단어 간 연관성 측정을 통한 문맥 철자오류 교정1)

최성기*, 김민호*, 권혁철*
*부산대학교 컴퓨터공학과
e-mail:view88@pusan.ac.kr

Context-sensitive Spelling Correction using Measuring Relationship between Words

Sung-Ki Choi*, Minho Kim*, Hyuk-Chul Kwon*
*Dept of Computer Science & Engineering, Pusan National University

요 약

한국어 텍스트에 나타나는 오류어의 유형은 크게 단순 철자오류와 문맥 철자오류로 구분할 수 있다. 이중 문맥 철자오류는 문맥의 의미·통사적 관계를 고려해야만 해당 어휘의 오류 여부를 알 수 있는 오류로서 철자오류 중 교정 난도가 가장 높다. 문맥 철자오류의 유형은 발음 유상성에 따른 오류, 오타 오류, 문법 오류, 띄어쓰기 오류로 구분할 수 있다. 본 연구에서는 오타 오류에 의해 발생하는 문맥 철자오류를 어의 중의성 해소와 같은 문제로 보고 교정 어휘 쌍을 이용한 통계적 문맥 철자오류 교정 방법을 제안한다. 미리 생성한 교정 어휘 쌍을 대상으로 교정 어휘 쌍의 각 어휘와 주변 문맥 간 의미적 연관성을 통계적으로 측정하여 문맥 철자오류를 검색하고 교정한다. 제안한 방법을 적용한 결과 3개의 교정 어휘 쌍 모두 90%를 넘는 정확도를 보였다.

1. 서론

컴퓨터, 인터넷과 스마트 폰이 융합된 정보환경은 소셜 네트워크 서비스(SNS)를 비롯한 새로운 정보유통 환경을 구축하였고, 모든 사람이 정보의 생산자이자 소비자가 되었다. 이에 따라 실수든 의도적이든 또는 무지든 문서에 포함된 철자 오류는 더욱 증가하고 있다. 여기에 더해 두벌식 자판, 세벌식 자판, 스마트 폰과 피처폰 등 다양한 입력 환경에 따라 입력 오류의 형태도 다양한 다른 특성을 보이면서 발생하고 있다. 여기에 더해 한류, 국제결혼의 증가와 같은 국제화에 따라 한국어를 사용하거나 배우는 외국인이 크게 늘고 있다.

이런 환경 변화에 따라 한국어 문서 교정기의 성능 향상에 대한 요구가 증대하고 있다. 그런데 기존의 규칙에 기반을 둔 철자 검사 기술로는 이런 변화에 적응하는 문서 교정기를 개발하기는 불가능하다. 그 가장 큰 이유는 '문맥 철자오류'의 교정과 검색이 어렵기 때문이다. 문맥 철자오류(context-sensitive spelling error)는 단순 철자오류(non-word spelling error)와는 단어가 사용된 문맥을 통해서 오류 여부를 알 수 있다.

문맥 철자오류는 현재 해결해야 할 중요한 대상이지만, 기존 한국어 문서 교정기 중 문맥 철자오류의 교정이 가능한 것은 부산대학교 개발한 시스템이 유일하다. 또 부산대학교가 개발한 시스템도 규칙에 기반을 둔 접근이므로 한국어 사용자가 자주 틀리는 정형화된 문맥 철자오류 외에는 고칠 수 없다.

<표 1> 문맥 철자오류 유형

유형	원인	예
homophone error (발음 유사성에 따른 오류)	철자는 다르나 발음이 같거나 유사하여 발생함	peace / piece 낫다/낮다
typographical error (오타 오류)	오타에 의해 발생함	form / from 가장/가정
grammatical error (문법 오류)	사용자가 문법의 차이를 정확히 알지 못해서 발생함	among / between 여기가 학교 보다 멀다.
cross word boundary error (띄어쓰기 오류)	단어 사이의 잘못된 공백 때문에 발생함	maybe / may be 대학생 선교회/대학 생선 교회

