐만 아니라 어절들의 형태소 분석 결과를 산출함으로써 후보 선택 단계에서 분석된 형태소들간의 공기 정보가 손쉽게 이용될 수 있도록 한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 한국어의 음절 특성을 이용한 편집 거리 계산을 설명하고, 제3장에서는 편집 거리에 기반한 어절 생성 사전의 형식화된 표현에 대해 기술한다. 제4장에서는 실험으로서 문자 인식기의 결과로부터 발생되는 오류 어절을 교정한다. 시스템의 성능을 평가하기 위해, 각 생성된 교정 후보들을 대상으로 품사 태깅과 공기 정보를 바탕으로 교정 후보의 우선 순위를 매기고 이에 의해 시스템의 정확성을 측정하며 실험의 결과를 분석한다. 마지막으로 제5장에서는 결론으로서 본 연구에 대해 정리한다.

2. 한국어의 음절 특성을 이용한 편집 거리 계산

편집 거리(edit distance)란 Levenshtein이 형식화한 두 문자열간에 유사성을 계산하는 방법으로, 한 문자열을 다른 문자열로 변환하는데 최소한으로 필요한 토큰의 도치(transposition)[1], 교체(substitution), 삭제(deletion), 삽입(insertion) 연산의 개수이다[15]. 도치 연산은 한 문자열과 인접한 두 개의 토큰의 위치를 바꾸며, 교체 연산은 한 문자열의 토큰을 다른 문자열의 토큰으로 바꾼다. 삭제 연산은 한 문자열의 토큰을 삭제하며, 삽입 연산은 한 문자열에 토큰을 삽입한다. 이 개념은 각각의 연산에 비용값(cost)을 주어 일반화할 수 있으며, 그 알고리즘의 시간 복잡도는 주어지는 문자열의 길이가 각각 N, M일 때 $O(NM)$이다[19]. 하지만 한국어의 음절 특징을 고려하여 연산의 종류를 단순화함으로써 더 빠른 편집 거리 계산이 가능하다.

한국어의 음절은 2차원 형태로 초성, 중성, 종성으로 이루어진다. 또한 초성과 종성에는 자음만이 올 수 있으며 중성에는 모음만이 올 수 있다. 그러므로 음절을 1차원 형태로 표현하여 초성, 중성, 종성을 하나의 토큰으로 간주하여 어절간의 편집 거리를 계산하면 필요 없는 연산이 발생하게 된다. 즉, 한국어의 한 음절은 초성, 중성, 종성이 모여 이루어지며, 초성과 종성은 자음, 중성은 모음으로 구성되고 이들이 한 음절 안에서 위치가 바뀌는 일이 없다. 그러므로 음절의 초성과 중성이 바뀌는 연산에 대해서는 고려할 필요가 없다는 점에 주의를 기울일 필요가 있다.

따라서, 본 논문에서는 각 자소를 토큰으로 간주하는 대신, 한글 각 음절을 3항의 벡터 형태로 표현하고 각각의 음절을 하나의 토큰으로 보아 편집 거리를 계산한다. 예를 들어 두 어절 '사랑하고'와 '사망하고'는 [(ㅅ,ㅏ,null)(ㄹ,ㅏ,ㅇ)(ㅎ,ㅏ,null)(ㄱ,ㅗ,null)], [(ㅅ,ㅏ,null)(ㅁ,ㅏ,ㅇ)(ㅎ,ㅏ,null)(ㄱ,ㅗ,null)]과 같은 토큰열로 각각 표현된다. 이때 편집 거리의 계산을 위한 연산에서는 위에서 설명한 바와 같이 자음-모음과 같이 이질적 자소에는 비교 연산이 적용되지 않고 자음-자음과 같은 동질적 자소에만 비교 연산이 적용된다. '사랑하고'와 '사망하고'의 예를 들면, 첫 번째 토큰 열과 두 번째 토큰 열의 두 번째 토큰을 비교하면 토큰의 첫 번째 요소가 틀려 비용 값은 1로 계산되고 편집 거리는 1이 된다. 더욱이 문자 인식이나 음성 인식에서 발생하는 철자 오류는 음절간에 도치 오류와 앞 음절의 종성과 뒤 음절의 초성이 바뀌는 도치 오류는 발생하지 않는다는 가정을 할 수 있다. 따라서 본 논문에서는 편집 거리 연산의 종류로 토큰간의 교체 연산만 고려한다.

