creative commons

COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

조선대학교
CHOSUN UNIVERSITY

[UCI]I804:24011-200000266192

2017년 2월

석사학위 논문

코사인 유사도 알고리즘을 이용한 한국어 철자오류 교정방법

조선대학교 산업기술융합대학원

소프트웨어융합공학과

박 승 현

조선대학교
CHOSUN UNIVERSITY

2017년 2월

석사학위논문

코사인 유사도 알고리즘을 이용한 한국어 철자오류 교정방법

박 승 현

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

코사인 유사도 알고리즘을 이용한 한국어 철자오류 교정방법

Method for Spelling Error Correction in Korean Using a Cosine Similarity Algorithm

2017년 2월 25일

조선대학교 산업기술융합대학원

소프트웨어융합공학과

박 승 현

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

코사인 유사도 알고리즘을 이용한 한국어 철자오류 교정방법

지도교수 김 판 구

이 논문을 공학석사학위신청 논문으로 제출함

2016년 10월

조선대학교 산업기술융합대학원

소프트웨어융합공학과

박 승 현

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

박승현의 석사학위논문을 인준함

위원장 조선대학교 교수 반성범 (인)

위 원 조선대학교 교수 신주현 (인)

위 원 조선대학교 교수 김판구 (인)

2016년 11월

조선대학교 산업기술융합대학원

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

목 차

ABSTRACT

Ⅰ. 서론 ····· 1
A. 연구 배경 및 목적 ····· 1
B. 연구 내용 및 구성 ····· 3

Ⅱ. 관련연구 ····· 4
A. 철자오류 교정 ····· 4
B. 한글 편집거리 알고리즘 ····· 7

Ⅲ. 코사인 유사도를 이용한 철자오류 교정 ····· 9
A. 철자오류 교정 ····· 9
B. 말뭉치 사전 구축 ····· 11
C. 철자오류 검출 ····· 13
1. 단어의 동시등장 빈도를 이용한 코사인 유사도 측정 ····· 13
2. 철자오류 검출 알고리즘을 사용한 검출 예시 ····· 15
D. 교정 단어 리스트 구축 ····· 18
1. 한글 편집거리 알고리즘 ····· 18
2. 철자오류 단어와 정상 단어의 편집거리 추출방법 ····· 19
E. 철자오류 교정 ····· 23

Ⅳ. 실험 및 결과 ····· 25
A. 철자오류 데이터 ····· 25
B. 철자오류 검출 ····· 28
C. 철자오류 교정 단어 리스트 구축 ····· 33

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

D. 철자오류 교정 ····· 36
E. 철자오류 교정 알고리즘의 성능 비교 ····· 37

V. 결론 및 제언 ····· 39

참고문헌 ····· 40

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

표 목차

[표 2-1] 문맥의존 철자오류의 유형과 발생원인 ····· 4
[표 3-1] 동시등장 빈도를 위한 예시 문장 ····· 13
[표 3-2] 철자오류 검출 알고리즘 ····· 14
[표 3-3] 철자오류 검출 예시 문장 ····· 14
[표 3-4] 예시 문장의 추출 단어 리스트 ····· 15
[표 3-5] 단어들의 동시 출현 빈도 ····· 16
[표 3-6] 단어들의 동시 등장 빈도를 이용한 코사인 유사도 ····· 16
[표 3-7] 단어들의 코사인 유사도 평균 ····· 17
[표 3-8] 편집거리 알고리즘 예시 단어 ····· 19
[표 3-9] 예시 단어들의 편집거리 ····· 20
[표 3-10] 철자오류가 발생한 단어 리스트 ····· 21
[표 3-11] SylED알고리즘 편집거리 통계 ····· 21
[표 3-12] PhoED알고리즘 편집거리 통계 ····· 22
[표 3-13] 추출된 '사임'의 교정단어 리스트 ····· 23
[표 3-14] 교정 단어의 좌우문맥 단어 동시 등장 빈도 ····· 23
[표 3-15] 교정 단어의 코사인 유사도 추출 ····· 24
[표 3-16] 교정 단어의 코사인 유사도 평균 ····· 24
[표 3-17] 철자오류 교정 알고리즘 ····· 24
[표 4-1] 트위터 데이터 수집 방법 및 순서 ····· 26
[표 4-2] 트위터 데이터 수집을 위한 자바 소스 코드 ····· 26
[표 4-3] 코사인 유사도 임계값에 따른 통계 표 ····· 30
[표 4-4] 코사인 유사도 임계값에 따른 통계 상세 표 ····· 31
[표 4-5] 음절 길이에 따른 철자오류 단어의 편집거리 ····· 33
[표 4-6] 기준을 갖춘 음절 길이에 따른 철자오류 단어의 편집거리 ····· 34
[표 4-7] 철자오류 교정 단어 리스트 구축을 위한 규칙 ····· 34
[표 4-8] 철자오류 교정 알고리즘의 성능 ····· 36

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그림 목차

[그림 2-1] KorED알고리즘 ····· 7

[그림 2-2] GrpSIM알고리즘 ····· 8

[그림 3-1] 시스템 구성도 ····· 9

[그림 3-2] 수집한 뉴스 기사 데이터 ····· 12

[그림 3-2] 구축한 말뭉치 사전 데이터 ····· 12

[그림 4-1] 수집한 트위터 데이터 ····· 27

[그림 4-2] 철자오류 데이터 ····· 28

[그림 4-3] 코사인 유사도 임계값에 따른 통계 그래프 ····· 30

[그림 4-4] 코사인 유사도 임계값에 따른 통계 상세 그래프 ····· 31

[그림 4-5] 철자오류 교정 알고리즘의 F-Measure 비교 그래프 ····· 37

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

ABSTRACT

Method for Spelling Error Correction in Korean Using a Cosine Similarity Algorithm

SeungHyeon Bak
Advisor : Prof. PanKoo Kim Ph.D.
Department of Software Convergence Engineering
Graduate School of Industry Technology Convergence,
Chosun University

Long time has passed since computers which used to be a means of research were commercialized and available for the general public. People used writing instruments to write before computer was commercialized. However, today a growing number of them are using computers to write instead. Computerized word processing helps write faster and reduces fatigue of hands than writing instruments, making it better fit to making long texts. However, word processing programs are more likely to cause spelling errors by the mistake of users. Spelling errors distort the shape of words, making it easy for the writer to find and correct directly, but those caused due to users' lack of knowledge or those hard to find may make it almost impossible to produce a document free of spelling errors. Even though people often write for chatting or on their SNS pages on the Internet, there are no set of spelling rules and manners for cyberspace and as a result, serious issues have not be raised to date. However, spelling errors in important documents such as theses or business proposals may lead to falling reliability. These spelling errors should be completely avoided, but it is not easy even for experts. Consequently, it is necessary to conduct research on high-level spelling error correction programs for the general

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

public. This study was designed to produce a system to correct sentence-level spelling errors to normal words with Korean alphabet similarity algorithm. On the basis of findings reported in related literatures that corrected words are significantly similar to misspelled words in form, spelling errors were extracted from a corpus. Extracted corrected words were replaced with misspelled ones to correct spelling errors with spelling error detection algorithm.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅰ. 서론

A. 연구 배경 및 목적

계산기에서 발전하여 탄생하게 된 컴퓨터는 인간이 직접 계산하는데 시간이 오래 걸리는 문제들을 빠른 시간 내에 처리할 수 있도록 개발되어졌다. 초기의 컴퓨터는 방 하나를 차지할 정도로 크기가 거대하였고, 거대한 크기만큼 개인이 함부로 소유할 수 없을 정도로 구매 비용이 막대하였다. 하지만 기술이 발전함에 따라 컴퓨터의 크기는 점점 더 소형화 되어졌으며 그로 인해 컴퓨터의 가격이 하락하면서 일반인들도 쉽게 구매할 수 있도록 사용화 되어졌다. 기존에 사람들은 문서를 작성하기 위하여 종이와 볼펜 같은 필기도구를 이용하여 문서를 작성하였으나, 컴퓨터가 상용화 된지 오랜 시간이 지난 현재에는 컴퓨터를 이용하여 문서를 작성하는 경우가 많아졌다. 컴퓨터를 이용하여 문서를 작성하는 방식은 필기도구를 이용하여 문서를 작성하는 방식에 비해 문서의 작성 속도가 월등히 빠르며 손에 가해지는 피로 또한 적기 때문에 작성에 시간이 오래 걸리는 장문의 문서를 작성할 경우 사람들은 일반적으로 필기도구를 이용하기보다 컴퓨터를 이용하여 문서를 작성하는 경우가 많다. 하지만 컴퓨터를 이용해 문서를 작성하는 방식은 작성자가 팬을 이용해 철자를 그리는 형태의 직접적인 방식이 아닌, 키보드의 자판을 누름으로서 철자를 입력하는 간접적인 방식으로 문서를 작성하기 때문에 철자오류가 발생할 확률이 높다. 보통 철자오류는 문서를 작성하는 작성자의 눈에 쉽게 발견되기 때문에 발생 즉시 교정되어진다. 하지만 철자오류의 종류에 따라 쉽게 발견하기 힘든 부류의 철자오류가 존재하며, 또한 작성자의 지식부족으로 인해 발생한 철자오류도 존재하기 때문에 철자오류가 존재하지 않는 문서는 작성하기 어렵다. 철자오류가 발생할 경우 문장의 내용을 곡해할 수 있기 때문에 작성이 완료된 문서는 읽는 대상에게 정확한 내용을 전달 위하여 반복적으로 철자오류 교정을 수행하지만, 그럼에도 불구하고 철자오류가 존재하는 경우가 많다. 철자오류는 문장의 내용을 곡해할 뿐만이 아니라 논문이라 사업 제안서와 같은 중요 문서에서 발견될 경우 문서와 연관된 연구나 실험 혹은 사업에 대한 신뢰도를 하락시키는 문제를 발생시킬 수 있다. 문제 발생을 막기 위해 철자오류는 철저히 교정되어야 하지만 철자오류를

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

완벽히 교정하는 것은 교정 지식이 풍부한 교열 전문가들 또한 수행하기 어렵다. 철자오류 교정은 교정 지식이 미비한 일반인들이 수행하기에는 큰 어려움을 겪고 있으며, 더군다나 일반인들이 철자오류를 완벽히 교정한다는 것은 불가능에 가깝다. 일반인들의 철자오류 교정을 보조하고자 문서를 작성하기 위해 제작된 워드프로세서와 같은 문서 작성 프로그램은 작성자의 철자오류 교정 보조 시스템을 제공하고 있으나, 문서 작성 프로그램에서 제공하는 한국어 철자오류 교정 시스템은 사전을 기반으로 하여 규칙에 맞지 않는 철자오류를 교정하고 사전에 존재하지 않는 단어들을 표시할 뿐 문맥에 맞지 않는 철자오류는 교정하지 못하고 있다. 이 때문에 문서 작성 프로그램 사용 중 발생한 철자오류의 일부는 교정 시스템에 발견되지 못한 채 문서에 남는 경우가 존재한다. 이와 같은 문제를 해결하기 위하여 철자오류 교정 시스템에 대한 추가적인 연구가 필요하다.

본 논문은 철자오류 교정을 수행하기 위해 기존의 코사인 유사도를 이용하여 철자오류를 검출하는 알고리즘을 개량하여 철자오류를 검출할 뿐만이 아니라 추가적으로 철자오류를 교정하는 방법에 대해 연구하는 논문이다. 기존의 철자오류 교정 방식은 철자오류가 발생한 단어의 철자를 하나씩 치환한 다음 주변 단어들과 동시 등장빈도를 구함으로서 철자오류를 교정한다. 하지만 이런 방식은 철자오류 교정을 완료하는데 소모되는 시간이 길어질 수 있다. 그렇기 때문에 철자오류를 교정하기 위한 방법 중 하나로 한글 편집거리 알고리즘을 사용하여 철자오류 교정 단어를 추출하게 된다. 본 논문에서는 제안하는 철자오류 교정 알고리즘의 성능을 향상시키기 위해 코사인 유사도의 임계값을 탐색하며, 최종적으로 철자오류 교정에 대한 성능을 제시한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 연구 내용 및 구성

본 연구에서는 코사인 유사도를 사용한 철자오류 검출 알고리즘과 한글 편집거리 알고리즘을 통합하여 철자오류를 교정하는 새로운 알고리즘을 제안하고, 또한 교정 정확도를 높이기 위해 수행한 연구의 내용을 보여주고 있다. 본 논문의 구성은 다음과 같다.

2장에서는 철자오류 검출과 철자오류 교정을 위해 수행되어진 기존 연구들과 한글 편집거리를 구하기 위해 수행되어진 기존 연구들에 대해 소개한다.

3장에서는 철자오류 교정에 대한 연구를 수행하기 위한 데이터에 대하여 언급하며, 더불어 철자오류 검출 및 교정에 사용할 알고리즘에 대해 설명한다.

4장에서는 소개한 철자오류 검출 및 교정 시스템에 대한 정확도를 높이기 위해 수행한 연구의 내용에 대해 설명한다. 또한 철자오류 교정을 위하 한글 편집거리 알고리즘을 이용해 교정단어를 추출하는 방법에 대해서 이야기한다.

