Journal of Korea Multimedia Society Vol. 20, No. 2, February 2017(pp. 371-381)
https://doi.org/10.9717/kmms.2017.20.2.371

문맥의존 철자오류 후보 생성을 위한 통계적 언어모형 개선

이정훈[†], 김민호[††], 권혁철[†††]

Improved Statistical Language Model for Context-sensitive Spelling Error Candidates

Jung-Hun Lee[†], Minho Kim[††], Hyuk-Chul Kwon[†††]

ABSTRACT

The performance of the statistical context-sensitive spelling error correction depends on the quality and quantity of the data for statistical language model. In general, the size and quality of data in a statistical language model are proportional. However, as the amount of data increases, the processing speed becomes slower and storage space also takes up a lot. We suggest the improved statistical language model to solve this problem. And we propose an effective spelling error candidate generation method based on a new statistical language model. The proposed statistical model and the correction method based on it improve the performance of the spelling error correction and processing speed.

Key words: Context-Sensitive Spelling Error Correction, Statistical Language Model, Spelling Error Candidate Generation, Natural Language Processing, Text Mining

1. 서 론

최근 빅 데이터(big data) 활용에 관한 관심이 증가하면서 심층학습(deep learning)과 같은 인공지능 관련 기술의 연구가 활발히 진행되고 있다. 특히 스마트폰을 중심으로 한 모바일 사용 환경발달로 텍스트와 같은 비정형 데이터(unstructured data)의 양이 폭발적으로 증가함에 따라, 이를 효과적으로 처리하기 위한 자연어처리 기술에 대한 수요가 그 어느 때보다 많다. 자연어처리 기술의 핵심은 텍스트의 형태적·의미적 분석에 기반을 둔 자연어 이해(natural language understanding)로서 그 중심에 문서 교정 기술이 있다. 텍스트에는 사용자의 실수든, 의도적이든, 또는 무지에 의해서든 철자 오류가 많이 포함되어 있으며, 이러한 오류는 자연어 이해의 정확도를 낮추는 주요한 요인이 된다.

철자오류의 유형은 크게 단순 철자오류(non-word spelling error)와 문맥의존 철자오류(context-sensitive spelling error)로 구분할 수 있다. 전자는 “결죄”와 같이 오류어가 실제로 존재하지 않는 어휘기 때문에 해당 어절을 형태적으로 분석하는 것만으로 쉽게 오류를 검출할 수 있다. 반면에 후자는 “요금결재”와 같이 “요금”과 함께 사용되었을 때 오류가 되기 때문에 해당 어절의 형태적·의미적 특성을 고

※ Corresponding Author : Hyuk-Chul Kwon, Address: (609-735) Busan, Rep. Republic of Korea, TEL : +82-51-510-2875, FAX : +82-51-516-6764, E-mail : hckwon@pusan.ac.kr
Receipt date : Jan. 13, 2017, Approval date : Feb. 6, 2017
[†] Dept. of Electrical and Computer Eng., Graduate School, Pusan National University
(E-mail : it_leejh@pusan.ac.kr)
[††] Dept. of Electrical and Computer Eng., Graduate School, Pusan National University
(E-mail : karma@pusan.ac.kr)
[†††] Dept. of Information Computer Science., College of Eng., Pusan National University
(E-mail : hckwon@pusan.ac.kr)
※ This work was supported by a 2-Year Research Grant of Pusan National University

려해야만 검출할 수 있어 검출이 매우 어렵다.

이러한 문맥의존 철자오류의 주요 원인에는 Table 1에서 보인 바와 같이 “발음 유사성에 따른 오류(errors due to similarity in pronunciation)”, “문법 오류(grammar error)”, “띄어쓰기 오류(word spacing error)”, “오타 오류(typos error)” 등이 있다. 한국어 철자오류 중 문맥의존 철자오류의 비율은 20%를 넘으며, 이는 영어에 비해 4~5배나 높은 비율이다[1, 2]. 한국어 문맥의존 철자오류의 비율이 영어와 비교하여 상대적으로 높은 이유는 현재 사용하는 한글 자판이 입력 오류에 아주 취약하기 때문이다. 영어 QWERTY 자판은 입력 오류로 한 자소가 자판 주위의 자소로 바뀌었을 때 단어가 될 확률이 3.11%이다. 그러나 표준 두벌식 한글 자판에서 한 자소가 주위의 자소로 잘못 입력되었을 때 단어가 될 확률은 14.87%다[3,4].

문맥의존 철자오류의 교정 방법은 크게 ① 규칙을 이용한 방법과 ② 통계적 방법으로 나눌 수 있다. 규칙을 이용한 방법은 발생 빈도가 높거나 정형화된 문맥의존 철자오류를 교정할 수 있는 확률이 높으나, 입력 오류로 일어나는 비정형화된 유형에 대한 교정은 어렵다. 반면에 통계적 방법은 반복성이 작은 문맥의존 철자오류에도 적용할 수 있고, 통계에 사용된 말뭉치를 바꿈으로써 단시간에 다양한 환경에 맞는 철자오류 교정 기술을 개발할 수 있어 활용도가 높다.

본 연구는 통계적 언어모형에 기반을 둔 문맥의존 철자오류 교정에 관한 연구로서 Fig. 1에서와 같이 크게 6단계로 이루어진 통계적 교정 방법을 사용한다. 이 교정 단계 중 성능과 속도에 영향을 주는 것은 4번째 단계인 문맥의존 철자오류 후보 생성 과정이다. 문맥의존 철자오류 후보 생성이란 교정 대상 단어가 오류어라고 가정하였을 때, 오류어 대신 해당 문맥에 쓰일 수 있는 대치어 후보를 생성하는 것을 말한다.