<표 1>은 문맥 철자오류의 유형을 구분한 것이다. 본 논문에서는 다양한 문맥 철자오류 중 정형성과 반복성이 약한 '입력 오류에 의한 문맥 철자 오류'를 대상으로 하며, 사용 방법론은 통계적 접근을 이용한다. 본 논문은 '통계적 문맥 철자 오류 교정' 연구의 초기 연구로서 교정 어휘 쌍을 이용한 통계적 문맥 철자오류 교정 방법을 제안한다. 이후 논문의 구성은 다음과 같다. 2장에서는 통계적 문맥 철자오류 교정의 연구현황을 분석하고, 3장에서는 본 논문에서는 제안하는 교정 어휘 쌍을 이용한 통계적 문맥 철

1) 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2012R1A2A2A06046730).

자오류 교정 방법을 설명한다. 4장에서는 교정 어휘 쌍의 생성과 생성한 교정 어휘 쌍에 대한 문맥 철자 오류 교정 실험 결과를 제시한다. 마지막으로 5장에서는 결론과 향후 연구에 대해 설명한다.

2. 관련 연구

문서 교정기에서 문맥 철자 오류를 교정하는 방법은 크게 규칙을 이용한 방법과 통계적 방법으로 나뉜다. 규칙을 이용한 방법은 사람이 직접 규칙을 만드는 방법과 기계학습을 이용하는 방법으로 나뉜다. 통계적 문맥 철자 오류 검사와 교정 방법은 영어를 대상으로 활발히 연구되었으며, 다음과 같이 크게 3가지를 들 수 있다.

첫 번째는 교정 어휘 쌍을 이용한 방법으로 기본적으로 어의 중의성 해결(word sense disambiguation, WSD) 방식과 같은 방법론을 이용한다. 즉 교정 어휘 쌍에 해당하는 단어가 중의적이라 보고, 통계적 방법으로 중의성을 해결한 후 그 결과와 원래 단어가 같으면 철자가 바르다고 보고, 아니면 문맥 철자 오류로 본다. A. R. Golding[1]은 좌우 10개 단어와 좌우의 문법 요소를 이용하여 오류가 없다고 추정되는 문서에서 통계치를 구하고, 오류가 있는 문서에서 비감독 학습을 하는 방법으로 연구를 진행했다. 예제는 자주 틀리는 21개 교정 어휘 쌍에 행해졌으며, 바르게 분류한 것은 95% 내외로 베이지안 방법보다는 3~4% 높았다.

두 번째 방법은 n-gram에 기반을 둔 언어모형을 사용하는 것으로 Mays[2]가 처음 도입했다. 이 방법은 대용량 말뭉치에서 어절 3-gram을 구하고, 이를 바탕으로 각 문장 또는 부분 문장의 확률을 계산한다. 그리고 그 문장 또는 부분 문장에서 빈도가 낮은 3-gram 중 철자 오류로 생성될 수 있으면서 확률이 높은 3-gram으로 대치한 문장이나 부분 문장의 확률을 원래 확률과 비교하여 문맥 철자 오류를 찾는 방법이다. 이 방법은 과거에는 주로 음성인식이나 문자인식 등의 후처리 방법으로 이용되었지만, 1조 어절이라는 천문학적인 영어 말뭉치에서 3-gram을 구하여 공개한 'Web 1T 3-grams'를 발표하면서부터 문맥 철자 오류 교정에 이용되기 시작했다. Islam은 자신들이 개발한 스트링 유사도 계산법을 이용하여 3-gram에 의한 문맥 철자 오류 교정 방법[2, 3]을 개발하였다. 그러나 이 방법은 정확도와 재현율이 50% 내외로 문맥 절자 오류 교정에는 아직 실용화하기 어렵다[4]. 더구나 1조 어절 이상의 대용량 자료에서 구한 정확도가 이 정도밖에 안 되며, 또한 필요한 메모리도 아주 크다.

세 번째 방법은 문서 전체를 분석하여 사용된 어휘가 문맥상으로 일관성을 유지하는지를 검증하는 방법이다. 이 방법은 어휘 간의 관계를 분석하기 위한 일종의 지식베이스가 필요하다. Hirst[5]는 영어 워드넷(Princeton WordNet, 이하 PWN)을 이용하여 이 문제에 접근했다. 그러나 아직 기술 부족으로 정확도와 재현율이 30% 내외밖에 나오지 않고, 전치사를 비롯한 기능어는 검사할 수 없는 약점이 있다.