마지막으로 5장에서는 본 연구에 대한 전체적인 결과와 향후 연구에 대해 제시하며 마무리한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

II. 관련 연구

A. 철자오류 교정

철자오류는 단어의 철자 중 본래 입력해야할 철자 대신 다른 철자를 입력하거나 추가적으로 철자를 입력 혹은 입력하지 못하여 단어가 다른 의미의 단어로 변하거나 혹은 의미가 없는 단어로 변하는 형태의 오류를 의미한다. 이렇게 발생한 철자오류 중 단순히 의미가 없는 단어로 변한 철자오류를 단순 철자오류, 그리고 다른 의미로 변하여 문맥에 맞지 않는 단어로 치환된 철자오류를 문맥의존 철자오류라고 한다[1]. 단순 철자오류는 '요금 결죄'와 같이 사전에 등록되지 않은 어휘를 사용한 오류를 의미하며 텍스트를 형태적으로 분석하는 것만으로도 쉽게 철자오류를 검출할 수 있다. 문맥의존 철자오류는 '요금 결재'와 같이 문맥의 의미나 관계를 고려해야만 오류의 여부를 알 수 있는 철자오류를 의미한다. 따라서 맞춤법 검사기의 성능은 문맥의존 철자오류의 검출과 교정이 큰 영향을 끼친다[2]. 그렇기 때문에 최근 철자오류 교정에 대한 연구는 문맥의존 철자오류에 대한 내용으로 활발히 진행 중이다[3]. 문맥의존 철자오류는 "발음 유사성에 따른 오류", "문법 오류", "띄어쓰기 오류", "오타 오류"의 4가지로 구분할 수 있다[4]. [표 2-1]은 문맥의존 철자오류의 유형과 발생 원인에 대한 예를 보여주고 있다.

[표 2-1] 문맥의존 철자오류의 유형과 발생원인

오류 유형	발생원인	예
발음 유사성에 따른 오류	철자는 다르지만, 발음이 같거나 유사하여 발생	낫다 / 낳다
문법 오류	사용자가 문법의 차이를 정확히 알지 못해 발생	정말 / 너무
띄어쓰기 오류	어휘 사이의 잘못된 공백 때문에 발생	대학생 선교회 / 대학 생선 교회
오타 오류	오타에 의해 발생	장치 / 정치

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

문맥의존 철자오류 교정은 크게 구문 분석에 기반을 둔 규칙을 이용한 방법과 n-gram과 같은 통계 모형을 이용한 통계적 방법으로 구분할 수 있다[5]. 규칙을 기반으로 한 철자오류 교정 방법은 고도의 교정 지식을 갖추고 있는 교열전문가에 의해 제작된 규칙을 통해 철자오류를 교정하게 된다. 하지만 규칙을 기반으로 한 철자오류 교정 방법은 추가된 규칙이 증가할수록 성능이 향상된다는 장점이 존재하지만 규칙의 유지 및 보수에 대해 막대한 비용이 필요하며, 시스템의 유지 기간 동안 반드시 교정 지식을 갖춘 교열전문가의 존재가 필수 불가결이라는 단점이 존재한다[6]. 통계 모형을 기반으로 한 철자오류 교정 방법은 대표적으로 n-그램 언어 모델을 사용한 방법, 교정 어휘 쌍을 이용한 방법, 그리고 노이지 채널 모델을 기반으로 한 방법이 있다.

n-그램 언어 모델을 이용한 방법은 말뭉치 사전에서 어절 n-그램을 구하고, 이를 이용하여 철자오류를 검사할 문장에서 어절의 동시 등장 확률을 계산함으로서 철자오류를 교정하는 방법이다[7]. 영어권 언어에서는 'Web 1T 3-Grams'라는 대용량 말뭉치 사전이 발표되면서 3-gram을 이용한 철자오류 교정 방법들이 연구되어졌다[8, 9].

한국에서는 단어와 조사간의 결합정보를 학습하기 위해 어절 n-gram을 이용하며, 또한 말뭉치 사전의 자료 부족 문제를 해결하기 위해 확률 추청 모델을 사용하여 철자오류 교정을 수행한 연구가 존재한다[10]. 하지만 일반적으로 한글로 작성된 문장은 어절이 복잡한 형태로 이루어졌기 때문에 어절 n-gram이 아닌 형태소 n-gram을 기준으로 한 철자오류 교정 방법이 활발히 연구되어지고 있다.

교정 어휘 쌍을 이용하는 방법은 기본적으로 어의 중의성 해결 방식과 같은 방법론을 이용한다[11, 12]. 교정 어휘 쌍을 이용하는 철자오류 교정 방법은 교정 어휘 쌍에 해당하는 어휘가 중의적이라고 보고, 통계적인 방법을 통해 중의성을 해결한 다음, 그 결과가 원래 어휘와 같다면 철자가 바르다고 보지만, 아니라면 문맥의존 철자오류라고 판단한다.

노이지 채널 모델에 기반을 둔 철자오류 교정 방법은 기계 번역, 광학 문자 인식, 음성 인식등과 같은 입력 데이터가 노이지 채널에 의해 데이터의 변형이 일어나 출력 데이터에 오랴가 발생하였다고 가정하여 출력으로부터 입력을 이끌어 내는 디코딩 문제로 간주하여 해결한다[13].

다수의 철자오류 교정 방법 연구는 영어권 언어를 대상으로 수행되었지만 한국

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

어를 대상으로 한 철자오류 교정 방법에 관한 연구 역시 다수 진행되었다[4, 14]. 그 중 한 연구는 교정 어휘 쌍을 이용한 철자오류 교정 방법을 한국어에 적용하여, 편집거리가 1에 해당하는 어휘들을 교정 어휘 쌍으로 선정한 후 어절 단위의 문맥 크기가 3인 교정 어휘 쌍의 어휘나 문맥에 나타난 공이 어휘 간의 확률을 계산하여 철자오류를 교정하였다[15]. 어떤 연구에서는 철자오류가 발생할 확률은 적으며, 작성자가 한번 발생시킨 철자오류는 재 발생할 확률이 극히 희박하기 때문에 단어의 출현빈도를 이용하여 철자오류는 검출하는 연구가 있다[17]. 또한 어떤 연구에서는 띄어쓰기와 철자오류 교정을 통합적으로 수행하기 위해서 문장을 자소 단위로 분리하여 통계적 모델을 이용함으로서 철자오류를 교정하였다[18].

기존의 코사인 유사도를 이용하여 철자오류를 교정하는 연구에서는 문장에서 등장하는 단어들의 의존 관계에 따라 단어들의 교정 어휘 쌍을 이룬 다음 코사인 유사도를 이용해 동시 출현 확률을 계산하여 철자오류를 교정하는 연구가 존재한다[16]. 하지만 이 연구의 경우 단어들을 의존 관계에 따라 묶을 경우 특정 단어는 교정 어휘 쌍을 이룬 단어의 개수가 줄어 철자오류 교정을 수행하지 못하는 상황이 올 수 있기 때문에 보다 확실히 철자오류 교정을 수행할 수 있는 방법을 구축하기 위하여 본 연구를 수행하게 되었다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 한글 편집거리 알고리즘

편집거리란 문자열 A를 문자열 B로 치환하기 위해 필요한 삽입이나 삭제, 치환과 같은 연산의 최소 개수를 의미한다[25]. 편집거리를 구하는 문제에 대해서는 동적 프로그래밍을 이용하여 최적값을 구하는 여러 방법들이 알려져 있다[19, 20]. 하지만 대부분의 편집거리에 대한 기존 연구들은 알파벳을 기반으로 한 영어권 언어들을 대상으로 이루어졌으며, 한글과 같이 복잡한 형태를 이루고 있는 언어들에 대해서는 많은 연구가 진행되지 않았다. 그렇기 때문에 [21]에서는 한글에 적합한 편집거리 알고리즘에 대해 연구를 진행하였다. 그로인해 한글의 편집거리를 음절 단위 기준으로 추출하는 SylED알고리즘과 편집거리를 음소 단위 기준으로 추출하는 PhoED알고리즘을 제안하였다. 그리고 두 알고리즘을 복합적으로 사용하여 두 단어의 음소 단위가 다르다면 α를, 음절 단위가 다르다면 β를 더하는 형태로 문자열의 길이가 다름에도 불구하고 편집거리를 구할 수 있는 KorED알고리즘 또한 제안하였다. [그림 1]은 KorED알고리즘이 한글 단어의 편집거리를 구하는 수식을 보여주고 있다.

Base Case

$$SED(\Lambda, B(j)) = j \times \beta \; for \; |B| \geq j \geq 0$$

$$SED(A(\mathrm{i}), \Lambda) = i \times \beta \; for \; |A| \geq i \geq 0$$

General Case

$$\begin{aligned} SED(A(i), B(j)) &= min(SED(A(i), B(j-1)) + \beta, \\ &\quad SED(A(i-1), B(j)) + \beta, \\ &\quad SED(A(i-1), B(j-1)) + \delta_S(\mathrm{A[i]}, B[j])) \end{aligned}$$

[그림 2-1] KorED알고리즘

KorED알고리즘을 제안 이후 보다 시간이 지나면서 미비한 점이나 성능 향상의 목적으로 추가적인 연구가 진행되었다. 그 결과로 단순히 생각하면 ‘ㄱ’과 ‘ㄲ’의

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

편집거리는 'ㄱ'과 'ㅃ'의 편집거기보다 작다고 판단이 되기 때문에 음소의 종류에 따라 서로 연관성이 있는 음소들의 경우 KorED알고리즘에서 제시한 음소의 거리 단위보다 작은 거리를 가지고 있다고 판단하여 새로운 GrpSIM알고리즘을 제안한 논문을 작성하였다[22]. [그림 2-2]는 GrpSIM알고리즘이 한글 단어의 편집거리를 구하는 수식을 보여주고 있다.

Base Case

$$\mathrm{SIM}(\Lambda, B(j)) = -j \times I_S \; for \; |B| \geq j \geq 0$$

$$SIM(A(\mathrm{i}), \Lambda) = -i \times I_S \; for \; |A| \geq i \geq 0$$

General Case

$$\begin{aligned} SIM(A(i), B(j)) \\ = max(&SIM(A(i), B(j-1)) + I_S, \\ &SIM(A(i-1), B(j)) + I_S, \\ &SIM(A(i-1), B(j-1)) + \delta_S(\mathrm{A[i]}, B[j])) \end{aligned}$$

[그림 2-2]GrpSIM알고리즘

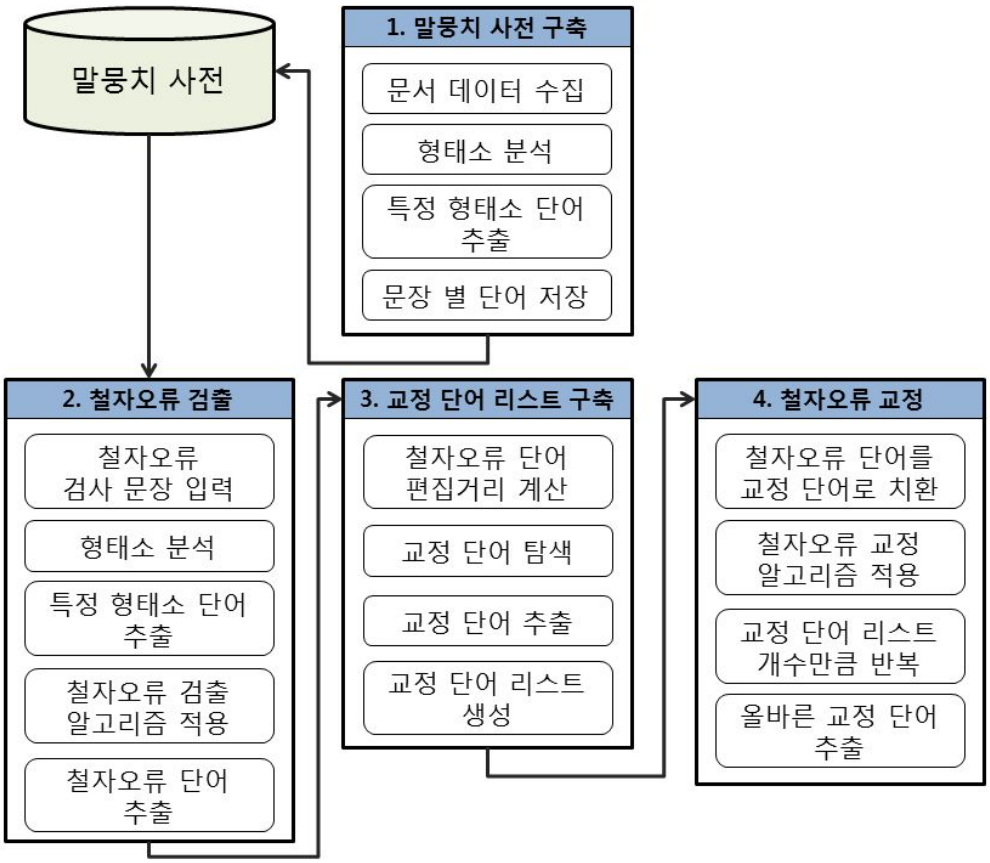
Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅲ. 코사인 유사도를 이용한 철자오류 교정

본 논문에서는 철자오류를 교정하기 위하여 코사인 유사도 알고리즘을 이용하여 문장에 존재하는 철자오류를 검출한 다음 한글 편집거리 알고리즘을 사용함으로서 교정 단어들을 추출하고 다시 코사인 유사도 알고리즘을 이용해 철자오류 교정을 수행한다. 본 장에서는 철자오류 교정을 수행하기 위한 순서와 사용한 알고리즘, 그리고 데이터에 대해서 자세히 서술한다.