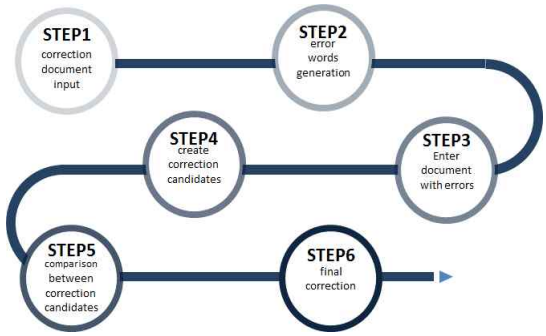

Fig. 1. Spelling error correction sequences.

문맥의존 철자오류 교정의 핵심이 되는 통계적 언어모형은 정보의 양이 많을수록 그 품질이 좋아지지만, 데이터가 많아질수록 처리 속도가 느려지고 저장 공간의 크기가 기하급수적으로 커진다. 이러한 한계점을 극복하고자 본 연구에서는 기존 통계적 언어모형의 자료구조와 알고리즘을 개선하여 데이터를 효율적으로 관리하고 처리 속도를 높인다. 이를 통해 최종적으로 문맥의존 철자오류 교정 후보의 생성 정확도와 교정 속도 그리고 교정의 재현율을 높일 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 통계적 언어모형에 기반을 둔 문맥의존 철자오류 교정 방법에 관해서 이야기하며, 3장에서는 본 논문에서 제안하는 새로운 통계적 언어모형과 이를 활용한 교정 후보 생성 방법에 관한 설명을 한다. 4장에서는 문맥의존 철자오류 교정 실험을 통해 제안하는 방법의 효율성을 검증하고 마지막으로 5장에서는 결론을 말한다.

2. 통계적 문맥의존 철자오류 교정 방법

2.1 통계적 언어모형과 노이지 채널모형

통계적 언어처리에서 가장 널리 사용하는 접근법은 Shannon이 발표한 노이지 채널모형(noisy channel model)에 기반을 둔 방법이다[5-7]. 이 방법은

Table 1. Context-sensitive spelling error type

error type	causes	example
errors due to similarity in pronunciation	Spelling is different but pronounced eqals or similar	peace / piece
grammar error	Occurs when the user does not know the exact grammatical difference	among / between
word spacing error	due to incorrect spacing between vocabularies	maybe / may be
typos error	caused by typos	from / form

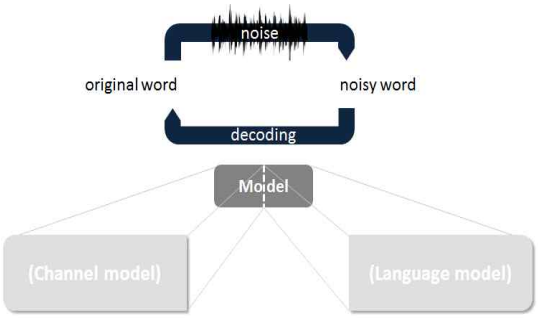

Fig. 2. Noisy channel model.

자연어처리 문제를 디코딩 문제(decoding problem)로 간주한다. Fig. 2에서 보듯이 노이지 채널모형은 채널에 존재하는 잡음(noise)에 의해 입력 데이터가 출력 데이터로 왜곡될 수 있다고 가정하고, 디코더(decoder)를 이용해 출력 데이터로부터 입력 데이터를 복원하는 모형이다. 가령 노이지 채널모형에 기반을 둔 한영 기계번역에서는 입력 데이터를 영어 문서, 출력 데이터를 한국어 문서로 가정한다. 즉, 영어 문서가 노이즈 채널을 통과하면서 한국어 문서로 왜곡된 것으로 간주하고, 왜곡된 데이터를 복원하는 것을 영한 번역으로 보는 것이다.

노이지 채널모형에 기반을 둔 복원 문제는 베이즈 이론(Bayes'theorem)에 의해 표현되며, 출력 데이터의 확률 $p(O)$는 상수이다. 수식 (1)에는 두 개의 확률분포가 존재하는데, 언어모형(language model)인 $p(I)$와 채널 확률(channel probability)인 $p(O|I)$이다.

$$\begin{aligned}\hat{I} &= \arg\max_{I}\ p(I|O) \\ &= \arg\max_{I}\ \frac{p(I)p(O|I)}{p(O)} = \arg\max_{I}\ p(I)p(O|I)\end{aligned} \tag{1}$$

언어모형은 자연언어의 생성이나 이해를 위해 사용하는 모형이며, 통계적 언어 모형화(Statistical Language Modeling)는 문자열의 확률을 예측할 수 있는 언어모형을 구축하는 작업이다. 이때, 통계적 언어모형은 주어진 문자열 s에 대한 확률분포 $p(s)$로 나타낼 수 있다. 통계적 언어모형의 종류에는 Gaussian mixture Language Model[8], Maximum entropy Language Model[9], Neural network Language Model[10], Syntactic/grammar- structured Language Model[11], 그리고 N-gram Language Model 등이 있다. 이 중에서 오늘날 가장 널리 활용되는 통계적 언어모형은 N-gram 모형으로서, 문자열 s에 대한 확률분포 $p(s)$에 대한 근사치를 수식 (2)와 같이 계산한다.

$$\begin{aligned}p(s) = p(w_1,\ldots,w_m) &= \prod_{i=1}^{m} p(w_i|w_1,\ldots,w_{i-1}) \\ &\approx \prod_{i=1}^{m} p(w_i|w_{i-(n-1)},\ldots,w_{i-1})\end{aligned} \tag{2}$$

수식 (3)에서 조건부 확률값은 학습말뭉치로부터 획득한 빈도를 활용하여 최대우도추정법(Maximum Likelihood Estimation; MLE)에 의해 추정된다.

$$p(w_i|w_{i-(n-1)},\ldots,w_{i-1}) = \frac{count(w_{i-(n-1)},\ldots,w_{i-1},w_i)}{count(w_{i-(n-1)},\ldots,w_{i-1})} \tag{3}$$

2.2 통계적 문맥의존 철자오류 교정 모형

노이지 채널모형은 언어모형 $p(I)$와 채널확률 $p(O|I)$를 어떻게 해석하느냐에 따라 다른 응용분야에 적용할 수 있다. 본 연구진의 기존 연구에서는 언어모형 $p(I)$를 사용자가 입력하려고 한 단어열 후보의 확률분포, 채널확률 $p(O|I)$를 철자오류의 발생률로 해석하여 수식 (5)~(9)로 정의되는 통계적 문맥의존 철자오류 교정 모형을 제안하였다[12-13].