본 논문에서는 교정 어휘 쌍을 이용한 통계적 방식으로 문맥 철자오류를 교정하는 방법을 제안한다.

3. 교정 어휘 쌍을 이용한 문맥 철자오류 교정

문맥 철자오류 유형 중 가장 빈번하게 발생하는 오류는 오타에 의해 발생하는 오류이다. 예를 들어, 자판을 이용하여 "오류 교정"을 입력할 때 글쇠 위치가 가까워 "*오류 교정*"을 "*오류 고정*"으로 입력할 수 있다. 그런데 "*교정*"에서 "ㅛ"를 위에 있는 "ㅗ"로 잘못 입력한 결과가 우리가 사용하는 단어인 "*고정*"이 되어 의미 분석 없이 이 오류를 찾기는 쉽지 않다. "*교정*"은 편집거리 1인 "*고정*", "*교장*", "*교전*", "*교종*" 따위로 잘못 입력되어도 오류를 교정하려면 의미 분석이 필요하다. 하지만 현재 개발된 의미분석 기술로 문맥 철자 오류를 교정하는 것은 불가능하다.

이에 따라 통계적 방법으로 이 문제에 접근하는 방법이 영어권에서는 다양하게 연구되었다. 이 중 가장 성능이 높으면서, 단순한 방법이 '교정 어휘 쌍'을 이용하는 방법이다. 즉, 문맥 철자 오류 어절의 주변 문맥에 있는 어절이 문맥 철자 오류 어절과 어울릴 확률은 상대적으로 낮다. 예를 들어 "*철자 오류 교정이 너무 어렵다*"를 "*철자 오류 교장이 너무 어렵다*"로 잘못 썼다면 문장이 아주 어색하다. 이 어색한 정도를 통계적으로 측정하여 어느 범위를 넘어서면 문맥 철자 오류로 판단하는 것이다.

어의 중의성 해소에서 단어의 의미를 파악하는 데 가장 중요한 단서는 주변 문맥에 함께 나타난 공기 어휘이다. 마찬가지로 문맥 철자오류에서도 해당 단어가 오류인지 아닌지를 판단하는 데 가장 중요한 단서는 공기 어휘이다. 본 논문에서는 철자가 비슷한 어휘들을 '교정 어휘 쌍'으로 선정하고, '교정 어휘 쌍'의 어휘들과 문맥에 나타난 공기 어휘 간 연관성을 측정하여 문맥 철자오류를 검색하고 교정한다. 문맥 철자오류 검정의 대상이 되는 단어(이하 검증단어)와 공기 어휘 간 연관성이 검증단어가 속한 교정 어휘 쌍의 다른 단어(이하 대치단어)와 공기 어휘 간 연관성보다 작다면 해당 단어를 오류로 판단한다.

<표 2는> '21세기 세종계획'의 최종 성과물인 '세종 형태소 분석 말뭉치'에서 추출한 통계 정보 중의 일부로써, 문장 내에서 틀리기 쉬운 단어인 '정치', '장치'와 함께 나타난 단어의 빈도를 내림차순으로 정렬하여 상위 10개만 나타낸 것이다. '정치'는 주로 '경제', '사회', '문화'와 같은 단어와 자주 나타나고, 반면에 '장치'는 '제도', '기억', '컴퓨터' 등과 같은 단어와 자주 나타난다. 그러나, '하다', '있다'와 같이 특정 단어와 상관없이 나타나는 단어도 있기 때문에 본 논문에서는 단어 간 연관성 측정을 위해 카이스퀘어 통계량을 사용한다. 그리고 <표 2>를 보면 '정치', '장치'의 공기 어휘 중에 동사나 형용사는 중복되는 것들이 많지만, 명사는 중복이 없는 것을 볼 수 있다. 이것을 봤을 때 주변 문맥 어휘 중에서 명사 어휘와의 연관성만 검사하는 것이 오류를 판단할 때 더 좋을 것으로 추측할