A. 시스템 구성도



[그림 3-1] 시스템 구성도

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[그림 3-1]는 본 논문에서 제안하는 철자오류 교정 시스템의 구성도를 보여주고 있다. 철자오류를 교정하기에 앞서 통계적인 방식을 통해 철자오류를 교정하기 위해서는 철자오류가 포함되어 있지 않은 문서를 수집하여 말뭉치 사전을 구축할 필요가 존재한다. 그렇기 때문에 본격적인 연구를 수행하기에 앞서 말뭉치 사전을 구축하게 된다. 말뭉치 사전 구축 이후 철자오류를 교정하기 위해 일차적으로 문장 내에서 철자오류 존재 유무를 검사하며, 철자오류 단어가 존재할 시 철자오류 단어를 추출하게 된다. 본 연구에서는 철자오류 교정을 위해 철자오류 단어와 편집거리가 작은 단어들을 말뭉치 사전에서 추출하여 교정 단어 리스트를 제작하여 교정을 수행하게 된다. 구축한 교정 단어 리스트의 단어들을 입력 받은 문장의 철자오류 단어와 치환한 다음 철자오류 교정 알고리즘을 수행함으로서 치환한 교정 단어의 알맞음 여부를 검사하여, 가장 수치가 높은 단어를 문장에 삽입하는 것으로 철자오류 교정을 끝마치게 된다.

Collection @ chosun

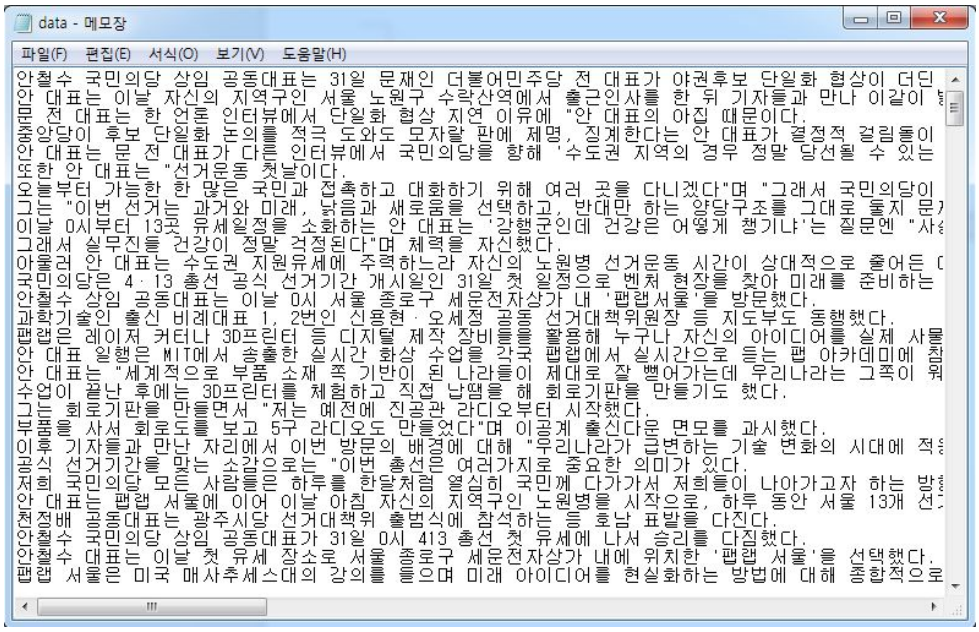
조선대학교
CHOSUN UNIVERSITY

B. 말뭉치 사전 구축

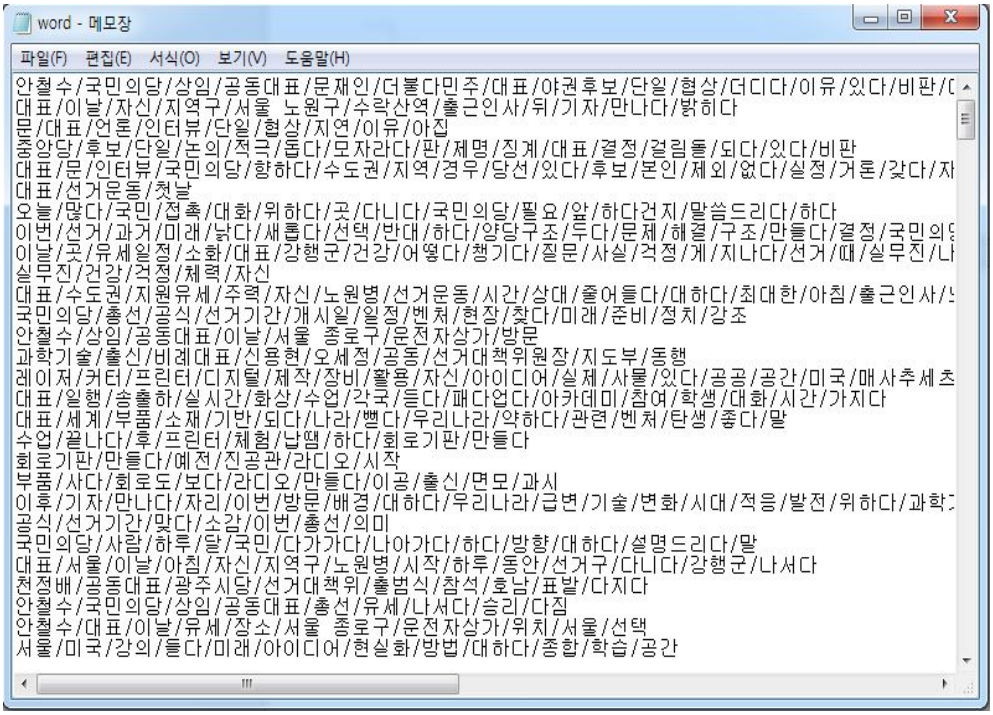
본 논문에서 제안하는 철자오류 교정 알고리즘은 통계적 기반의 철자오류 교정 방식이다. 통계적 기반의 철자오류 교정 방식은 단어들의 동시 등장 확률을 구하기 위해 말뭉치 사전을 구축할 필요가 있으며, 통계적인 철자오류 교정 방식의 교정 과정 중 특히 조심해야 될 과정이 바로 말뭉치 사전 구축 과정이다. 통계적인 방식의 철자오류 교정 모델은 철자오류 교정을 받고 있는 단어가 좌우 문맥에서 등장하는 단어들과의 동시 등장 확률을 계산함으로서 수행되는 형태의 교정 모델을 의미한다[24]. 그렇기 때문에 통계적인 방식의 철자오류 교정 모델을 사용하기 위해서는 단어의 동시 등장 확률을 구하기 위해 말뭉치 사전의 구축이 필수불가결한 문제이다. 하지만 말뭉치 사전의 구축이 철자오류 교정 과정 중 가장 중요한 이유는 말뭉치 사전을 구축하기 위해 수집한 문서에서 철자오류가 포함되어 있다면, 특정 단어의 철자오류 검사 시 철자오류 단어와 동시에 등장한다고 계산이 될 수 있기 때문에 말뭉치 사전을 구축할 때에는 되도록 철자오류가 포함되어 있지 않는 문서들 위주로 수집해야 한다. 그렇기 때문에 대다수의 연구에서는 말뭉치 사전을 구축하기 위해 교과서나 교재, 사전 그리고 뉴스 기사와 같은 문서들을 수집함으로서 말뭉치 사전을 구축하게 된다. 앞서 언급한 문서들 또한 철자오류가 존재하지 않는 완벽한 문서라고 할 수는 없지만, 해당 문서들에서 등장하는 철자오류는 독자들에게 잘못된 지식을 넘겨주거나 특정 기업이나 사람에 대한 명성 혹은 신뢰도에 큰 타격을 끼칠 수 있기 때문에 다른 문서에 비해 특히 철자오류 검사에 세심한 과정을 거치게 된다. 그로인해 앞서 언급한 문서들에서는 철자오류 발생률이 극히 희박하기 때문에 말뭉치 사전을 구축하기 위해서 자주 수집되는 문서의 종류 중 하나이다. 본 논문은 뉴스 기사들 중 네이버 뉴스에서 2016년 4월 총선의 기사들을 수집하여 말뭉치 사전을 구축하게 되었다. 이 때 수집한 뉴스 기사의 분량은 약 7000여개의 문장에 달하며, 또한 철자오류 검출과 교정 시 문장별로 동시 등장 빈도를 구하여 알고리즘에 적용하기 때문에 문장별로 분리하여 말뭉치 사전을 구축하였다. [그림 3-2]는 본 논문에서 말뭉치 사전을 구축하기 위해 수집한 네이버 뉴스 데이터의 일부를 보여주고 있으며, [그림 3-3]은 수집한 데이터에서 형태소들을 추출하여 문장별로 분류한 말뭉치 사전의 일부를 보여주고 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY



[그림 3-2] 수집한 뉴스 기사 데이터



[그림 3-3]구축한 말뭉치 사전 데이터

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

C. 철자오류 검출

본 장에서는 코사인 유사도를 이용해 철자오류 검출을 수행하는 과정에 대해 소개한다. 철자오류를 검출하기 위해서는 우선 문장에서 등장하는 모든 단어들을 추출하기 위해 형태소 분석을 할 필요가 있다. 품사가 추출된 단어들 중 철자오류 검출이 수행되는 품사는 동사, 형용사, 명사 같은 품사 단어들로, 이 단어들은 문장의 의미를 결정짓는데 높은 역할을 맡고 있으므로 중점적으로 철자오류 검사를 수행하게 된다. 부사, 접속사와 같은 품사 단어들은 주로 통계적인 방식이 아닌 규칙기반 방식을 통해 철자오류 교정이 수행되며, 모든 품사의 단어들을 통계적인 방식을 통해 교정을 수행하게 되면 시스템의 작동 시간이 기하급수적으로 늘어나는 단점이 존재하기 때문에 문장에 등장하는 모든 단어를 통계적 방식으로 철자오류를 검출하는 것은 현실성이 떨어진다.

1. 단어의 동시등장 빈도를 이용한 코사인 유사도 측정

본 절에서는 단어의 동시등장 빈도를 이용해 코사인 유사도를 계산함으로서 철자오류가 발생한 단어를 검출하는 방법에 대해 설명한다.

[표 3-1] 문맥의존 철자오류의 유형과 발생원인

A	어머니는 동생을 칭찬하셨다.

[표 3-1]은 단어의 동시등장 빈도를 사용하여 코사인 유사도를 구하는 방법을 설명하기 위해 제시한 예문이다. 그 중 문장 A에 등장하는 단어의 동시등장 빈도는 다음과 같이 계산된다. ‘어머니’라는 단어 t_1은 ‘동생’이라는 단어 t_2와 ‘칭찬이라는 단어 t_3와 한 문장 내에 동시에 등장하고 있다. 그렇기 때문에 t_1과 t_2가 동시에 발생하는 빈도 f_{12}와 단어 t_1과 단어 t_3가 동시에 등장하는 빈도 f_{13}은 각각 ’1‘이 된다는 것을 알 수 있다. 단어의 동시등장 빈도를 나타내는 f_{ij}는 절대빈도라고 하며, 이전 연구들에서는 절대빈도를 이용하여 철자오류를 검출하였다. 그렇기 때문에 이전 연구들에서는 철자오류 검출 방법은 등장 경향이 유사하더라도 절대빈

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

도의 영향을 많이 받기 때문에 다른 결과가 출력되었지만, 코사인 유사도를 이용하여 철자오류를 검출할 경우, 절대빈도의 영향이 받지 않기 때문에 코사인 유사도를 이용하여 철자오류를 검출하게 된다. 수식 (1)은 두 단어의 동시 출현 확률을 구하기 위해 사용한 코사인 유사도 수식을 보여주고 있다.

$$SIM(t_i,t_j) = \frac{\sum_{k=1(k \neq i,j)}^{n} f_{ik} \times f_{jk}}{\sqrt{\sum_{k=1(k \neq i,j)}^{n} f_{ik}^2} \times \sqrt{\sum_{k=1(k \neq i,j)}^{n} f_{jk}^2}} \quad (1)$$

수식 (1)은 문장에서 등장하는 i번째 단어 t_i와 j번째 단어 t_j를 쌍으로 묶어 두 단어의 동시 출현 확률을 구하는 수식을 보여주고 있다. f_{ik}는 단어 t_i와 단어 t_k의 동시 출현 빈도를 나타내며, k는 i나 j가 되지 않기 때문에 두 단어 t_i와 t_j의 동시 등장 빈도, 즉 절대빈도를 이용하지 않는다. 코사인 유사도는 0부터 1까지의 값을 가질 수 있으며, 유사도가 0에 가까울수록 두 객체는 서로 독립적 이라는 의미가 되고, 1에 가까울수록 유사하다고 할 수 있다. 그렇기 때문에 수식 (1)을 통해서 나온 두 단어의 동시 등장 확률이 0에 가까울수록 수식 (1)에 적용한 두 단어 중 한 단어가 철자오류가 발생한 단어라고 할 수 있다. 하지만 단순히 수식 (1)만 가지고는 수식에 적용한 두 단어 중 어느 단어가 철자오류가 발생한 단어인지 알 수 없다. 그렇기 때문에 특정 문장에서 등장하는 단어 t_1이 철자오류 단어인지 아니면 정상 단어인지 여부를 가리기 위해 단어 t_1을 제외한 문장에 등장하는 다른 단어들인 t_2부터 문장에 등장하는 마지막 단어 t_n까지의 코사인 유사도를 계산한 다음 평균값을 추출함으로서 철자오류를 검출하게 된다. [표 3-2]는 철자오류를 검출하는 알고리즘을 보여주고 있다.