편집 거리 값은 다음에서 보듯이 각각의 토큰 쌍에 대한 교체 연산의 비용 값을 계산하는 것으로, 길이가 N인 두 문자열 X, Y에 대한 편집거리 $ed(X(N), Y(N))$는 다음 (1)과 같이 계산된다. 또한 대치되는 음소의 개수를 C로 할 때 알고리즘의 복잡도는 $O(CN)$이다.

$$ed(X(i), Y(i)) = Cost(X<i>, Y<i>) + ed(X(i-1), Y(i-1))$$
$$ed(X(0), Y(0)) = 0 \quad (1)$$

X와 Y는 비교가 되는 두 어절로 $X(i)$는 $X<0:i>$를 말하며 $X<i>(0 \le i \le N)$는 어절을 이루고 있는 토큰이다.

1) 도치 연산은 Boguraev and Pustejovsky (1996)에서는 언급되지 않았다.

비용값 $Cost(X<i>, Y<i>)$는 두 토큰을 비교하는 단위 연산으로 0에서 3이내의 값을 계산한다.

편집 거리 연산에 기반하여 틀린 어절에 대한 후보 어절을 생성할 때 수행되는 편집 거리 연산의 횟수는 매우 많다. Oflazer(1966)는 그의 실험에서 이것에 대한 통계 자료를 보여준다. 그는 실험에서 약 24,000개의 어근을 가지는 사전을 사용했고, 형태소 분석이 다르더라도 표층 형태가 같으면 같은 단어로 보아 중복해서 단어를 생성하지 않았으며, 후보 어절을 생성하기 위한 어근들의 부분적인 정보가 주어진 상태에서 계산을 하였다. 거리 차의 한계 값을 1로 했을 때 발생하는 연산의 횟수는 약 2,500번이었고, 거리 차를 2로 했을 때 발생하는 횟수는 약 21,000번이었다. 이는 위 연산이 본 시스템에서 얼마나 자주 수행되는지를 보여준다.

3. 편집 거리에 기반한 어절 생성 사전과 철자 교정

3.1 어절 생성 사전의 표현

본 절에서는 철자 교정을 위해 편집 거리를 기반으로 탐색되는 어절 생성 사전에 대해 기술한다. 전술한 바와 같이 교착어인 한국어는 한 어근이 여러 개의 형식 형태소들과 결합되어 다양한 어절들을 생성하므로 단어들의 집합인 사전은 정적인 단어 리스트가 아닌 형태소들을 결합하여 어절들을 생성할 수 있는 생성기로 표현되어야 한다.

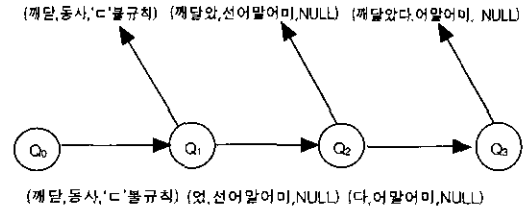
어절 생성 사전 G는 다음과 같은 유한 상태 변환기 M (FST ; finite state transducer)로 표현된다.

$M = (Q, \Sigma, \Sigma', R, F, \delta, \lambda, q_0, D, T, Ir)$

Q : 형태소 분석기 상태들의 집합.
Σ : 입력 알파벳. Σ' : 출력 알파벳.
R : Σ의 부분집합으로 어간들의 집합
F : Q의 부분 집합으로 최종 상태(final state)들의 집합.
q_0 : 시작 상태, $q_0 \in Q$.
δ : 전이 함수, $\delta : Q \times \Sigma \rightarrow Q$
λ : 출력 함수, $\lambda : \Sigma' \times \Sigma \rightarrow \Sigma'$
D : 형태소들의 집합 T : 품사들의 집합
Ir : 불규칙 유형의 집합

입력 알파벳 Σ의 각각의 원소들은 유한 상태 변환기의 마지막 세 요소인 형태소, 품사, 형태소의 불규칙 유형 즉, $(m_i, p_j, irregular\ Type_k)$로 이루어지고 $m_i \in D$, $p_j \in T$, $irregular\ Type_k \in Ir$이다. 출력함수 λ는 문자열 w_j가 생성된 상태에서 형태소 m_i이 입력되었을 때 불규칙 현상과 음운 현상을 통해 w_j와 m_i가 결합했을 때의 표층형태 w_k를 출력하는 함수이다.