<표 2> '정치', '장치'와 가장 많이 나타나는 어휘

순위	공기 어휘	'정치'와의 공기 출현 빈도	공기 어휘	'장치'와의 공기 출현 빈도
1	경제(명사)	1276	있다(동사)	170
2	사회(명사)	1206	제도(명사)	136
3	하다(동사)	1094	위하다(동사)	114
4	있다(동사)	921	이다(형용사)	77
5	대하다(동사)	721	하다(동사)	75
6	문화(명사)	562	기억(명사)	72
7	문제(명사)	558	없다(형용사)	63
8	위하다(동사)	539	되다(동사)	60
9	되다(동사)	524	마련(명사)	59
10	우리(명사)	489	컴퓨터(명사)	56

수 있다. <표 3>은 '세종 형태소 분석 말뭉치'에서 추출한 통계 정보를 바탕으로, '장치'가 등장한 문장 내에서 '장치'의 주변 문맥 어휘 중 명사 어휘와의 카이스퀘어 통계량을 '정치'와도 계산한 결과이다. <표 3>의 값을 보면 전체적으로 '정치'와의 카이스퀘어 통계량이 훨씬 큰 것을 알 수 있다.

<표 3> '정치', '장치'와 문맥 내 공기 어휘 사이의 카이스퀘어 통계량

공기 어휘	'정치'와의 공기 출현 빈도	'장치'와의 공기 출현 빈도	'장치'와의 카이스퀘어 통계량	'정치'와의 카이스퀘어 통계량
경우	154	21	287.84	26.49
혜택	9	1	16.39	0.09
이용	62	29	33.39	117.09
필요	108	21	226.75	79.24

일반적으로 공기 어휘의 수는 1개 이상이기 때문에 검증단어와 공기 어휘 간 카이스퀘어 통계량과 대응단어와 공기 어휘 간 카이스퀘어 통계량을 비교할 때 카이스퀘어 통계량을 비교하는 방법이 필요하다. 가장 단순한 방법은 모든 카이스퀘어 통계량의 합, 곱, 혹은 평균을 구해 비교하는 것이다. 즉, 검증단어 tw_1이 속한 교정 어휘 쌍을 $TW=\{tw_1,\dots,tw_k\}$이라 하고, tw_1이 쓰인 문맥 내 공기 어휘를 $CW=\{cw_1,\dots,cw_m\}$라고 할 때, 다음 수식을 만족하면 tw_1은 문맥 철자오류가 아니다.

$$argmax_{tw\in TW}\sum_{cw\in CW}\chi^2(cw,tw)=tw_1 \qquad (1)$$

수식 (1)은 카이스퀘어 통계량의 합을 사용한 수식이며, 곱이나 평균을 사용한 수식은 각각 수식 (2)와 (3)으로 나

$$argmax_{tw\in TW}\prod_{cw\in CW}\chi^2(cw,tw)=tw_1 \qquad (2)$$

타낼 수 있다.그러나 수식 (1), (2), (3) 모두 특정한 단어

$$argmax_{tw\in TW}\frac{\sum_{i=1}^{m}\chi^2(cw_m,tw)}{m}=tw_1 \qquad (3)$$

하나 때문에 결과가 달라질 수 있다는 단점이 있다. 즉, 카이스퀘어 통계량 값이 매우 큰 하나의 공기 어휘 때문에 다른 공기 어휘가 무시될 수 있다. 물론, 카이스퀘어 통계량 값이 매우 크다는 것은 중요한 단서가 될 수 있지만, 카이스퀘어 통계량이 특정 신뢰도를 넘는 값을 가지는 단어들이 얼마나 많은지가 더 큰 단서가 된다.