수식 (1)에서 입력 데이터 I를 사용자가 입력하고자 한 문서의 단어열 W, 출력 데이터 O를 실제 사용자가 보고 있는 문서의 단어열 Y로 치환하면 수식 (5)와 같은 통계적 문맥의존 철자오류 교정 모형이 된다. 입력 단어열 W와 출력 단어열 Y에서 노이지 채널을 지나면서 바뀐 단어가 T라고 하고, 다른 단어에는 변화가 없다고 가정하면, 문맥의존 철자오류 교정은 확률 $p(W)p(Y|W)$를 최대로 하는 T를 선택하는 문제가 된다. 따라서 수식 (5)는 수식 (6)과 같이 바뀐다.

$$\hat{W} = \arg\max_{W}\ p(W)p(Y|W) \tag{5}$$

$$\hat{T} = \arg\max_{T}\ p(w_1,\cdots,T,\cdots,w_n)p(Y|W) \tag{6}$$

2.1에서 설명한 N-gram 모형을 언어모형으로 사용한다고 하였을 때, 단어열 W에 대한 확률분포 $p(W)$에 대한 근사치를 수식 (7)로 계산할 수 있다. 이때, T를 포함하지 않는 N-gram은 상수항이므로 수식에서 제외할 수 있다. 수식 (4)에서 L_C와 R_C는 T가 단어열 t의 위치에 나타날 때 함께 단어열에 나타나는 주변 문맥 단어로서 각각 왼쪽 문맥과 오른쪽 문맥을 나타낸다. 함수 $f(T)$의 공역은 노이지 채널의

잡음에 의해 T로 출력될 수 있는 모든 단어의 집합이다. N은 N-gram 모형의 차수를 나타내는 수로서 $N=3$인 3-gram 모형을 사용하여 근사치를 구하면 수식 (8)과 같이 된다.

$$\hat{T}=\underset{w_t\in T}{\arg\max}\prod_{k=t-(N-1)}^{t+(N-1)}p(w_k|w_{k-(N-1)},\cdots,w_{k-1})p(Y|W)$$

$$\text{단, }\begin{cases}w_k\in L_C & \text{if } k<t\\ w_k\in f(T) & \text{if } k=t\\ w_k\in R_C & \text{if } k>t\end{cases} \tag{7}$$

$$\hat{T}=\underset{w_t\in f(T)}{\arg\max}\ p(w_t|w_{t-2},w_{t-1})p(w_{t+1}|w_{t-1},w_t)\ p(w_{t+2}|w_t,w_{t+1})p(Y|W) \tag{8}$$

한편, 채널 확률 $p(Y|W)$에서 입력 단어열과 출력 단어열의 길이가 같고($m=n$), 각 문자열에서 단어의 발생이 독립이라면 가정하면 철자오류 발생률 $p(Y|W)$는 수식 (9)와 같이 된다.

$$p(Y|W)\approx\prod_{k=1}^{n}p(y_k|w_k) \tag{9}$$

$$p(Y|W)=\begin{cases}1-\varepsilon & Y=W\\ \varepsilon & otherwise\end{cases}$$

철자오류 발생률은 철자오류의 유형에 따라 달라질 수 있으나, 입력 오류에 의한 철자오류 발생률을 수치화하기 어렵다. 본 연구진의 기존 연구에서는 수식 (9)에서 정의하였듯이, 모든 철자오류 발생률이 같다고 가정하였다. 대신 철자오류 발생률을 사용자가 목적에 맞게 설정한 값으로 정의하여 문서환경에 적응적인 통계적 교정 모형을 만들 수 있었다.

일반적으로 철자오류 교정의 성능은 오류어 검출(detection)과 대치어 선택(correction)에 대한 정확도(precision)와 재현율(recall)로 평가한다. 오류어 검출의 정확도는 시스템이 오류라고 판단한 어휘 중 실제 오류어의 비율로 측정하며, 오류어 검출의 재현율은 문서에 포함된 전체 오류어 중 시스템이 검출한 오류어의 비율로 측정된다. 일반적으로 정확도와 재현율은 서로 반비례하며, 기존 연구에서 영어 문서 교정에 대한 오류어 정확도와 재현율은 50% 내외이다[14-17].

본 연구진의 기존 연구에서는 철자오류 발생률을 조정함으로써 교정 모형의 정확도와 재현율을 사용자 목적에 맞게 조정할 수 있음을 보였다. 즉, 철자오류 발생률을 높게 설정할수록 오류어라고 판단하는 어휘가 많아지므로 재현율은 높아지지만, 정확도는 낮아진다. 극단적인 예로 철자오류 발생률을 100%로 설정하면, 오류 검출 재현율은 항상 100%가 된다. 반면에 철자오류 발생률을 낮게 설정할수록 오류어라고 판단하는 어휘가 적어지므로 재현율은 낮아지고, 정확도는 높아진다.

Fig. 3은 노이지 채널모형에 기반을 둔 통계적 문맥의존 철자오류 교정에 관한 실제 예이다. "… where the core of school…"라는 문장이 입력되었을 때, 교정 시스템은 각각의 단어가 오류어일 가능성을 확인하게 된다. 즉, "core"가 오류어일 경우, "core"의 대치어가 될 수 있는 대치어 후보는 "dore", "cord"이기 때문에 이들 3개 단어를 이용하여 확률을 구한 다음 확률값을 비교한다.