본 논문에서 불규칙 유형 집합인 Ir은 국어학에서 분류하는 9가지의 불규칙 활용뿐만 아니라, 자동적 교체가 일어나는 현상과 전산언어학의 관점에서 불규칙 용언으로 간주되는 불규칙 11가지를 포함한 20가지의 불규칙 활용 형태로 구성된다[1]. 예를 들어 어절 "깨달았다"는 어절 생성 사전에서 다음과 같은 패스를 거쳐 생성된다(그림 1).



(그림 1) "깨달았다" 어절 생성 패스

(깨닫, 동사, 'ㄷ'불규칙), (었, 선어말어미, NULL), (다, 어말어미, NULL)는 입력 알파벳의 원소이고 (깨달았, 선어말어미, NULL)은 (깨닫, 동사, 'ㄷ'불규칙)과 (었, 선어말어미, NULL)이 결합할 때 'ㄷ'불규칙과 모음조화 현상이 반영되어 생성되는 문자열이다. (깨달았다, 어말어미, NULL)는 (깨달았, 선어말어미, NULL)과 (다, 어말어미, NULL)이 결합하여 생성되는 문자열로 출력함수에 의해 상태가 전이될 때 생성된다. 출력 알파벳의 원소로 출력 알파벳의 품사와 불규칙 정보는 뒤의 형태소의 것에 따른다. 출력된 문자열의 품사가 체언, 조사, 어말 어미와 같이 어절의 끝에 올 수 있는 품사일 경우 최종 상태로 결정이 된다.

어절 생성 사전 G의 동작을 형식적으로 설명하기 위해 먼저 전이 함수 δ를 상태와 형태소 열을 인자로 취하는 확장 전이 함수 δ^*로 확장시키고, 출력 함수 λ를 표층형태와 형태소 열을 인자로 취하는 확장 전이 함수 λ^*로 확장시킨다.

[정의 1] 어절 생성 사전 G에 대하여 확장 전이 함수 δ^*를 다음과 같이 정의한다.

$\delta^* : Q \times \Sigma^* \rightarrow Q$,
$\delta^*(p, \varepsilon) = p$, ε : *null string*
$\delta^*(p, um) = \delta(\delta^*(p, u), m)$,
$m \in \Sigma$, $u \in \Sigma^*$

[정의 2] 어절 생성 사전 G에 대하여 확장 출력 함수 λ^*를 다음과 같이 정의한다.

$\lambda^* : \Sigma' \times \Sigma^* \rightarrow \Sigma'$,
$\lambda^*(r, \varepsilon) = r$, ε : *null string*
$\lambda^*(r, um) = \lambda(\lambda^*(r, u), m)$,
$r \in R$, $m \in \Sigma$, $u \in \Sigma^*$

이제 정의 1과 정의 2에 의해 하나의 어휘 형태소는 형식 형태소와의 결합을 통해 하나의 어절로 생성된다. 예를 들어, "깨닫"이라는 동사어간에 "고"라는 어미가 결합하여 "깨닫고"라는 어절이 생성되게 된다.

어절 생성 사전의 전이 함수는 형태소들의 결합 관계를 나타내는 것으로 결합 가능한 형태소들 간에는 전이가 존

재한다. 전이를 결정하기 위해 기존의 사전이 갖고 있는 품사, 불규칙, 종성의 종류 등의 정보만을 고려하면 어절 생성 사전은 과다한 패스의 검색을 수행해야 한다. 예를 들어 '사랑(명사) + 이(조사)', '사랑(명사) + 시(접사)', '사랑(명사) + 히(접사)', '사랑(명사) + 씨(접사)', '사(수사) + 라지(어미)', '사(수사) + 랄지(어미)[2]', '사(동사) + 라지(어미)' 등의 전이가 발생할 수 있는데 '사랑시', '사랑히', '사랑씨'는 잘 발생하지 않는 어절이며, 어떤 것들은 접미사의 의미 제약 정보를 사용하면 발생하지 않는 후보다. 이와 같은 과다한 패스 탐색을 방지하기 위해서는 사전의 정보가 매우 복잡해지고 이러한 정보를 손으로 코딩하면 많은 시간과 노력이 필요하고 일관성과 정확성을 보장하기 힘들다.