[표 3-2] 철자오류 검출 알고리즘

$$AVR(t_i,t_j) = \frac{\sum_{j=1(j \neq i)}^{n} SIM(t_i,t_j)}{n-1}$$

$$\text{if } AVR(t_i,t_j) \leq Critical\ Value$$

$$t_i \equiv Spelling\ Error\ Word$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

2. 철자오류 검출 알고리즘을 사용한 검출 예시

단순히 말로만 설명할 뿐만이 아니라 본 논문에서 설명하는 철자오류 검출 알고리즘에 대한 이해를 높이기 위해 어떤 방식을 통하여 철자오류가 검출되는지에 대한 과정을 제대로 알 필요가 있다. 그렇기 때문에 보다 심도 깊은 이해를 돕기 위하여 위에서 설명한 철자오류 검출 알고리즘을 이용하여 특정 문장에서 철자오류를 검출하는 방법에 대해 예시를 이용하여 자세히 설명한다. 본 장에서 철자오류 검출에 대한 예시를 위해 사용할 문장은 [표 3-3]와 같다.

[표 3-3] 철자오류 검출 예시 문장

예시 문장	안철수 국민의당 사임(상임) 공동대표가 31일 0시 413 총선 첫 유세에 나서 승리를 다짐했다.

[표 3-3]에서 보여주고 있는 예시 문장은 철자오류 교정을 위해 수집한 데이터들 중 철자오류가 포함되어 있는 실제 문장을 가지고 온 것이다. 예시 문장에서 등장하는 '사임'이라는 단어는 본래 '상임'이라는 단어가 입력됐어야 했지만, 'ㅇ'이라는 음소가 입력되지 못하면서 '사임'이라는 철자오류가 발생하였다. 본 논문에서는 철자오류 교정을 위해 동사, 명사, 형용사와 같이 문장의 의미에 큰 영향을 주는 단어들만 추출하여 철자오류 교정을 수행하기로 하였으므로, 예시 문장에서 중요한 형태소 단어들만 추출하면 그 결과는 [표 3-4]과 같이 나오게 된다.

[표 3-4] 예시 문장의 추출 단어 리스트

추출 단어	안철수	국민의당	사임	공동대표	총선	유세	나서다	승리	다짐

수식 (1)을 적용하여 코사인 유사도를 계산하기 위해서는 말뭉치 사전에서 추출된 단어들에 대한 동시 등장 빈도를 구할 필요가 있다. [표 3-3]는 [표 3-4]에 등장하는 단어들이 말뭉치 사전에서 추출된 동시 등장 빈도를 나타내고 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 3-5] 단어들의 동시 출현 빈도

	안철수	국민의당	사임	공동대표	총선	유세	나서다	승리	다짐
안철수	**45**	33	1	20	41	26	9	13	8
국민의당	33	**117**	0	31	76	71	7	9	3
사임	1	0	**7**	0	0	0	0	1	0
공동대표	20	31	0	**96**	72	65	27	8	2
총선	41	76	0	72	**81**	53	1	5	7
유세	26	71	0	65	53	**77**	17	6	1
나서다	9	7	0	27	17	16	**32**	10	3
승리	13	9	1	8	5	6	10	**15**	4
다짐	8	3	0	2	7	1	3	4	**9**

[표 3-5]에서 굵게 표시된 숫자는 해당 단어가 문서에서 전체적으로 출현한 횟수를 의미하고 있다. 철자오류 단어인 '사임'의 단어 동시 출현 빈도만을 보더라도 상당히 낮다는 것을 알 수 있다. [표 3-5]에서 추출된 단어의 동시 출현 빈도를 이용하여 수식 (1)을 적용시키면 [표 3-6]과 같은 결과가 추출된다.

[표 3-6] 단어들의 동시 등장 빈도를 이용한 코사인 유사도

	안철수	국민의당	사임	공동대표	총선	유세	나서다	승리	다짐
안철수	**0.0**	0.96	0.20	0.93	0.94	0.92	0.84	0.83	0.88
국민의당	0.96	**0.0**	0.26	0.97	0.88	0.88	0.85	0.68	0.72
사임	0.20	0.26	**0.0**	0.18	0.26	0.20	0.34	0.59	0.69
공동대표	0.93	0.97	0.18	**0.0**	0.87	0.87	0.94	0.71	0.71
총선	0.94	0.88	0.26	0.87	**0.0**	0.99	0.86	0.84	0.65
유세	0.92	0.88	0.20	0.87	0.99	**0.0**	0.86	0.79	0.71
나서다	0.84	0.85	0.34	0.94	0.86	0.86	**0.0**	0.79	0.69
승리	0.83	0.68	0.59	0.71	0.84	0.79	0.79	**0.0**	0.86
다짐	0.88	0.72	0.69	0.71	0.65	0.71	0.71	0.86	**0.0**

수식 (1)을 이용한 코사인 유사도 추출의 목적은 서로 다른 한 쌍의 단어를 이용해 철자오류를 검출하는 것이 목적이므로 같은 단어끼리 묶어서 코사인 유사도를 추출할 필요가 없다. 그렇기 때문에 같은 단어들끼리 묶어서 코사인 유사도를 추출할 필요가 없기 때문에 같은 단어 쌍 끼리의 코사인 유사도는 '0.0'값이 나오게 된다. 그리고 철자오류를 검출하기 위해 각 단어들이 코사인 유사도 평균값을 추출하게 되면 [표 3-7]과 같은 결과가 나오게 된다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 3-7] 단어들의 코사인 유사도 평균

단어	코사인 유사도 평균
안철수	0.79
국민의당	0.75
사임	0.34
공동대표	0.75
총선	0.76
유세	0.76
나서다	0.75
승리	0.66
다짐	0.73

코사인 유사도 평균값 추출 결과, 철자오류 단어인 '사임'이 가장 낮게 나왔다. 하지만 평균값 수치가 가장 낮았다고 철자오류로 검출하게 되면, 철자오류가 아님에도 불구하고 문장에서 등장하는 단어 중 가장 낮은 수치를 기록했다는 이유 하나만으로 정상 단어가 철자오류라고 판단하여 추출될 경우가 생겨버린다. 그렇기 때문에 철자오류 검출에 대한 정확도와 재현율이 높은 임계값을 설정할 필요가 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

D. 교정 단어 리스트 구축

철자오류 검출이 성공적으로 수행이 완료 되었다면, 검출된 철자오류 단어를 본래 입력 단어로 교정해야할 필요가 존재한다. 본래 철자오류를 교정하는 기존 연구에서는 철자오류가 발생한 단어의 자소를 다른 자소로 치환한 이후 다시 좌우 문맥에 등장하는 단어들과의 동시 빈도를 구함으로서 철자오류 교정을 수행하게 된다[2, 6, 10]. 하지만 이전 연구에서 수행된 교정 단어 탐색 방식은 알고리즘 수행 시간을 크게 증가된다는 단점이 존재한다. 6개의 자소로 이루어진 철자오류 단어를 교정하기 위해서는 원래 입력되어있는 자소를 대치가 가능한 다른 자소로 치환한 다음 좌우 문맥에 등장하는 단어들과의 동시 등장 확률을 계산한다고 생각할 경우 처리 시간이 길다는 것은 금방 알 수 있게 한다. 하지만 단순히 생각하면 사전에 존재하는 단어들 중 철자오류가 발생한 단어와 편집거리가 작은 단어들만 추출하여 좌우문맥 단어들과 동시 등장 확률을 구하게 되면 철자오류 교정이 걸리는 처리시간이 줄어들겠다는 것을 알 수 있다. 그렇기 때문에 본 논문에서는 한글 편집거리 알고리즘을 사용하여 철자오류 단어와 편집거리가 작은 단어들을 말뭉치 사전에서 추출함으로서 철자오류 교정을 수행하게 된다.

1. 한글 편집거리 알고리즘

기존의 편집거리 알고리즘에 대한 연구는 영어권 언어를 중심으로 활발히 연구되었으나, 한글과 같은 복잡한 형태의 언어에 대한 연구는 미흡했다. 또한 기존의 영어권 언어를 대상으로 한 편집거리 알고리즘은 한글에 적용시키기도 어려웠다. 그래서 한 연구자는 한글의 편집거리를 구하기 위한 알고리즘을 연구하여 제안하였다[21, 22, 23]. 이 연구자가 제안한 한글 편집거리 알고리즘은 한글의 편집거리 기준을 음절 단위로 잡은 SylED알고리즘, 그리고 한글의 편집거리 기준을 음소 단위로 잡은 PhoED알고리즘, 그리고 한글의 편집거리를 음절과 음소 모두 포함한 KorED알고리즘 있으며, 마지막으로 KorED알고리즘에서 대치되는 자음이나 모음의 종류에 따라 편집거리를 다르게 잡은 GrpSIM알고리즘이 존재한다. 한글 편집거리 알고리즘을 이용하여 철자오류 교정 단어를 추출하기 위해서는 실제로 철자오류 단어와 정상 단어의 편집거리에 대한 정확한 정보를 통해 추출할 필요가 존재한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

그렇기 때문에 본 장에서는 철자오류 단어와 정상 단어의 편집거리에 대한 통계를 분석하여 철자오류 교정을 위해 한글 편집거리 알고리즘의 정확한 사용 방법을 제시하고 교정 단어 리스트를 구축하는 방법에 대해 소개한다.

2. 철자오류 단어와 정상 단어의 편집거리 추출 방법

본 장에서는 철자오류 단어와 정상 단어의 편집거리를 구하기 위해 사용한 알고리즘과 방법에 대해 소개한다. 철자오류 단어와 정상 단어의 편집거리 통계는 한글의 편집거리 단위를 음절 단위를 통하여 편집거리를 구하는 SylED알고리즘과 편집거리의 단위를 음소 단위를 통하여 편집거리를 구하는 PhoED알고리즘을 이용하여 구하게 된다. 철자오류가 발생한 이유는 단어를 구성하는 음소의 일부가 변경되어서 발생한 경우기 일반적이기 때문에 철자오류 단어와 정상 단어의 편집거리에 대한 통계를 추출할 경우 음절에 대해 고려할 경우는 적다보니 음절 단위와 음소 단위를 모두 고려하여 편집거리를 구하는 KorED알고리즘을 통한 편집거리 통계는 구하지 않았다. 또한 철자오류 단어와 정상 단어의 편집거리 통계를 통해 적당한 임계값을 구하여 사전에서 임계값에 해당하는 단어들을 추출하여 교정 단어 리스트를 구축하는 것이 목적이기 때문에 자음이나 모음의 종류에 따라 편집거리를 다르게 잡는 GrpSIM알고리즘 또한 고려하지 않았다. 하지만 실제로 철자오류가 발생한 단어가 정상 단어에 대한 음절 편집거리에 대한 자료가 필요하기 때문에 음절 단위로 편집거리를 구하는 SylED알고리즘을 이용하여 편집거리 통계를 구하게 되었다. [표 3-8]은 SylED알고리즘과 PhoED알고리즘을 이용해 편집거리를 구하는 방법을 설명하기 위한 예시 단어들이다.

[표 3-8] 편집거리 알고리즘 예시 단어

A	과 학
B	화 학
C	수 학

SylED알고리즘을 이용해 한글의 편집거리를 구하게 되면 문자열의 같은 순서의 음절의 동일 유무에 따라 편집거리가 증가하게 된다. 단어 A와 단어 B의 편집거리

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

를 SylED알고리즘을 이용해 구하게 된다면, 두 단어의 두 번째 음절인 '학'은 두 단어 모두 동일하므로 두 번째 음절로 인해 편집거리가 증가하게 되지는 않는다. 하지만 단어 A의 첫 번째 음절인 '과'와 단어 B의 첫 번째 음절은 '화'는 동일하지 않으므로 편집거리가 '1'증가하게 된다. 그렇기 때문에 단어 A와 단어 B의 최종적인 편집거리는 '1'로 계산된다. 또한 단어 A와 단어 C 혹은 단어 B와 단어 C의 편집거리 또한 앞서 설명한 것과 똑같은 형태로 편집거리가 추출되므로 세 개의 예사 단어들의 편집거리를 '1'로 계산된다. 하지만 PhoED알고리즘을 이용하여 편집거리를 구할 경우 SylED알고리즘을 통해 구한 편집거리와는 다른 값이 나오게 된다. 단어 A와 단어 B의 편집거리를 PhoED알고리즘을 이용해 구할 경우 SylED알고리즘과 마찬가지로 두 단어의 두 번째 음절은 '학'은 동일하기 때문에 두 번째 음절에서 편집거리는 발생하지 않는다. 단어 A의 첫 번째 음절인 '과'와 단어 B의 첫 번째 음절인 '화'가 동일하지 않기 때문에 첫 번째 음절에 의해 편집거리가 발생하게 되는데, 단어 A의 첫 번째 음절 '과'는 초성인 'ㄱ'과 중성인 'ㅘ', 그리고 종성인 공백으로 분리될 수 있으며, 단어 B의 첫 번째 음절 '화'는 초성인 'ㅎ'과 중성인 'ㅘ', 그리고 종성인 공백으로 분리 할 수 있다. 이 중 단어 A의 첫 번째 음절의 초성인 'ㄱ'이 단어 B의 첫 번째 음절의 초성인 'ㅎ'과 다르므로 편집거리가 '1' 증가하게 된다. 그 이외의 다른 음소는 차이가 없으므로 단어 A와 단어 B의 최종 편집거리는 '1'이 된다. 단어 A와 단어 C의 편집거리를 PhoED알고리즘을 이용해 구할 경우 단어 A와 단어 B의 편집거리와 마찬가지로 단어 A와 단어 C의 두 번째 음절인 '학'은 동일하여 편집거리가 발생하지 않는다. 하지만 단어 A의 첫 번째 음절의 초성인 'ㄱ'과 중성인 'ㅘ'가 단어 C의 첫 번째 음절의 초성인 'ㅅ'과 중성인 'ㅜ'과 다르기 때문에 각각 편집거리가 '1'씩 증가하면서 단어 A와 단어 C의 최종 편집거리는 '2'가 된다. 예시 단어들의 모든 편집거리는 [표 3-9]와 같다.