본 논문에서는 검증단어와 대치단어 각각 자신과의 카이스퀘어 통계량 값이 특정 신뢰도를 넘는 값을 가지는 단어들의 개수를 세고, 그 개수를 비교하여 대응단어 쪽의 개수가 3개 이상 많으면 오류로 판단하고, 차이가 1개나 2개가 나면 카이스퀘어 통계량이 가장 큰 어휘를 하나씩 뽑아 그 카이스퀘어 통계량의 차를 이용해 오류 여부를 판단하는 것을 기본 알고리즘으로 하였다. 여기서 앞에서 밝혔듯이 오류 검증단어와 대응단어의 카이스퀘어 통계량 임계값을 다르게 적용하는 것이 정확도가 더 높으므로 각각 2.71과 7.88로 카이스퀘어 통계량 임계값을 따로 적용하였다.

또한, 여기에 카이스퀘어 통계량이 2.71 이하가 되면 신뢰도가 낮아 의미가 없다고 판단하여 각 단어와의 카이스퀘어 통계량이 모두 2.71 이하일 경우에는 해당 어휘를 오류 여부를 판단할 때 사용하지 않도록 하는 방법을 알고리즘에 추가로 적용하였다. 카이스퀘어 통계량 하한값은 2.71 이하의 다양한 값을 사용해 보았으나 성능 변화가 미미하여 본 논문에서는 2.71 값을 하한 임계값으로 설정하고 실험을 진행하였다.

4. 실험 결과 및 분석

4.1 실험 환경

본 논문에서는 평가를 위해 '21세기 세종계획'의 최종 결과물인 '세종 형태 분석 말뭉치(약 1500만 어절)'를 이용하였다. 이 말뭉치에서 명사, 형용사, 동사를 추출하여 각 어휘 간 공기 빈도를 알 수 있도록 사전화하였다. 그리고 성능 평가를 위해 세종 말뭉치에서 3개의 틀리기 쉬운 교정 어휘 쌍(정치-장치, 정비-장비, 사정-사장)을 포함하는 모든 문장을 추출한 후, 이 중에서 어휘마다 100개의 문장을 따로 떼어내어 평가 말뭉치를 구축하였다. <표 4>는 평가할 3개 교정 어휘 쌍의 구성을 나타낸 것이다.

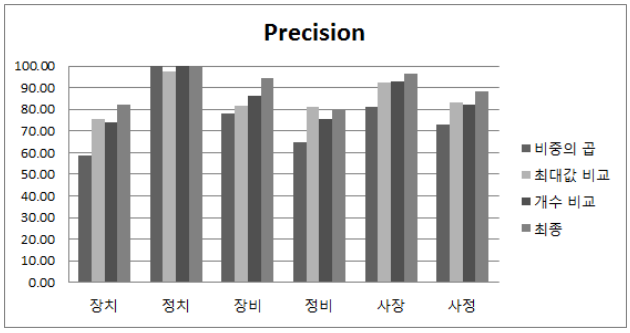
<표 4> 말뭉치의 구성

어휘쌍 (어휘1-어휘2)	학습 (어휘1-어휘2)	평가 (어휘1-어휘2)
장비-정비	812-681	100-100
사정-사장	2888-2821	100-100
정치-장치	13415-1795	100-100

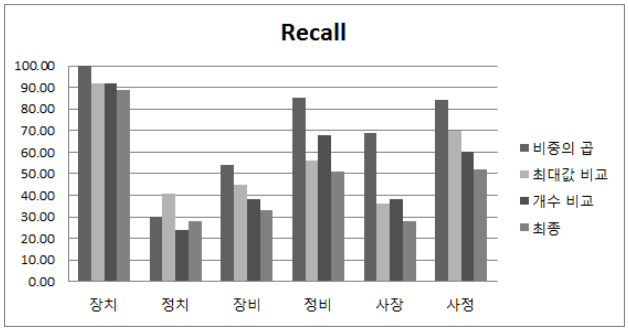
여기서 평가 데이터는 교정 어휘 쌍 별로 200문장의 2개의 데이터, 총 6개의 평가 데이터를 사용한다. 각 데이터에 포함된 문장은 모두 같은 검증 어휘가 나타나는 문장으로서 즉, 정답 문장 100개, 오류 문장 100개로 이루어진 데이터이다.

4.2 실험 방법 및 결과

(그림 1]과 (그림 2)는 윈도우 사이즈가 5일 때 평가 데이터에 나타난 교정 어휘 쌍별 정확도(Precision)와 재현율(Recall)을 구한 것이다.