교정 대상 단어인 "core"가 문맥에서 차지하는 위치를 "*"라고 하였을 때, 3개의 tri-gram인 "where the *"(left), "the * of"(middle), "* of school"(right)을 이용하여 각 대치어 후보에 대한 확률이 계산된다. 예를 들어, "core"의 대치어 후보인 "cord"에 대한 확률은 "where the cord"의 빈도 26,298과 "the cord of"의 빈도 26,573 그리고 "cord of school"의 빈도 3,054에 의해 10.8064[1]가 된다. Fig. 3의 예에서는 "core"에 대한 확률값이 다른 후보에 대한 확률값보다 높으므로 오류어로 판단하지 않는다. 만약에 오류어 후보인 "core"보다 확률값이 더 높게 나오는 대치어 후보가 존재한다면, 교정 시스템은 "core"를 오류로 판단하게 된다.

3. 통계적 언어모형을 위한 자료구조와 검색 알고리즘 개선

본 논문에서는 제안하는 통계적 언어모형은 N-gram 언어모형으로서 교정 후보가 포함된 문장의 확률을 계산하는 데 활용된다. N-gram 언어모형에 기반을 둔 문맥의존 철자오류 교정에서 교정 성능과 처리 속도에 영향을 미치는 가장 큰 요소는 대용량 N-gram 통계 사전의 검색이다.

본 논문에서는 교정 후보의 생성과 확률 계산을 위한 가장 기본적인 연산을 'Default 연산'이라고 정의한다. Default 연산은 N-gram 통계 사전에서 특정

1) Fig. 3의 예에서는 철자오류 발생률을 3%로 가정하였다. 또한, 확률의 곱이 0이 되는 것을 방지하고자 확률의 합으로 수식을 변형하기 위해 로그(log)를 수식에 적용하였다.

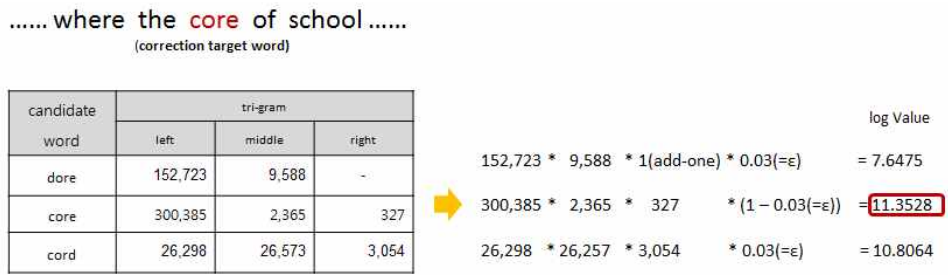

Fig. 3. The example of context-sensitive spelling error correction.

N-gram을 찾는 것에 기반을 둔 알고리즘이다. 예를 들어, 3개의 tri-gram ($a\ b\ *$), ($a * b$), ($* \ a\ b$)가 공통으로 포함하고 있는 대상 어휘 '*'를 찾는 연산이다.

문맥의존 철자오류 교정에서 Default 연산이 교정 성능과 처리 속도에 영향을 미치는 이유는 다음과 같다. N-gram에서 N의 차수가 3인 tri-gram을 사용한다고 가정하고, 대상 단어 T에 대응하는 단어 w_i를 찾는 문제를 생각해보자. T에서 w_i를 구하는 문제는 교정 어휘쌍(confusion matrix)을 이용하여 철자 오류를 찾거나, 시소러스를 이용해 유사 단어나 형제어 따위를 찾는 문제 등 다양한 문제로 해석할 수 있다.

현재 가장 널리 사용하는 방법은 T에서 $f'(T)$, 즉 T에서 생성 가능한 모든 어절을 생성하고, 이 어절을 바탕으로 N-gram을 모두 사전에서 찾는 방법이다. 단, 함수 f'는 T에 연산을 적용해 가능한 어절(candidate word)을 모두 찾아서 집합으로 넘겨주는 함수다. 그러나 이 방법은 사전 검색이 엄청나게 많다. $f'(T)$의 모든 원소에 대해 각각 3번의 사전 검색을 해야 한다. 즉, (c_{i-2}, c_{i-1}, w_i), (c_{i-1}, w_i, c_{i+1}), 그리고 (w_i, c_{i+1}, c_{i+2})를 찾아야 한다. 만약 $f'(T)$가 n개의 원소라면 $3 * n$번 사전 검색이 필요하며, 이 연산이 모든 어절에 대해 행해져야 한다. 그러나 Default 연산을 활용하면 최소한의 사전 검색만으로 모든 N-gram을 검색할 수 있다.

Default 연산을 위해서는 먼저 N-gram 정보를 (N-1)-gram에 Default를 적용해 구할 수 있는 연산을 도입한다. 그리고 Default에 해당하는 값을 연속으로 저장해 찾을 수 있게 한다. 예를 들어 $<w_1, w_2, *>$는 "$w_1 w_2$"로 시작하는 모든 tri-gram의 세 번째 위치의 어절과 빈도를 가지게 된다. 이 연산을 바탕으로 먼저 교정 대상 단어 T의 위치에 올 수 있는 모든 어절을 구한 결과를 수식 (10)에서 정의한 CL(candidate lexicon)이라 한다.