본 논문에서는 전이함수 δ의 구축을 위해 품사 태그된 말뭉치를 사용했다. 즉, 형태소간의 결합 제약 정보를 품사 부착 말뭉치로부터[3] 자동으로 추출하여 이용한다. 추출되는 정보는 두 형태소가 형태소분석의 결과로 하나의 어절을 구성하는데 사용되면 두 형태소간에 전이가 발생하는 것을 의미하며, δ의 원소가 된다.

3.2 철자 교정

오류 어절에 임계값 t 이내로 철자 교정을 하는 것은 오류 어절과의 편집 거리가 t 이내인 어절들을 생성하는 것이다.

[정의 3] 후보들의 집합 $C(E, t)$는 오류 어절 E와 편집 거리 차가 임계값 t 이하인 어절들의 집합으로 다음과 같이 정의한다.

$$C(E, t) = \{E' | E' = \lambda^*(r, um), \delta^*(q_0, rum) \in F, u \in \sum^*, r \in R, m \in \sum, 0 < D(|E'|, |E|) \le t\}$$

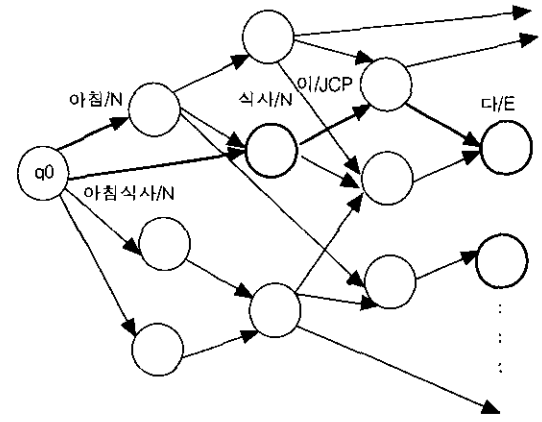
한국어에서 오류 어절에 대한 후보들을 제시하기 위해서 첫 단계로 오류 어절로부터 추출할 수 있는 가능한 모든 어간들을 계산한다. 모든 가능한 어간들이 체언인 경우에는 어절의 Prefix와 임계값 t 이내의 편집 거리로 계산되는 어간을 후보 어간으로 선택하면 된다. 하지만 어간이 용언인 경우에는 어간의 형태가 불규칙현상과 축약 현상으로 변경될 수 있으므로 편집 거리가 t+2 이내인 것들을 어간으로 선택한다. 본 시스템에서는 어간들의 집합 R은 Trie로 구성되어 오류로부터 가능한 어간들을 효율적으로 계산한다.

[정의 4] 후보 어절의 어간들의 집합 $R(E, t)$는 오류 어절 E의 접두어(prefix)와 편집 거리 차가 임계값 t+2 이하인 어절들의 집합으로 다음과 같이 정의한다.

$$R(E, t) = \{r \mid ed(E:1[i], r) \le t+2, 1 \le i \le |E|, r \in R\}$$

정의된 오토마타 G은 알파벳 $\sum$ 안에 있는 심벌을 레이블로 갖는 방향 그래프(directed graph)로 볼 수 있다.

오류 어절에 대해 편집 거리 t 이내의 후보 어절을 생성한다는 것은 시작 상태에서 최종 상태까지 주어진 어절과 편집 거리 차가 t 이내인 가능한 모든 수열을 발견하는 것이다. 예를 들어 (그림 2)의 굵은 선은 오류 어절 '아침직사다'에 대해 후보 어절을 생성하는 패스를 나타낸다. (그림 2)에서 생성 중인 어절은 대부분의 경우 입력 어절과 편집 거리 차를 발생시킨다. 이 때 거리차가 t+2 이상으로 발생한 부분의 그래프 탐색을 수행할 필요가 없다. 이는 더 이상의 수열의 계산은 입력 어절과 t+2 미만으로 편집 거리가 계산되지 않기 때문이다.[4] 이렇게 하여 후보 어절 생성에 필요한 탐색 범위를 전체 그래프의 매우 작은 부분으로 제한한다. 오류 어절 E가 어절 생성 사전 G에 입력되었을 때 후보 어절의 형태소 분석 결과는 정의 5의 정규 집합(regular set)으로 표현된다.