[표 3-9] 예시 단어들의 편집거리

알고리즘	SylED			PhoED		
단어	A	B	C	A	B	C
A	0	1	1	0	1	2
B	1	0	1	1	0	2
C	1	1	0	2	2	0

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

철자오류 단어와 정상 단어와의 편집거리에 대한 통계를 추출하기 위해 키보드 입력 데이터를 수집하는 KeyLogger프로그램[26]과 온라인 또는 작성이 완료된 문서에서 철자오류 데이터들을 수집하였다. 수집한 철자오류 데이터는 총 227개의 단어들이 수집되었으며, [표 3-10]은 수집한 철자오류 단어의 일부를 보여주고 있다.

[표 3-10] 철자오류가 발생한 단어 리스트

	철자오류 발생 단어	정상 입력 단어
1	갖이	같이
2	곡이	고기
3	울다	놀다
4	마니	아니
5	쏟아니다	쏟아지다
6	보르다	모르다
7	니미	이미
8	멍망	엉망
9	티보드	키보드
10	련구	연구
11	엉앙	엉망
12	말머라	말머리
13	분다	준다
14	얼미	멀미
15	맹면	냉면

또한 수집한 철자오류 단어들을 이용해 차례대로 편집거리를 추출하였다. 먼저 SylED알고리즘을 이용해 추출한 철자오류 편집거리는 [표 3-11]과 같다.

[표 3-11] SylED알고리즘 편집거리 통계

편집거리	개수	퍼센트
1	203	89.4%
2	24	10.6%
합	227	100%

수집한 데이터 중 1개의 음절에서 철자오류가 발생한 경우는 전체 데이터 중 약

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

90%에 해당하였으며, 2개의 음절에 철자오류가 발생한 경우는 약 10%의 비율을 차지하였다. 수집한 데이터 중 3음절 이상의 길이를 가진 단어의 개수는 절반 이상을 차지하였으나 3음절 이상 철자오류가 발생한 경우는 존재하지 않았다. 그리고 PhoED알고리즘을 이용해 추출한 철자오류 편집거리는 [표 3-12]와 같다.

[표 3-12] PhoED알고리즘 편집거리 통계

편집거리	개수	퍼센트
1	201	88.5%
2	26	11.5%
합	227	100%

PhoED알고리즘을 이용해 철자오류 단어들의 편집거리를 구하게 되면 SylED알고리즘을 구한 것과 비슷한 결과가 나오게 된다. 다만 편집거리가 2인 경우가 SylED 알고리즘에 비해 2개 더 나오게 됐는데, 이 경우는 하나의 음절에 2개의 음소에서 철자오류가 발생하였기 때문에 이런 결과가 나오게 되었다. 이 결과를 통해서 철자오류가 발생할 경우 한 음절의 하나의 음소가 치환, 추가, 혹은 삭제되기 때문이라는 것을 알 수가 있다. 철자오류가 발생한 음소가 2회인 경우고 있다는 것은 교정단어를 사전에서 추출할 경우 단순히 편집거리가 1인 단어만을 추출할 경우 옳은 교정단어를 추출하지 못할 수 있다는 것을 알 수 있다. 그렇기 때문에 사전에서 올바른 교정 단어들을 추출하기 위해서 단어의 문자열 길이에 따른 철자오류 편집거리에 대한 통계를 제시함으로서 교정 정확도와 재현율을 높일 수 있도록 해야 한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

E. 철자오류 교정

한글 편집거리 알고리즘을 통해 철자오류가 발생한 단어와 편집거리가 적은 교정 단어들을 사전에서 추출하여 리스트를 구축하였다면 리스트에 포함되어 있는 단어들을 통해 철자오류 교정을 수행하여야 한다. 본 장에서는 [표 3-3]의 문장을 예시로 이용하여 철자오류 교정을 수행하는 것을 설명한다. [표 3-2]에서 등장하는 철자오류 단어는 '사임'이라는 단어로, 본래 입력해야 될 정상 단어는 '상임'이다. 실제로 '사임'이라는 단어를 이용해 사전에서 편집거리가 '1'인 단어들을 추출하게 되면 [표 3-13]과 같은 단어들이 추출하게 된다.

[표 3-13] 추출된 '사임'의 교정 단어 리스트

편집거리	철자오류 단어	정상 단어	추출 교정 단어
1	사임	상임	상임
			사이
			짜임

또한 세 단어의 말뭉치 사전에서 등장 빈도는 [표 3-12]와 같다.

[표 3-14] 교정 단어의 좌우문맥 단어 동시 등장 빈도

	안철수	국민의당	-	공동대표	총선	유세	나서다	승리	다짐
상임	32	15	51	43	17	22	19	7	2
사이	5	4	7	4	0	2	0	0	0
짜임	0	0	1	0	0	0	0	0	0

[표 3-13]에서 '-'으로 표시한 공간은 '상임', '사이', '짜임'의 세 단어가 들어가는 공간이다. [표 3-13]에 등장하는 세 개의 교정 단어들 중 '짜임'이라는 단어는 말뭉치 사전에서 한번 등장하기는 하지만 [표 3-3]의 좌우문맥 단어들과는 동시에 등장한 경우가 없기 때문에 올바른 교정 단어에서 제외된다. 그렇기 때문에 '상임'과 '사이' 두 단어 중 올바른 교정 단어를 선택하게 된다. 두 단어 중 한 단어를 선택하는 방법은 철자오류 검출에 사용하였던 수식 (1)을 사용하여 올바른 교정 단

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

어를 선택하게 된다. 하지만 철자오류를 검출하기 위해서 수식 (1)을 사용할 경우 수식 (1)의 계산 결과가 임계값 이하이며 또한 가장 낮은 수치를 기록한 단어를 철자오류 단어로 판단하여 검출하게 되었다. 하지만 철자오류 교정은 반대로 가장 높은 수치를 기록한 단어라 올바른 교정 단어라고 판단하여 교정하게 된다. 두 단어의 코사인 유사도를 추출하면 [표 3-15]과 같은 결과가 나오게 된다.

[표 3-15] 교정 단어의 코사인 유사도 추출

	안철수	국민의당	-	공동대표	총선	유세	나서다	승리	다짐
상임	0.79	0.75	0.0	0.82	0.89	0.85	0.93	0.91	0.76
사이	0.68	0.56	0.0	0.49	0.92	0.78	0.70	0.84	0.64

또한 두 단어의 유사도 평균은 [표 3-16]과 같다.

[표 3-16] 교정 단어들의 코사인 유사도 평균

단어	코사인 유사도 평균
상임	0.84
사이	0.70

두 단어 중 가장 높은 코사인 유사도 평균을 추출한 단어는 '상임'이다. 실제로 [표 3-3]에서 등장하는 철자오류 단어 '사임'은 '상임'이라는 철자오류 단어이므로 '상임'이라는 단어를 '사임'과 치환하면 성공적으로 철자오류를 교정했다고 할 수 있다. 위와 같은 내용을 통해 철자오류 교정에 대한 알고리즘을 지정하면 수식 (3)과 같다.

[표 3-17] 철자오류 교정 알고리즘

$$AVR(t_i, t_j) = \frac{\sum_{j=1(j \neq i)}^{n} SIM(t_i, t_j)}{n-1}$$

$$\text{if } AVR(t_i, t_j) \equiv Max\,Value$$

$$t_i \equiv True\ Correction\ Word$$

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

Ⅳ. 실험 및 결과

본 장에서는 철자오류 교정에 대한 연구를 수행하기 위해 첫 번째로 철자오류 데이터를 수집한 방법과 두 번째로는 철자오류를 검출하기 위해서 가장 정확도와 재현율이 높은 코사인 유사도 임계값을 추출한 과정에 대해서 소개한다. 또한 추출한 코사인 유사도 임계값을 통하여 검출하게 된 철자오류 단어를 이용하여 교정 단어 리스트를 구축하기 위해 수행한 방법과 더불어 마지막으로 철자오류 교정을 수행함으로서 최종적으로 본 논문에서 제안한 철자오류 교정 시스템에 대한 성능이 어떤지 평가하게 된다.

A. 철자오류 데이터

철자오류 교정에 대한 연구를 수행하기 위해서는 철자오류가 포함되어 있는 데이터가 필요하다. 본 논문에서는 자연스럽게 철자오류가 포함되어 있는 문장을 수집하기 위하여 온라인상에 존재하는 총선과 관련된 기존 문서 및 뉴스 댓글 혹은 SNS와 같은 웹 문서들을 통하여 철자오류가 포함되어 있는 문장들을 수집하게 되었다. 또한 보다 자세한 결과를 추출하기 위해 문장에 포함된 어절 중 하나의 어절에서만 철자오류가 발생한 문장들만 수집하게 되었다. 하지만 철자오류가 자연스럽게 포함되어 있는 문장은 수집한 문서에서 전체적인 분량의 약 0.005%의 수준 밖에 되지 않을 정도로 철자오류 데이터는 수집하기 굉장히 어려운 데이터이다[27]. 그렇기 때문에 실제 연구에서 사용된 철자오류 데이터는 100개 정도의 적은 숫자의 데이터를 이용하여 연구를 수행하게 되었다. 철자오류가 발생한 데이터를 수집할 수 있는 SNS는 여러 종류가 있지만 본 논문에서는 그 중 트위터에서 데이터를 수집하게 되었다. 트위터에서 데이터를 수집하기 위해서는 일단 트위터의 데이터를 수집할 수 있는 Open API를 이용해야 하며, 또한 트위터에서 데이터 수집을 하겠다는 신청서를 작성하고 Access_token을 발급받아야만 데이터를 수집할 수 있다. 트위터에서 데이터를 수집하는 방법과 순서는 [표 4-1]과 같다. 또한 [표 4-2]는 트위터에서 데이터를 수집하기 위해 작성한 자바 소스코드의 일부를 보여주고 있다. 그리고 [그림 4-1]은 트위터의 철자오류 데이터를 수집하기 위해 작성한 자바 소스를 이용하여 수집한 철자오류 데이터의 일부를 보여주고 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 4-1] 트위터 데이터 수집 방법 및 순서

순서	내용
1	트위터 개발자 모드에서 제공하는 Open API를 다운 받은 이후 트위터에서 데이터를 받기 위한 자바 소스에 삽입함으로서 트위터 데이터를 수집하기 위한 코드를 완성한다.
2	트위터 API를 사용하기 위해서는 트위터의 승인이 필요하므로, 트위터 개발자 페이지에서 OAuth 인증절차를 거쳐 트위터에 접속할 수 있는 consumer_token, consumer_secret, access_token, access_secret 값을 발급받는다.
3	수집할 트위터 데이터는 말뭉치 사전을 구축하는데 수집한 문서 데이터인 '2016년 총선'과 연관되어야 하므로, 트위터가 작성된 일자를 총선이 열렸던 기간과 내용에 해당되는 데이터들을 수집하여 텍스트 파일을 생성해 저장하게 된다.