$$CL = <w_{-2}, w_{-1}, *> \cup <w_{-1}, *, w_1> \cup <*, w_1, w_2> \quad (1)$$

Fig. 4는 수식 (10)에서 정의한 Default 연산의 방식을 도식화한 것으로서, "a b * c d"라는 문장에서

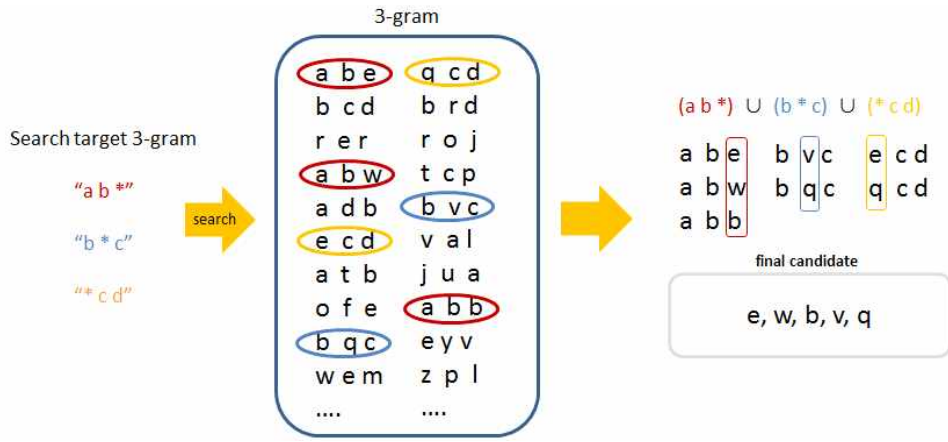

Fig. 4. Union calculation in Default operation.

(a b *) ∪ (b * c) ∪ (* c d)를 만족하는 모든 "*"를 찾는 과정을 보여준다. 말뭉치로부터 추출한 tri-gram의 빈도 사전에서 "a b *"를 만족하는 후보는 3개("a b e", "a b w", "a b b"), "b * c"를 만족하는 후보는 2개("b v c", "b q c"), 그리고 "* c d"를 만족하는 후보는 2개("q c d", "e c d")를 찾을 수 있다. 이렇게 검색된 tri-gram을 합집합 연산을 통해 정렬한다면 중복된 단어가 제거되면서 "*"에 해당하는 최종 단어인 "e, w, b, v, q"를 얻게 된다. 이때, 전체 tri-gram의 검색을 위한 사전 접근 횟수는 총 7회며, 합집합 연산을 통해 추출된 최종 단어의 개수는 총 5개이다.

Fig. 5에서는 Default 연산을 위해 구성한 자료구조의 예를 보여준다. 일반적으로 말뭉치에서 추출한 tri-gram의 수는 매우 크기 때문에 통계 정보의 순차적 검색은 많은 시간이 필요하다. 본 연구에서는 역 N-gram과 Trie 자료구조를 활용한 N-gram 고속 탐색 방법을 제안한다. 먼저 Fig. 5에서 오른쪽 그림의 "Court which have"라는 3-gram을 예로 설명하자면 세 단어 중에 어느 위치가 "*"가 될지 모르기 때문에 총 3가지 나누어 생성한다. "Court which have"의 경우 다음과 같다.

"(*Court) which have"
"Court (*which) have"
"Court which (*have)"

다음 단계에서는 각 "*"위치에 있는 단어들을 가장 오른쪽으로 옮기게 된다.

"(*Court) which have" → "which have (*Court)"
"Court (*which) have" → "Court have (*which)"
"Court which (*have)" → "Court which (*have)"

Fig. 6은 위의 N-gram과 역N-gram의 저장과 검색을 위한 자료구조를 보여준다.

Fig. 6에서 볼 수 있듯이 1-gram은 다음과 같이 $[s_1, N(s_1), point_1]$으로 저장한다. 여기서 $N(s_1)$은 s_1 (1-gram)의 출현빈도를 뜻한다. $point_1$은 s_1으로 시작하는 모든 2-gram이 저장된 저장소를 가리킨다. 2-gram은 $point_1 \rightarrow [s_2, N(s_1,s_2), N(s_2,s_1), point_2]$처럼 기억한다. $N(s_1, s_2)$는 2-gram인 $(s_1,\ s_2)$의 개수다. 당연히 $N(s_2, s_1)$은 2-gram인 $(s_2,\ s_1)$의 개수다. 마지막으로 세 번째 단계는 $point_2 \rightarrow [s_3, N(s_1,s_2,s_3), N(s_3,s_1,s_2), N(s_3,s_2,s_1)]$로 기억한다. 단, $N(s_3, s_1, s_2)$는 $N(s_2, s_1, s_3)$로 저장할 수도 있다.

문맥의존 철자오류 교정에서 교정 후보의 생성은 교정 성능을 좌우하는 작업으로서 이론적으로는 하나의 오류어에 대한 대치어 후보는 해당 오류어를 제외한 실세계에 존재하는 모든 단어이다. 즉, 사전의 크기가 k일 때, 대치어 후보의 수는 $k-1$이 된다.

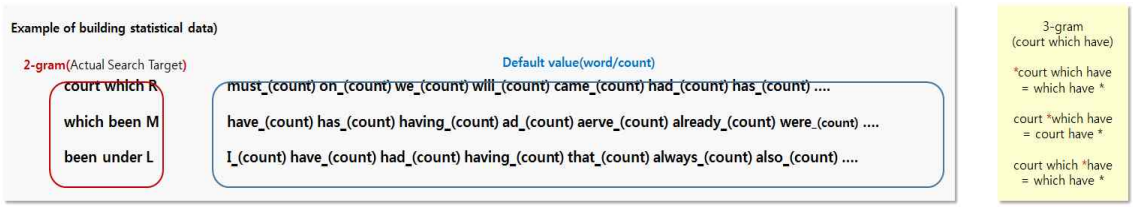

Fig. 5. Example of building statistical data.

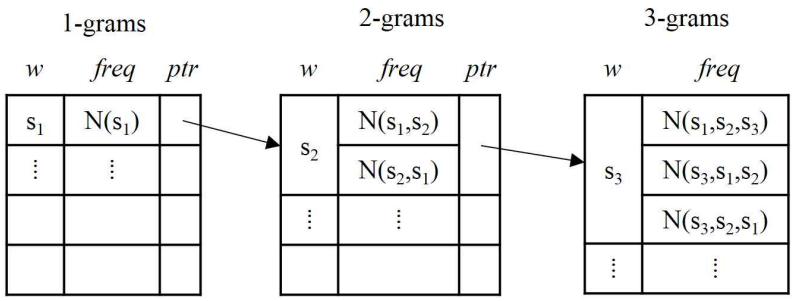

Fig. 6 Data structure for retrieving and storing N-gram and reverse N-gram.