(그림 2) 어절 생성기 G

[정의 5] 오류 어절 E와 편집 거리 t 이내인 후보 어절들의 형태소 분석 결과 R(E,t)를 다음과 같이 정의한다.

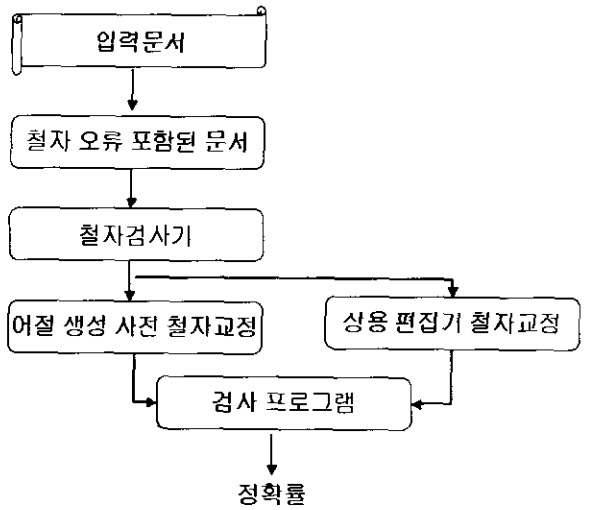
$$R(E, t) = \{m_1 \cdots m_n \mid \delta^*(q_0, m_1 \cdots m_n) \in F, D(|\lambda^*(m_1, m_2 \cdots m_n)|, |E|) \le t, m_1 \in R, m_i \in \sum, 1 \le i \le n\}$$

4. 실 험

4.1 실험 방법

본 시스템을 평가하기 위해 실험에서는 문자 인식의 후처리로서 본 철자 교정을 적용하였다. 제안되는 방법의 성능 평가를 위한 실험 방법은 (그림 3)과 같다. 문자 인식기를 통해 입력 문서에 대한 철자 오류가 포함된 문서를 만든 후 철자검사기를 사용하여 틀린 어절을 추출한다. 다음

2) 서술격조사 '이'가 생략된 형태
3) 대한민국 국어정보베이스
4) 체언인 경우에는 허용치를 t로 한다.



(그림 3) 실험 절차

으로 어절 생성 사전을 이용하여 철자 교정을 시도하여 본 논문에서 제안한 방법을 정확률 측면에서 다른 상용 문자 편집기의 철자 교정과 비교 분석한다.

입력 문서는 어절 생성 사전을 구축하는데 사용된 품사 부착 말뭉치와 국민학교 교과서로부터 임의로 추출된 문장으로 구성된다. 실험한 문자 인식기의 정확률은 노이지가 적은 문서를 대상으로 하였을 때 90%정도로서 결과적으로 오류가 있는 실험 대상 어절은 약 3000개였다.

4.2 사전과 태거 구현

본 시스템이 사용하고 있는 형태소의 범주 체계는 <표 1>과 같이 14개, 사전의 어휘 수는 약 6만 개로 이루어져 있다. 철자 교정에서는 품사들이 세분화될 필요가 없으므로 세분화된 품사로 인한 중의성을 하나로 묶은 14개의 범주로 충분하다.

또, 생성된 후보 어절들 중에서 가장 적절하다고 판단되는 해를 결정하기 위해 두 가지 방법을 이용하였다. 그 첫번째 단계는 품사태깅으로 본 논문에서는 HMM(Hidden Markov Model)을 이용한다.

〈표 1〉 본 논문에서 사용한 품사 범주

N(체언) 대명사, 명사,수사	P(조사)	F(외국어)
V(용언) 형용사, 동사	E(어말 어미)	ADV(부사)
PPE(선어말 어미)	DET(관형사)	JCP(서술격조사)
EXCL(감탄사)	NMZ(명사형 어미)	PUNC(문장부호)
SFX(접미사)	AUX(보조 용언)	

이 때, 묶여 있는 각 중분류 수준의 품사는 세분화된 품사로 확장되는데, 세분화된 품사는 단어 사전으로부터 할당된다. 일례로 체언으로 분석된 '시'의 경우는 일반명사와 단위성 불완전명사가 할당될 것이다.