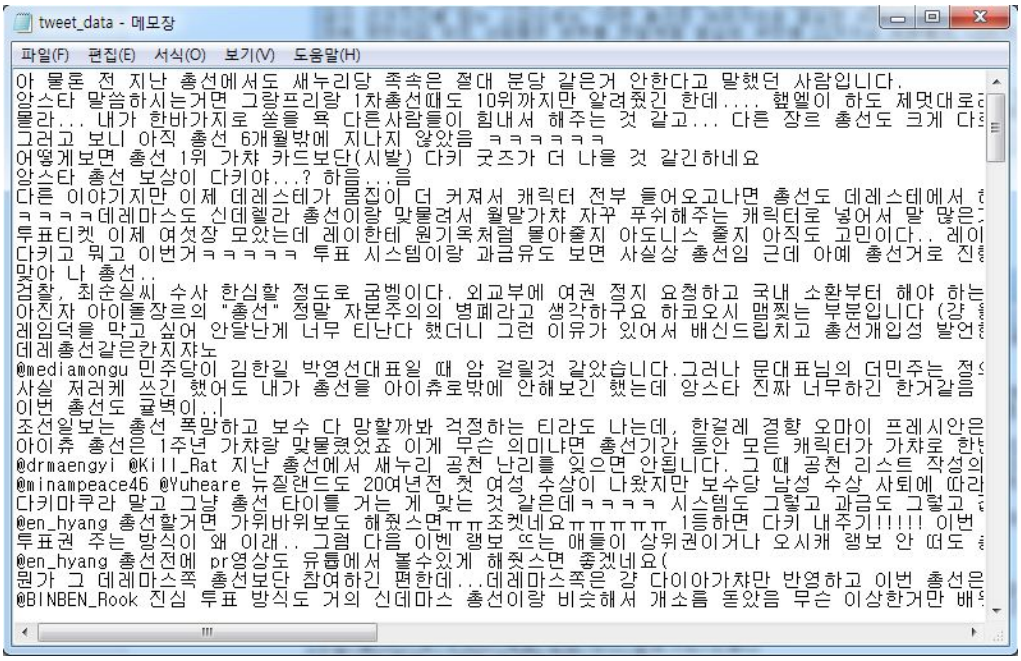
[표 4-2] 트위터 데이터 수집을 위한 자바 소스 코드

내용

```
public class MainClass {

String CONSUMER_KEY = "ZgHS95KDrE8cxL0QtAbnIvq6u";
String CONSUMER_SECRET = "1G5YACGYewzBKUeSL90W3rDYx6RSMMyrPTfQ1HlhaoG8dcr";

String ACCESS_TOKEN = "3247542852-30mcB33k40bgBgjvk8ipfCfW7eHhho7NphhxKV9";
String ACCESS_SECRET = "wKLxRZDe54R4auEICxwzdC44fZeHx2M3Z6UoYPVLFUNpd";

        static Twitter twitter;
        static AccessToken accessToken;

        static Scanner scan = new Scanner(System.in);

        public static void main(String[] args) {
                twitter = new TwitterFactory().getInstance();
                twitter.setOAuthConsumer(CONSUMER_KEY, CONSUMER_SECRET);
                accessToken = new AccessToken(ACCESS_TOKEN, ACCESS_SECRET);
                twitter.setOAuthAccessToken(accessToken);

                searchTweet();
        }
```


Collection @ chosun

조선대학교
CHOSUN UNIVERSITY


tweet_data - 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

아 물론 전 지난 총선에서도 새누리당 족속은 절대 분당 갈은거 안한다고 말했던 사람입니다.
앙스타 말씀하시는거면 그랑프리랑 1차총선때도 10위까지만 알려줬긴 한데.... 햅엘이 하도 제멋대로
몰라... 내가 한바가지로 쏟을 욕 다른사람들이 힘내서 해주는 것 같고... 다른 장르 총선도 크게 다
그러고 보니 아직 총선 6개월밖에 지나지 않았음 ㅋㅋㅋㅋㅋㅋ
어떻게보면 총선 1위 가챠 카드보단(시발) 다키 굿즈가 더 나올 것 같긴하네요
앙스타 총선 보상이 다키야...? 하음...음
다른 이야기지만 이제 데레스테가 몸집이 더 커져서 캐릭터 전부 들어오고나면 총선도 데레스테에서
ㅋㅋㅋㅋ데레마스도 신데렐라 총선이랑 맞물려서 월말가챠 자꾸 푸쉬해주는 캐릭터로 넣어서 말 많은
투표티켓 이제 여섯장 모았는데 레이한테 원기옥처럼 몰아줄지 아도니스 줄지 아직도 고민이다.. 레이
다키고 뭐고 이번거ㅋㅋㅋㅋㅋ 투표 시스템이랑 과금유도 보면 사실상 총선임 근데 아예 총선거로 진
맞아 나 총선..
검찰, 최순실씨 수사 한심할 정도로 굼벵이다. 외교부에 여권 정지 요청하고 국내 소환부터 해야 하는
아진자 아이돌장르의 "총선" 정말 자본주의의 병폐라고 생각하구요 하코오시 맴찢는 부분입니다 (강
레임덕을 막고 싶어 안달난게 너무 티난다 했더니 그런 이유가 있어서 배신드립치고 총선개입성 발언
데레총선같은칸지쟈노
@mediamongu 민주당이 김한길 박영선대표일 때 암 걸릴것 같았습니다.그러나 문대표님의 더민주는 정
사실 저러케 쓰긴 했어도 내가 총선을 아이츄로밖에 안해보긴 했는데 앙스타 진짜 너무하긴 한거같음
이번 총선도 글벽이..|
조선일보는 총선 폭망하고 보수 다 망할까봐 걱정하는 티라도 나는데, 한겔레 경향 오마이 프레시안은
아이츄 총선은 1주년 가챠랑 맞물렸었죠 이게 무슨 의미냐면 총선기간 동안 모든 캐릭터가 가챠로 한
@drmaengyi @Kill_Rat 지난 총선에서 새누리 공천 난리를 잊으면 안됩니다. 그 때 공천 리스트 작성의
@minampeace46 @Yuheare 뉴질랜드도 20여년전 첫 여성 수상이 나왔지만 보수당 남성 수상 사퇴에 따라
다키마쿠라 말고 그냥 총선 타이틀 거는 게 맞는 것 같은데ㅋㅋㅋㅋ 시스템도 그렇고 과금도 그렇고
@en_hyang 총선할거면 가위바위보도 해줬스면ㅠㅠ조켓네요ㅠㅠㅠㅠㅠ 1등하면 다키 내주기!!!!! 이번
투표권 주는 방식이 왜 이래.. 그럼 다음 이벤 랭보 뜨는 애들이 상위권이거나 오시캐 랭보 안 떠도
@en_hyang 총선전에 pr영상도 유튜에서 볼수있게 해줬스면 좋겠네요(
뭔가 그 데레마스쪽 총선보단 참여하긴 편한데...데레마스쪽은 걍 다이아가챠만 반영하고 이번 총선은
@BINBEN_Rook 진심 투표 방식도 거의 신데마스 총선이랑 비슷해서 개소름 돋았음 무슨 이상한거만 배

[그림 4-1] 수집한 트위터 데이터

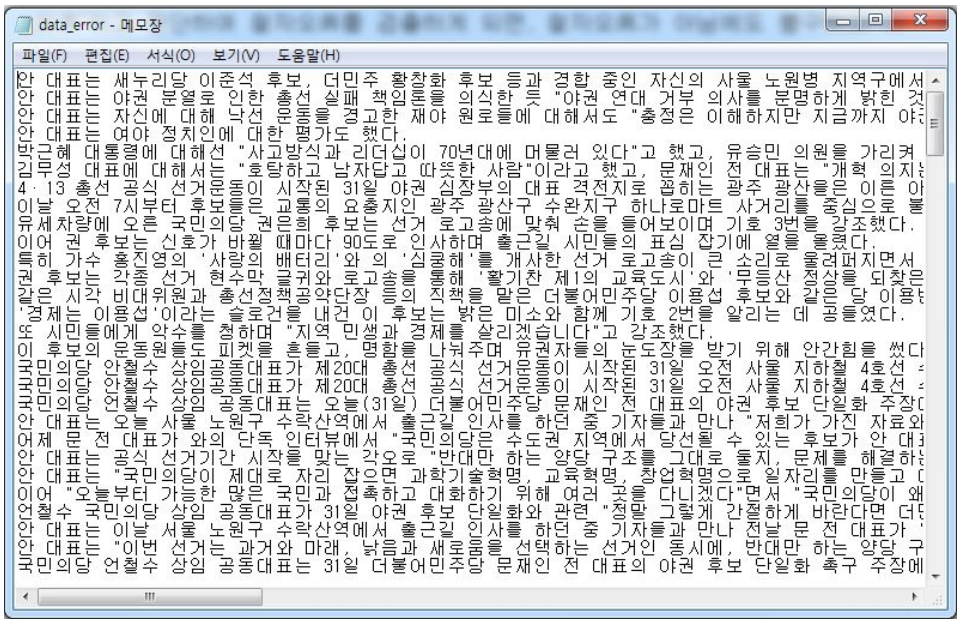
수집이 완료된 트위터 데이터는 SNS라는 플랫폼의 특성상 작성자가 일부러 글을 손상시키는 통신체나 감정을 표시하는 이모티콘에 의해서 문장에 대한 형태소 분석이 어렵다. 그렇기 때문에 수집한 트위터 데이터는 곧바로 철자오류 데이터로 사용하는 것이 아닌, 심각한 통신체를 수정하고 이모티콘은 삭제를 함으로서 철자오류 데이터로 사용하게 된다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

B. 철자오류 검출

철자오류 교정에 사용할 수 있는 철자오류 데이터 수집이 완료되었다면 철자오류 교정을 위해 우선적으로 철자오류 검출 과정에 들어서게 된다. 철자오류 검출은 앞서 설명한 수식 (2)를 통하여 철자오류를 검출하게 되지만, 철자오류를 검출하기 위해서는 코사인 유사도 평균의 임계값을 설정할 필요가 존재한다. 임계값을 지정하지 않은 채 단순히 '가장 낮은 수치의 코사인 유사도 평균을 기록한 단어가 철자오류'라고 판단하여 철자오류를 검출하게 되면, 철자오류가 아님에도 불구하고 가장 낮은 수치의 코사인 유사도 평균을 기록하였단 이유 하나만으로 철자오류가 검출 될 가능성이 존재한다. 그렇기 때문에 본 절에서는 검출율이 가장 높은 코사인 유사도 평균의 임계값을 찾게 된다. 철자오류 검출에 사용하게 될 문서 데이터는 총 200개의 문장으로 이루어져 있으며 그 중에 100개의 문장은 철자오류로 이루어 져 있다. [그림 4-2]는 철자오류 검출에 사용한 문서 데이터의 일부를 보여주고 있다.



[그림 4-2] 철자오류 데이터

코사인 유사도 평균의 임계값은 교정 시스템의 성능을 평가하는데 주로 사용되

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

는 정확도(Precision)와 재현율(Recall)로 추출한다[28]. 하지만 각각 하나의 요소만으로 임계값을 추출하기엔 정확도와 재현율 모두 중요한 평가요소이므로 정확도와 재현율을 모두 반영한 평가방법인 F-Measure로 구해보도록 한다. 정확도와 재현율, 그리고 F-Measure을 구하는 수식은 각각 (2), (3), (4)와 같다.

$$\mathrm{Precision}(\text{정확도}) = \frac{\text{바르게 교정한 수}}{\text{교정한 데이터 수}} \tag{2}$$

$$Recall(\text{재현율}) = \frac{\text{바르게 교정한 수}}{\text{오류 데이터 수}} \tag{3}$$

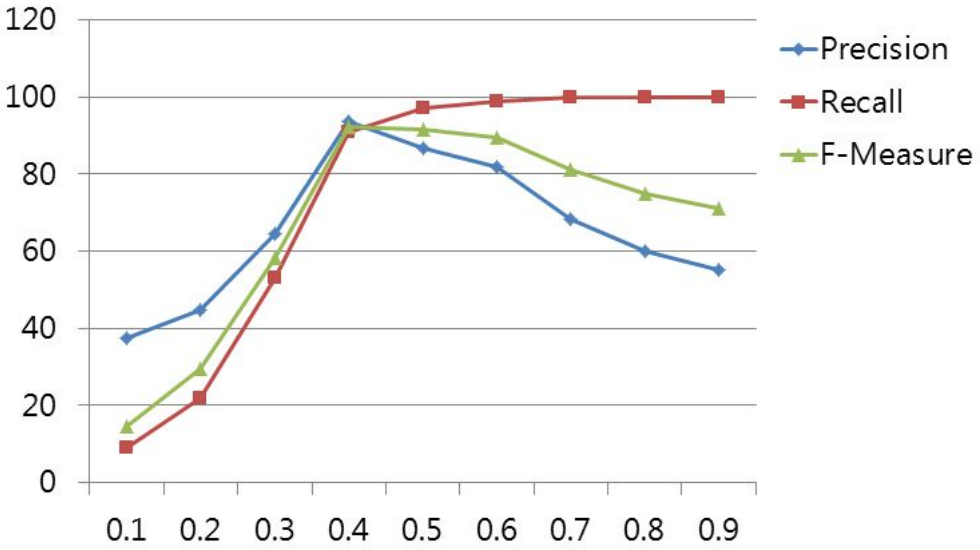
$$F-Measure = 2 \cdot \frac{\mathrm{Precision} \cdot Recall}{\mathrm{Precision} + Recall} \tag{4}$$

정확도와 재현율, 그리고 F-Measure를 구하는 방법은 다음과 같다. 철자오류 교정 연구를 위한 철자오류 데이터에는 총 200개의 문장으로 구성되어 있으며 그 중 100개의 문장이 철자오류가 포함되어 있는 문장이라고 하였다. 이 때 철자오류 검출 알고리즘을 적용함으로써 130개의 문장에 철자오류가 포함되어 있다고 판단하였으며, 그 중 80개의 문장이 실제로 철자오류가 포함되어 있는 문장이었다고 하자. 그러면 정확도는 철자오류를 검출한 130개의 문장에서 80개를 올바르게 교정하였으므로 '61.5%'의 값이 나오게 된다. 재현율의 경우 100개의 철자오류 중 실제로 80개의 철자오류가 포함되어 있는 문장을 검출하였으므로 '80%'의 값을 가지게 된다. 이렇게 출력된 정확도와 재현율을 수식 (4)에 적용시키면 F-Measure의 값은 '69.5%'가 나오게 된다. 철자오류를 검출하는 알고리즘은 알고리즘 마다 각기 시스템의 차이로 인해 성능이나 기능이 제각각이며 어떤 알고리즘은 정확도가 높으나 재현율이 낮을 수 있으며 또 다른 알고리즘은 정확도는 낮지만 반대로 재현율이 높을 수 있다. 이렇게 다른 두 알고리즘의 성능을 비교하려면 정확도와 재현율을 동시에 고려하여 값을 매기는 F-Measure이 존재하며, 본 논문에서 철자오류 검출을 위해 사용하는 알고리즘의 최적의 임계값을 추출하기 위하여 F-Measure을 이용하게 된다. 코사인 유사도는 0부터 1까지의 수치를 가질 수 있으며, 최적의 임계값을 추출하기 위해 임계값을 0부터 1까지 각각 0.1단위로 나누어 각 값에 따른 F-Measure값을 기록하게 된다. 가장 높은 값을 발견하게 되면 다시 그 기록을 중심으로 0.01단위로 탐색하여 결과적으로 F-Measure가 높게 나

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

온 임계값을 추출하게 된다. [그림 4-3]과 [표 4-3]은 각 코사인 유사도 임계값 설정에 따른 철자오류 검출의 정확도와 재현율, 그리고 F-Measure을 보여주고 있다.