기존 연구에서는 오류어에 따라 대치어 후보를 미리 정의하거나, 혹은 오류어와 대치어 간 편집거리는 1이라는 제약을 두어 N-gram 탐색에 의한 처리 속도의 저하를 방지하였다.

Table 2는 본 연구에서 제안하는 자료구조의 효율성을 검증하고자 기존의 검색 방식과 비교한 결과이다. 기존 연구의 대치어 생성 방식에서는 대치어가 10개가 있을 때 이들을 포함한 3-gram의 검색을 10번 하게 되고, 문장의 길이가 5일 경우 총 3개의 3-gram이 존재하므로 10 × 3 = 30번의 검색이 이루어진다. 하지만 Default 연산에서의 검색 구조는 대치어 후보를 포함한 모든 N-gram 따로 검색하지 않고 주위 문맥의 어절만을 이용하여 검색을 하게 된다. 또한, 검색을 통해서 오류어에 대한 대치어 후보 "*"를 얻게 된다. 즉, 길이 5인 문장에서는 총 3번의 접근만으로 모든 대치어 후보를 얻을 수 있다.

Table 2에서 사용된 문장은 브라운 말뭉치에서 무작위로 추출한 2,000문장으로서, 각 어절 당 후보가 수십 개가 존재할 수 있으므로 문장의 어절 수보다 더 많은 검색이 이루어질 것이다. 그렇지만 Default 연산 구조에서의 검색은 문맥의 길이에 따라 횟수가 제한되므로 본 논문에서 사용하는 길이인 5에서는 총 3번의 검색만 이루어진다. Table 2에서 나와 있듯이 전체 기존 방식이 전체 후보의 3-gram 빈도정보를 검색하는 속도는 96초이지만 Default 연산 구조에서의 속도는 4.6초로 획기적으로 줄었음을 알 수 있다.

말뭉치에서 추출한 모든 3-gram에 앞의 단계를 거치게 된다면 데이터의 양은 실제 3-gram의 3배가 될 것이다. 모든 "*"의 정렬을 완료 후 "*"를 제외한 나머지 두 어절을 2-gram 형태로 간주하여 전체적인 정렬을 하게 되는데 정렬을 거치고 난 뒤에는 동일한 2-gram의 "*"자리에 서로 다른 각각의 어절들이 존재할 것이다. 이런 데이터들은 Fig. 5의 하단의 예와 같이 특정한 실제 데이터에서 "*"의 위치를 알려주는 태그를 붙여서 나타낸다.

이렇게 나눈 이유는 검색의 편리성과 "*"를 한 방향으로 모아서 중복되는 부분의 제거를 통해 저장 공간의 절약을 할 수 있기 때문이다. 첫 과정에서 전체 3-gram을 세배로 늘린 후에 정렬을 통해 일부 데이터를 병합하는 과정을 거치는 것만으로 데이터의 양이 줄어든 것을 Table 3에서 보여주고 있다. 실제 실험에서 사용된 데이터는 Google Web 1T에 포함된 전체 3-gram으로서 실제 3-gram의 개수와 정렬과정에서 "*"가 위치하는 3가지 데이터를 각각의 병합과정까지 끝낸 후의 용량을 보여준다.

Table 3의 정제 과정에서 웹에서 사용하는 태그와 같이 불용어라고 판단되는 부분을 제거하였지만, 실제 사용되는 데이터의 양과 비교한다면 그렇게 많은 부분을 차지하지 않았기 때문에 60GB에서 34.7GB로 데이터의 양이 줄어든 것은 저장 문제만을 따지고 보자면 상당한 효율이 있음을 알 수 있다.

4. 실험결과 및 고찰

본 실험의 교정은 전체 어절을 대상어로 한다. 학습 데이터로는 Google Web 1T를 사용하며, 평가데이터로는 브라운 말뭉치에서 무작위로 선택한 2,000문장을 사용한다. 철자오류 발생률은 1%부터 9%까지 1%씩 값을 변화하면서 진행하였다. 그리고 대치어 후보를 생성하는 과정에서 교정 대상 어절과의 편집거리를 이용한 실험도 진행하였다. Default 연산은 교정 말뭉치의 3-gram을 정제하여 미리 "*"의 자리에 위치하는 해당 어절의 집합을 구축하였기 때문

Table 2. Capacity change according to default operation data structure

	Basic search speed	Speed in the Default operation structure
Document Full Search Time(sec)	96	4.6
Search time per word(sec)	0.00169	0.00008

Table 3. Capacity change according to Default operation data structure

"*" position	basic	left	middle	right
corpus 3-gram (Google 1T)	60.0GB	10.9GB	11.9GB	11.9GB
		sum : 34.7GB		

에 이전 방식과 같이 임의로 후보를 생성하지 않았다. Default 연산을 이용해서 3-gram의 검색 후 “*”에 해당하는 후보를 얻은 뒤 이들을 교정 대상 어절과 유사성을 비교하여 교정 후보로 사용한다.