품사 태깅으로부터 나오는 결과에는 연어 정보를 적용한다. 본 시스템에서 이용되는 품사 태깅은 최적 n개의 결과를 산출하며 상위 n개의 결과로부터 단어들 간의 연어 제약을 사용하여 해를 결정한다. 연어 제약으로는 단어들 간의 공기 정보가 사용되고 공기 정보를 얻기 위해 구문 정보 부착 말뭉치가 사용될 수 있다. 그러나 구문 정보 부착 말뭉치를 구축하는 일은 많은 시간과 노력이 필요한 작업으로 현재 이용 가능한 구문 부착 말뭉치는 불과 10만 어절 정도이다. 또 본 시스템에서 사용한 사전의 체언 수는 약 40,000개이고, 용언의 수는 약 20,000개이다. 여기에 문법적 관계가 조사에 의해 이루어지는데 이 중 10개의 조사만을 고려한다고 해도 총 $40,000*20,000*10 = 8\times 10^9$개의 공기쌍을 가능하게 한다. 따라서 적은 양의 말뭉치로부터 구한 공기 정보는 전체 어휘 수를 고려할 때 매우 적다고 할 수 있다. 따라서 본 연구에서는 약간의 오류를 포함하지만 대량의 원시 말뭉치로부터 추출된 공기 정보를 이용함으로써 자료 희귀성(data sparseness)을 극복하고자 한다.[5] 본 연구에서는 윤준태(1997)의 연구에서 구축된 공기 정보를 이용하였다. 이용된 공기 정보는 3,000만 원시 말뭉치로부터 추출된 약 200만 쌍의 <동사-명사-조사>로 이루어진 어휘 관계와 이들의 빈도수로 이루어져 있다. <표 2>는 이러한 공기 정보의 예를 보여 준다. 표에서 PPOS는 서술어의 품사이며 NPOS는 명사류의 품사이다. V, C, N은 각각 동사, 명사의 서술형(명사+지정사), 명사를 의미한다.

〈표 2〉 서술어-명사-조사 공기 사전의 예

PRED	PPOS	NOUN	NPOS	PP	FREQ
cm이	C	반지름	N	가	8
cm이	C	변	N	가	21
cm이	C	세 로	N	를	1
가	V	가 계	N	를	1
가	V	가 계	N	에	30
가	V	군 대	N	에	81
먹	V	감 주	N	를	1

4.3 실험 결과와 분석

어절 생성 사전을 구축하는데 사용된 품사 부착 말뭉치와 국민학교 교과서로부터 임의로 추출된 30,000 개의 어절을 입력으로 하였을 때 약 3,000개의 어절이 잘못 인식되었다. 잘못 인식된 어절의 약 45%는 편집거리가 1이었고 50%는 2, 나머지 5%가 3 이상이었다.

<표 3>은 틀린 어절에 대해 후보 어절을 생성하기 위해 계산되는 편집거리 연산 횟수와 생성된 후보 어절 개수를 나타낸다.

<표 3>에서 보듯이 본 시스템에서 제안하는 방법은 Oflazer(1966)의 방법보다 생성되는 후보 어절의 개수가 많았다. 왜냐하면 본 시스템에서는 같은 어절일지라도 형태소

7) 연세대학교 컴퓨터과학과 한글정보처리 연구실 어휘관계 사전

분석이 다르면 다른 후보로 여겨지기 때문이다. 또한 한국어에서는 하나의 어절이 여러 개의 형태소 분할되며 서로 다른 태그가 할당되는 중의성이 편재하고 용언의 경우에는 음운의 불규칙 현상과 축약현상으로 인해 주어진 한계값보다 더 큰 값으로 후보 어절을 생성해야 하기 때문이다.