[그림 4-3] 코사인 유사도 임계값에 따른 통계 그래프

[표 4-3] 코사인 유사도 임계값에 따른 통계 표

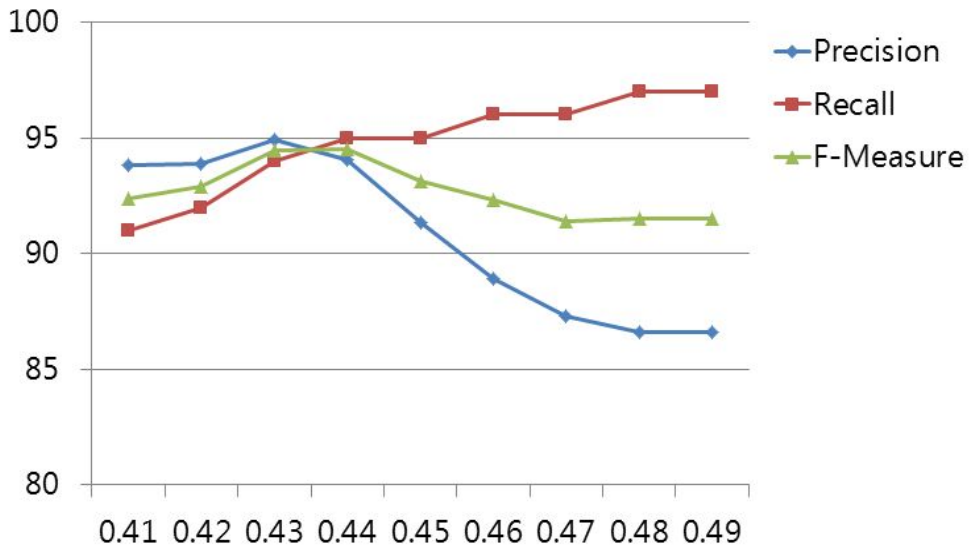
	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Precision	37.5	44.89	64.63	93.81	86.6	81.81	68.49	59.88	55.24
Recall	9	22	53	91	97	99	100	100	100
F-Measure	14.52	29.53	58.24	92.38	91.5	89.59	81.3	74.9	71.17

[그림 4-3]과 [표 4-3]을 보면 알 수 있듯이, 정확도는 계속 높아지다가 어느 시점으로 하여 값이 떨어지며, 재현율은 계속 올라간다는 것을 알 수 있다. 그 이유는 철자오류를 검출하기 위한 코사인 유사도의 임계값이 높아질수록 많은 수의 단어들을 철자오류라 판단하여 검출하게 되지만, 검출 수가 평가 데이터에 포함되어 있는 단어의 개수를 넘어서게 되면 그 기점으로 정확도가 떨어지게 된다. 다만 재현율의 경우 계속 증가하게 되는데, 그 이유는 임계값이 높아질수록 단어의 검출

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

량이 늘어나게 되고, 그로인해 검출된 단어에서 실제로 철자오류가 포함되어 있는 비율이 지속적으로 증가하게 되기 때문이다. 앞과 같은 사실로 인해 철자오류 단어의 검출을 위한 코사인 유사도의 임계값을 추출하기 위해서는 정확도와 재현율이 전부 관여하는 F-Measure을 이용하는 것이 옳다. 코사인 유사도의 임계값을 '0.1'로 분리하여 추출한 결과에서 철자오류의 검출율이 가장 높게 출력된 코사인 유사도의 임계값은 '0.4'이며 그 다음으로 가장 높게 출력된 코사인 유사도의 임계값은 '0.5'이다. 이로 인해 가장 높은 철자오류의 검출율은 '0.4'에서 '0.5'사이의 코사인 유사도에서 나올 확률이 존재한다. 그렇기 때문에 '0.4'에서 '0.5'사이의 코사인 유사도에서 철자오류 검출율을 확인하기 위해 [그림 4-4]와 [표 4-4]같이 코사인 유사도의 임계값을 '0.01'단위로 분리하여 철자오류 검출율을 확인해 보았다.



[그림 4-4] 코사인 유사도 임계값에 따른 통계 상세 그래프

[표 4-4] 코사인 유사도 임계값에 따른 통계 상세 표

	0.41	0.42	0.43	0.44	0.45	0.46	0.47	0.48	0.49
Precision	93.81	93.87	94.94	**94.05**	91.34	88.89	87.27	86.6	86.6
Recall	91	91	94	**95**	95	96	96	97	97
F-Measure	92.38	92.92	94.47	**94.52**	93.13	92.3	91.42	91.5	91.5

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[그림 4-4]와 [표 4-4]에서 알 수 있듯이 ‘0.4’와 ‘0.5’의 코사인 유사도 사이에 존재하는 값 중 코사인 유사도 임계값을 ‘0.44’로 지정할 시 가장 높은 철자오류 검출율을 보인다는 것을 알 수 있다. 이와 같은 결과를 통해 본 논문에서는 ‘0.44’를 철자오류 검출을 위한 코사인 유사도의 임계값으로 사용하게 된다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

C. 철자오류 교정 단어 리스트 구축

철자오류를 검출하였다면 철자오류를 교정하기 위해 사전에서 철자오류 단어와 편집거리가 작은 단어들을 뽑아 리스트를 구축하여 철자오류 교정을 수행할 필요가 있다. 철자오류가 발생한 단어와 본래 입력해야할 정상 단어의 최소 편집거리는 '1'이 되지만, 철자오류가 발생한 단어와 정상 단어의 편집거리가 '1'이상인 경우도 존재하기 때문에 단순히 철자오류와 편집거리가 '1'인 단어들만 추출하여 교정 단어 리스트를 구축할 경우 철자오류 교정율이 하락하는 원인이 될 수 있다. 하지만 그렇다고 해서 편집거리에 제한 없이 철자오류가 발생한 단어의 교정 단어들을 사전에서 추출하게 되면 다수의 단어들이 수집되면서 철자오류 교정 알고리즘의 처리 시간이 기하급수적으로 증가하는 상황이 발생한다. 그렇기 때문에 교정 단어 리스트 구축 시 최소편집거리는 '1'로 지정하며 최대 편집거리에 일정한 제한을 둘 필요가 존재한다. 그렇기 때문에 철자오류 단어의 길이에 따라 정상 단어와의 편집거리 변화가 어떻게 되는지 확인해야할 필요가 존재한다. [표 4-5]는 수집한 철자오류 단어의 문자열 길이에 따른 정상 단어와의 편집거리를 나타내고 있다.

[표 4-5] 음절 길이에 따른 철자오류 단어의 편집거리

음절길이 편집거리	2	3	4
1	86	107	8
2	1	22	3
합	87(38%)	129(57%)	11(5%)

[표 4-5]는 철자오류 단어와 정산 단어의 편집거리에 대한 통계 자료를 추출하기 위하여 수집한 227개의 단어를 통해 작성된 자료이다. 문자열의 길이는 음소와 음절을 기준으로 잴 수 있지만, 문자열의 길이를 음소 단위로 구하게 되면 단어마다 음절을 구성하는 음소가 종성의 여부에 따라서 큰 차이가 발생할 수 있기 때문에 음절 단위를 기준으로 철자오류 단어와 정상 단어의 편집거리에 대해서 구하게 되었다. 또한 [표 4-5]의 통계 자료를 추출하기 위해 수집한 철자오류 단어만으로는 각기 수집한 철자오류 단어의 음절 길이에 대한 비율 및 개수가 다르기 때문에

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

철자오류 단어의 음절 길이에 따른 정상 단어와의 편집거리에 대해 제대로 된 통계를 추출할 수가 없다. 그렇기 때문에 추출한 통계 자료에서 음절 길이가 '3'인 철자오류 단어를 기준으로 잡고 편집거리가 '2'인 철자오류 단어의 통계 자료에는 '1.5'를, 그리고 편집거리가 '4'인 철자오류 단어의 통계 자료에는 '11.5'를 곱함으로서 세 음절길이에 따른 통계 자료를 비슷한 수준으로 올려 평가해 보았다. [표 4-6]은 비슷한 수준으로 숫자를 끌어올린 통계자료를 보여주고 있다.

[표 4-6] 기준을 갖춘 음절 길이에 따른 철자오류 단어의 편집거리

음절길이 편집거리	2	3	4
1	129	107	92
2	2	22	34
합	131	129	126

[표 4-6]을 보면 알 수 있듯이 철자오류 단어의 음절 길이가 길어질수록 편집거리가 '1'인 철자오류 단어의 개수가 줄어들고 있으며 편집거리가 '2'인 단어의 개수가 늘어난다는 것을 알 수 있다. 어느 시점에서는 철자오류 단어와 정상 단어의 편집거리가 '3'인 단어도 등장할 수 있으나, 평가 데이터에서는 음절 길이가 '4' 이상인 단어가 존재하지 않기 때문에 이 부분은 무시하게 된다. 또한 [표 4-6]의 데이터는 막연히 생각할 수 있는 단어의 길이 길어질수록 철자오류가 발생할 음소의 개수가 많을 것이라는 생각을 확정할 수 있는 자료가 된다. [표 3-11]과 [표 3-12] 그리고 [표 4-6]에서 볼 수 있는 정보를 취합하면 철자오류 교정을 위한 단어를 추출하기 위해 [표 4-7]과 같은 규칙을 세울 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[표 4-7] 철자오류 교정 단어 리스트 구축을 위한 규칙

1.	최소 하나의 음소에서 철자오류가 발생하므로 음소 단위를 기준으로 하여 교정 단어들을 추출한다.
2.	한 음절에 편집거리가 2회 이상 발생한 단어는 추출할 필요가 없다.
3.	음절길이가 2인 철자오류 단어는 편집거리가 2인 교정 단어를 추출할 필요가 없다.
4.	음절길이가 3 이상인 철자오류 단어는 편집거리가 2인 교정 단어를 추출할 필요가 존재한다.

[표 4-7]에서 볼 수 있듯이 철자오류 교정 단어 리스트를 구축하기 위해 3개의 규칙을 지정하였다. 1번 규칙은 [표 3-11]과 [표 3-12]에서 볼 수 있는 음절 기준으로 철자오류 단어와 정상 단어의 편집거리를 구할 경우와 음소 기준으로 철자오류 단어와 정상 단어의 편집거리를 구할 경우 편집거리가 '1'인 경우와 '2'인 경우의 숫자가 비슷하다는 것은 한 음절에 하나의 음소가 철자오류가 발생하였기 때문에 비슷한 편집거리 결과가 나왔다고 할 수 있다. 한 음절에 2회의 철자오류가 발생한 경우도 존재하지만 전체적인 비율로 보면 1%도 되지 않는 수준이기 때문에 무시하도록 한다. 2번 규칙은 [표 4-6]을 통해 철자오류 단어가 2 이하일 경우 2회 이상 철자오류가 거의 발생하지 않았기 때문에 지정한 규칙이며, 3번 규칙은 2번 규칙과 마찬가지로 [표 4-6]을 통해 철자오류가 발생한 경우가 3음절 이상일 경우 2회 이상 철자오류가 발생한 경우가 생겨나기 때문에 지정한 규칙이다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

D. 철자오류 교정

철자오류 교정 단어들의 리스트를 구축하였다면 구축한 철자오류 교정 단어 리스트의 단어들을 철자오류가 발생한 단어와 치환한 이후 철자오류 교정 알고리즘을 적용시키는 것으로 철자오류 교정을 수행하게 된다. 철자오류 교정은 철자오류 검출과 마찬가지로 수식 (2)의 코사인 유사도를 이용하여 철자오류 교정을 수행하게 된다. 하지만 철자오류 검출과 철자오류 교정의 차이점은 철자오류 검출가 코사인 유사도에 임계값을 지정해 두어 수식 (2)의 결과값이 미리 지정해 둔 임계값 이하이면서 동시에 가장 낮은 수치를 기록한 것을 철자오류 단어라 판단하여 검출하였지만, 철자오류 교정에서는 임계값을 지정하지 않고 단순히 코사인 유사도가 높은 단어를 올바른 교정 단어라 판단하고 교정을 수행한다. 철자오류 교정에 대한 성능을 평가하기 위해 철자오류 검출 때와 마찬가지로 정확도와 재현율, 그리고 F-Measure를 구함으로서 철자오류 교정에 대한 알고리즘의 성능을 평가하게 된다. [표 4-8]은 철자오류 교정 알고리즘을 적용하여 추출된 정확도와 재현율, 그리고 F-Measure값을 보여주고 있다.