Table 4는 오류율 3%에서의 기존 교정 방식과 Default 연산을 적용한 교정 방식을 비교한 것이다. 검출 성능에는 큰 차이가 있으며 교정에서도 높은 차이를 보인다. 두 실험은 교정 시스템 외의 모든 환경을 동일하게 진행하였으므로 두 시스템 간의 결과값 차이는 신뢰성이 높다. 결론적으로 두 교정 시스템의 차이는 자료구조의 개선에 따른 오류 어절 및 교정 후보에 사용되는 후보 어절의 생성을 3-gram에서 얻으므로 기존의 방식에서보다 자료의 누락이 크게 줄었기 때문이다. 예로 “the″라는 교정 대상 어절을 교정하였을 경우 기존에는 편집거리와 키보드 근접거리를 고려하여 “rhe″를 생성하였다. 그리고 교정 말뭉치의 1-gram에 존재한다면 “rhe″를 대치어 후보 어절로 선택하였다. 그리고 “the″의 자리를 “rhe″로 대치하여 문장 확률을 구하고 ″the″가 포함된 문장이 “rhe″가 포함된 문장과 확률비교를 하게 된다.

Table 5는 Default 연산을 적용한 실험으로서 철자오류 발생률이 문맥의존 철자오류 교정에 미치는 영향을 분석하기 위한 실험이다. 실험결과에서 볼 수 있듯이 철자오류 발생률이 높일수록 검출 및 교정의 정확도는 낮아지고 재현율은 높아진다. 이는 교정 대상 어휘와 대치어 간 확률값 비교에서 대치어가 포함된 문장의 확률값이 더 높아지도록 영향을 미치기 때문에 전반적으로 오류어로 판단하게 되는 교정 대상 어휘가 많아지기 때문이다. 극단적인 예로 철자오류 발생률을 100%로 설정하면, 오류 검출 재현율은 항상 100%가 된다. 반면에 철자오류 발생률을 낮게 설정할수록 오류어라고 판단하는 어휘가 적어지므로 재현율은 낮아지고, 정확도는 높아지는 것을 알 수 있다.

Table 6은 교정 대상 후보와 대치어 간 편집거리

Table 4. Comparison table

system	basic		Default operation	
	detection	correction	detection	correction
정밀도	98.62%	96.73%	99.46%	97.94%
재현율	76.17%	74.71%	87.30%	85.96%
F1	85.95%	84.31%	92.98%	91.57%

Table 5. Error Rate Change Test Chart

	detection	correction	detection	correction	detection	correction
Error Rate	1		2		3	
precision	95.89%	94.98%	95.18%	93.71%	94.67%	93.23%
recall	65.70%	65.45%	69.64%	68.57%	72.26%	71.16%
F1	77.97%	77.50%	80.43%	79.19%	81.96%	80.71%
Error Rate	4		5		6	
precision	94.20%	92.72%	93.98%	92.49%	93.73%	92.23%
recall	74.54%	73.37%	76.10%	74.89%	77.13%	75.89%
F1	83.22%	81.92%	84.10%	82.76%	84.62%	83.27%
Error Rate	7		8		9	
precision	93.43%	91.93%	93.25%	91.77%	93.06%	91.54%
recall	78.42%	77.16%	79.26%	77.99%	80.18%	78.87%
F1	85.27%	83.90%	85.69%	84.32%	86.14%	84.73%

Table 6. Edit distance adjustment experiment

	Detection	Correction	Detection	Correction	Detection	Correction
Edit Distance	1		2		3	
Precision	94.67%	93.23%	94.12%	89.88%	92.73%	86.21%
Recall	72.26%	71.16%	75.19%	71.80%	75.39%	70.09%
F1	81.96%	80.71%	83.59%	79.83%	83.17%	77.32%

가 문맥의존 철자오류 교정에 미치는 영향을 분석한 실험결과이다. 철자오류 발생률은 3%로 설정하였으며, 편집거리가 늘어남에 따라 교정의 성능이 조금씩 떨어짐을 볼 수 있다. 참고로 교정 대상 어절을 구할 때 편집거리가 늘어날수록 더 많은 어절이 나타나게 되고 이렇게 나타난 어절들과의 비교가 많아질수록 실제 확률비교에서 정답 어절이 아닌 다른 어절로 교정이 이루어지는 가능성이 커지므로 교정 성능이 낮아지게 된다. 이런 경향은 교정 대상 어절의 길이가 짧을수록 더 빈번하게 나타난다. 예로 "if" 편집거리가 1인 단어를 생각할 때 "is"와는 편집거리가 1이지만 전혀 다른 어절이기 때문에 실험결과에 영향을 미치게 된다. 따라서 단순히 편집거리를 고정하는 것보다는 교정 대상 어절의 길이에 따라 단계적 편집거리를 적용하는 것이 철자오류 교정 성능에 더 유리할 것으로 예상한다.

5. 결 론

본 연구는 통계적 문맥의존 철자오류 교정에 관한 연구로서 통계적 언어모형의 자료구조와 알고리즘 개선을 통한 교정 성능과 처리 속도의 향상을 목표로 하였다. 일반적으로 통계적 언어모형의 성능은 정보의 양이 많을수록 좋은 결과를 보여주지만, 데이터가 많아질수록 처리 속도가 느려지고 저장 공간의 크기가 기하급수적으로 커진다. 이런 점을 극복하고자 본 연구에서는 기존의 통계적 언어모형의 자료구조와 알고리즘을 개선하여 데이터를 효율적으로 관리하고 처리 속도를 높였다. 또한, 이를 통해 최종적으로 문맥의존 철자오류 교정 후보의 생성 정확도와 교정 속도를 높일 수 있었다.

본 연구에서 제안하는 새로운 통계적 언어모형은 Default 연산이라고 정의한 과정을 통해 n-gram 데이터의 검색을 위한 데이터 접근 횟수와 데이터 저장 용량을 상당히 낮출 수 있었다. 또한, Default 연산을 이용한 교정 후보의 생성은 기존의 통계적 철자오류 교정에서 널리 사용되는 미리 정의된 교정 후보 쌍이 아닌 실제 교정 대상 어절과 공기하는 주위문맥에 존재하는 어절을 후보로 선택하므로 교정의 재현율을 높일 수 있었다. 실제 문맥의존 철자오류 연구가 활발히 이루어진 영어권 연구에서도 문맥의존 철자오류 교정의 정확도와 재현율이 50% 내외임을 비춰보았을 때, 본 연구에서 제안하는 방법의 효과를 알 수 있다.