〈표 3〉 철자 교정을 위한 편집거리 연산 횟수와 생성된 후보 어절 개수

	평균 편집거리 연산 횟수		평균 생성된 후보 갯수		
	t = 1	t = 2	t = 1	t = 2	t = 3
본 시스템	2,612.8	22,101.3	11.9	90.2	599.4
Oflazer(1966)	2498.4	20680.4	3.6	52.0	

시스템의 교정 정확도를 알기 위해 정확률을 고려하는데, 정확률은 어절 정확률(EP)과 단어의 정확률(WP)로 나누어 볼 수 있으며 이들은 각각 다음과 같이 계산된다.

$$EP = \frac{\text{교정 후 원 어절과 일치하는 어절의 수}}{\text{전체 오류 어절의 수}} * 100$$

$$WP = \frac{\text{원 어절내의 단어와 일치하는 교정된 어절내 단어의 수}}{\text{전체 오류 어절내의 단어의 수}} * 100$$

<표 4>는 잘못 인식된 어절에 대해 고쳐진 결과의 예를 상용 문서 편집기 A의 결과와 같이 보여주고 있다. 음영이 있는 부분은 교정이 잘못된 것을 말한다. <표 4>에서 보듯이 편집기 A의 철자 교정의 정확률은 매우 낮은데 비해, 어절 생성 사전을 이용한 철자 교정은 언어와 오류 유형에 의존적인 정보를 사용하지 않고 편집거리에 의해 사전이 탐색되면서 후보 어절을 생성하여 상용 편집기보다 정확률이 매우 높았다. <표 5>는 철자 교정 대상인 3,000개의 어절에 대해 본 시스템에 대한 정확률을 나타낸다.

〈표 4〉 오류 어절에 대한 교정 결과 예

의도된 결과	인식 결과	한계값	본 시스템		편집기 A	
			교정 단어	교정 어절	교정 단어	교정 어절
동침치	퐁침치	2	동침(N)	동침이	퐁침치(N)	퐁침치
연음	여으	2	이(JCP)	연음	여느(DET)	여는
유혹	유혹이	1	유혹(N)	유혹이	유혹이(N)	유혹이
내려가다	내력가다	2	내리다(V)	내려가다	내력가다(N)	내력가다
다스리는	다스리튼	1	다스리다(V)	다스리는	다스리다(V)	다스리튼
그	프	2	푸다(V)	퍼	푸다(V)	퍼

<표 5>에서 보는 바와 같이 실험 결과 어절의 정확률은 약 83%이고 단어의 정확률은 약 93%로 나타났다. 또한 문자 인식 오류의 약 5%는 편집 거리가 3인 부분에서 발생한 에러로 이러한 오류는 탐색의 효율을 위해 편집거리를 2로 설정하였으나 문제의 오류를 해결하기 위해서는 편집 거리를 보다 늘려야 할 것이다. 거리 차가 2미만인 것만을 대상으로 하는 경우 어절의 교정율은 약 87%이고 단어의 교정율은 약 98%가 된다.

〈표 5〉 실험 결과 나타난 정확률

편집거리 3인 오류 포함하는 경우		편집거리 2미만인 오류만을 대상으로 한 경우	
EP	WP	EP	WP
83%	93%	87%	93%

5. 결 론

본 논문에서는 지금까지 두 어절 간 음절 특성이 고려된 편집 거리 계산을 기반으로 탐색되는 어절 생성 사전을 이용한 철자 교정을 제안하였다. 어절 생성 사전은 유한 상태 오토마타로 형식화되어 언어와 오류 유형에 의존적인 정보를 이용하지 않고 후보 어절을 생성할 수 있었고, 전이 함수는 말뭉치로부터 구축되어 기존의 사전 정보의 미흡함으로 인한 후보의 과잉 생성을 막을 수 있었으며 쉽게 구축될 수 있었다. 또한 후보 어절과 같이 형태소 분석 결과를 계산하여 태깅이나 공기 패턴의 상호 정보치 같은 방법을 바로 적용할 수 있었다. 3,000개의 어절에서 실험한 결과 어절의 교정율은 약 83%이고 단어의 교정율은 약 93%였다.

참 고 문 헌

[1] 강승식, 김영택, "한국어 형태소 분석기에서 불규칙 용언의 분석 모형", 한국정보과학회 논문지, 제19권 제2호, pp.151-163, 1992.

[2] 김영택, "자연 언어 처리", 교학사, 1994.

[3] 남윤진, 옥철영, "말뭉치 분석에 기반한 명사파생 접미사의 사전 정보 구축".