(그림 1) 실험 대상 6개 어휘에 대한 Precision 비교



(그림 2) 실험 대상 6개 어휘에 대한 Recall 비교

기본 알고리즘의 성능을 평가하고자 일반적으로 어의 중의성 해소에서 사용하는 방법인 '문맥 단어와의 카이스퀘어 통계량의 비중의 곱을 비교하는 방법'과 '카이스퀘어 통계량 최대값이 5배 이상 차이 날 때 오류로 보는 방법'과 그리고 '카이스퀘어 통계량이 특정 값 이상인 어휘의 개수 차이가 1개 이상일 경우 오류로 보는 방법'도 실험해 보았다. 실험 결과를 보면 대부분 어휘에서 비중의 곱을 비교하는 방법보다 다른 방법들이 정확도(Precision)가 높고 재현율(Recall)이 낮은 것을 볼 수 있다. 최대값을 비교하는 방법과 개수를 비교하는 방법은 서로 성능이 거의 비슷하였지만, 두 방법을 결합한 최종적인 방법에서는 확실히 정확도(Precision)가 높아지고 재현율(Recall)이 낮아지는 것을 볼 수 있다.

<표 5>의 전체 평균을 살펴보면 본 논문에서 사용한 최종 방법이 정확도(Precision)가 제일 높으며 재현율(Recall)이 제일 낮은 것을 볼 수 있다. 하지만 맞춤법 검사기에서는 정확도(Precision)가 매우 중요한 요소이고, 재현율(Recall)이 46%라는 것은 규칙 기반 맞춤법 검사의 재현율이 10%를 못 넘는다는 것을 볼 때 월등하게 향상된 성능이므로 본 논문에서 사용한 방법이 의미가 있다고 볼 수 있다.

<표 5> 전체 평균 비교

구분	Precision	Recall
비중의 곱	76.03	70.33
최대값 비교	85.27	56.67
개수 비교	85.16	53.33
최종 방법	90.18	46.83

5. 결론 및 향후 연구

본 논문에서는 기존에 상용화되어 있는 한국어 맞춤법 검사기에서 사용하는 규칙 기반 문맥상 철자오류 교정 방법의 재현율을 극적으로 향상시켜 실용성을 높여보고자 통계적 방법을 이용한 문맥 기반 오류 교정 방법을 제안하였다. 문서에 나타나는 문맥 철자오류의 발생률을 50%라도 가정하였을 때, 90.18%의 정확도와 46.83%의 재현율을 보였다. 본 연구의 의의로써 기존에 상용화된 한국어 맞춤법 검사기는 규칙 기반이라 재현율을 향상하는데 한계가 있었지만, 통계적 방법을 사용한 맞춤법 검사에서 그 발전 가능성을 보았다.

향후 연구 계획으로는 가장 먼저 아직 시도해보지 못한 다양한 오류 검출 방법에 대한 실험이 필요하다. 그리고 데이터를 증가시키면서 어휘별로 데이터 증가에 따른 성능의 변화를 관찰하는 실험도 필요하다. 또한, 실험 자료를 더 수집하여 좀 더 신뢰할 수 있는 실험 결과를 도출하는 것도 필요하며 지금은 오류율을 50%로 가정하였지만, 실제 오류율은 훨씬 작으므로 실제 오류율에서의 성능을 평가하는 것도 필요하다.

참고문헌

[1] Golding, Andrew R. and Dan Roth and J. Moon(ey and Claire Cardie (1999) "A Winnow-Based Approach to Context-Sensitive Spelling Correction," Machine Learning, Vol.34, pp.107-130.

[2] Islam, Aminul and Diana Inkpen (2008) "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.

[3] Islam, Aminul and Diana Inkpen (2009) "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.

[4] Wilcox-O'Hearn, Amber, Graeme Hirst, and Alexander Budanitsky (2008) "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.

[5] Hirst, Graeme and Alexander Budanitsky (2005) "Correcting real-word spelling errors by restoring lexical cohesion", Natural Language Engineering, Vol.11, No.1, pp.87-111.