본 연구결과는 의미 처리기, 문장분석기, 어의 중의성 해결, 개체명(named entity) 인식 등 자연언어처리의 기본모듈 개발과 성능 향상의 기반이 될 것으로 기대한다. 또한, 음성인식, 문자인식, 이동형 단말기를 위한 검색 등 통계적 언어모형에 기반을 둔 다양한 자연언어처리 응용분야에서 요소기술로 활용되어 관련 기술의 성능을 향상할 수 있을 것으로 기대한다.

향후 연구에서는 영어가 아닌 한국어 문맥의존 철자오류 교정을 위해 통계적 언어모형을 적용하고, 최근 각광받고 있는 심층학습 기반 언어모형과의 결합 가능성을 검토할 예정이다.

REFERENCE

[1] C.W. Young, C.M. Eastman, and R.L. Oakman, "An Analysis of Ill-formed Input in Natural Language Queries to Document Retrieval Systems," *Information Processing and Management,* Vol. 27, No. 6, pp. 615-622, 1991.

[2] A.M. Wing and A.D. Baddeley, "Spelling Errors in Handwriting: A Corpus and Distributional Analysis," *Cognitive Processes in Spelling*, Academic Press, London, 1980.

[3] H.S. Choi, A.S. Yoon, and H.C. Kwon, "Improving Recall for Context-Sensitive Spelling

Correction Rules Through Integrated Constraint Loosening Method," *Korean Institute of Information Scientists and Engineers Transactions on Computing Practices*, Vol. 21, No. 6, pp. 412-417, 2015.

[4] H.S. Choi, H.C. Kwon, and A.S. Yoon, "Improving Recall for Context-Sensitive Spelling Correction Rules using Conditional Probability Model with Dynamic Window Sizes," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 42, No. 5, pp. 629-636, 2015.

[5] D. III Hal and D. Marcu. "A Noisy-channel Model for Document Compression," *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics,* pp. 449-456, 2002.

[6] Kolak, Okan, and P. Resnik. "OCR Error Correction Using a Noisy Channel Model," *Proceedings of the Second International Conference on Human Language Technology Research,* pp. 257-262, 2002.

[7] D. Yuret, and M.A. Yatbaz. "The Noisy Channel Model for Unsupervised Word Sense Disambiguation," *Computational Linguistics*, Vol. 36, No. 1, pp. 111-127, 2010.

[8] M.J. Kim, S.Y. Suk, K.S. Kim, H.Y. Jung and H.Y. Chung, "Hybrid Method using Frame Selection and Weighting Model Rank to improve Performance of Real-time Text-Independent Speaker Recognition System based on GMM," *Journal of Korea Multimedia Society*, Vol. 5, No. 5, pp. 515-522, 2002.

[9] A.L. Berger, S.A.D. Pietra, and V.J.D. Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics*, Vol. 22, No. 1, pp. 39-71, 1996.

[10] T. Mikolov, S. Kombrink, A. Deoras, L. Burget and J. Cernocky, "Extensions of Recurrent Neural Network Language Model," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 5528-5531, 2011.

[11] M. Collins, B. Roark, and M. Saraclar, "Discriminative Syntactic Language Modeling for Speech Recognition," *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 507-514, 2005.

[12] M.H. Kim, S.K. Choi, and H.C. Kwon, "Context-sensitive Spelling Error Correction Using Eojeol N-gram," *Journal of Korean Institute of Information Scientists and Engineers*, Vol. 41, No. 12, pp. 1081-1089, 2014.

[13] M.H. Kim, S.K. Choi, J.Z. Jin, and H.C. Kwon, "Adaptive Context-Sensitive Spelling Error Correction Techniques for the Extremely Unpredictable Error Generating Language Environments," *Proceedings of 2015 IEEE International Conference on Computer and Information Technology*, pp. 927-928, 2015.

[14] A. Islam and D. Inkpen, "Semantic Text Similarity Using Corpus-based Word Similarity and String Similarity," *ACM Transactions on Knowledge Discovery from Data*, Vol. 2, No. 2, pp. 1-25, 2008.

[15] A. Islam and D. Inkpen, "Real-Word Spelling Correction Using Google Web 1T 3-grams," *Proceeding of International Conference on Natural Language Processing and Knowledge Engineering*, Vol. 3, pp. 1241-1249, 2009.

[16] A. Islam and D. Inkpen, "Real-word Spelling Correction Using Google Web 1T n-gram Data Set," *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pp. 1689-1692, 2010.

[17] A. Wilcox-O'Hearn, G. Hirst, and A. Budanitsky, "Real-word Spelling Correction with Trigrams: A Reconsideration of the Mays, Damerau, and Mercer Model," *Proceedings of 9th International Conference on Intelligent Text Processing and Computational Linguistics*, Vol. 4919, pp. 605-616, 2008.

이 정 훈

2014년 경상대학교 컴퓨터공학부 공학사
2017년 부산대학교 전기전자컴퓨터공학과 석사
관심분야 : 자연언어처리, 기계학습

김 민 호

2007년 부산대학교 정보컴퓨터공학부 학사
2009년 부산대학교 컴퓨터공학과 석사
2009년~현재 부산대학교 전자전기컴퓨터공학과 박사과정
관심분야 : 자연언어처리, 정보검색, 인공지능, 기계학습

권 혁 철

1982년 서울대학교 컴퓨터공학과 학사
1984년 서울대학교 컴퓨터공학과 석사
1987년 서울대학교 컴퓨터공학과 박사
1992년~1993년 (미)Stanford 대학 CSLI 방문 교수
1987년~현재 부산대학교 정보컴퓨터공학부, 인지과학협동과정 교수
관심분야 : 인간언어공학, 정보검색, 인공지능