[4] 박영환, 송만석, "말뭉치에 기반한 형태소 분석기 및 철자 검사기의 구현", 연세대학교 석사학위논문, 1992.

[5] 심철민, 권혁철, "언어 정보에 기반한 한국어 철자 검사와 교정기의 구현", 정보 과학회 논문지, 제23권 제7호, pp.776-785, 1996.

[6] 유진희, 이종혁, 이근배, "형태소 분석과 언어평가를 이용한 문자인식 후처리", 정보과학회 논문지, 제22권 제6호, pp.880-891, 1995.

[7] 윤준태, "공기 관계 기반 어휘 정보를 이용한 한국어 구문 분석", 연세대학교 박사학위 논문, 1997.

[8] 이병훈, 윤준태, 송만석, "말뭉치를 기반으로 한 한국어 철자 교정기의 구현", 한글 및 한국어 정보처리 학술발표논문집, pp.285-293, 1993.

[9] 이하규, "어말-어두 공기 정보를 이용한 한국어 어휘 중의성 해소", 정보과학회 논문지, 제24권 제1호, pp.82-89, 1997.

[10] 최기선, "국어 정보 베이스 CD", KAIST, 1999.

[11] 최현배, "우리말본", 정음문화사, 1989.

[12] 황영숙, 박봉래, 임해창, "한국어의 음절 결합 특성 및 통사적 어휘 특성을 이용한 문자인식 후처리 시스템", 한글 및 한국어 정보처리, pp.175-182, 1997.

[13] Allen, James, Natual Language Understanding, The Benjamin/Cummings, 1994.

[14] Boguraev, Branimir and Pustejovsky, James, Corpus Processing for Lexical Acquitsition, The MIT press, 1996.

[15] Du, M. W. and Chang, S. C., "A model and a fast algorithm for multiple errors spelling correction," Acta Information,

(29) : pp.281-302, 1992.
[16] Golding, Andrew R., "A Baysian hybrid method for context-sensitive spelling correction," cmp-lg, 1996.
[17] Hopcroft, John E. and Ullman, Jeffrey D., Introduction to Automata Theory, Languages, and Computation, Addison-Wesley, 1979.
[18] Kukich, Karen, "Automatically Correcting Words in Text," ACM Computing Surveys, Vol.24, No.4, pp.377-438, 1992.
[19] Oflazer. Kemal, "Error-tolerant Finite-state Recognition with Applications to Mopological Analysis and Spelling Correction," ACL, Vol.22, pp.73-89, 1996.
[20] Ross, Sheldon M., Introduction to Probability and Statistics For Engineers And Scientists, Wiley, 1987.
[21] Wagner, Robert A. and Fischer, Michael J., "The String-to-String Correction Problem," Journal of the ACM, Vol.21, No.1, 1974.

이 영 신

e-mail : youngsinlee@lgic.co.k
1996년 충북대학교 컴퓨터공학과(공학사)
1998년 연세대학교 대학원 컴퓨터과학과 (공학석사)
1998년~1999년 사이버토크 주식회사 자연어처리 연구소 연구원
1999년~현재 LG전자/정보통신 네트워크연구소 연구원
관심분야 : 철자 교정, 음성인식, VoIP(Voice over IP)

박 영 자

e-mail : pyoungja@us.ibm.com
1988년 연세대학교 컴퓨터과학과(이학사)
1988년~1991년 한국국방정보분석 연구소 연구원
1993년 연세대학교 컴퓨터과학과(이학 석사)
1998년 연세대학교 컴퓨터과학과(공학 박사)
1988~현재 Text Analysis and Natural Language Engineering IBM T.J. Watson Research Center
관심분야 : Statistical Natural Language Processing, Digital Library & Information Retrieval, Machine Learning, Artificial Intelligence

송 만 석

e-mail : mssong@december.yonsei.ac.kr
1963년 한남대학교 수학과(이학사)
1968년 연세대학교 대학원 수학과(이학석사)
1972년 University of Wisconsin, Madison, Wisconsin(M.S)
1978년 University of Michigan, Ann, Arbor, Mischigan(Ph.D)
1975년~1981년 국방과학연구소 책임연구원
1981년~현재 연세대학교 공과대학 기전공학부 교수
관심분야 : 자연어처리, 수치해석