[표 4-8] 철자오류 교정 알고리즘의 성능

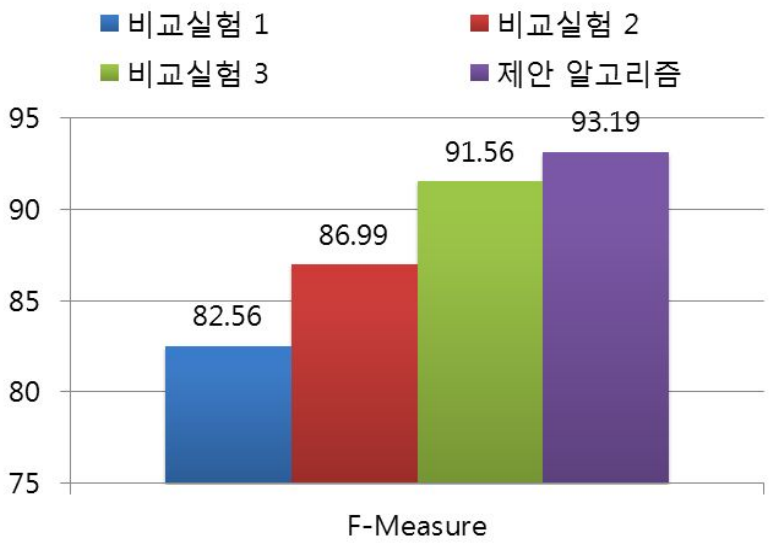
	Value
Precision	97.80
Recall	89
F-Measure	93.19

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

E. 철자오류 교정 알고리즘의 성능 비교

본 논문에서 제안한 철자오류 교정 알고리즘에 대한 성능을 평가하기 위해 기존에 코사인 유사도의 자세한 임계값과 교정 단어 리스트 규칙 없이 수행했던 연구[29]와 코사인 유사도를 이용하여 철자오류를 교정했던 다른 연구[16]와 비교하게 된다. 철자오류 교정 알고리즘은 알고리즘에 따라 정확도가 높은 대신 재현율이 낮은 알고리즘이 있거나 반대로 정확도가 낮은 대신 재현율이 높은 알고리즘도 존재하기 때문에 성능 비교는 F-Measure을 비교함으로써 평가한다. [그림 4-5]는 네 종류의 철자오류 교정 알고리즘에 대한 F-Measure을 보여주고 있다. 첫 번째 그래프 바는 이전에 연구를 수행한 철자오류 교정 알고리즘이며, 두 번째 그래프 바는 기존의 코사인 유사도를 이용한 철자오류 교정 알고리즘, 그리고 세 번째 그래프 바는 부산대학교에서 진행한 철자오류 교정 알고리즘을 나타내며, 마지막으로 네 번째 그래프 바는 본 논문에서 제안하는 철자오류 교정 알고리즘을 나타내고 있다.



[그림 4-5] 철자오류 교정 알고리즘의 F-Measure 비교 그래프

첫 번째 그래프 바를 나타내는 이전에 수행한 연구에서 사용한 알고리즘에 비해 성능이 높게 나온 이유는 이전에 수행했던 연구에서는 본 논문과는 달리 최적의

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

코사인 유사도 임계값을 지정하지 않았으며 철자오류 교정 단어들을 추출하기 위한 명확한 규칙 없이 단순히 편집거리가 '1'인 단어들로만 추출하여 철자오류 교정을 수행하였기 때문에 성능에 대한 차이가 크게 나오게 되었다. 두 번째 그래프 바를 나타내는 기존의 코사인 유사도 알고리즘을 이용한 연구에서는 철자오류 교정을 위해 문장에서 등장한 단어들의 연어 관계를 통해 특정 단어와 관계가 있는 단어들만 묶어 코사인 유사도를 구함으로서 관계가 없거나 혹은 관계가 적은 단어들은 코사인 유사도를 구할 수 없었기 때문에 본 논문에서 제안한 알고리즘과 성능 차이가 발생하였다. 세 번째 그래프 바를 나타내는 부산대에서 수행한 연구 같은 경우, 본 논문과의 차이는 크게 발생하지는 않았으며, 부산대에서 수행한 연구에서 교정할 수 있는 철자오류 단어의 폭이 크기 때문에 어떤 알고리즘이 더 우수하다고 할 수 없다. 하지만 부산대에서 수행한 알고리즘에서 철자오류 교정 단어를 추출하는 방법은 철자오류가 발생한 단어의 음소를 하나씩 변경하며 문장에서 등장하는 단어들과의 동시 등장 확률을 구함으로서 수행되기 때문에 처리시간이 긴 반면 본 논문에서 제안한 철자오류 교정 단어 추출 방법은 철자오류 단어의 길이에 따라 적합한 교정 단어들을 사전에서 추출하여 동시 등장 확률을 구하는 것이기 때문에 단순히 하나의 음소를 변경하여 교정 단어를 추출하는 방법에 비해 처리시간이 굉장히 짧기 때문에 처리 시간에 한해서는 부산대에서 수행한 연구의 알고리즘에 비해 본 논문에서 제안한 알고리즘이 우수하다고 할 수 있다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

V. 결론 및 제언

본 논문에서는 기존의 코사인 유사도 알고리즘을 이용하여 철자오류를 교정하는 알고리즘과 철자오류 교정 단어를 추출하기 위하여 한글 편집거리 알고리즘을 이용해 철자오류 단어와 편집거리가 작은 단어들을 추출하는 알고리즘을 복합한 새로운 철자오류 교정 알고리즘에 대해서 제안하였다.

제안한 철자오류 교정 시스템에서 철자오류 검출 및 교정을 위해 사용한 코사인 유사도 알고리즘은 여러 철자오류 교정 시스템 중 교정 어휘 쌍을 묶어서 수행하는 통계 기반 방식의 철자오류 교정 모델이라고 할 수 있다. 그렇기 때문에 철자오류 교정 알고리즘에 대한 성능을 평가하기 위해 말뭉치 사전을 구축하였으며 동시에 SNS의 데이터를 수집하여 철자오류 데이터 또한 수집하게 되었다.

본 논문에서 제안한 철자오류 교정 알고리즘은 성공적으로 수행을 완료하였으며, 기존의 코사인 유사도 알고리즘을 이용한 철자오류 교정 연구와 이전에 수행했던 철자오류 교정 연구와 비교하면 철자오류 검출에 대한 코사인 유사도를 명백히 설정하였으며, 또한 철자오류 교정 단어 리스트를 구축하기 위한 규칙을 추가함으로서 7%에서 10% 이상의 성능 향상의 결과를 보여주었다. 이는 상당히 높은 수준의 알고리즘 성능 향상을 보여주고 있다고 할 수 있다. 또한 그 이외의 철자오류 교정 알고리즘들과 성능을 비교해 보아도 높은 결과를 이룩했다는 것을 알 수 있다.

일반적인 철자오류 교정에 대한 알고리즘은 문장에서 등장하는 각 단어들의 최대 등장 빈도를 구하기 위해 나이브 베이지안의 방식을 많이 사용하고 있으나 코사인 유사도 알고리즘을 사용해서 철자오류를 교정한 본 연구와 같이 각양각생의 방법을 이용하여 철자오류 교정 연구를 수행할 수 있다. 하나의 방법으로만 철자오류 교정을 수행하게 된다면, 해당 연구는 하나의 알고리즘으로만 틀에 박힌 단순한 연구로만 남게 될 수 있다. 차후 철자오류 교정을 수행할 경우 하나의 알고리즘에 틀 박힌 연구가 아닌 다종다양한 알고리즘을 이용하여 철자오류 교정 연구를 수행할 필요가 존재한다.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

참고문헌

[1] Andrew R. Golding, “A Window-Based Approach to Context-Sensitive Spelling Soccrection”, Machine Learning, Vol.34, pp.107-130, 1999.

[2] 최현수, 윤애선, 권혁철, “조사제약 조건의 완화에 의한 문맥의존 철자오류 교정의 재현율 향상 방식”, 정보과학회논문지 : 소프트웨어 및 응용, 제 41권 제 3호, pp.249-256, 2014.

[3] 최현수, 권혁철, 운애선, “동적 윈도우를 갖는 조건부 확률 모델을 이용한 한국어 문맥의존 철자오류 교정 규칙의 재현율 향상”, 정보과학회논문지 제 42권 제 5호, pp.629-636, 2015.

[4] Minho Kim, Jingzhi Jin, Aesun Yoon, Hyuk-Chul Kwon, “Statistical Context-Sensitive Spelling Correction Using Typing Error Rate”, Proceedings of The 2014 IEEE 16th International Conference on Computational Science and Engineering(CSE), Vol.16, No.1, pp.1242-1246, 2013.

[5] 서한영, “관계어에 기반을 둔 확률추정방법을 이용한 문맥 의존 철자오류 교정”, 부산대학교, 석사학위논문, 2014.

[6] 김경식, 최성기, 권혁철, “극한 언어사용 환경에 적응적인 문맥의존 철자오류 교정 기법”, 한국정보과학회 학술발표논문집, Vol.2015, No.6, 2015.

[7] Eric Mays, Fred J. Damerau, Robert L. Mercer, “Context Based Spelling Correction”, Imformation Processing & Management, Vol.23, Mo.5, pp.517-522, 1991.

[8] Aminul Islam, Diana Inkpen, “Semantic Text Similarity Using Corpus-Based Word Similarity and String Similarity”, ACM Transaction on Knowledge Discovery from Data(TKDD), Vol.2, No.2, pp.1241-1249, 2008.

[9] Aminul Islam, Diana Inkpen, “Real-Word Spelling Correction Using Google Web 1T 3-Grams”, Proceedings of The 2009 Conference on Empirical Methods in Natural Language Processing, Vol.3, No.3, pp.1241-1249, 2009.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[10] 김민호, 권혁철, 최성기, “어절 N-gram을 이용한 문맥의존 철자오류 교정”, 정보과학회논문지, Vol.41, No.12, pp.1081-1089, 2014.

[11] Andrew R. Golding, Dan Roth, “A Window-Based Approach to Context-Sensitive Spelling Correction”, Machine learning, Vol.34, No.1-3, pp.107-130, 1999.

[12] Andrew R. Golding, “A Bayesian Hybrid Method for Context-Sensitive Spelling Correction”, Proceeding of the 3rd Workshop on Very Large Corpora, Vol.3, No.1, pp.39-53, 1995.

[13] Mark D. Kernighan, Kenneth W. Church, William A. Gale, “A Spelling Correction Program Based on a Noisy Channel Model”, Proceedings of The 13th Conference on Computational Linguistics, Vol.2, No.1, 1990.

[14] Minho Kim, Sung-Ki Choi, Hyuk-Chul Kwon, “Context-Sensitive Spelling Error Correction Using Inter-Word Semantic Relation Analysis”, Proceeding of The 50th International Conference on Infomation Science and Applications(ICISA 2014), Vol.5, No.1, pp.209-302, 2014.

[15] 김민호, 김경식, 권혁철, “교정 어휘 쌍을 이용한 통계적 문맥 철자오류 교정”, 2013 한국컴퓨터종합학술대회, 제 20권, 제 1호, pp.607-609, 2013.

[16] 김동주, “철자오류에 기인한 가의미 오류의 검출 및 교정 방법”, 한국컴퓨터정보학회논문집, Vol.18, No.10, pp.173-182, 2013.

[17] 김동주, “문서내 단어간 비교를 통한 철자오류 검출”, 한국컴퓨터정보학회논문집, Vol.16, No.12, pp.83-92, 2011.

[18] 노형종, 차정원, 이근배, “띄어쓰기 및 철자 오류 동시교정을 위한 통계적 모델”, 정보과학회논문지 : 소프트웨어 및 응용, Vol.32, No.2, pp.131-139, 2007.

[19] Dan Gusfield, “Algorithm on Strings, Trees, and Sequences”, Computer Science and Computational Biology, Cambridge University Press, January 2007.

[20] Robert A. Wagner, Michael J. Fisher, “The String-to-String Correction Problem”, Journal of the ACM 21(1), pp.168-173, 1974.

[21] 노강호, 김진욱, 김은상, 박근수, 조환규, “한글에 대한 편집 거리 문제”, 정보과학회논문지 : 시스템 및 이론, Vol.37, No.2, pp.103-109, 2010.

Collection @ chosun

조선대학교
CHOSUN UNIVERSITY

[22] 노강호, 박근수, 조환규, 장소원, “음소의 분류 체계를 이용한 한글 편집거리 알고리즘”, 정보과학회논문지 : 시스템 및 이론, Vol.37, No.6, pp.323-329, 2010.

[23] 노강호, 박근수, 조환규, 장소원, “음소의 1차원 배열을 이용한 한글 유사도 및 편집거리 알고리즘”, 정보과학회논문지 : 컴퓨팅의 실제 및 레터, Vol.17, No.10, pp.519-526, 2011.

[24] 이세희, 김학수, “음절 통계를 이용한 경량화된 철자 오류 교정 모델”, 한국정보과학회 학술발표눈문집, Vol.36, No.1, pp.84-85, 2009.

[25] Gonzalo Navarro, “A Guided Tour to Approximate String Matching”, ACM Computing Surveys, Vol.33, No.1, pp.31-88, 2001.

[26] https://ko.wikipedia.org/wiki/%ED%82%A4%EB%A1%9C%EA%B9%85

[27] 임한규, 김응모, “철자오류 통계자료에 근거한 철자오류 교정시스템”, 정보과학회논문지, Vol.2, No.6, pp.839-846, 1995.

[28] Christopher D. Manning, Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, 2001.

[29] 박승현, 김판구, “한글 유사도 알고리즘을 이용한 한국어 철자오류 교정방법 연구”, 제 4회 융합·스마트미디어시스템 워크샵, 2016.

Collection @ chosun